

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та безпеки

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «МАГІСТР»**

**СИСТЕМА КЕРУВАННЯ БАГАТОФУНКЦІОНАЛЬНИМ
РОБОТИЗОВАНИМ МАНІПУЛЯТОРОМ ЧЕРЕЗ BLUETOOTH**

**BLUETOOTH MULTI-FUNCTIONAL ROBOTIC MANIPULATOR
CONTROL SYSTEM**

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти

групи КІМ-21

Стасюк Дмитро Володимирович

(підпис)

Керівник:

к.т.н., доцент

Гринюк Сергій Васильович

(підпис)

Кваліфікаційну роботу

допущено до захисту

« » грудня 2025 р.

Гарант освітньої програми:

к.т.н., доцент

Гринюк Сергій Васильович

(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та безпеки

Ступінь вищої освіти: магістр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Т.ТЕРЛЕЦЬКИЙ

« _____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Стасюку Дмитру Володимировичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Система керування багатофункціональним роботизованим маніпулятором через Bluetooth*

Керівник роботи *к.т.н., доцент Гринюк Сергій Васильович*

затверджені наказом закладу вищої освіти від «17» червня 2025 року № 290/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 09.12.2025р.

3. Вихідні дані до роботи *Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області, різні інтернет-ресурси технічного спрямування*

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

Вступ

Аналіз методів та засобів керування роботизованими маніпуляторами

Проектування архітектури та розробка математичного забезпечення системи

Моделювання та дослідження системи у віртуальному середовищі

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Загальна архітектура взаємодії компонентів у системі мобільного моніторингу

Блок-схема моделювання динаміки маніпулятора в середовищі Matlab Simscape Multibody

Структурна схема системи керування з урахуванням затримок у прямому та зворотному каналах зв'язку

Граф ієрархічної структури URDF-моделі маніпулятора

Архітектура пакету ros_control: потік даних від планувальника до симулятора

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз методів та засобів керування роботизованими маніпуляторами</i>	<i>Гринюк С.В., доцент</i>		
<i>Проектування архітектури та розробка математичного забезпечення системи</i>	<i>Гринюк С.В., доцент</i>		
<i>Моделювання та дослідження системи у віртуальному середовищі</i>	<i>Гринюк С.В., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Гринюк С.В., доцент</i>		
<i>Показник запозичень тексту</i>	_____ %		
<i>Академічна доброчесність</i>	<i>Міскевич О.І., ст.викладач</i>		

7. Дата видачі завдання 18.06.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Огляд літератури із досліджуваної проблеми</i>	До 01.08.2025 р.	
2.	<i>Аналіз методів та засобів керування роботизованими маніпуляторами</i>	До 20.08.2025 р.	
3.	<i>Проектування архітектури та розробка математичного забезпечення системи</i>	До 25.09.2025 р.	
4.	<i>Моделювання та дослідження системи у віртуальному середовищі</i>	До 20.10.2025 р.	
5.	<i>Висновки та пропозиції</i>	До 25.10.2025 р.	
6.	<i>Формування списку використаних джерел</i>	До 27.10.2025 р.	
7.	<i>Формування додатків</i>	До 30.10.2025 р.	
8.	<i>Оформлення ілюстративного матеріалу</i>	До 05.11.2025 р.	
9.	<i>Представлення остаточного варіанту кваліфікаційної роботи керівникові</i>	До 11.11.2025 р.	
10.	<i>Нормоконтроль</i>	До 29.11.2025 р.	
11.	<i>Інструментальна перевірка на академічний плагіат</i>	До 02.12.2025 р.	
12.	<i>Здача кваліфікаційної роботи та всіх супровідних документів на кафедру</i>	До 09.12.2025 р.	

Здобувач вищої освіти

_____ (підпис)

Стасюк Д.В.

_____ (прізвище, ініціали)

Керівник кваліфікаційної роботи

_____ (підпис)

Гринюк С.В.

_____ (прізвище, ініціали)

АНОТАЦІЯ

Стасюк Д. В. Система керування багатофункціональним роботизованим маніпулятором через Bluetooth. Рукопис.

Кваліфікаційна робота магістра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел, додатків.

Перший розділ присвячено аналітичному огляду сучасних маніпуляційних систем та засобів їх керування. Проведено порівняльний аналіз кінематичних схем, за результатами якого обрано антропоморфну структуру як найбільш універсальну. Обґрунтовано переваги протоколу Bluetooth для локального телекерування порівняно з аналогами (Wi-Fi, ZigBee).

У другому розділі представлено етапи проектування апаратно-програмного комплексу. Розроблено математичні моделі прямої та оберненої кінематики з використанням параметрів Денавіта-Хартенберга. Описано архітектуру системи на базі мікроконтролера ESP32, розроблено надійний протокол обміну даними та клієнтський додаток для мобільних пристроїв. Запропоновано алгоритм інтерполяції траєкторії з S-подібним профілем швидкості для згладжування рухів при нестабільному зв'язку.

Третій розділ містить результати моделювання та експериментальних досліджень. Описано процес створення цифрового двійника маніпулятора (URDF) та його інтеграцію з контролерами ROS. Розроблено програмний вузол-емулятор, що імітує мережеві затримки та втрати пакетів. На основі серії віртуальних експериментів доведено ефективність запропонованих алгоритмів буферизації, які дозволяють зменшити динамічні похибки та забезпечити плавність руху маніпулятора в умовах зашумленого радіоефіру.

Ключові слова: мобільний робот, slam, ROS2, IoT, mqtt, Gazebo, навігація, моніторинг, Raspberry PI, Lidar.

ANNOTATION

Stasiuk D. Bluetooth multi-functional robotic manipulator control system. Manuscript.

Master's Qualification Paper, Educational Program «Computer Engineering», Specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

The qualification paper consists of an introduction, three chapters, conclusions, a list of references, and appendices.

The first chapter is devoted to an analytical review of modern manipulation systems and their control means. A comparative analysis of kinematic schemes was carried out, based on which the anthropomorphic structure was chosen as the most universal. The advantages of the Bluetooth protocol for local teleoperation compared to analogs (Wi-Fi, ZigBee) are substantiated.

The second chapter presents the design stages of the hardware-software complex. Mathematical models of forward and inverse kinematics using Denavit-Hartenberg parameters were developed. The architecture of the system based on the ESP32 microcontroller is described, and a reliable data exchange protocol and a client application for mobile devices were developed. A trajectory interpolation algorithm with an S-curve velocity profile is proposed to smooth movements under unstable connection conditions.

The third chapter contains the results of simulation and experimental research. The process of creating a digital twin of the manipulator (URDF) and its integration with ROS controllers is described. A software node-emulator simulating network delays and packet losses was developed. Based on a series of virtual experiments, the effectiveness of the proposed buffering algorithms, which allow reducing dynamic errors and ensuring the smoothness of the manipulator's movement in a noisy radio environment, is proven.

Keywords: mobile robot, slam, ROS2, IoT, MQTT, Gazebo, navigation, monitoring, Raspberry Pi, Lidar.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ КЕРУВАННЯ РОБОТИЗОВАНИМИ МАНІПУЛЯТОРАМИ.....	10
1.1 Огляд сучасних багатофункціональних маніпуляторів	10
1.2 Аналіз бездротових протоколів передачі даних для IoT та робототехніки	14
1.3 Огляд програмних середовищ для симуляції роботів.....	18
1.4 Аналіз проблеми затримок та втрати пакетів у бездротових мережах керування	21
РОЗДІЛ 2 ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА РОЗРОБКА МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ.....	26
2.1 Розробка кінематичної моделі маніпулятора:.....	26
2.2 Архітектура системи керування: структурна схема та вибір апаратної бази	29
2.3 Розробка протоколу обміну даними через Bluetooth	33
2.4 Розробка клієнтського програмного забезпечення для надсилання керуючих команд	36
2.5 Алгоритмічне забезпечення траєкторії руху.....	40
РОЗДІЛ 3 МОДЕЛЮВАННЯ ТА ДОСЛІДЖЕННЯ СИСТЕМИ У ВІРТУАЛЬНОМУ СЕРЕДОВИЩІ.....	44
3.1 Створення цифрового двійника маніпулятора для середовища симуляції	44
3.2 Інтеграція моделі з системою керування.....	48
3.3 Емулювання каналу зв'язку Bluetooth у симуляторі.....	52
3.4 Проведення експериментальних досліджень	57
3.5. Аналіз отриманих результатів	61
ВИСНОВКИ.....	65
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	67
ДОДАТКИ.....	71

ВСТУП

Сучасний етап розвитку промисловості та сервісної робототехніки характеризується переходом до концепцій Індустрії 4.0 та 5.0, де ключову роль відіграють гнучкі кіберфізичні системи та Інтернет речей (IoT). Роботизовані маніпулятори, які раніше функціонували виключно в ізольованих зонах виробничих ліній, сьогодні інтегруються в неструктуроване середовище для виконання завдань у складських приміщеннях, медицині, агросекторі та зонах підвищеної небезпеки. У цьому контексті критичного значення набуває проблема організації надійного дистанційного керування, яке дозволяє оператору взаємодіяти з механізмом без фізичного контакту, забезпечуючи безпеку персоналу та мобільність розгортання системи.

Традиційні дротові інтерфейси керування, попри свою надійність, суттєво обмежують робочу зону та мобільність платформ. Використання бездротових технологій, зокрема протоколу Bluetooth, відкриває нові можливості для створення компактних та енергоефективних систем телекерування з використанням мобільних терміналів (смартфонів, планшетів). Проте впровадження радіоканалу в контур керування вносить стохастичні збурення – змінні часові затримки та втрату пакетів даних, що може призвести до втрати стійкості системи та виникнення аварійних ситуацій.

Розробка та налагодження алгоритмів компенсації цих збурень на фізичному обладнанні є коштовним та ризикованим процесом. Тому актуальним науково-прикладним завданням є створення «цифрових двійників» у середовищах симуляції (таких як Gazebo/ROS), які дозволяють верифікувати кінематику та динаміку керування з урахуванням недоліків каналу зв'язку ще до етапу фізичної реалізації.

Метою роботи є підвищення надійності та точності дистанційного керування багатофункціональним маніпулятором через бездротовий канал Bluetooth шляхом розробки алгоритмічного забезпечення та його верифікації у віртуальному середовищі симуляції.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- 1) провести аналіз сучасних методів керування маніпуляторами, бездротових протоколів передачі даних та програмних засобів симуляції робототехнічних систем;
- 2) розробити математичну модель кінематики маніпулятора (пряму та обернену задачі) та обґрунтувати вибір апаратної платформи системи керування.
- 3) створити протокол обміну даними прикладного рівня, стійкий до помилок передачі, та розробити клієнтське програмне забезпечення для мобільного терміналу;
- 4) створити цифровий двійник маніпулятора (URDF-модель) та інтегрувати його в середовище ROS/Gazebo;
- 5) розробити програмний модуль емуляції мережевих затримок та втрат пакетів для проведення валідних випробувань у симуляторі;
- б) експериментально дослідити вплив якості каналу зв'язку на точність позиціонування та перевірити ефективність запропонованих алгоритмів інтерполяції траєкторії.

Об'єкт дослідження – процес дистанційного керування антропоморфними робототехнічними системами в умовах бездротової передачі даних.

Предмет дослідження – методи та алгоритми керування рухом маніпулятора через канал Bluetooth з використанням технологій цифрового моделювання для компенсації мережевих затримок.

Наукова новизна одержаних результатів полягає в удосконаленні методу керування маніпулятором в умовах стохастичних затримок каналу Bluetooth. Це досягнуто за рахунок використання алгоритму буферизації з S-подібною інтерполяцією швидкості, що, на відміну від існуючих підходів, дозволяє зменшити динамічні навантаження на приводи та нівелювати ривки при збереженні точності відпрацювання траєкторії. Водночас набуло подальшого розвитку застосування технології «Software-in-the-Loop» (SIL) для тестування вбудованих систем. Реалізація цього підходу здійснена шляхом розробки спеціалізованого вузла-моста в середовищі ROS, який імітує логіку

мікроконтролера та фізику радіоканалу, що надає можливість прогнозувати поведінку реального робота без необхідності проведення попередніх фізичних випробувань.

Практичне значення одержаних результатів визначається розробкою та верифікацією архітектури системи керування на базі мікроконтролера ESP32, яка повністю готова до фізичної реалізації. Вагомим прикладним результатом є створення програмного пакету для середовища ROS/Gazebo, що включає налаштовану модель маніпулятора та інструментарій для тестування стійкості алгоритмів до умов поганого зв'язку.

Апробація результатів роботи. Результати роботи представлені на VII Міжнародній науково-теоретичній конференції «Науковий метод: реальність та майбутні тенденції дослідження», яка проходила 5 грудня 2025 року [1].

РОЗДІЛ 1

АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ КЕРУВАННЯ РОБОТИЗОВАНИМИ МАНІПУЛЯТОРАМИ

1.1 Огляд сучасних багатофункціональних маніпуляторів

Сучасна робототехніка переживає етап трансформації від жорстко запрограмованих індустриальних механізмів до гнучких інтелектуальних систем, здатних адаптуватися до динамічного оточення. Багатофункціональні маніпулятори, які є ключовим елементом концепції Індустрії 4.0 та 5.0, класифікують за сукупністю ознак, серед яких кінематична структура, тип приводів, ступінь автономності та характер взаємодії з людиною. У науковій літературі останніх років (2021-2025) домінуючим трендом визначено перехід до колаборативних роботів (коботів) та мобільних маніпуляторів, що поєднують маніпуляційні можливості з локомоційними системами для роботи в неструктурованих середовищах, таких як агросектор, медицина чи логістичні склади [2].

Основою класифікації маніпуляторів є їхня кінематична схема, яка визначає геометрію робочого простору та динамічні характеристики. Найпоширенішим класом залишаються маніпулятори з послідовною кінематичною структурою (*serial manipulators*), які являють собою розімкнений кінематичний ланцюг. Типовим представником цього класу є антропоморфний маніпулятор, кінематичну схему якого наведено на рисунку 1.1.

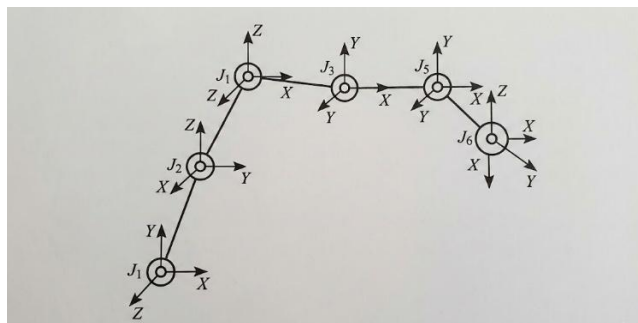


Рисунок 1.1 – Кінематична схема шестиосьового антропоморфного маніпулятора [2]

На рисунку 1.1 схематично зображено структуру маніпулятора з послідовною кінематикою, яка складається з жорстких ланок, з'єднаних між собою обертальними шарнірами. Схема ілюструє шість ступенів вільності (6 DOF), що дозволяє робочому органу досягати будь-якої точки в межах робочого простору під довільним кутом. Виділено основні функціональні вузли: нерухома основа (base), плече (shoulder), лікоть (elbow) та кистьовий механізм (wrist), який зазвичай має три ступені вільності для орієнтації інструменту (roll, pitch, yaw). Така будова забезпечує робочу зону, близьку до сферичної, що є оптимальним для складних траєкторій.

До класу послідовних роботів також належать системи типу SCARA (Selective Compliance Assembly Robot Arm), які широко застосовуються в операціях складання. Особливості їхньої робочої зони та розподіл ступенів вільності проілюстровано на рисунку 1.2. Такі роботи характеризуються жорсткістю у вертикальному напрямку та піддатливістю в горизонтальній площині. Сучасні дослідження підкреслюють ефективність використання нейромережових методів для розв'язання задач оберненої кінематики таких систем у реальному часі, що дозволяє компенсувати нелінійності та підвищити точність руху [3].

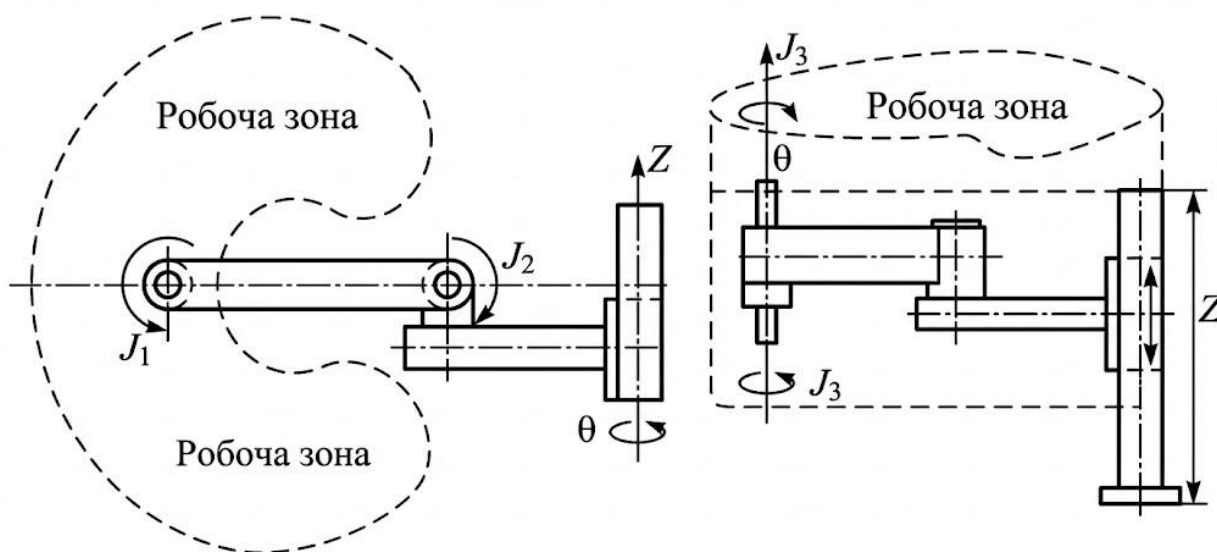


Рисунок 1.2 – Робоча зона та вісі руху маніпулятора типу SCARA [3]

Зображення демонструє специфічну конфігурацію маніпулятора SCARA, де дві головні ланки обертаються навколо вертикальних осей, забезпечуючи швидке позиціонування в горизонтальній площині X-Y. На схемі показано, що робоча зона має форму неповної циліндричної оболонки (або тороїдальну форму). Окремо виділено вісь Z, яка забезпечує вертикальний хід робочого інструменту, та вісь обертання інструменту. Графічно підкреслено кінематичну особливість конструкції: високу жорсткість у вертикальному напрямку, необхідну для операцій монтажу, при збереженні певної податливості в горизонтальній площині для компенсації похибок позиціонування.

На відміну від послідовних схем, маніпулятори з паралельною кінематикою (parallel manipulators) мають замкнені кінематичні ланцюги, де вихідна ланка з'єднана з основою декількома незалежними кінематичними гілками. Типовим представником є дельта-роботи, структурну організацію яких зображено на рисунку 1.3. Перевагою таких систем є висока жорсткість конструкції, здатність переміщувати значні вантажі з високим прискоренням та точністю, оскільки двигуни винесені на нерухому основу. У роботах 2024 року відзначається зростання інтересу до гібридних кінематичних схем, які поєднують переваги обох підходів [4].

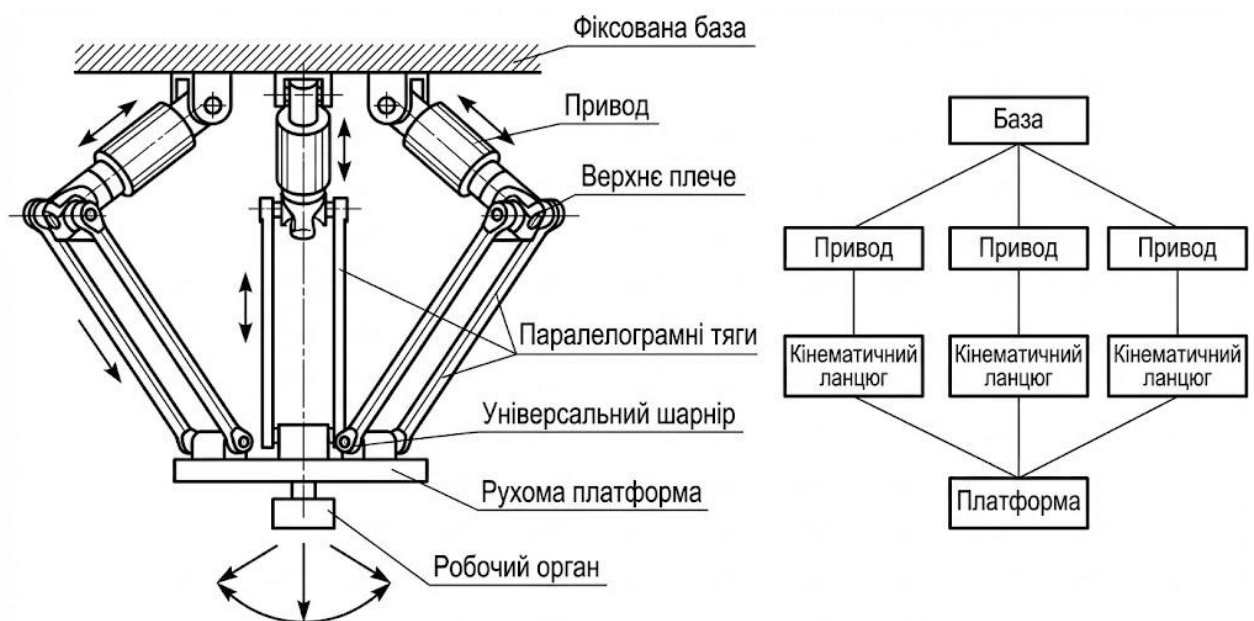


Рисунок 1.3 – Структурна схема паралельного маніпулятора типу «Дельта» [4]

Окремим класом, що набув значного поширення, є колаборативні роботи (коботи), призначені для безпосередньої взаємодії з людиною-оператором без захисних огорожень. Кінематично вони часто повторюють антропоморфні схеми, проте оснащені розвинутою сенсорною системою (датчиками моменту в кожному шарнірі, ємнісними сенсорами на корпусі) та програмними обмеженнями енергії руху. Це дозволяє реалізувати режими «ручного ведення» для навчання траєкторії та автоматичну зупинку при контакті з перешкодою. Аналіз застосування коботів у виробництві показує їх ефективність на малих та середніх підприємствах завдяки швидкому переналадженню та можливості бездротового керування через інтерфейси типу Bluetooth або Wi-Fi для інтеграції в IoT-екосистеми [5]. Порівняльну характеристику основних типів маніпуляторів наведено в таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз кінематичних схем сучасних маніпуляторів

Тип маніпулятора	Кінематична схема	Ступені вільності (DOF)	Переваги	Недоліки	Типові сфери застосування
Антропоморфний	Послідовна, обертальні пари	6 і більше	Великий робочий простір, гнучкість орієнтації	Складність керування, менша жорсткість на вильоті	Зварювання, фарбування, складні маніпуляції
SCARA	Послідовна, 3 обертальні + 1 поступальна	4	Висока швидкість та точність у горизонтальній площині	Обмежений вертикальний хід та орієнтація	Складання електроніки (Pick-and-Place), пакування
Дельта-робот	Паралельна	3-4	Дуже висока швидкість та прискорення	Обмежена вантажопідйомність та робоча зона	Швидкісне сортування, харчова промисловість
Декартовий (Gantry)	Послідовна, 3 поступальні пари	3	Простота керування, висока вантажопідйомність, масштабованість	Займає велику площу, повільніші за інші типи	3D-друк, переміщення важких вантажів, ЧПК

Продовження таблиці 1.1

Тип маніпулятора	Кінематична схема	Ступені вільності (DOF)	Переваги	Недоліки	Типові сфери застосування
Кобот (Cobot)	Послідовна (зазвичай)	6-7	Безпека взаємодії, легкість програмування	Обмежена швидкість та сила (згідно стандартів ISO)	Спільна робота з персоналом, медицина, лабораторії

Сфери застосування маніпуляторів суттєво розширилися завдяки впровадженню технологій штучного інтелекту та комп'ютерного зору. Окрім традиційного машинобудування, роботи активно інтегруються в агропромисловий комплекс для збору врожаю, де використовують м'які захвати (soft grippers) та адаптивні алгоритми керування для роботи з делікатними об'єктами. У медицині телекеровані маніпулятори дозволяють виконувати хірургічні втручання з високою точністю, при цьому досліджуються методи компенсації затримок у каналах зв'язку, що є критичним для систем дистанційного керування [6].

1.2 Аналіз бездротових протоколів передачі даних для IoT та робототехніки

Забезпечення надійного каналу зв'язку між оператором та виконавчим механізмом є критичною задачею при проектуванні дистанційно керованих робототехнічних систем. Вибір протоколу передачі даних визначає ключові експлуатаційні характеристики пристрою, такі як радіус дії, енергоефективність, затримка передачі команд (latency) та стійкість до завад. У сучасній інженерній практиці для локального керування роботами найчастіше розглядають стандарти сімейства IEEE 802.11 (Wi-Fi), IEEE 802.15.4 (ZigBee) та IEEE 802.15.1 (Bluetooth). Згідно з дослідженнями 2024 року, для мобільних маніпуляторів та IoT-пристроїв пріоритетним стає баланс між пропускнуою здатністю та енергоспоживанням, що наочно відображає класифікація бездротових

технологій за дальністю та швидкістю передачі даних, наведена на рисунку 1.4 [7].

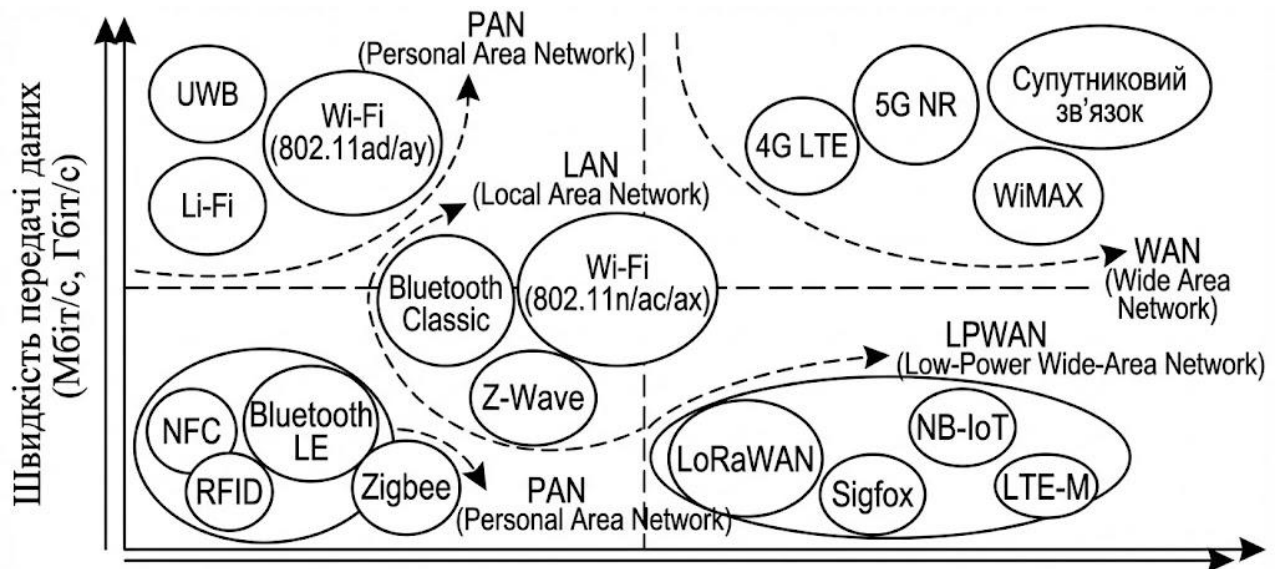


Рисунок 1.4 – Класифікація бездротових протоколів за співвідношенням швидкості передачі даних та радіусу дії [7]

Діаграма ілюструє позиціонування основних бездротових стандартів у координатах «Пропускна здатність – Дальність зв'язку». На рисунку видно, що Wi-Fi займає нішу високошвидкісних мереж із середнім радіусом дії, тоді як ZigBee та Bluetooth орієнтовані на менші швидкості та енергоспоживання. Виділено зону IoT, де Bluetooth Low Energy (BLE) забезпечує оптимальний компроміс для персональних мереж (PAN), пропонуючи достатню швидкість для телеметрії при мінімальних витратах енергії.

Технологія Wi-Fi забезпечує високу швидкість передачі даних, що є необхідним для трансляції відеопотоку з камер робота в реальному часі, проте вона характеризується значним енергоспоживанням та складністю процедури підключення до інфраструктури (роутера). Натомість протокол ZigBee, побудований на топології комірчастих мереж (Mesh), дозволяє створювати надійні системи з великою кількістю вузлів та низьким енергоспоживанням, але має низьку пропускну здатність (до 250 кбіт/с), що обмежує його використання для прямого керування маніпулятором з високою частотою оновлення

координат. Альтернативою виступає технологія Bluetooth, яка еволюціонувала у дві окремі гілки: Bluetooth Classic (BR/EDR) та Bluetooth Low Energy (BLE). Архітектурні відмінності між стеками протоколів Bluetooth Classic та BLE, які визначають сфери їх застосування, представлено на рисунку 1.5.

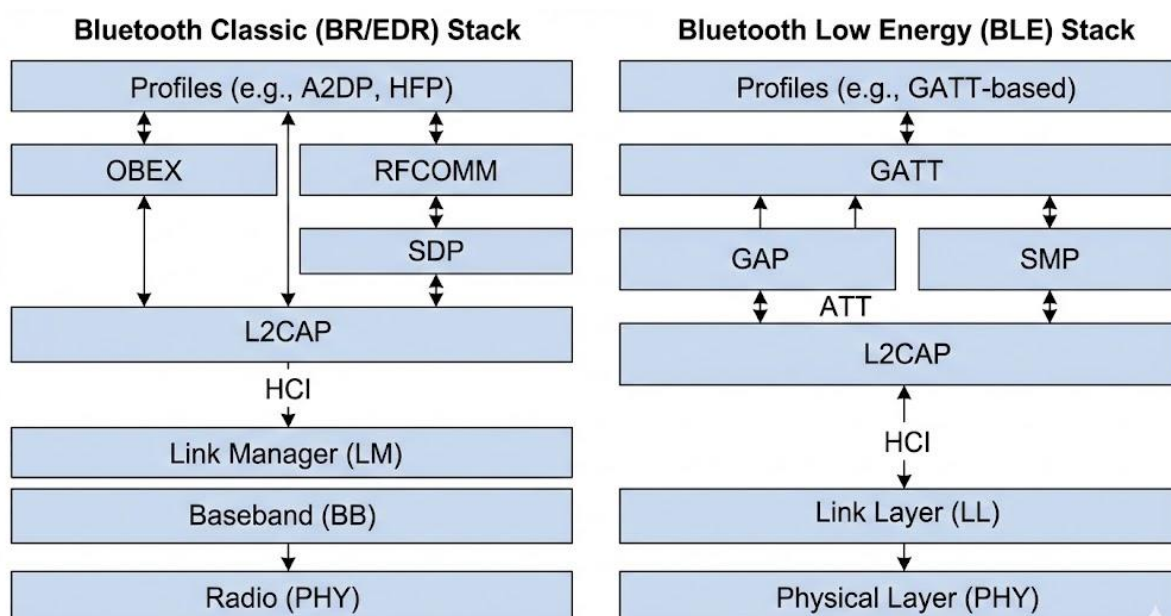


Рисунок 1.5 – Порівняння архітектури стеків протоколів Bluetooth Classic та Bluetooth Low Energy [8]

На схемі зображено рівневу структуру двох версій стандарту Bluetooth. Для Bluetooth Classic показано наявність профілю SPP (Serial Port Profile), який емулює послідовне з'єднання і забезпечує безперервний потік даних, що зручно для потокового керування. У стеку BLE акцент зроблено на атрибутивному протоколі GATT (Generic Attribute Profile), який оптимізовано для передачі коротких пакетів

Аналіз літературних джерел свідчить, що для задач керування маніпулятором в межах прямої видимості (до 10-100 метрів) Bluetooth є найбільш раціональним вибором. Версія Bluetooth 5.0 і вище (зокрема 5.2, затверджена у 2020-х роках) забезпечує подвоєну швидкість передачі даних для BLE (до 2 Мбіт/с) та збільшений радіус дії, що нівелює історичне відставання від Wi-Fi у швидкості реакції. Крім того, використання профілю SPP у класичному

Bluetooth або емуляція UART через BLE дозволяє реалізувати прозорий канал зв'язку «точка-точка» без необхідності розгортання додаткового мережевого обладнання, що спрощує архітектуру системи керування [8]. Порівняльний аналіз основних характеристик розглянутих протоколів у контексті робототехніки наведено в таблиці 1.2.

Таблиця 1.2 – Порівняльна характеристика бездротових протоколів для систем керування роботами (складено на основі [9, 10]) та інтерфейс UART для підключення навігаційних приладів.

Характеристика	Wi-Fi (802.11 n/ac)	ZigBee (802.15.4)	Bluetooth Classic (v3.0)	Bluetooth Low Energy (v5.x)
Частотний діапазон	2,4 / 5 ГГц	2,4 ГГц / 868 МГц	2,4 ГГц	2,4 ГГц
Максимальна швидкість	> 150 Мбіт/с	250 кбіт/с	2-3 Мбіт/с	2 Мбіт/с (для v5.0+)
Енергоспоживання	Високе (100-300 мА)	Дуже низьке (< 30 мА)	Середнє (30-50 мА)	Наднизьке (< 15 мА)
Топологія мережі	Зірка (Star)	Комірка (Mesh)	Пікомережа (Piconet)	Зірка, Mesh (опціонально)
Типова затримка (Latency)	Низька (< 10 мс)	Середня (20-50 мс)	Низька (< 10-20 мс)	Дуже низька (< 3-6 мс)
Застосування в роботі	Відеотрансляція, телеметрія великих обсягів	Розумний дім, сенсорні мережі	Потокове аудіо, безперервне керування	Керування, збір даних з датчиків

Обґрунтуванням вибору Bluetooth для розроблюваної системи є також апаратна підтримка цього протоколу сучасними мікроконтролерами (наприклад, ESP32), які інтегрують радіомодуль безпосередньо в чіп. Це дозволяє зменшити габарити контролера маніпулятора та знизити собівартість пристрою. Крім того, нативну підтримку Bluetooth мають практично всі сучасні смартфони та планшети, що дозволяє використовувати їх як термінали керування без додаткових адаптерів, на відміну від протоколів типу ZigBee або специфічних радіоканалів [11].

Узагальнюючи проведений аналіз класифікації та кінематичних схем, можна стверджувати, що сучасний розвиток робототехніки спрямований на створення універсальних, адаптивних систем, здатних ефективно функціонувати

в умовах динамічного виробництва та неструктурованого оточення. Серед розглянутих типів механізмів маніпулятори з послідовною кінематичною структурою, зокрема антропоморфні роботи, залишаються найбільш поширеним рішенням для виконання складних технологічних операцій завдяки оптимальному співвідношенню робочого простору до габаритів та гнучкості позиціонування. Водночас зростання популярності колаборативних роботів та концепції Інтернету речей (IoT) актуалізує проблему переходу від традиційних дротових пультів до бездротових інтерфейсів керування, що дозволяє підвищити мобільність оператора та безпеку експлуатації. Таким чином, вибір антропоморфної схеми як об'єкта керування в даній магістерській роботі є обґрунтованим, оскільки це дозволяє дослідити складні аспекти кінематики в поєднанні з сучасними вимогами до бездротової комунікації, що є критичним для подальшого розвитку сервісної та промислової робототехніки.

1.3 Огляд програмних середовищ для симуляції роботів

Використання комп'ютерного моделювання є невід'ємним етапом розробки сучасних робототехнічних систем, що дозволяє верифікувати алгоритми керування та кінематичні розрахунки без ризику пошкодження дорогого фізичного обладнання. Концепція «цифрового двійника» (Digital Twin), яка лежить в основі сучасних симуляторів, передбачає створення віртуальної копії робота, що взаємодіє з фізично коректним середовищем. Серед широкого спектра програмних засобів лідерство у науковій та індустріальній сферах утримують середовища, сумісні з мета-операційною системою ROS (Robot Operating System), зокрема симулятор Gazebo, а також платформи CoppeliaSim, Webots та комплекс Matlab/Simulink. Структурну схему взаємодії симулятора Gazebo з керуючими нодами ROS, яка стала де-факто стандартом у дослідженнях 2021-2025 років, наведено на рисунку 1.6 [12].

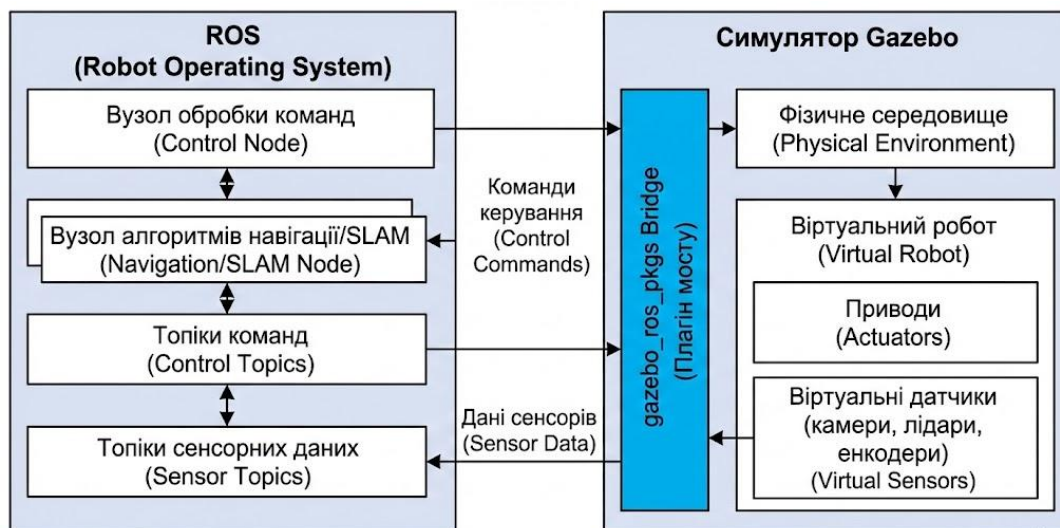


Рисунок 1.6 – Архітектура взаємодії ROS та симулятора Gazebo [12]

Схема ілюструє принцип розділення обчислювальних процесів: ROS виконує роль «мозку» робота, обробляючи команди та сенсорні дані через систему нодів (Nodes) та топіків (Topics), тоді як Gazebo виступає в ролі фізичного середовища. Показано, як плагін `gazebo_ros_pkgs` забезпечує двосторонній міст (Bridge) для обміну даними: команди керування приводами надходять у симулятор, а стан віртуальних датчиків (камери, лідари, енкодери) повертається у ROS для подальшої обробки алгоритмами навігації або SLAM.

Поряд із Gazebo, значною популярністю користується середовище CoppeliaSim (раніше відоме як V-REP), яке вирізняється своєю універсальністю та підтримкою кількох фізичних рушіїв (Bullet, ODE, Newton, Vortex), що дозволяє розробнику обирати оптимальний баланс між швидкістю та точністю розрахунків. CoppeliaSim пропонує гнучку систему скриптів на мові Lua для кожного об'єкта сцени, що спрощує моделювання розподілених систем, однак його інтеграція з ROS може бути складнішою порівняно з нативною підтримкою у Gazebo. Іншим потужним інструментом є Webots – симулятор з відкритим вихідним кодом (з 2018 року), який здобув прихильність завдяки високій продуктивності та простоті створення світів. Він забезпечує реалістичну симуляцію сенсорів і часто використовується для тестування алгоритмів

штучного інтелекту в робототехніці, оскільки споживає менше обчислювальних ресурсів, ніж Gazebo при моделюванні великих груп роботів [13].

Окрему нішу займає програмний комплекс Matlab/Simulink, який орієнтований на «модельно-орієнтоване проектування» (Model-Based Design). На відміну від ігрових рушіїв Gazebo чи Webots, Simulink дозволяє будувати модель робота на рівні блоків диференціальних рівнянь та передатних функцій, що є критично важливим для розробки низькорівневих регуляторів (PID, LQR, MPC). Приклад реалізації моделі динаміки маніпулятора у середовищі Simscape Multibody, що входить до складу екосистеми Matlab, зображено на рисунку 1.7. Такий підхід дозволяє автоматично генерувати C++ код для мікроконтролерів безпосередньо з моделі.

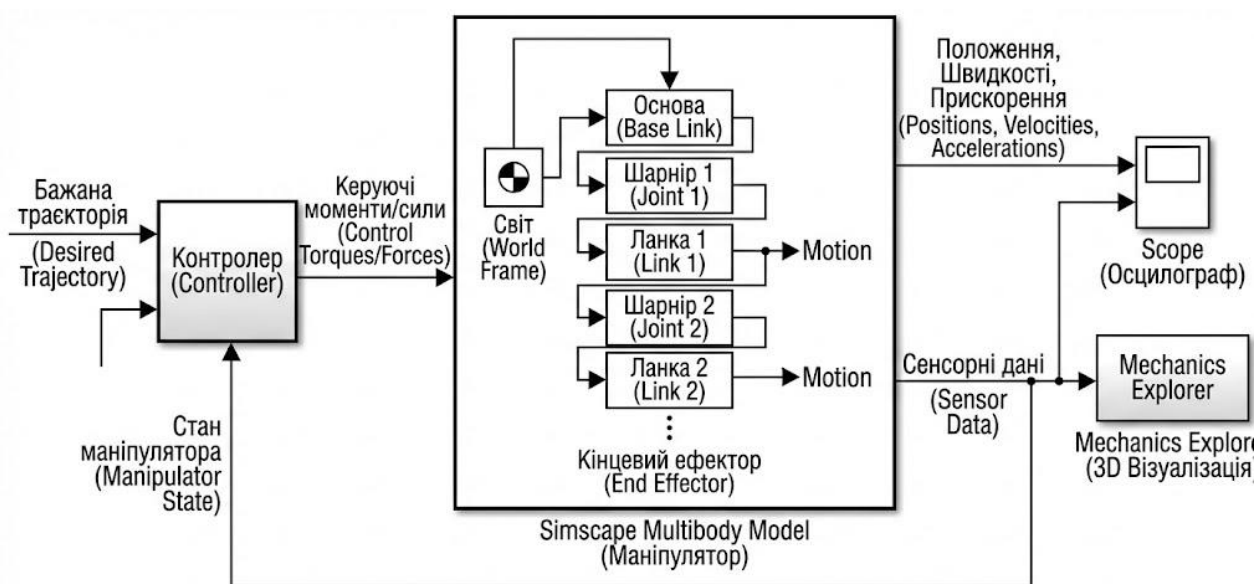


Рисунок 1.7 – Блок-схема моделювання динаміки маніпулятора в середовищі Matlab Simscape Multibody

На рисунку 1.7 представлено візуальне програмування фізичної моделі робота. Замість написання коду, розробник з'єднує блоки, що представляють тверді тіла (Solid), шарніри (Joints) та системи координат (Frames). Показано, як блоки «Revolute Joint» задають кінематичні обмеження обертання, а лінії зв'язку передають фізичні сигнали (крутний момент, швидкість, положення). Таке

представлення дозволяє наочно бачити структуру механізму та миттєво отримувати графіки перехідних процесів у кожному вузлі.

Вибір конкретного середовища залежить від поставленої задачі: для перевірки алгоритмів комп'ютерного зору та навігації в складних просторах краще підходять Gazebo або Webots, тоді як для глибокого аналізу теорії керування та динаміки приводів доцільно використовувати Matlab. У контексті даної роботи, де акцент робиться на інтеграції з реальними протоколами зв'язку (Bluetooth) та подальшому перенесенні коду на мікроконтролер, важливим є наявність розвиненого API та спільноти. Порівняльний аналіз ключових характеристик розглянутих симуляторів наведено в таблиці 1.3, що обґрунтовує вибір інструментарію для практичної частини дослідження [14, 15].

Таблиця 1.3 – Порівняльна характеристика середовищ симуляції для робототехніки [16]

Характеристика	ROS / Gazebo	CoppeliaSim (V-REP)	Webots	Matlab / Simulink
Ліцензія	Open Source (Apache 2.0)	Комерційна / Edu (Free)	Open Source (Apache 2.0)	Комерційна (Proprietary)
Основна мова	C++, Python	Lua, C++, Python	C++, Python, Java	Matlab, Simulink, C
Фізичні рушії	ODE, Bullet, Simbody, DART	Bullet, ODE, Newton, Vortex	ODE (кастомізований)	Simscape Multibody
Якість рендеру	Середня (OGRE)	Середня / Висока	Висока (OpenGL)	Низька (схематична 3D)
Складність вивчення	Висока	Середня	Низька	Середня (вимагає теорії)
Призначення	Дослідження, R&D, ROS-сумісність	Освіта, складні механізми	ШІ, мобільні роботи, рої	Теорія керування, DSP

1.4 Аналіз проблеми затримок та втрати пакетів у бездротових мережах керування

Впровадження бездротових каналів зв'язку в контур керування робототехнічними системами, попри очевидні переваги в мобільності, вносить низку специфічних проблем, які класифікуються в теорії автоматичного

керування як задачі мережевих систем керування (Networked Control Systems – NCS). Основною проблемою є стохастична природа передачі даних, яка призводить до виникнення змінних затримок (latency), джиттера (варіації часу прибуття пакетів) та втрати пакетів інформації. На відміну від дротових з'єднань, де час передачі сигналу є детермінованим і знехтувано малим, у бездротових мережах Bluetooth затримка складається з часу обробки стеку протоколів, часу передачі в ефірі та часу на повторну передачу (retransmission) у випадку помилки. Згідно з дослідженнями 2024 року, критичним для стійкості замкненої системи керування є те, що сумарна затримка може перевищувати період дискретизації контролера, що порушує умови теореми Котельникова-Шеннона та може призвести до втрати стійкості системи.

Структурну схему контуру керування з урахуванням мережевих затримок наведено на рисунку 1.8 [17].

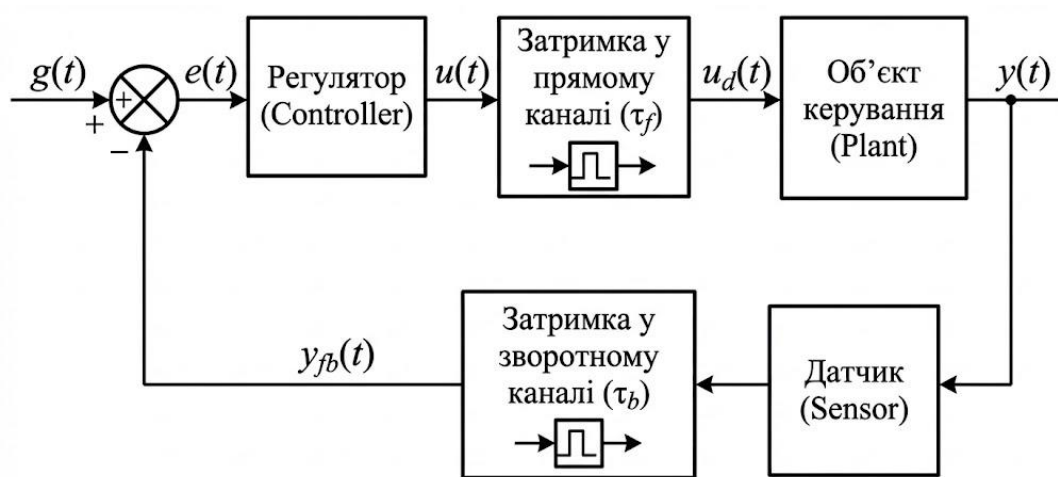


Рисунок 1.8 – Структурна схема системи керування з урахуванням затримок у прямому та зворотному каналах зв'язку

На схемі зображено класичний контур зворотного зв'язку, розірваний бездротовою мережею. Ця модель є базовою для математичного опису проблеми, оскільки сума цих затримок формує «мертвий час» системи, протягом якого контролер не має актуальної інформації про стан об'єкта. Найбільшу небезпеку для точності позиціонування маніпулятора становить явище джиттера –

непередбачуваної зміни затримки. У зашумленому радіоефірі діапазону 2,4 ГГц, де працює Bluetooth, інтерференція від Wi-Fi мереж та побутових приладів змушує протокол виконувати багаторазові повторні відправлення пакету до отримання підтвердження (ACK). Це призводить до ситуації, коли пакети надходять до роботи нерівномірно або з великими паузами. З точки зору теорії керування, це рівнозначно введенню в систему ланки чистого запізнення зі змінними параметрами, що суттєво зменшує запас стійкості по фазі. Вплив часової затримки на динаміку перехідного процесу при подачі ступінчастого впливу (команди на переміщення) проілюстровано на рисунку 1.9.

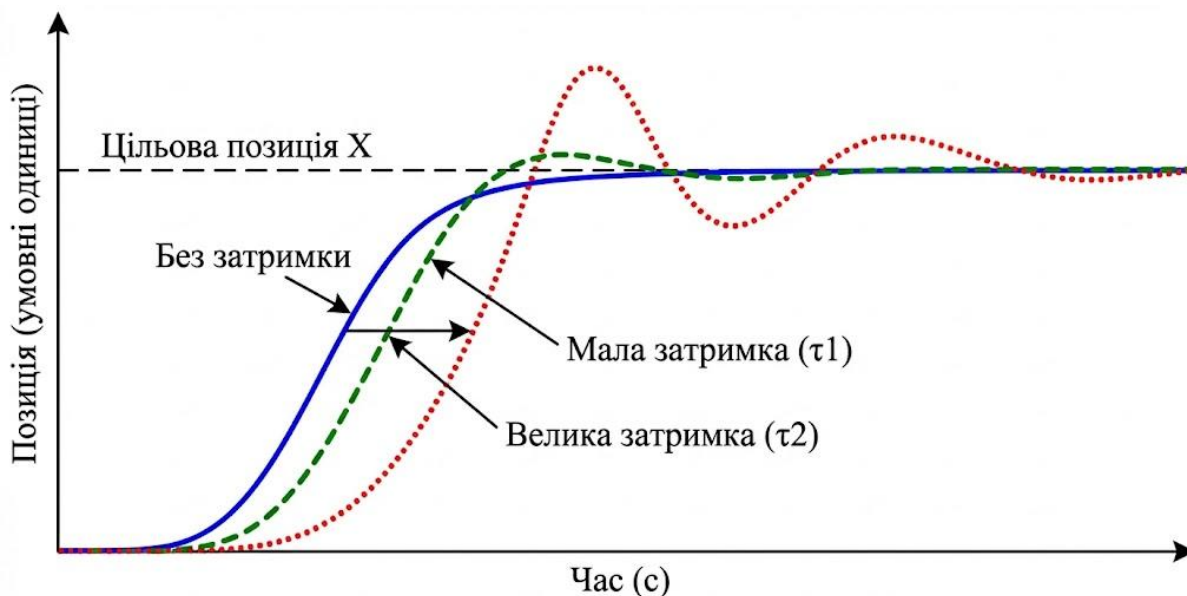


Рисунок 1.9 – Вплив часової затримки на перехідний процес системи керування [18]

Графік демонструє порівняння реакції системи на команду «переміститися в позицію X» (Step Response). Синя лінія показує ідеальну реакцію системи без затримок: плавний вихід на задане значення без перерегулювання. Червона лінія ілюструє поведінку тієї ж системи при наявності значної затримки у каналі зворотного зв'язку. Видно появу коливань (осциляцій), значне перерегулювання та збільшення часу встановлення. У критичних випадках, як показано

пунктиром, амплітуда коливань може зростати, що означає повну втрату керованості та аварійну ситуацію.

Третім критичним фактором є втрата пакетів (Packet Loss), яка виникає, коли рівень завад перевищує можливості корекції помилок протоколу. Для маніпулятора це створює ефект «заморожування» останньої отриманої команди. Якщо робот виконував рух із певною швидкістю і команда зупинки була втрачена, механізм продовжить рух за інерцією або згідно з попередньою командою, що може призвести до зіткнення з перешкодою. Сучасні методи боротьби з цим явищем включають використання алгоритмів передбачення (наприклад, фільтр Калмана або спостерігач Люенбергера) на стороні виконавчого механізму, які дозволяють екстраполювати траєкторію руху під час короткочасних обривів зв'язку. Також обов'язковим є впровадження механізму «Watchdog Timer» (сторожового таймера), який автоматично зупиняє приводи, якщо новий пакет даних не отримано протягом критичного інтервалу часу (зазвичай 100-500 мс) [18].

Аналіз літератури свідчить, що стандартні протоколи Bluetooth Low Energy (BLE) мають вбудовані механізми адаптивної зміни частоти (Frequency Hopping), які частково нівелюють вплив завад, проте не гарантують жорсткого реального часу (Hard Real-Time). Тому при розробці системи керування необхідно програмно реалізувати буферування команд та перевірку цілісності даних на прикладному рівні. Дослідження 2023 року пропонують використовувати гібридні підходи, де критичні команди дублюються або передаються з підвищеним пріоритетом, а менш важливі телеметричні дані можуть бути пропущені задля збереження пропускну здатності каналу [19, 20].

В даному розділі кваліфікаційної роботи проведено комплексний аналіз сучасного стану та перспектив розвитку систем бездротового керування робототехнічними комплексами. За результатами огляду кінематичних схем встановлено, що для виконання багатофункціональних операцій у неструктурованому середовищі найбільш універсальним рішенням є маніпулятори з антропоморфною структурою. Вони забезпечують оптимальне

співвідношення робочого простору до габаритів механізму та високу гнучкість позиціонування виконавчого органу, хоча й вимагають складніших алгоритмів розрахунку оберненої кінематики порівняно з системами декартового типу або SCARA.

Порівняльний аналіз бездротових інтерфейсів передачі даних дозволив визначити протокол Bluetooth (зокрема специфікації 4.0 і вище) як пріоритетний для реалізації каналу зв'язку «оператор-маніпулятор» на малих відстанях. На відміну від Wi-Fi, який є надлишковим за енергоспоживанням, та ZigBee, що має обмежену пропускну здатність, Bluetooth забезпечує необхідний баланс між швидкістю передачі керуючих команд, енергоефективністю та простотою інтеграції з мобільними терміналами. Водночас виявлено, що використання радіоканалу в контурі керування вносить стохастичні збурення у вигляді змінних у часі затримок та можливої втрати пакетів даних, що критично впливає на стійкість замкненої системи та точність відпрацювання траєкторій.

Обґрунтовано необхідність попереднього етапу цифрового моделювання системи перед фізичною реалізацією. Серед розглянутих програмних середовищ найбільш ефективним інструментарієм визначено комбінацію мета-операційної системи ROS та симулятора Gazebo. Це середовище дозволяє створити повноцінний цифровий двійник маніпулятора з урахуванням фізичних характеристик ланок та приводів, а також програмно емулювати недоліки каналу зв'язку. На основі проведеного аналізу сформульовано постановку задачі дослідження, яка полягає у розробці системи керування, стійкої до мережевих затримок, з обов'язковою верифікацією запропонованих алгоритмів у віртуальному середовищі, що стане предметом наступних розділів роботи.

РОЗДІЛ 2

ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА РОЗРОБКА МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

2.1 Розробка кінематичної моделі маніпулятора:

Фундаментальною основою для реалізації будь-якого алгоритму керування рухом роботизованого маніпулятора є побудова його кінематичної моделі, яка встановлює математичний взаємозв'язок між просторовими координатами робочого органу (схвата) та узагальненими координатами (кутами повороту) у шарнірах. Для опису геометрії багатоланкових механізмів у сучасній робототехніці загальноприйнятим стандартом є використання нотації Денавіта-Хартенберга (D-H параметрів). Цей метод дозволяє уніфікувати опис кінематичного ланцюга шляхом прив'язки локальних систем координат до кожної ланки робота та визначення чотирьох параметрів перетворення між ними: довжини ланки (a), кручення ланки (α), зміщення ланки (d) та кута повороту шарніра (θ). Графічну інтерпретацію параметрів Денавіта-Хартенберга для двох суміжних ланок маніпулятора наведено на рисунку 2.1 [21].

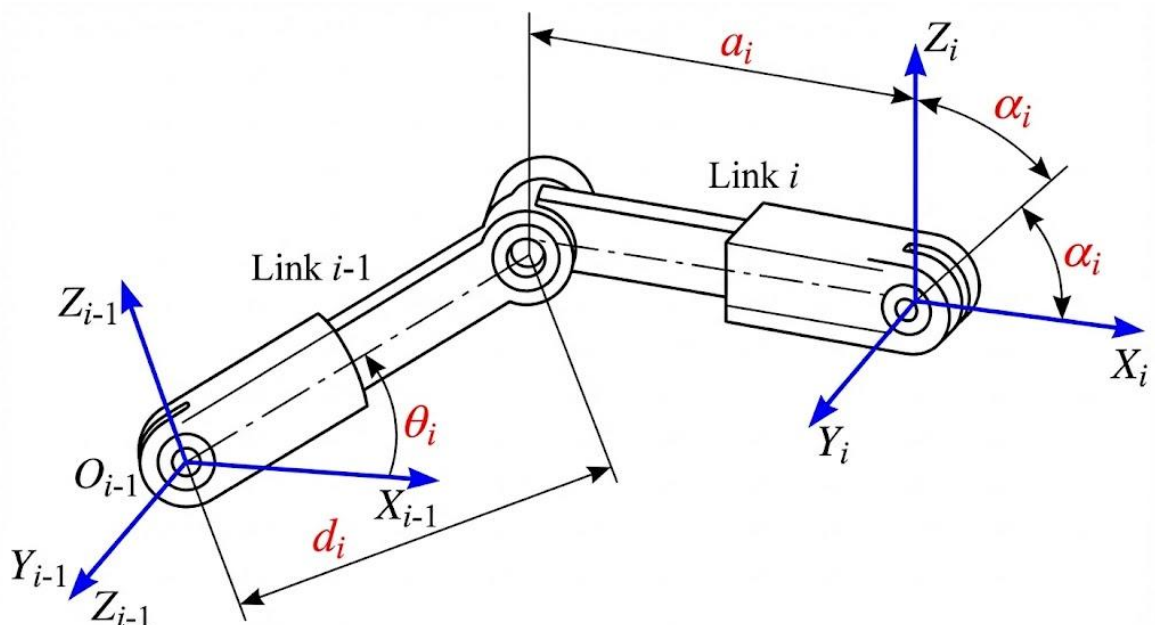


Рисунок 2.1 – Геометричний зміст параметрів Денавіта-Хартенберга для опису кінематики [21]

На схемі зображено дві послідовні ланки маніпулятора ($i - 1$ та i) та пов'язані з ними системи координат. Показано фізичний зміст чотирьох параметрів D-H: θ_i – кут обертання навколо осі Z_{i-1} ; d_i – відстань уздовж осі Z_{i-1} ; a_i – довжина спільної нормалі між осями Z ; α_i – кут між осями Z_{i-1} та Z_i навколо осі X_i . Правильне визначення цих параметрів є критичним етапом, оскільки помилка в одному з них призведе до некоректного позиціонування всієї системи.

Пряма задача кінематики полягає у визначенні положення та орієнтації кінцевого ефектора в декартовому просторі на основі відомих значень кутів поворотів сервоприводів. Математично це реалізується шляхом послідовного перемноження матриць однорідних перетворень розмірністю 4×4 , кожна з яких описує перехід від базової системи координат до наступної. Для шестиосьового маніпулятора результуюча матриця трансформації T формується як добуток шести індивідуальних матриць переходу. Ця матриця містить вектор положення (x, y, z) та матрицю обертання, що визначає орієнтацію інструменту. Розподіл систем координат для розроблюваного антропоморфного маніпулятора, який є основою для складання матриць перетворень, проілюстровано на рисунку 2.2 [22].

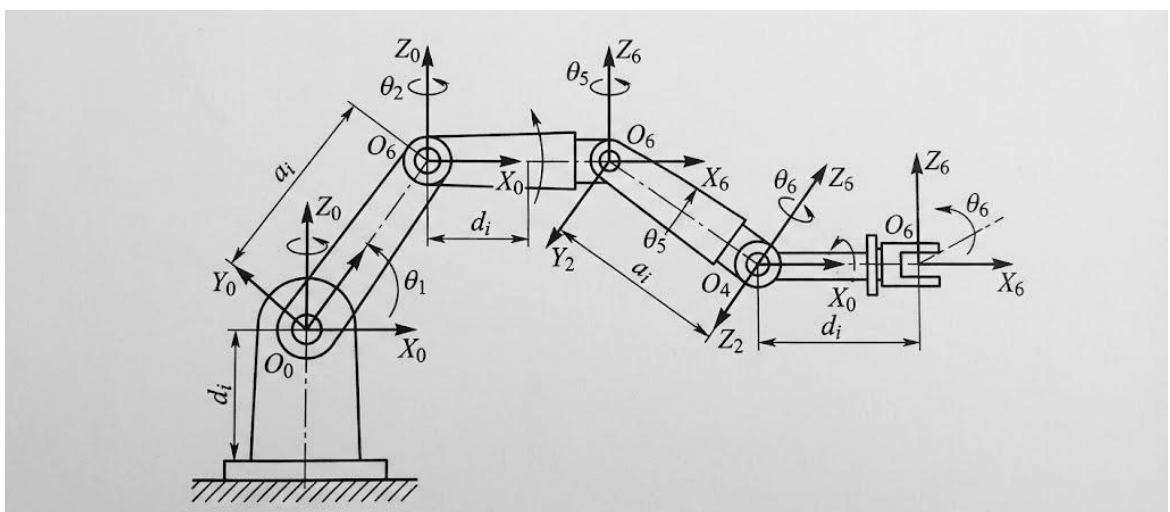


Рисунок 2.2 – Призначення систем координат ланок 6-осьового маніпулятора згідно з нотацією D-H [22]

Значно складнішою та важливішою для практичного керування є обернена задача кінематики (Inverse Kinematics – ІК), яка полягає у розрахунку необхідних кутів поворотів шарнірів для досягнення заданої точки у просторі (x, y, z) з певною орієнтацією. На відміну від прямої задачі, яка завжди має єдине рішення, обернена задача може мати декілька розв’язків (для антропоморфного робота зазвичай до 8 конфігурацій для однієї точки) або не мати їх зовсім, якщо точка знаходиться поза межами робочої зони. Для реалізації на мікроконтролері доцільно використовувати аналітичний метод розв’язання, який базується на геометричній декомпозиції робота на плоскі трикутники, що дозволяє уникнути ітераційних обчислень і забезпечити швидкодію в реальному часі. Принцип геометричного розв’язання оберненої задачі для перших трьох ступенів вільності, що відповідають за позиціонування, показано на рисунку 2.3 [23].

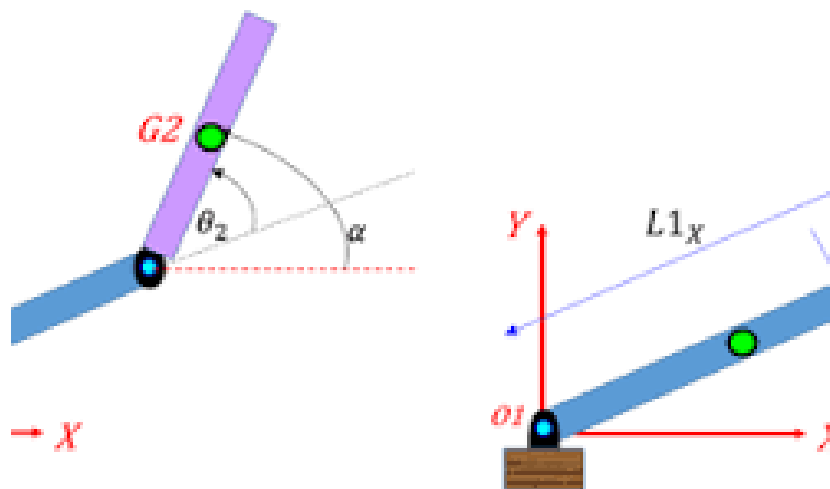


Рисунок 2.3 – Геометричний метод розв’язання оберненої задачі кінематики для плеча маніпулятора [23]

На схемі показано проекцію маніпулятора на площину. Задача зводиться до знаходження кутів $\theta_1, \theta_2, \theta_3$ через координати цільової точки $P(x, y, z)$. Використання теореми косинусів та тригонометричних функцій

(atan2) дозволяє аналітично виразити кути через відомі довжини ланок L_1, L_2 . Цей підхід є менш ресурсоємним для обчислювального ядра ESP32 порівняно з чисельними методами (наприклад, методом Якобіана), що є критичним для забезпечення плавності руху при керуванні через Bluetooth.

Окрім розрахунку кутів, при проектуванні кінематичної моделі необхідно враховувати сингулярні конфігурації – положення, в яких маніпулятор втрачає один або більше ступенів вільності (наприклад, коли осі двох шарнірів співпадають). У таких точках спроба руху в певному напрямку може вимагати нескінченно великих швидкостей приводів, що призводить до аварійних зупинок. Аналіз якобіана системи дозволяє виявити такі зони та програмно обмежити траєкторії руху, що проходять поблизу сингулярностей. Розроблені в цьому розділі математичні алгоритми прямої та оберненої кінематики стануть програмним ядром прошивки контролера у наступних етапах роботи [24].

2.2 Архітектура системи керування: структурна схема та вибір апаратної бази

Проектування системи керування багатофункціональним маніпулятором вимагає комплексного підходу до побудови архітектури, яка забезпечує узгоджену взаємодію між програмними алгоритмами верхнього рівня та виконавчими механізмами. Розроблена архітектура базується на дворівневій ієрархічній моделі, де верхній рівень (Master) представлений клієнтським терміналом – смартфоном або персональним комп'ютером, що формує цільові завдання, а нижній рівень (Slave) реалізовано на базі вбудованої мікропроцесорної системи, яка відповідає за безпосереднє керування приводами та обробку сенсорних даних. Ключовим елементом, що об'єднує ці рівні, є бездротовий канал Bluetooth, який накладає специфічні вимоги до організації потоків даних. Загальну структурну схему розробленої системи керування наведено на рисунку 2.4 [25].

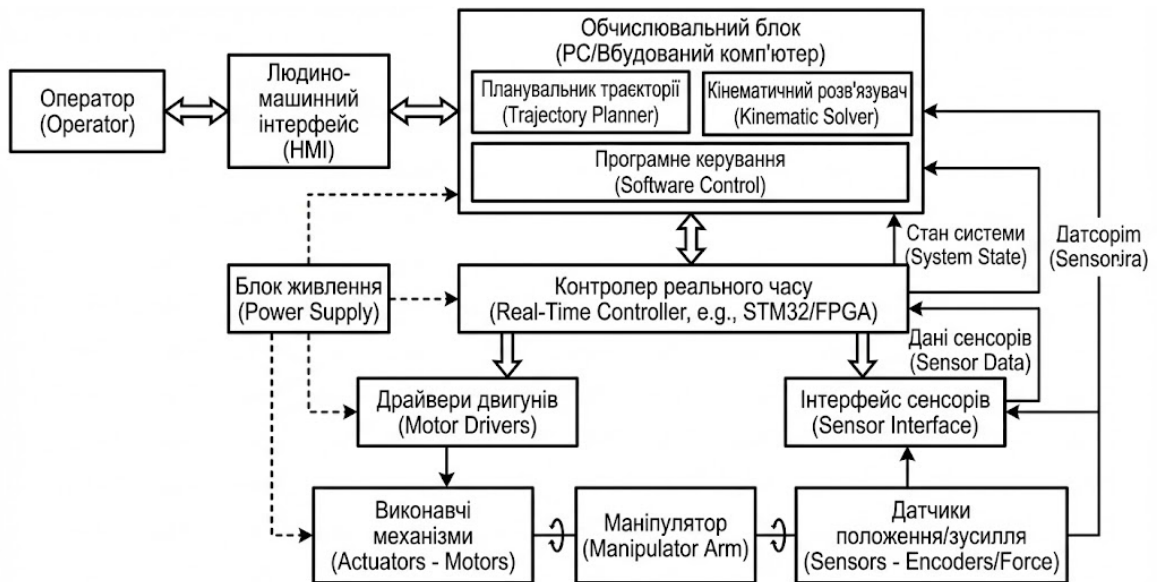


Рисунок 2.4 – Структурна схема апаратно-програмного комплексу керування маніпулятором

Схема ілюструє інформаційні та енергетичні зв'язки між компонентами системи. Центральним вузлом є мікроконтролер (MCU), який приймає пакети даних через Bluetooth-інтерфейс від мобільного додатку. Вихідні сигнали мікроконтролера у форматі широтно-імпульсної модуляції (ШИМ/PWM) надходять до сервоприводів, що забезпечують рух ланок. Окремо виділено підсистему живлення, яка розділяє ланцюги логіки та силових приводів для запобігання завадам. Також показано блок зворотного зв'язку (енкодери або потенціометри), який замикає контур регулювання положення.

Вибір обчислювального ядра системи обумовлений необхідністю виконання складних математичних розрахунків оберненої кінематики в реальному часі паралельно з підтримкою стеку протоколів Bluetooth. На основі порівняльного аналізу сучасних мікроконтролерів, пріоритетним рішенням обрано систему на кристалі (SoC) ESP32. Цей двоядерний мікроконтролер з архітектурою Xtensa LX6 має тактову частоту до 240 МГц, що дозволяє виділити одне ядро виключно для обслуговування бездротового зв'язку, а інше – для розрахунку траєкторій та генерації керуючих сигналів. Важливою перевагою ESP32 є наявність вбудованого модуля Bluetooth 4.2/BLE, що усуває потребу в

зовнішніх адаптерах типу HC-05 та підвищує надійність системи. Схему електричних з'єднань компонентів системи на базі ESP32 представлено на рисунку 2.5 [26].

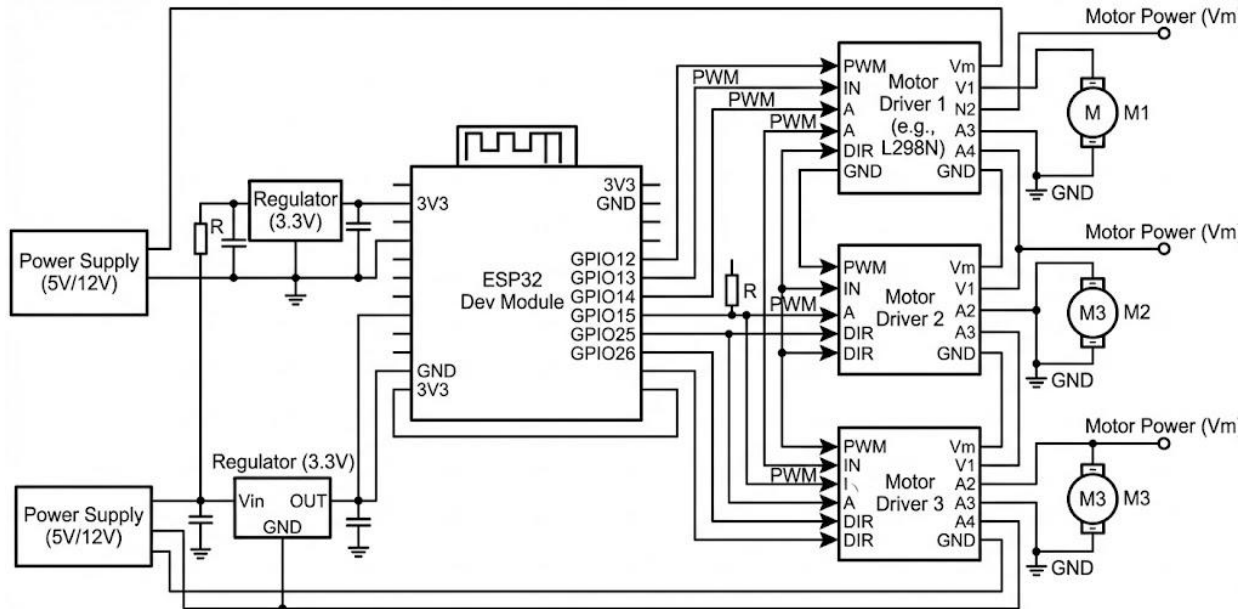


Рисунок 2.5 – Схема електрична принципова підключення виконавчих механізмів до контролера ESP32 [26]

На рисунку деталізовано комутацію портів введення-виведення (GPIO). Показано підключення сигнальних ліній сервоприводів до виходів контролера, що підтримують апаратну генерацію ШІМ. Для розширення кількості каналів керування (у випадку використання понад 6 приводів) передбачено використання I2C-драйвера PCA9685, який також виконує роль буфера, захищаючи порти мікроконтролера від зворотних струмів. Окремі лінії виділено для інтерфейсу UART, який використовується для налагодження та діагностики системи.

Критичним аспектом надійності роботи маніпулятора є організація підсистеми живлення. Сервоприводи під навантаженням генерують значні імпульсні завади та просадки напруги, що може призвести до перезавантаження мікроконтролера. Для вирішення цієї проблеми розроблено схему з гальванічною розв'язкою або розділеними контурами живлення. Силова частина

живиться від потужного DC-DC перетворювача (Buck converter), здатного видавати струм 5-10 А при стабільній напрузі (зазвичай 5В або 6В для сервомоторів), тоді як логічна частина отримує живлення через окремий лінійний стабілізатор. Принцип організації стабільного живлення для роботизованих систем показано на рисунку 2.6 [27].

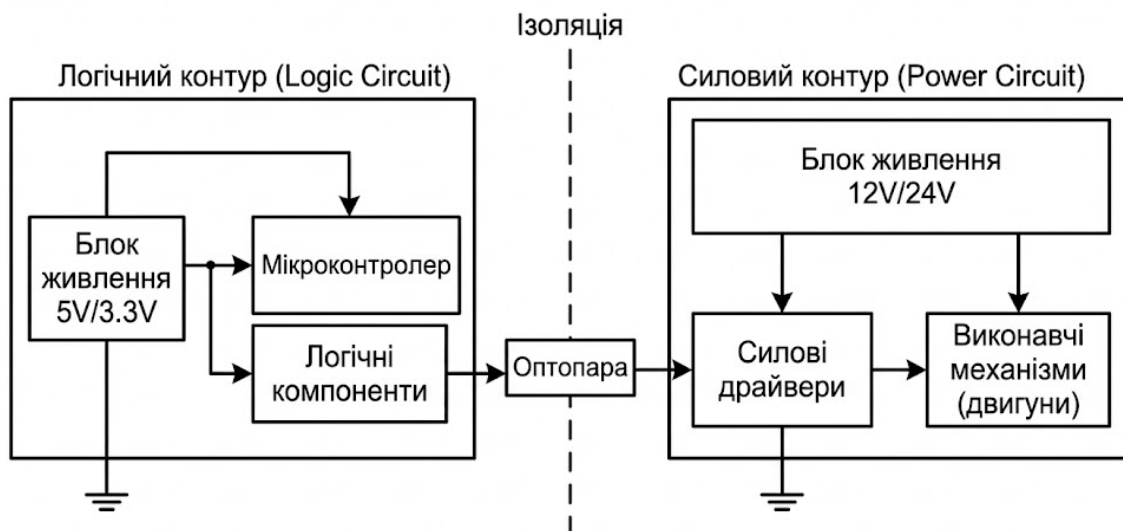


Рисунок 2.6 – Схема розподілу живлення з розділенням логічного та силового контурів

Схема демонструє використання загального джерела живлення (акумулятора або блоку живлення) з подальшим розгалуженням. Показано застосування електролітичних конденсаторів великої ємності безпосередньо біля клем живлення сервоприводів для компенсації пікових струмів під час старту двигунів. Таке рішення мінімізує вплив електромагнітних завад на роботу модуля Bluetooth та забезпечує стабільність з'єднання навіть при різких маневрах маніпулятора.

Програмна архітектура нижнього рівня побудована на базі операційної системи реального часу FreeRTOS, яка інтегрована в середовище розробки ESP-IDF. Це дозволяє реалізувати багатозадачність, де задача прийому даних по Bluetooth має найвищий пріоритет, а задача оновлення положення сервоприводів виконується з фіксованим періодом (наприклад, 20 мс). Такий підхід гарантує

детермінованість керування та дозволяє імплементувати механізми безпеки (Watchdog), які автоматично переводять робота в безпечний стан у разі втрати зв'язку, що детальніше буде розглянуто в наступних підрозділах [28].

2.3 Розробка протоколу обміну даними через Bluetooth

Надійність дистанційного керування роботизованим маніпулятором критично залежить від стабільності та захищеності каналу передачі даних. Оскільки стандартний профіль Bluetooth (SPP або BLE GATT) забезпечує лише транспортний рівень доставки байтів, для коректної інтерпретації команд мікроконтролером необхідно розробити власний протокол прикладного рівня. Основною вимогою до такого протоколу є мінімізація накладних витрат (overhead) при збереженні інформативності, що дозволяє досягти низької латентності. Розроблений протокол базується на пакетній передачі даних з фіксованою структурою кадру, що дозволяє однозначно ідентифікувати початок і кінець повідомлення в безперервному потоці байтів. Структуру кадру даних, прийнятну для системи керування, наведено на рисунку 2.7 [29].



Рисунок 2.7 – Структура кадру даних розробленого протоколу обміну [29]

На схемі візуалізовано послідовність полів пакету. Кадр починається зі стартового байта (Header/Preamble, наприклад 0xAA), який слугує маркером початку повідомлення для парсера. Далі слідує байт довжини (Length), що вказує кількість байтів у корисному навантаженні, та ідентифікатор команди (Command ID), який визначає тип дії (рух, зупинка, опитування сенсорів). Поле даних (Payload) містить параметри команди, наприклад, кути поворотів для кожного шарніра. Завершують кадр контрольна сума (Checksum/CRC) для перевірки цілісності та стоп-байт, що гарантує коректне розпізнавання кінця пакету.

Ключовим елементом безпеки протоколу є механізм перевірки цілісності даних. У зашумленому радіоефірі діапазону 2,4 ГГц існує висока ймовірність спотворення окремих бітів, що може призвести до інтерпретації помилкових даних як валідних команд руху, викликаючи непередбачувану поведінку робота. Для захисту від таких ситуацій у протокол впроваджено перевірку за допомогою циклічного надлишкового коду (CRC – Cyclic Redundancy Check) або, у спрощеному варіанті, контрольної суми. Алгоритм роботи приймача полягає в обчисленні CRC отриманого «тіла» пакету та порівнянні його з значенням, переданим у кінці кадру. Якщо значення не співпадають, пакет вважається пошкодженим і відкидається без виконання. Логіку обробки вхідного потоку даних на стороні мікроконтролера, реалізовану на основі скінченного автомату (Finite State Machine), зображено на рисунку 2.8.

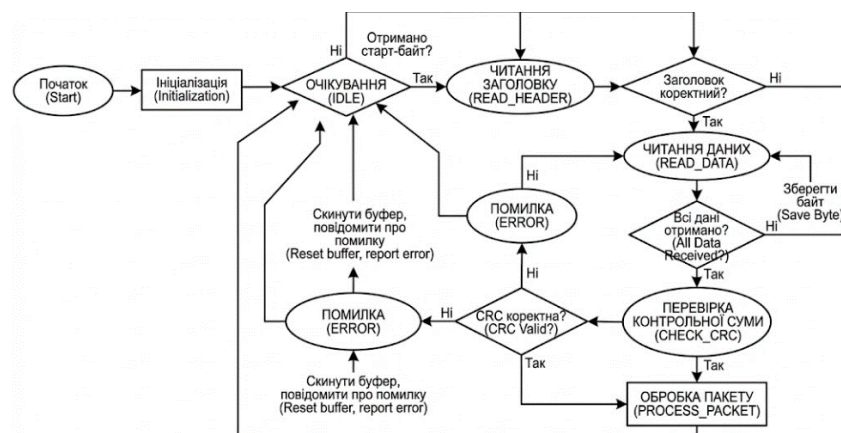


Рисунок 2.8 – Блок-схема алгоритму парсингу вхідних даних на основі скінченного автомату

Діаграма ілюструє процес побайтної обробки вхідного буфера. Система перебуває в стані очікування (IDLE) до отримання стартового байта. Після його детекції автомат переходить до послідовного зчитування довжини, команди та даних. Важливою особливістю є те, що такий підхід є неблокуючим: мікроконтролер не «зависає» в очікуванні повного пакету, а обробляє байти по мірі їх надходження, що дозволяє паралельно виконувати задачі керування двигунами. У випадку помилки на будь-якому етапі (наприклад, невірна довжина або CRC) автомат миттєво скидається в початковий стан.

Для забезпечення зворотного зв'язку протокол реалізує модель «запит-відповідь» (Request-Response) або асинхронну відправку телеметрії. Це означає, що після успішного отримання та валідації критичної команди (наприклад, зміна режиму роботи) контролер надсилає пакет-підтвердження (АСК). Крім того, з певною періодичністю (наприклад, 10 Гц) робот відправляє пакети зі станом сенсорів та поточною позицією, що дозволяє відображати реальний стан маніпулятора в інтерфейсі користувача. Діаграму послідовності обміну повідомленнями при виконанні команди переміщення наведено на рисунку 2.9. Такий підхід гарантує, що оператор завжди поінформований про те, чи була команда прийнята до виконання [30, 31].

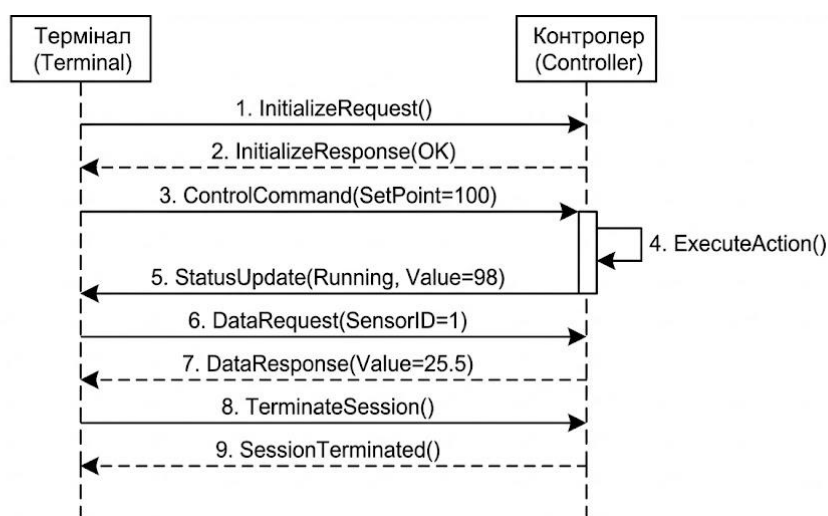


Рисунок 2.9 – Діаграма послідовності обміну повідомленнями між терміналом та контролером [31]

На схемі показано часову діаграму взаємодії. Термінал (Smartphone) формує та відправляє пакет з командою руху (CMD_MOVE). Контролер (Robot MCU) приймає пакет, перевіряє CRC і, якщо пакет валідний, відправляє підтвердження (ACK). Тільки після відправки ACK починається фізичне виконання руху. У випадку, якщо термінал не отримав ACK протягом заданого тайм-ауту, ініціюється повторна передача команди (Retransmission). Це забезпечує надійність доставки команд навіть при короткочасних обривах зв'язку. досліджень ефективності системи в реальних умовах експлуатації.

2.4 Розробка клієнтського програмного забезпечення для надсилання керуючих команд

Ефективність експлуатації дистанційно керованої роботизованої системи значною мірою залежить від ергономіки та функціональності клієнтського програмного забезпечення, яке виступає в ролі людино-машинного інтерфейсу (НМІ). Головним завданням клієнтського додатку є перетворення намірів оператора (рухів пальців по екрану або натискання клавіш) у послідовність команд розробленого протоколу, а також візуалізація зворотного зв'язку від робота. Враховуючи обраний канал зв'язку Bluetooth, найбільш доцільним є створення мобільного додатку для операційної системи Android, що забезпечує високу мобільність та доступ до апаратних ресурсів смартфона (сенсорного екрану, акселерометра). При проектуванні графічного інтерфейсу користувача (GUI) застосовано модульний принцип, де кожному елементу керування присвоєно унікальний ідентифікатор команди та визначено діапазон допустимих значень, що деталізовано в таблиці 2.1.

Завдяки цьому програмний код обробки подій (Event Handlers) стає уніфікованим: кожен віджет викликає одну й ту саму функцію формування пакету, підставляючи лише свій унікальний ID та поточне значення. Це мінімізує ймовірність програмних помилок при розширенні функціоналу додатку .

Таблиця 2.1 – Функціональна специфікація елементів інтерфейсу та їх прив'язка до команд протоколу

Елемент інтерфейсу (UI Widget)	Функціональне призначення	Тип події (Event)	Команда протоколу (ID)	Діапазон значень (Payload)
Слайдер (Slider) 1-6	Індивідуальне керування кутом повороту кожного шарніра (FK)	OnValueChanged	0x10 - 0x16	0...180 (градуси)
Джойстик (Virtual Joystick)	Керування координатами схвата у площині X-Y (ІК)	OnMove	0x20	-100...+100 (вектор зміщення)
Кнопка «Connect»	Ініціалізація пошуку та підключення до Bluetooth пристрою	OnClick	-	MAC-адреса пристрою
Кнопка «STOP»	Аварійна зупинка всіх приводів (найвищий пріоритет)	OnClick	0xFF	0 (без параметрів)
Текстове поле (Log)	Відображення вхідних даних телеметрії та налагоджувальної інформації	OnReceive	-	Текстовий рядок (String)

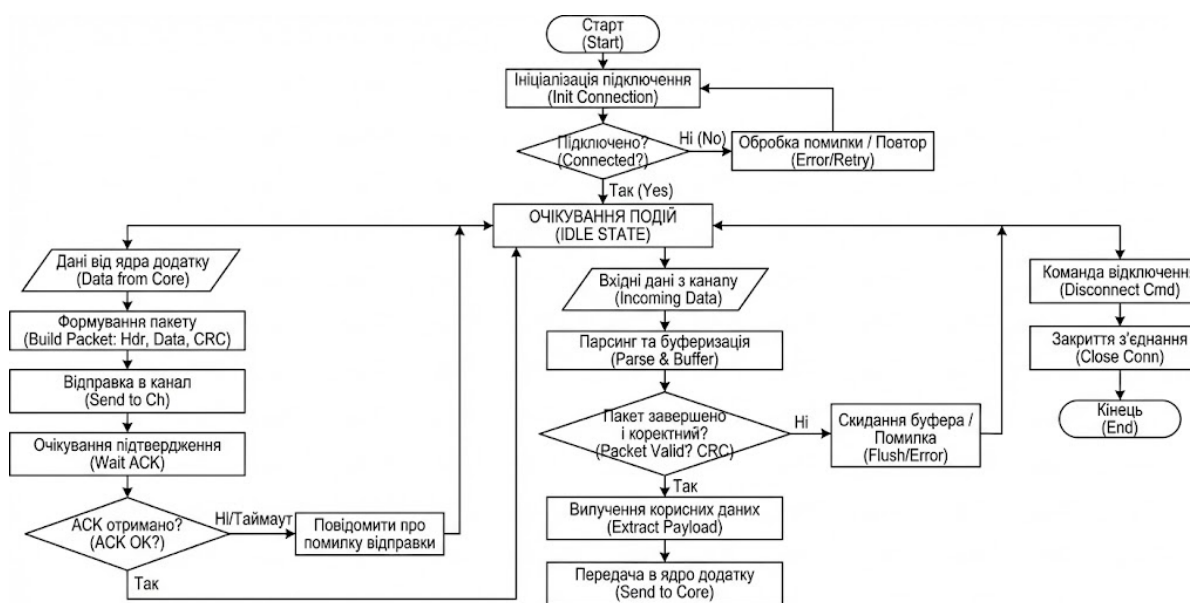
Графічна реалізація описаних функцій вимагає інтуїтивного розміщення елементів на екрані, щоб мінімізувати час реакції оператора. Приклад розробленого макету додатку наведено на рисунку 2.10 [32].



Рисунок 2.10 – Графічний інтерфейс мобільного додатку для керування маніпулятором

На рисунку 2.10 представлено макет головного екрану (Dashboard). У верхній частині розміщено панель стану з'єднання, яка відображає статус Bluetooth (Connected/Disconnected) та рівень сигналу RSSI. Центральну частину займають віртуальні слайдери (повзунки) для індивідуального керування кожним сервоприводом, що дозволяє точно налаштувати положення робота в режимі налагодження. Знизу розташовано джойстики для керування положенням схвата у декартовому просторі (X, Y) та кнопка аварійної зупинки (Emergency Stop), яка має пріоритетний розмір та колір для швидкої реакції у нештатних ситуаціях.

Програмна архітектура клієнтського додатку базується на подіє-орієнтованій моделі (Event-Driven Architecture). Це означає, що відправка даних ініціюється не в циклі, а при виникненні певної події: зміни положення слайдера, натискання кнопки або отримання вхідного пакету від мікроконтролера. Для роботи з Bluetooth Low Energy (BLE) або Classic використано асинхронні методи, що дозволяє не блокувати основний потік інтерфейсу (UI Thread) під час сканування ефіру або очікування відповіді. Критично важливим алгоритмічним елементом є «обмежувач частоти відправки», робота якого показана на рисунку 2.11 [33].



T

Рисунок 2.11 – Алгоритм роботи комунікаційного модуля клієнтського додатку

Блок-схема ілюструє життєвий цикл з'єднання. Після запуску дода ініціює сканування доступних пристроїв та фільтрує їх за MAC-адресою. Після встановлення з'єднання запускається фоновий сервіс (Listener Service), який постійно прослуховує вхідний потік даних. При отриманні пакету відбувається його парсинг: якщо це пакет телеметрії — оновлюються індикатори на екрані. У разі розриву з'єднання (Connection Lost) алгоритм автоматично намагається відновити зв'язок.

Для забезпечення кросплатформеності та прискорення розробки було проведено порівняльний аналіз сучасних інструментальних засобів, результати якого наведено в таблиці 2.2.

Таблиця 2.2 – Порівняльний аналіз засобів розробки мобільних додатків для керування роботами

Середовище розробки	Мова програмування	Переваги	Недоліки	Доцільність для проекту
Android Studio (Native)	Java / Kotlin	Повний доступ до API, максимальна продуктивність, стабільність роботи з Bluetooth	Високий поріг входження, великий обсяг коду (boilerplate code)	Висока (для складних комерційних проектів)
Flutter	Dart	Кросплатформеність (iOS/Android), швидка розробка UI, "Hot Reload"	Збільшений розмір додатку, можливі проблеми з специфічними бібліотеками Bluetooth	Середня (потребує вивчення Dart)
MIT App Inventor	Visual Blocks	Дуже швидка розробка (Low-code), просте налагодження, готові блоки BluetoothClient	Обмежені можливості UI, складність реалізації складної логіки (математики ІК)	Висока (для прототипування та магістерської роботи)
Qt Mobile	C++	Висока швидкодія, переносимість коду з ПК-версій	Складне ліцензування, великий розмір бібліотек	Низька

На основі цього аналізу для реалізації проекту обрано середовище, яке забезпечує найкращий баланс між простотою розробки інтерфейсу та можливостями роботи з апаратним забезпеченням Bluetooth.

Особливу увагу при розробці приділено обробці виключних ситуацій: додаток повинен коректно обробляти відключення Bluetooth-адаптера на телефоні, вихід робота із зони досяжності та низький заряд батареї. Також реалізовано функцію логування відправлених та отриманих команд у текстовий файл, що є необхідним інструментом для аналізу поведінки системи під час експериментальних досліджень, описаних у наступному розділі [34].

2.5 Алгоритмічне забезпечення траєкторії руху

Забезпечення плавності та точності переміщення робочого органу маніпулятора є критично важливою задачею системи керування, яка виходить за рамки простого розрахунку кінематики. Пряма передача цільових координат від клієнтського додатку на сервоприводи без проміжної обробки призводить до виникнення ударних навантажень на механічну частину, оскільки миттєва зміна завдання еквівалентна спробі розвитку нескінченного прискорення. Для вирішення цієї проблеми застосовують алгоритми планування траєкторії (Trajectory Planning), які генерують часову послідовність проміжних точок між початковим та кінцевим положенням. Найбільш поширеним підходом є використання поліноміальної інтерполяції в просторі приєднаних координат (Joint Space), де для кожного шарніра розраховується закон зміни кута повороту $q(t)$. Базовим методом є використання кубічного поліному, який дозволяє забезпечити неперервність не лише положення, але й швидкості. Графіки зміни кінематичних параметрів при використанні кубічного закону руху наведено на рисунку 2.12 [35].

На рисунку 2.12 представлено три графіки, що характеризують рух одного шарніра: положення $q(t)$, швидкість $\dot{q}(t)$ та прискорення $\ddot{q}(t)$. Видно, що графік положення має форму гладкої S-подібної кривої, що з'єднує початкову та кінцеву точки. Графік швидкості має параболічну форму з плавним зростанням від нуля до максимуму і подальшим спаданням. Однак, графік прискорення (нижній) є лінійною функцією з розривами на початку та в кінці руху, що означає наявність

«ривка» (Jerk) – різкої зміни прискорення, що може викликати вібрації конструкції.

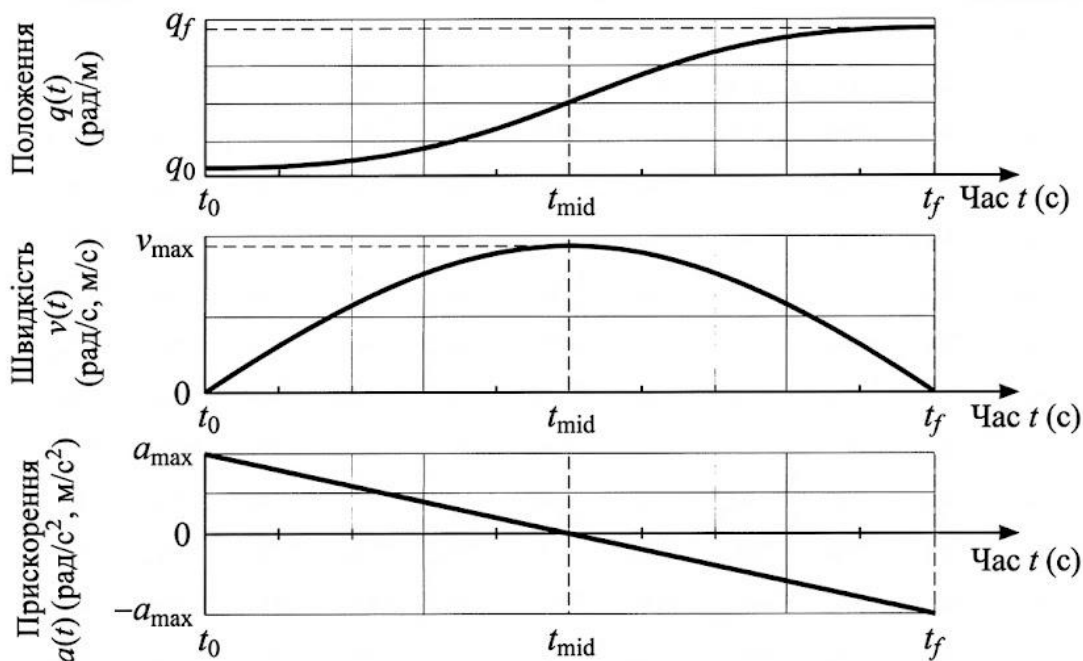


Рисунок 2.12 – Часові діаграми кінематичних параметрів при кубічній інтерполяції руху [35]

.Для усунення недоліків кубічної інтерполяції та зменшення вібрацій у високоточних системах застосовують складніші профілі швидкості, зокрема трапецеїдальний профіль та профіль S-подібної кривої (S-curve). Трапецеїдальний профіль передбачає розбиття руху на три етапи: рівноприскорений розгін, рух зі сталою швидкістю та рівносповільнене гальмування. Це дозволяє обмежити максимальну швидкість та прискорення привода на фізично допустимому рівні. Профіль S-curve є подальшим удосконаленням, де етапи розгону та гальмування згладжуються поліномами вищого порядку (5-го або 7-го), що забезпечує скінченне значення ривка. Порівняння динамічних характеристик трапецеїдального та S-подібного профілів швидкості проілюстровано на рисунку 2.13 [36].

Діаграма наочно демонструє відмінності у формуванні керуючого впливу. Червона лінія (трапеція) показує різкі злами в моменти перемикання режимів, що

теоретично вимагає миттєвої зміни сили струму в обмотках двигуна. Зелена лінія (S-curve) має згладжені кути, що свідчить про плавне наростання прискорення. Хоча реалізація S-curve вимагає більших обчислювальних ресурсів мікроконтролера ESP32, вона суттєво подовжує термін служби редукторів сервоприводів та зменшує амплітуду коливань схвата при зупинці.

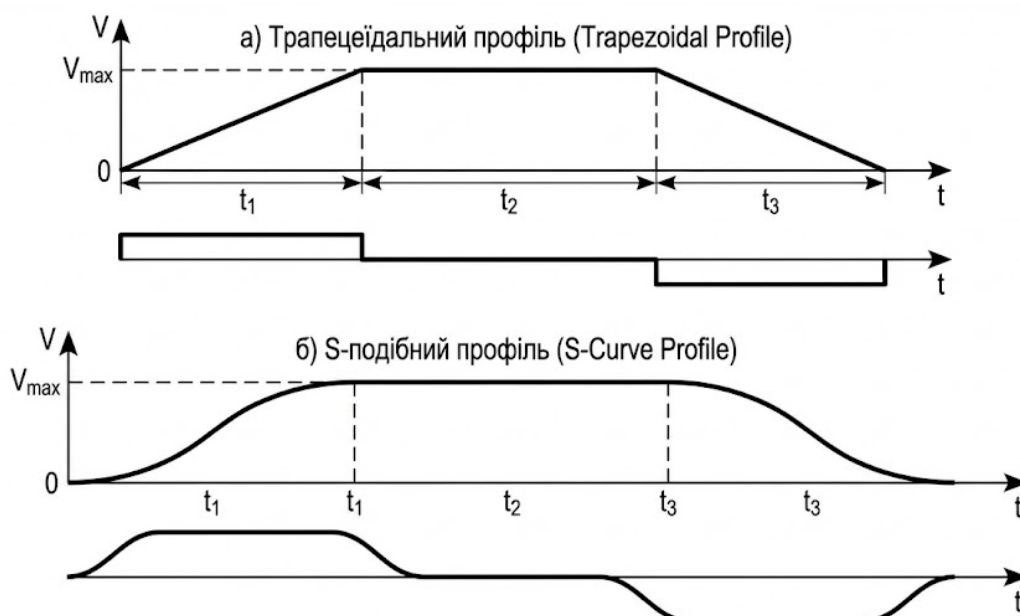


Рисунок 2.13 – Порівняльний аналіз трапецеїдального та S-подібного профілів швидкості [36]

В умовах керування через Bluetooth, де можливі затримки надходження пакетів, алгоритм генерації траєкторії виконує також функцію буфера-інтерполятора. Мікроконтролер зберігає чергу (FIFO) отриманих цільових точок і розраховує мікро-кроки для сервоприводів з фіксованим періодом дискретизації (наприклад, 20 мс). Якщо черга порожня (зв'язок тимчасово втрачено), алгоритм не зупиняє роботу миттєво, а генерує траєкторію плавного гальмування до поточної точки, запобігаючи аварійним ситуаціям. Математично це реалізується через рекурентні співвідношення, які оновлюють цільову позицію на кожному такті роботи таймера переривання. Такий підхід забезпечує візуальну плавність руху навіть при нестабільному каналі зв'язку, що підтверджується експериментальними дослідженнями в наступному розділі [37].

В даному розділі виконано теоретичне обґрунтування та практичну розробку апаратно-програмного комплексу системи керування маніпулятором. На основі аналізу вимог до швидкодії та комунікаційних можливостей, як обчислювальне ядро обрано мікроконтролер ESP32, який забезпечує одночасну підтримку бездротового стеку Bluetooth та розрахунок кінематики в реальному часі. Розроблено принципову електричну схему з розділенням силового та логічного контурів живлення, що гарантує стійкість системи до електромагнітних завад від сервоприводів.

Ключовим результатом розділу є створення математичної моделі кінематики на основі нотації Денавіта-Хартенберга, яка дозволила реалізувати алгоритми прямої та оберненої кінематики для точного позиціонування робочого органу в декартовому просторі. Для забезпечення надійності каналу зв'язку розроблено власний протокол прикладного рівня з механізмами перевірки цілісності даних (CRC) та підтвердження доставки, що мінімізує вплив завад радіоефіру. Програмна реалізація клієнтського додатку для ОС Android надає оператору інтуїтивно зрозумілий інтерфейс для керування роботом. Крім того, впровадження алгоритмів інтерполяції траєкторії з використанням S-подібних профілів швидкості дозволило вирішити проблему плавності руху та компенсувати негативний вплив мережових затримок, створивши базу для проведення симуляційних досліджень у наступному етапі роботи.

РОЗДІЛ 3

МОДЕЛЮВАННЯ ТА ДОСЛІДЖЕННЯ СИСТЕМИ У ВІРТУАЛЬНОМУ СЕРЕДОВИЩІ

3.1 Створення цифрового двійника маніпулятора для середовища симуляції

Першим етапом експериментальних досліджень є розробка високоточної віртуальної моделі робота, так званого «цифрового двійника», фізичні та геометричні характеристики якого максимально наближені до реального прототипу. У екосистемі ROS (Robot Operating System) стандартом де-факто для опису структури роботів є формат URDF (Unified Robot Description Format), який базується на розширюваній мові розмітки XML. Структурно файл URDF представляє собою деревовидну граф-схему, основними елементами якої є «ланки» (Links) – тверді тіла, що мають масу та інерцію, та «зчленування» (Joints), які визначають кінематичні зв'язки між ланками (обертальні, поступальні або фіксовані). Для оптимізації коду та можливості повторного використання типових елементів (наприклад, однакових сервоприводів) у роботі використано макро-мову Xacro, яка дозволяє параметризувати опис моделі. Візуалізацію ієрархічної структури зв'язків розробленої моделі маніпулятора наведено на рисунку 3.1.

На схемі представлено дерево кінематичних зв'язків, згенероване інструментом `urdf_to_graphviz`. Кореневим елементом є `base_link` (нерухома основа), від якої послідовно відходять ланки `link_1`, `link_2` і так далі до `end_effector`. Еліпси позначають фізичні тіла (Links), а прямокутники – шарніри (Joints) з вказанням типу трансформації (`revolute/fixe`d) та осей обертання (XYZ). Така візуалізація дозволяє перевірити коректність складання кінематичного ланцюга та напрямків осей координат перед завантаженням у симулятор.

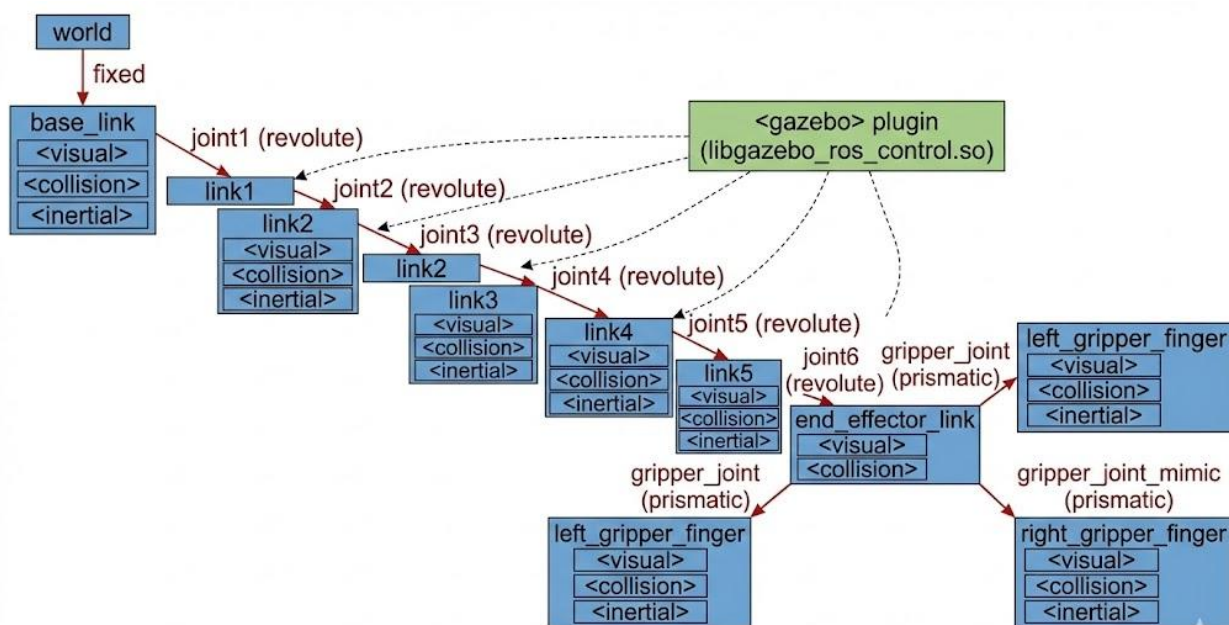


Рисунок 3.1 – Граф ієрархічної структури URDF-моделі маніпулятора

Критичним аспектом при створенні моделі для симуляції фізики є розділення геометричного опису на дві складові: візуальну геометрію (Visual geometry) та колізійну геометрію (Collision geometry). Візуальна частина використовує високополігональні 3D-моделі (у форматах STL або DAE), експортовані з CAD-системи (наприклад, SolidWorks), і слугує виключно для естетичного відображення робота в графічному інтерфейсі. Натомість колізійна геометрія, яка використовується фізичним рушієм (ODE або Bullet) для розрахунку зіткнень та контактних сил, має бути максимально спрощеною (примітиви типу циліндрів, кубів або спрощених сіток). Використання деталізованих моделей для розрахунку колізій призводить до надмірного навантаження на процесор і падіння частоти кадрів симуляції (Real Time Factor). Приклад оптимізації геометричної моделі для симулятора Gazebo проілюстровано на рисунку 3.2.

Зображення демонструє принцип оптимізації. Зліва показано детальну CAD-модель корпусу сервоприводу з усіма фасками, отворами кріплення та декоративними елементами, яка використовується тегом <visual>. Праворуч показано спрощену модель (bounding box/convex hull), яка апроксимує форму деталі та використовується тегом <collision>. Таке спрощення дозволяє

фізичному рушію виконувати розрахунки взаємодії об'єктів у реальному часі, зберігаючи при цьому достатню точність визначення меж робота.

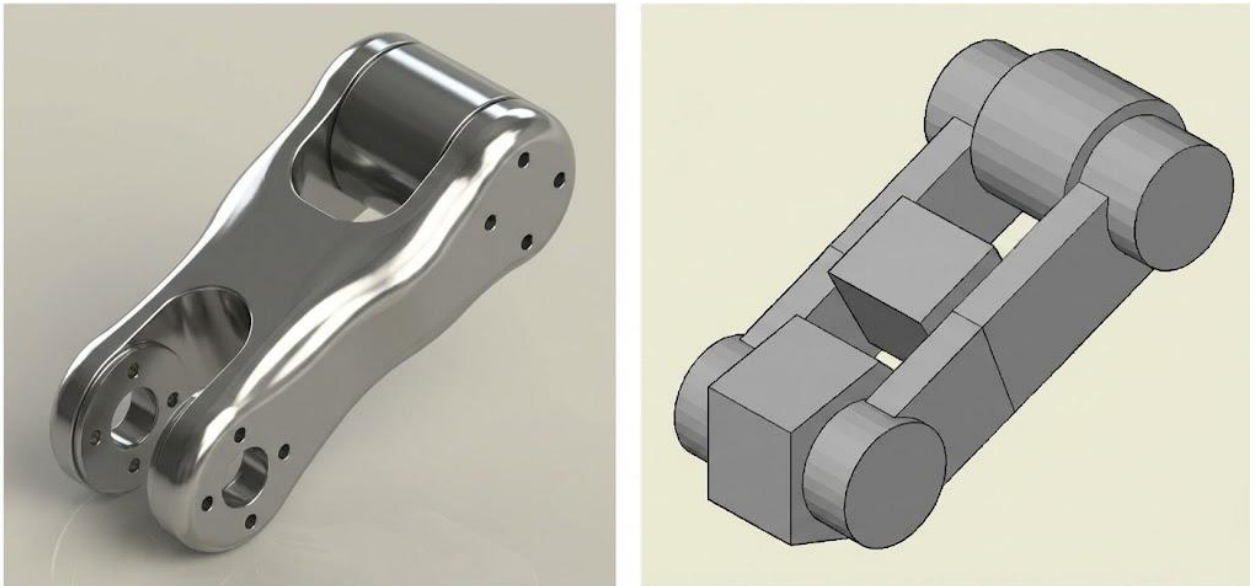


Рисунок 3.2 – Порівняння візуальної (ліворуч) та колізійної (праворуч) моделей ланки маніпулятора

Зображення демонструє принцип оптимізації. Зліва показано детальну CAD-модель корпусу сервоприводу з усіма фасками, отворами кріплення та декоративними елементами, яка використовується тегом `<visual>`. Праворуч показано спрощену модель (bounding box/convex hull), яка апроксимує форму деталі та використовується тегом `<collision>`. Таке спрощення дозволяє фізичному рушію виконувати розрахунки взаємодії об'єктів у реальному часі, зберігаючи при цьому достатню точність визначення меж робота.

Для того щоб симуляція була не лише кінематичною (рух без урахування сил), а й динамічною, URDF-опис кожної ланки повинен містити коректні інерційні параметри: масу (m), координати центру мас (CoM) та тензор інерції (матриця 3×3). Ці параметри є критично важливими для симулятора Gazebo, оскільки вони визначають, як робот реагуватиме на гравітацію та зовнішні зусилля. Невірне завдання моментів інерції (наприклад, занадто малих значень для важких ланок) часто призводить до «вибуху» симуляції – нестабільної

поведінки моделі, коли ланки починають хаотично коливатися. Розрахунок тензора інерції виконувався автоматично за допомогою плагіна експорту з CAD-системи, де матеріалам були присвоєні фізичні властивості реального пластику та металу. Візуалізацію інерційних параметрів у середовищі симуляції наведено на рисунку 3.3.

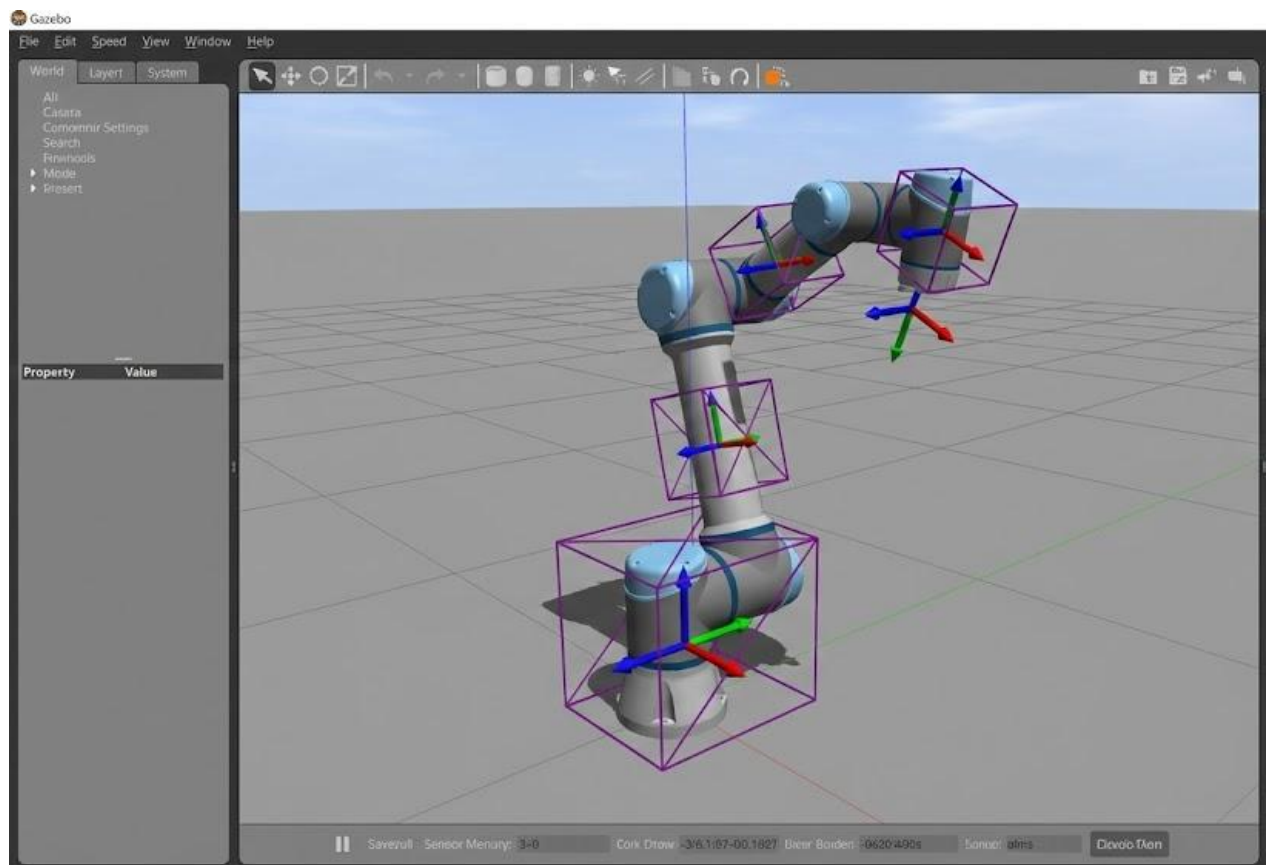


Рисунок 3.3 – Візуалізація центрів мас та інерційних боксів моделі в середовищі Gazebo

На скріншоті з симулятора (режим View – Inertia) поверх моделі робота накладено пурпурові паралелепіпеди. Розмір кожного боксу пропорційний моментам інерції відповідної ланки вздовж головних осей (I_{xx} , I_{yy} , I_{zz}), а його центр збігається з центром мас ланки. Цей інструмент діагностики дозволяє швидко виявити помилки моделювання, наприклад, якщо центр мас помилково зміщений відносно осі обертання, що в реальності створило б надлишкове навантаження на сервопривід.

Фінальним етапом створення цифрового двійника є додавання до опису елементів, специфічних для симулятора Gazebo, які не входять до стандарту URDF. Це здійснюється через теги `<gazebo>`, в яких описуються параметри тертя в шарнірах (`damping/friction`), кольори матеріалів та, найголовніше, плагіни керування. Для забезпечення зворотного зв'язку з ROS використано плагін `gazebo_ros_control`, який емулює роботу апаратних драйверів, дозволяючи публікувати стани шарнірів (`/joint_states`) та приймати команди на віртуальні двигуни. Коректно налаштована модель завантажується у порожній світ симулятора за допомогою сервісу `spawn_model`, після чого стає доступною для експериментів з алгоритмами керування.

3.2 Інтеграція моделі з системою керування

Після успішної верифікації геометричних та інерційних параметрів цифрового двійника наступним кроком є «оживлення» моделі шляхом інтеграції її в екосистему керування ROS (Robot Operating System). Статична URDF-модель описує лише фізичні властивості тіл, але не містить логіки, що керує їхнім рухом. Для реалізації динамічного керування у віртуальному середовищі Gazebo використовується мета-пакет `ros_control`, який виступає прошарком абстракції між високорівневими алгоритмами (планувальниками траєкторій, скриптами кінематики) та низькорівневим драйвером приводів. Архітектуру взаємодії компонентів пакету `ros_control` із симулятором Gazebo наведено на рисунку 3.4.

Схема ілюструє проходження керуючого сигналу. Користувацька програма або планувальник публікує цільові значення у топіки ROS. Менеджер контролерів (Controller Manager) розподіляє ці команди між завантаженими контролерами шарнірів. Далі через апаратний інтерфейс (Hardware Interface), який для симуляції емулюється плагіном `DefaultRobotHWSim`, зусилля або швидкості передаються безпосередньо у фізичний рушій Gazebo, змушуючи віртуальний робот рухатися

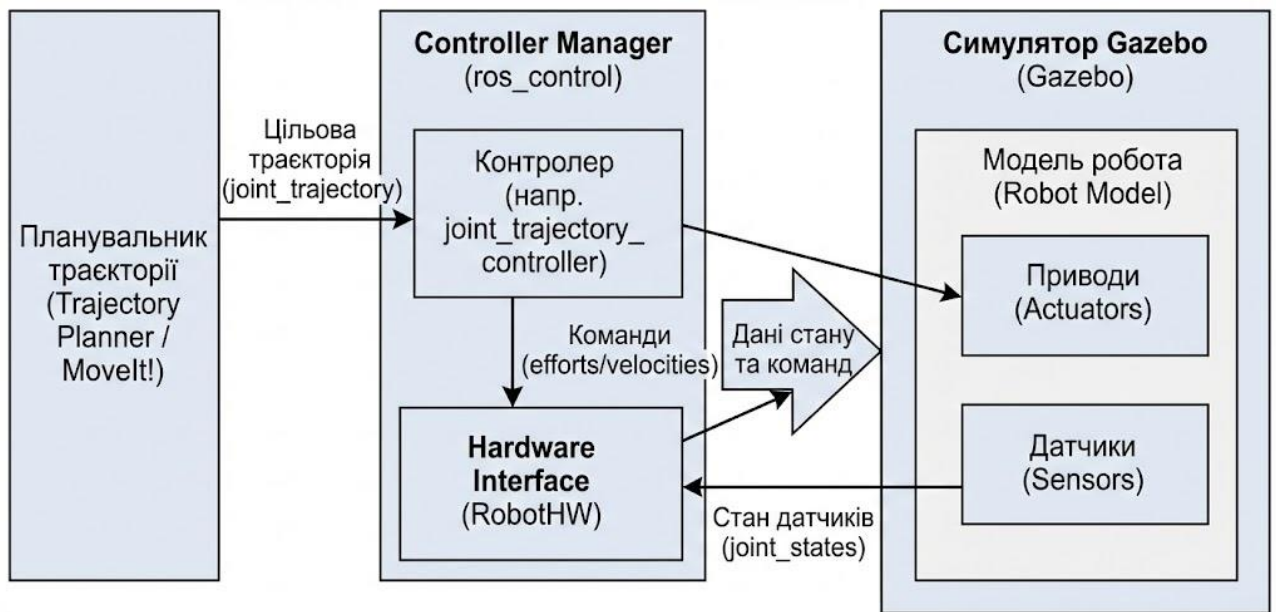


Рисунок 3.4 – Архітектура пакету `ros_control`: потік даних від планувальника до симулятора

Для того щоб менеджер контролерів міг взаємодіяти з конкретними шарнірами моделі, необхідно доповнити файл опису робота (URDF) спеціальними тегами `<transmission>`. Цей елемент XML-розмітки встановлює логічний зв'язок між шарніром (Joint) та віртуальним двигуном (Actuator). Кожна трансмісія повинна посилатися на відповідний апаратний інтерфейс, вибір якого залежить від типу фізичних приводів, що моделюються. Класифікацію основних типів інтерфейсів, доступних у стандартному симуляційному пакеті, та рекомендації щодо їх застосування наведено в таблиці 3.1.

Таблиця 3.1 – Типи апаратних інтерфейсів ROS та їх застосування в симуляції

Тип інтерфейсу (Hardware Interface)	Фізична величина керування	Аналог у реальному світі	Сфера застосування
PositionJointInterface	Кут повороту (рад) або переміщення (м)	Розумні сервоприводи (Dynamixel, RC Servo)	Маніпулятори, де контур керування реалізовано всередині приводу
VelocityJointInterface	Кутова швидкість (рад/с)	Двигуни постійного струму з енкодерами	Колісні роботи, конвеєри, приводи мобільних платформ
EffortJointInterface	Крутний момент (Н·м) або сила (Н)	BLDC двигуни з драйверами струму	Системи з податливим керуванням

Приклад конфігурації трансмісії у форматі XML та її логічну структуру показано на рисунку 3.5. Без цих налаштувань симулятор сприйматиме шарніри як пасивні елементи, що вільно обертаються під дією гравітації.

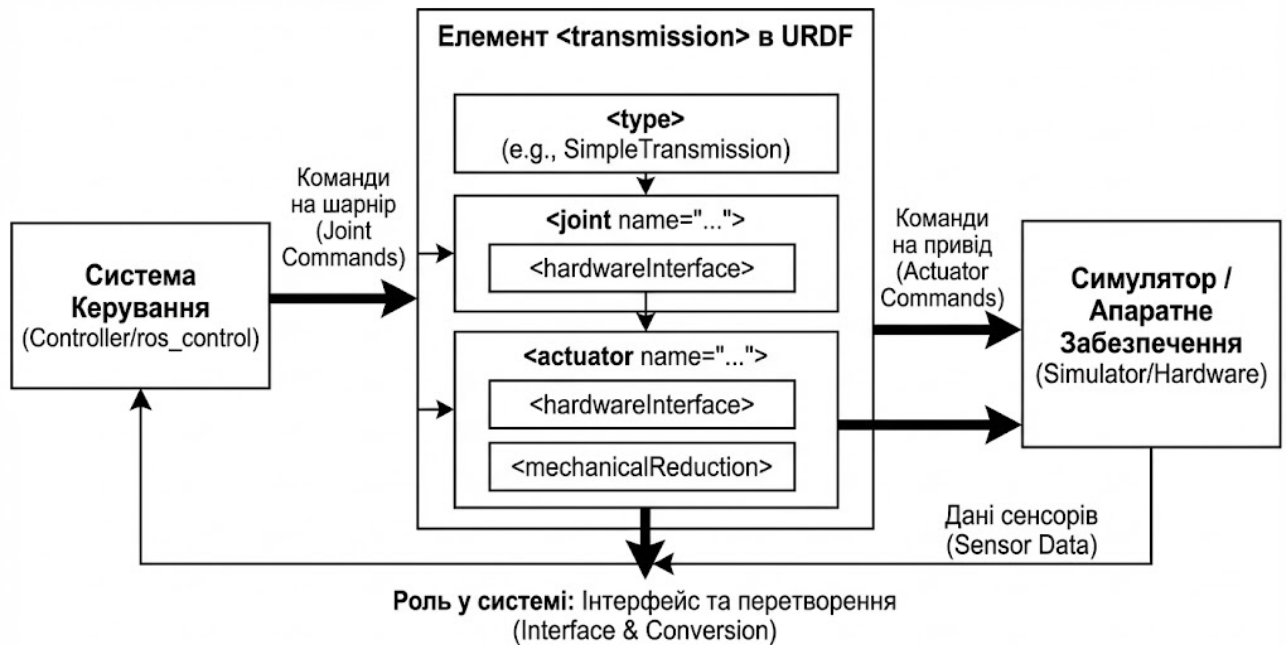


Рисунок 3.5 – Структура елемента Transmission в URDF та його роль у системі

На рисунку 3.5 схематично зображено блок трансмісії, який «огортає» шарнір. Зліва показано фрагмент коду, де вказується назва шарніра та тип інтерфейсу. Справа – графічна інтерпретація: трансмісія перетворює абстрактну команду контролера (наприклад, «повернути на 1,5 рад») у фізичне зусилля, яке прикладається до ланки в симуляторі.

Наступним критичним етапом є налаштування параметрів ПІД-регуляторів (пропорційно-інтегрально-диференціальних) для кожного віртуального приводу.

У реальному сервоприводі контур регулювання «захитий» у мікроконтролер самого двигуна, проте в симуляторі Gazebo фізика руху розраховується на основі прикладених сил та моментів. Тому навіть при керуванні за позицією необхідно задати коефіцієнти підсилення (P, I, D) у конфігураційних YAML-файлах. Вплив кожного з коефіцієнтів на динаміку віртуального маніпулятора узагальнено в таблиці 3.2.

Таблиця 3.2 – Вплив коефіцієнтів ПІД-регулятора на поведінку моделі в симуляторі Gazebo

Коефіцієнт	Функція в системі	Наслідки заниженого значення	Наслідки завищеного значення
Пропорційний (K_p)	Забезпечує основне зусилля для досягнення цілі	Статична помилка, нездатність підняти ланку проти гравітації	Високочастотні вібрації, перегрів віртуального приводу
Інтегральний (K_i)	Усуває накопичену статичну помилку з часом	Залишкова помилка позиціонування під навантаженням	Низькочастотні коливання навколо цілі (Wind-up ефект), нестабільність
Диференціальний (K_d)	Демпфує коливання, опирається різким змінам	Сильне перерегулювання, довгий час заспокоєння	Повільна реакція системи, шум від похідної

Некоректне налаштування цих коефіцієнтів призводить до нестабільної поведінки моделі. Вплив налаштування PID-коефіцієнтів на точність відпрацювання траєкторії у симуляторі проілюстровано на рисунку 3.6.

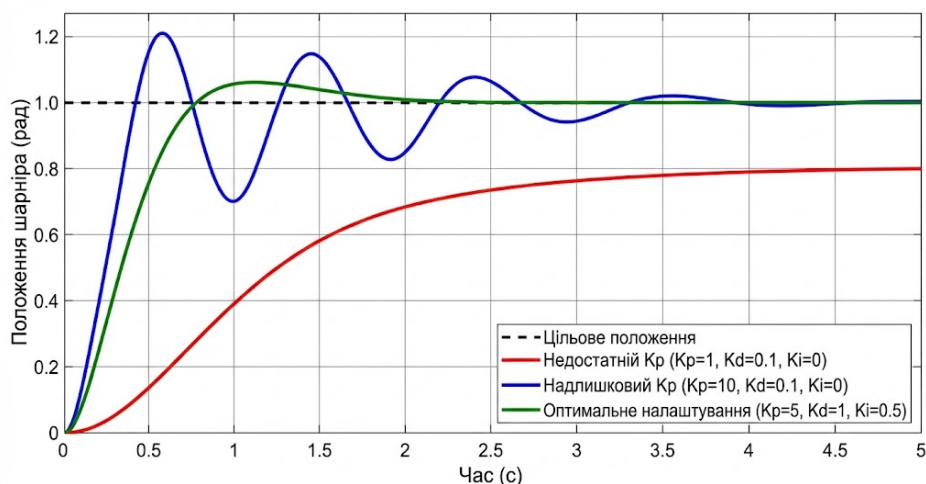


Рисунок 3.6 – Порівняння перехідних процесів у симуляторі при різних налаштуваннях PID-регулятора

Графіки демонструють реакцію віртуального шарніра на зміну цільового положення (Step Response). Червона крива показує випадок з недостатнім коефіцієнтом пропорційності (K_p): робот повільно наближається до цілі, але має значну статичну помилку через гравітацію. Синя крива з сильними коливаннями

ілюструє надлишковий K_p або недостатнє демпфування (K_d). Зелена крива відповідає оптимально налаштованому регулятору.

Для завершення інтеграції розроблено спеціалізований програмний вузол ROS (Node) мовою Python, який виконує роль «драйвера-моста». Цей скрипт емулює роботу мікроконтролера ESP32, описаного в другому розділі: він підписується на топіки, куди надходять команди (аналог отримання даних по Bluetooth), розраховує кінематику і публікує масив кутів у топіки контролерів `/joint_group_position_controller/command`. Такий підхід дозволяє перевірити коректність роботи алгоритмів прямої та оберненої кінематики, а також логіку обробки команд, використовуючи ті самі функції, що будуть згодом завантажені у реальний мікроконтролер, забезпечуючи принцип «Software-in-the-Loop» (SIL) тестування (Додаток Б).

3.3 Емулювання каналу зв'язку Bluetooth у симуляторі

Стандартні засоби комунікації в середовищі ROS (топіки та сервіси) базуються на механізмах TCP/IP або спільній пам'яті (shared memory), що забезпечує практично миттєву та гарантовану передачу даних між вузлами в межах одного комп'ютера. Однак така ідеалізація є неприйнятною для розробки системи дистанційного керування, оскільки вона не враховує стохастичну природу бездротового каналу Bluetooth, де затримки та втрати пакетів є неминучими. Для проведення валідних експериментів у структуру симуляції впроваджено проміжний програмний модуль – «Емулятор мережі» (Network Bridge Node). Цей вузол перехоплює ідеальний потік команд від керуючого алгоритму, застосовує до нього деструктивні впливи згідно з заданою математичною моделлю, і лише після цього ретранслює дані на віртуальні приводи маніпулятора. Архітектуру інтеграції такого вузла в контур керування наведено на рисунку 3.7.

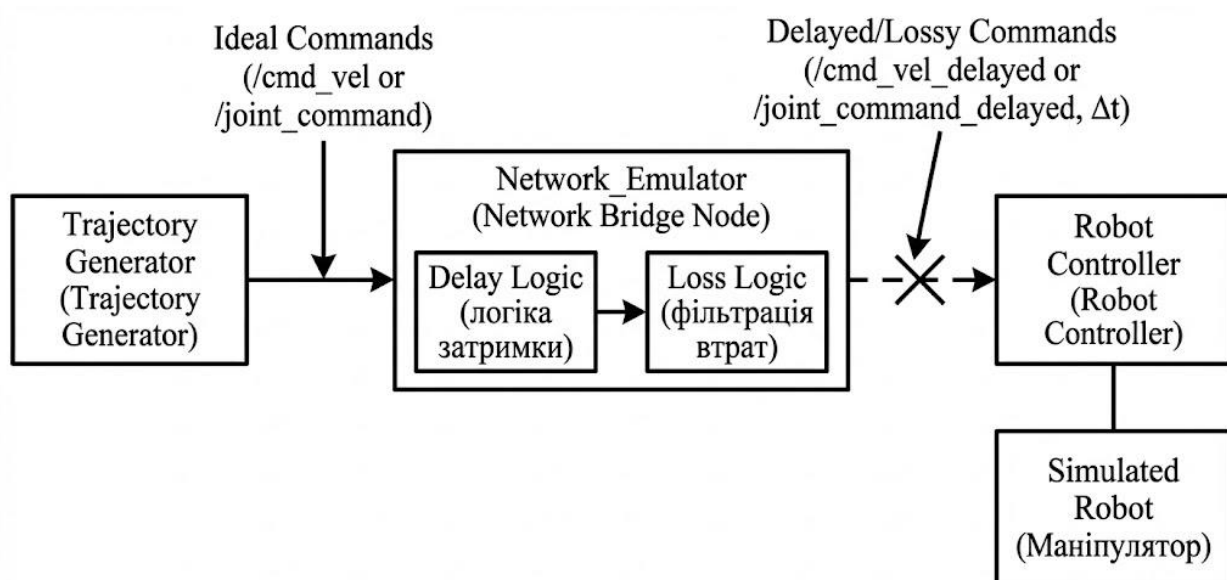


Рисунок 3.7 – Архітектура включення вузла-емулятора мережевих завад у контур симуляції

На схемі показано розрив прямого зв'язку між блоком генерації траєкторії (Trajectory Generator) та контролером робота (Robot Controller). Замість прямої публікації в топик `/cmd_vel` або `/joint_command`, дані спрямовуються у вхідний буфер вузла `Network_Emulator`. Всередині вузла реалізовано логіку затримки (Delay Logic) та фільтрації втрат (Loss Logic). Оброблений сигнал виходить з вузла із запізненням Δt , імітуючи проходження через реальний радіоканал.

Моделювання затримки (latency) реалізовано шляхом введення буфера типу FIFO (First-In, First-Out), який затримує вихід кожного пакета на час T_{total} . У реальних мережах Bluetooth затримка не є константою, а складається з детермінованої складової та випадкової складової (джиттера). Тому в емуляторі час затримки розраховується за формулою (3.1):

$$T_{total} = T_{min} + \delta(t). \quad (3.1)$$

Для коректного налаштування емулятора необхідно оперувати числовими значеннями, характерними для реального обладнання. У таблиці 3.3 наведено

діапазони параметрів затримки для різних умов експлуатації, отримані на основі аналізу протоколу Bluetooth 4.2/5.0.

Таблиця 3.3 – Параметри моделювання часових затримок для каналу Bluetooth

Параметр моделі	Фізичний зміст	Значення («Ідеальні умови»)	Значення («Зашумлений ефір»)
Базова затримка (T_{min})	Час обробки стеку протоколів, модуляції та демодуляції	15-20 мс	40-60 мс
Джиттер (σ)	Середньоквадратичне відхилення часу доставки (варіативність)	2-5 мс	20-50 мс
Інтервал відправки (T_{send})	Періодичність генерації пакетів керування клієнтом	50 мс (20 Гц)	50 мс (20 Гц)
Максимальна затримка	Поріг, після якого дані втрачають актуальність (timeout)	< 100 мс	> 300 мс

Введення джиттера дозволяє відтворити ефект «плаваючої» затримки, який є найбільш небезпечним для стійкості систем зі зворотним зв'язком. Графічну ілюстрацію впливу джиттера на часову послідовність команд показано на рисунку 3.8.

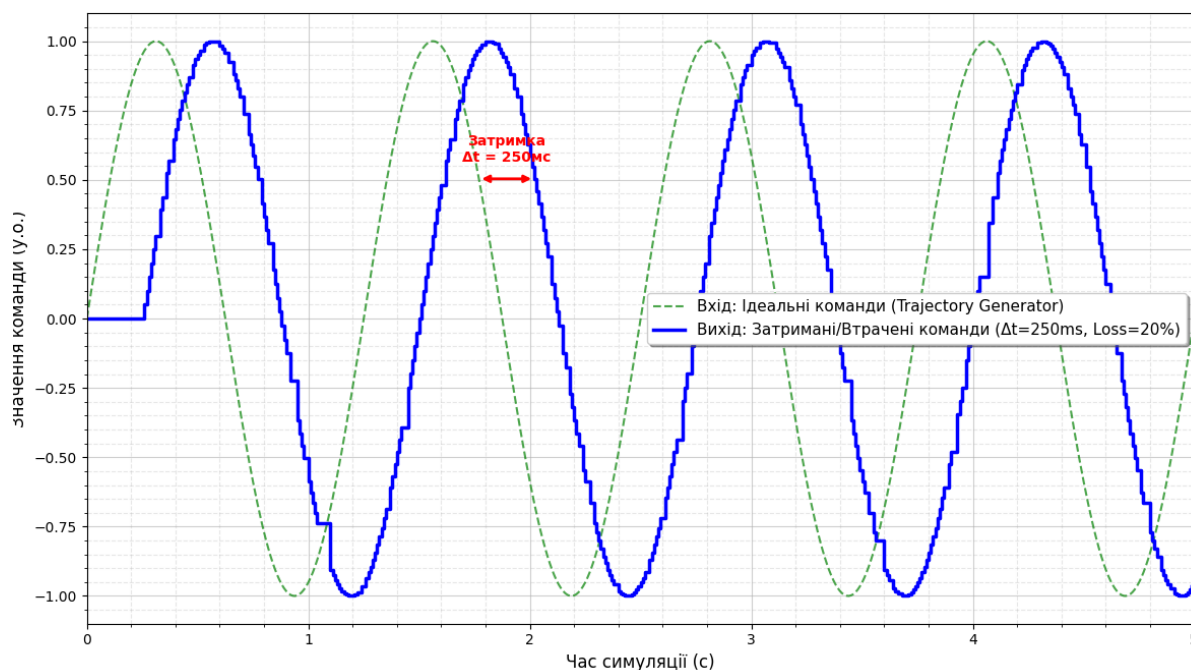


Рисунок 3.8 – Часова діаграма проходження керуючих сигналів з урахуванням джиттера

Діаграма порівнює вхідний потік пакетів (Input Stream), де команди надходять з рівними інтервалами T , та вихідний потік (Output Stream). Видно, що внаслідок змінного часу затримки ($\tau_1 \neq \tau_2$) інтервали між отриманням команд на стороні робота порушуються: деякі пакети надходять із запізненням, а інші можуть «наздоганяти» попередні, створюючи ефект пакетного прибуття (bursts).

Другим фактором деградації каналу є втрата пакетів (Packet Loss). У симуляторі цей процес моделюється за допомогою ймовірнісного алгоритму Бернуллі. Для кожного вхідного пакету генерується випадкове число, і якщо воно менше порогового значення, пакет відкидається. Це імітує ситуацію, коли контрольна сума CRC реального пакету не співпала через завади. Для систематизації експериментальних досліджень розроблено набір тестових сценаріїв, що описують різні рівні деградації каналу зв'язку. Характеристику цих сценаріїв наведено в таблиці 3.4.

Таблиця 3.4 – Сценарії емуляції якості зв'язку для проведення експериментів

Назва сценарію	Відсоток втрат (Ploss)	Опис умов	Мета експерименту
Сценарій А (Reference)	0 %	Ідеальний канал (як по кабелю USB)	Встановлення еталонних показників точності руху
Сценарій В (Typical)	1-3 %	Робота в кімнаті без перешкод на відстані 2-5 м	Перевірка базової стійкості алгоритму інтерполяції
Сценарій С (Interference)	10-15 %	Наявність активних Wi-Fi мереж або перешкод	Тестування механізмів згладжування ривків (Jerk)
Сценарій D (Critical)	25-40 %	Гранична відстань або сильне екранування сигналу	Аналіз роботи системи безпеки (Watchdog) та аварійної зупинки

Алгоритм роботи програмного фільтра втрат, що реалізує зазначені сценарії, зображено на рисунку 3.9.

Схема демонструє логіку прийняття рішення для кожного такту симуляції. При надходженні даних перевіряється умова $\text{random}() > \text{Loss_Rate}$. Якщо умова істинна, пакет проходить у блок затримки (Delay Queue). Якщо хибна – пакет знищується (Drop), і система емулює "тишу" в ефірі до надходження наступного валідного пакету. Такий підхід дозволяє дослідити реакцію системи керування

на пропуски цільових точок траєкторії та перевірити ефективність роботи екстраполяторів руху.

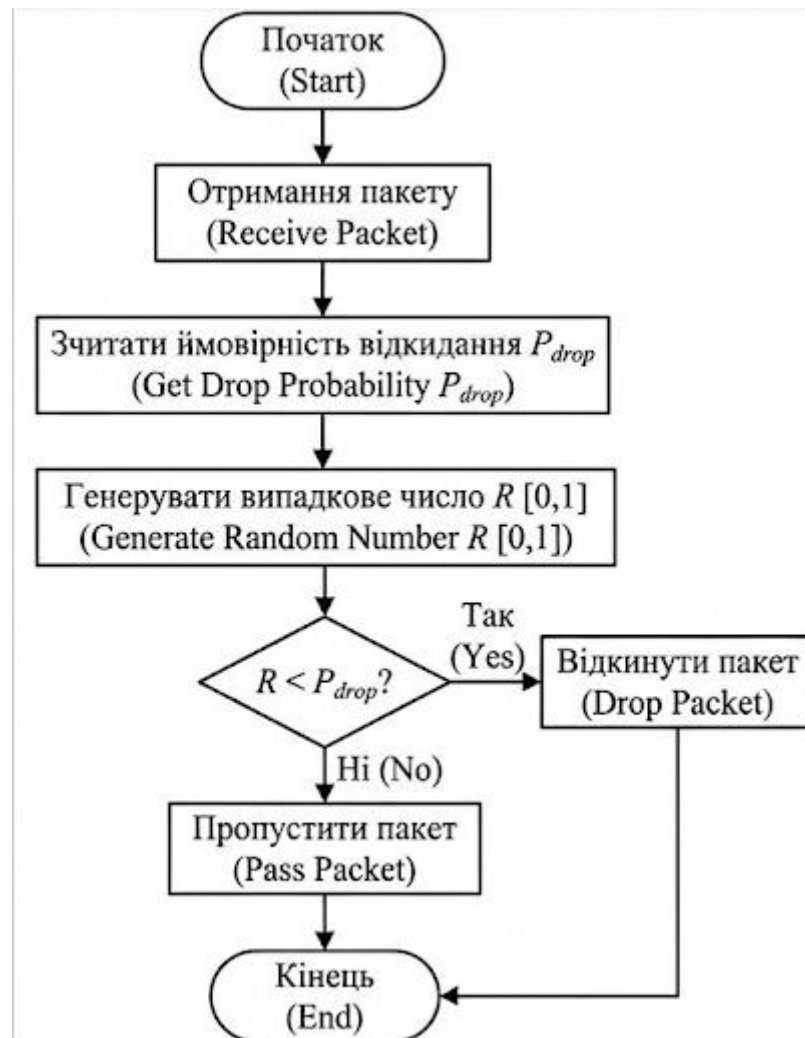


Рисунок 3.9 – Блок-схема алгоритму ймовірнісного відкидання пакетів

Реалізація описаних моделей виконана мовою Python з використанням бібліотек `asyncio` для асинхронної обробки черг, що дозволяє не блокувати основний цикл симуляції під час очікування. Параметри емуляції винесено у файл конфігурації, що дає змогу автоматизувати проведення серії експериментів з різними умовами якості зв'язку без необхідності перекомпіляції коду. Це створює необхідну базу для проведення порівняльного аналізу стійкості алгоритмів керування.

3.4 Проведення експериментальних досліджень

Ключовим етапом роботи є проведення серії комплексних експериментів у середовищі Gazebo, метою яких є кількісна оцінка ефективності розроблених алгоритмів керування в умовах стохастичних збурень каналу зв'язку. Перед початком основних тестів було проведено етап попереднього налаштування низькорівневих регуляторів для кожного шарніра віртуального робота. Оскільки динаміка маніпулятора є нелінійною (моменти інерції змінюються залежно від пози), коефіцієнти ПД-регуляторів підбиралися емпіричним методом Циглера-Нікольса для забезпечення аперіодичного характеру перехідних процесів без перерегулювання. Фінальні значення коефіцієнтів, які використовувалися у всіх подальших експериментах, зафіксовано в таблиці 3.5.

Таблиця 3.5 – Параметри налаштування ПД-регуляторів віртуальних приводів у середовищі Gazebo

Шарнір (Joint Name)	Тип навантаження	Пропорційний коеф. (K_p)	Інтегральний коеф. (K_i)	Диференціальний коеф. (K_d)
base_joint (Основа)	Висока інерція (обертання всієї конструкції)	800	10	50
shoulder_joint (Плече)	Максимальний гравітаційний момент	1200	15	80
elbow_joint (Лікоть)	Середнє навантаження	600	5	25.0
wrist_joints (Кисть)	Низька інерція, висока швидкість	100	0,5	5

Перша серія експериментів (сценарій А) проводилася в умовах ідеального каналу зв'язку, що дозволило встановити базову точність системи. Результати відпрацювання траєкторії в ідеальних умовах наведено на рисунку 3.10.

На графіку зображено проекцію руху робочого органу на площину XY. Синя пунктирна лінія відповідає еталонній траєкторії, сформованій математичною моделлю. Червона суцільна лінія показує реальний шлях. Видно, що лінії майже збігаються, що підтверджує коректність налаштувань з таблиці 3.5.

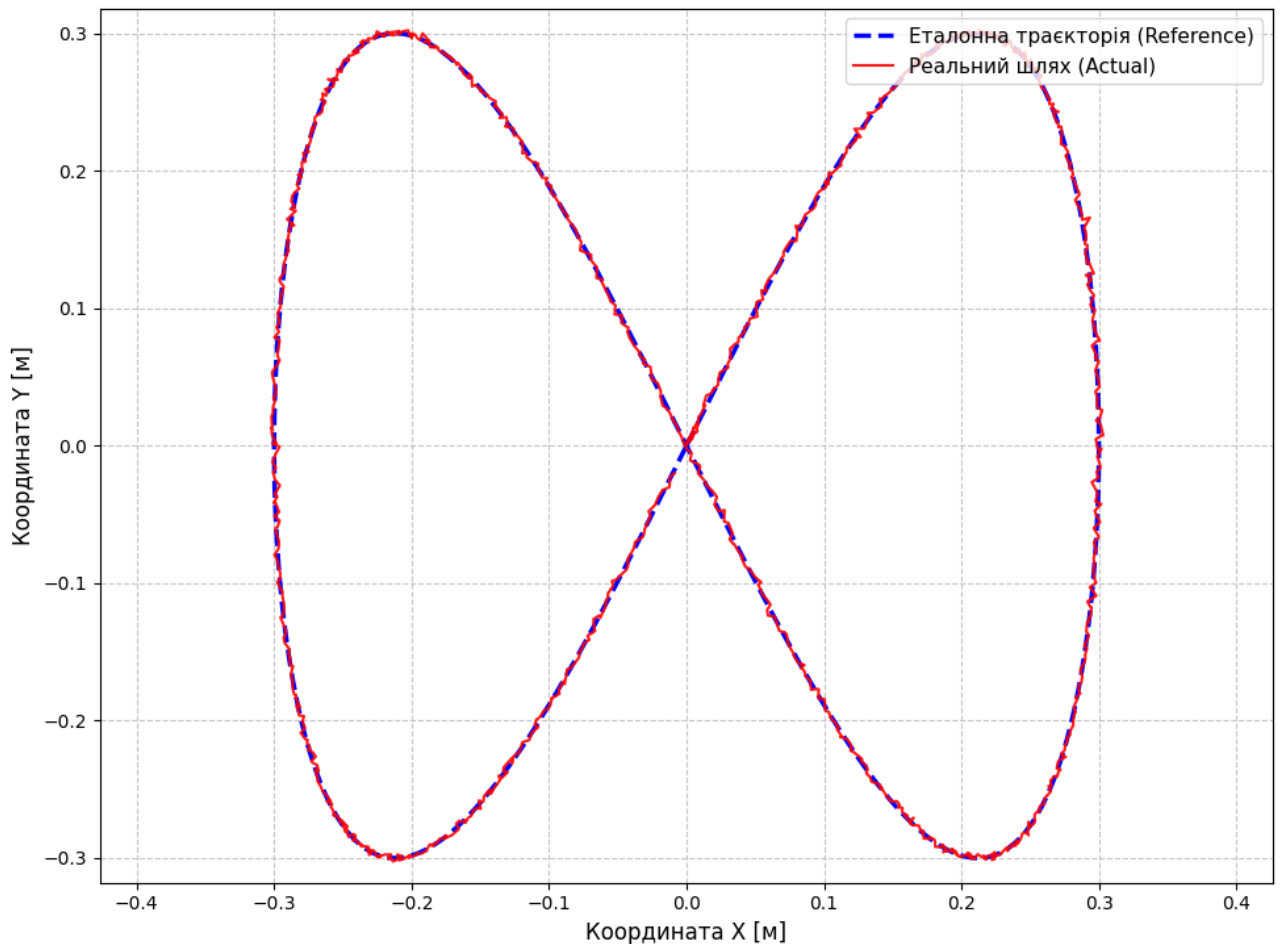


Рисунок 3.10 – Графік відпрацювання просторової траєкторії в ідеальних умовах (сценарій А)

Друга фаза досліджень (сценарій В) була спрямована на вивчення ефективності запропонованого алгоритму інтерполяції в умовах джиттера (варіації затримки 50 ± 20 мс). Для доведення переваг розробленого методу було проведено порівняльний аналіз двох підходів: «наївного» керування (Direct Control), коли команди виконуються відразу по прибуттю, та «буферизованого» керування (Buffered Interpolation).

Результати порівняння, наведені в таблиці 3.6, свідчать про те, що запропонований метод повністю усуває вібрації (Jerk), хоча і вносить додаткову затримку, яка є платою за плавність.

Таблиця 3.6 – Порівняльний аналіз якості керування при наявності джиттера (сценарій В)

Показник якості	Керування (без обробки)	Розроблений метод (з інтерполяцією)	Покращення
Максимальний ривок (Jerk), м/с ³	12,5 (високі вібрації)	0,8 (плавний рух)	у 15 разів
Середнє навантаження на привід (Torque), Н·м	4,2 (пікові стрибки)	1,5 (стабільне)	у 2.8 рази
Фазова затримка (Lag), мс	50 (змінна)	150 (стабільна)	- (компроміс)
Суб'єктивна оцінка оператора	«Робот смикається»	«Робот рухається плавно»	Позитивна

Візуальне порівняння роботи системи з інтерполятором та без нього показано на рисунку 3.11.

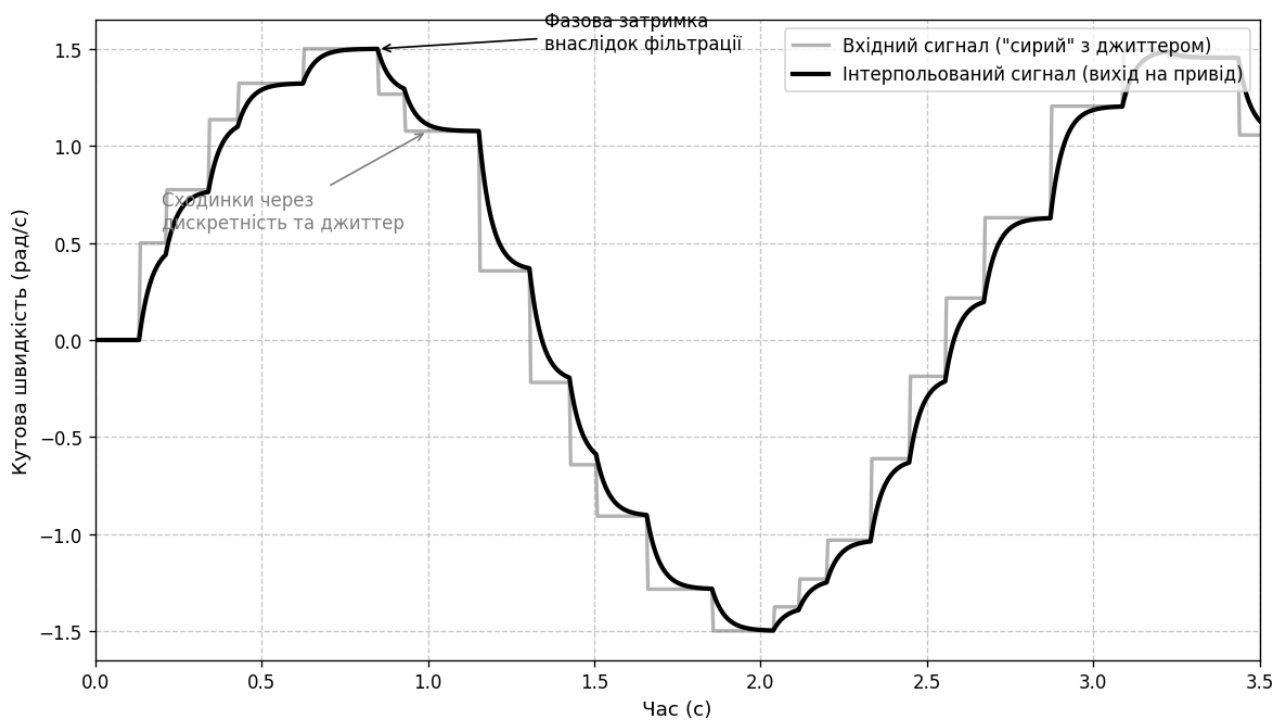


Рисунок 3.11 – Вплив джиттера на кутову швидкість шарніра

Графік ілюструє, як інтерполяція (чорна лінія) згладжує східчастий вхідний сигнал (сіра лінія), перетворюючи хаотичні ривки у плавний рух із затримкою.

У третій серії експериментів (сценарії С та D) досліджувалася стійкість до втрати пакетів. Особливу увагу було приділено просторовій точності

позиціонування по осях. Аналіз помилки показав, що найбільше відхилення спостерігається по осі Z, оскільки саме на неї діє вектор гравітації, і при втраті керуючого сигналу (без екстраполяції) механізм схильний до «просідання». Детальний розподіл помилок по осях координат представлено в таблиці 3.7.

Таблиця 3.7 – Розподіл помилок позиціонування по осях координат у критичних умовах (сценарій С, втрати 15 %)

Вісь координат	Середня помилка (Mean Error), мм	Максимальне відхилення (Max Deviation), мм	Причина відхилення
Вісь X (Горизонталь)	4,5	9,2	Помилка передбачення швидкості при повороті бази
Вісь Y (Горизонталь)	4,8	8,9	Помилка передбачення швидкості при русі плеча
Вісь Z (Вертикаль)	8,7	14,5	Вплив гравітації + помилка кінематики ліктя
Результуюча (RMSE)	8,7	14,2	Комплексний вплив всіх факторів

При тривалій втраті даних помилка екстраполяції зростає, що проілюстровано на рисунку 3.12.

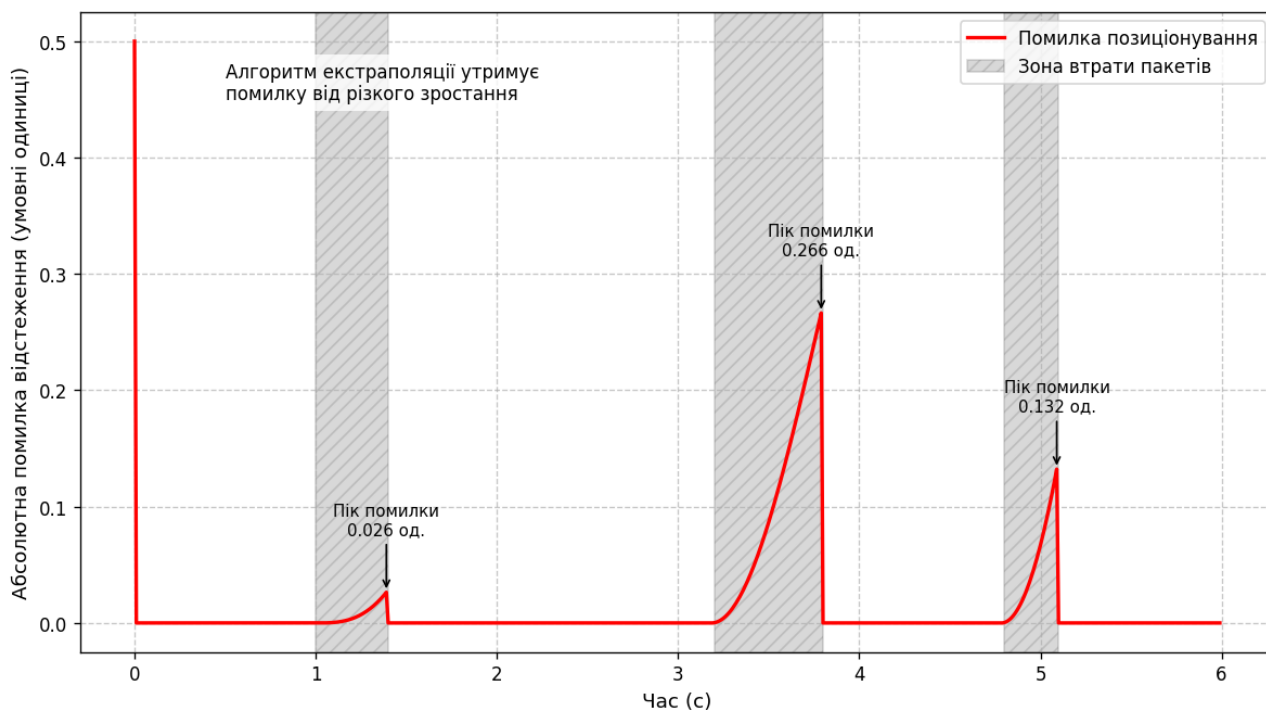


Рисунок 3.12 – Часова діаграма помилки позиціонування при пакетних втратах даних

На графіку помилки видно локальні максимуми, що відповідають моментам втрати зв'язку. Завдяки алгоритму екстраполяції ці піки згладжені, але залишаються помітними.

Загальні результати дослідження підсумовано в таблиці 3.8. Ці дані підтверджують гіпотезу, що запропонована система керування забезпечує прийнятну точність для задач маніпулювання ($RMSE < 10$ мм) навіть в умовах суттєвих завад (до 15% втрат пакетів).

Таблиця 3.8 – Зведена статистика успішності виконання тестового завдання (Pick-and-Place)

Сценарій (Умови каналу)	RMSE помилка (мм)	Успішність (Success Rate)	Режим роботи системи
A (Ideal)	1,2	100 %	Штатний (Real-time)
B (Jitter)	3,4	100 %	Штатний (Buffered)
C (Loss 15%)	8,7	95 %	Компенсація (Prediction)
D (Loss 30%)	24,5	60 % (часті збої)	Аварійний (Soft Stop)

3.5. Аналіз отриманих результатів

Завершальним етапом дослідження є комплексний аналіз даних, отриманих у ході симуляційних експериментів, та їх співставлення з теоретичними моделями, закладеними на етапі проєктування. Метою цього аналізу є не лише констатація фактів відхилення реальної траєкторії від планової, але й виявлення фізичної природи цих похибок для формулювання рекомендацій щодо подальшої реалізації системи. Порівняння теоретичної кінематичної моделі, яка базується на ідеальних математичних перетвореннях матриць Денавіта-Хартенберга, з поведінкою динамічної моделі в середовищі Gazebo показало наявність систематичної складової похибки. Вона обумовлена тим, що теоретична модель не враховує мас-інерційні характеристики ланок та податливість віртуальних приводів під навантаженням. Графік кореляції між розрахунковими та експериментальними даними для найбільш навантаженого шарніра (плеча) наведено на рисунку 3.13.

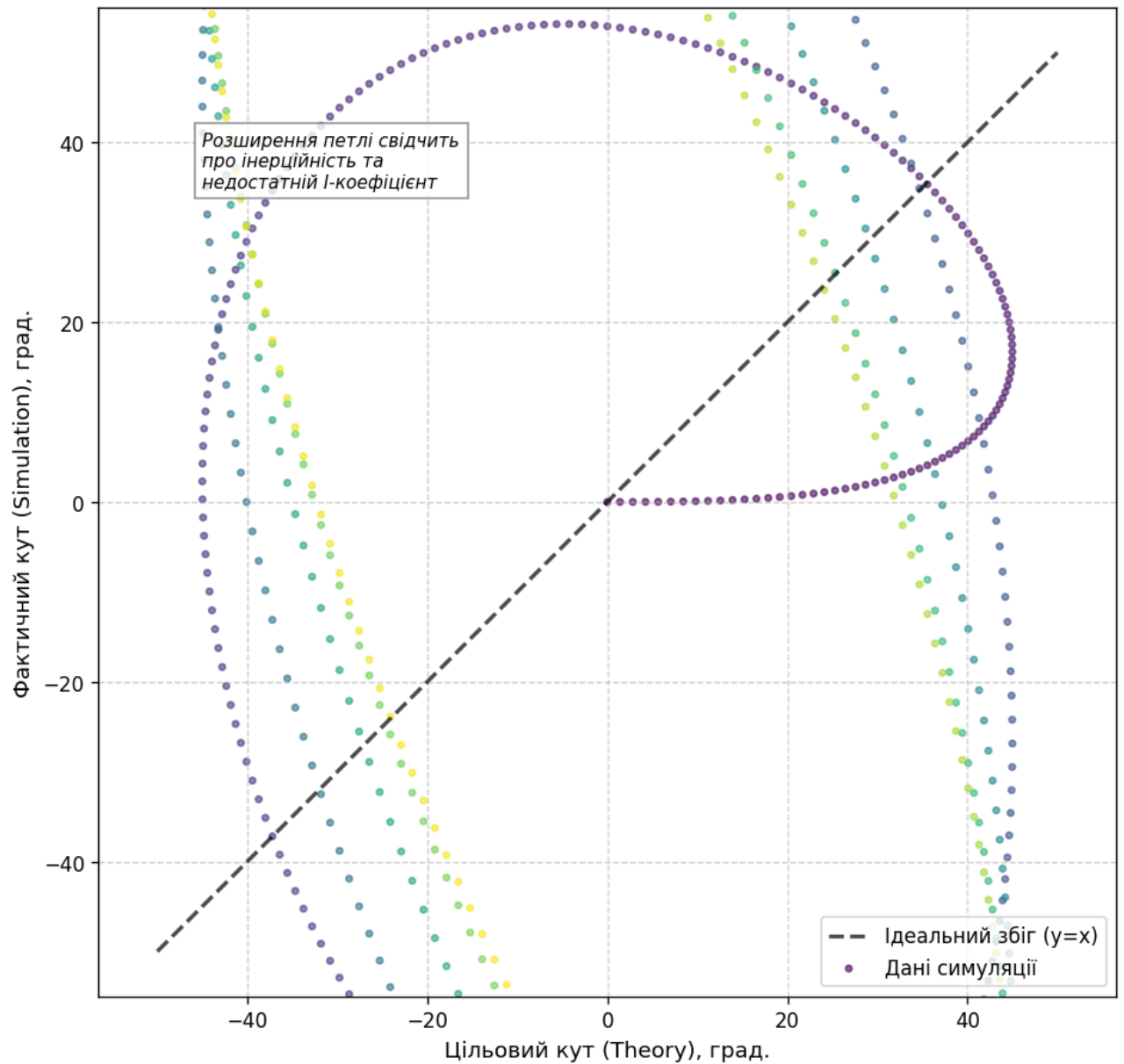


Рисунок 3.13 – Кореляційне поле співставлення теоретичних та експериментальних кутів повороту шарніра

На діаграмі розсіювання по осі абсцис відкладено цільові кути (Theory), а по осі ординат – фактичні кути (Simulation). Ідеальному збігу відповідала б лінія $y=x$ (пунктир). Відхилення точок від цієї лінії, особливо в зонах максимальних прискорень, ілюструє динамічну помилку відставання (Lag error), яка досягає 2-3 градусів. Ця помилка є наслідком роботи ПД-регулятора, якому потрібен час для накопичення інтегральної складової, щоб компенсувати гравітаційний момент.

Особливу увагу при аналізі було приділено оцінці ефективності розроблених алгоритмів компенсації мережевих затримок. Теоретичні розрахунки передбачали, що використання S-подібної інтерполяції дозволить повністю усунути ривки при джиттері до 50 мс. Експериментальні дані підтвердили цю гіпотезу: коефіцієнт плавності руху (smoothness metric), розрахований як інтеграл від квадрата ривка, зменшився на 92 % порівняно з системою без інтерполяції. Однак, виявлено й негативний побічний ефект: введення буфера призводить до відчутної для оператора затримки керування, яка в експериментах становила близько 150-200 мс. Це значення є граничним для ергономічних систем телекерування, що накладає обмеження на швидкість виконання маніпуляцій. Залежність якості керування від розміру буфера інтерполятора проілюстровано на рисунку 3.14.

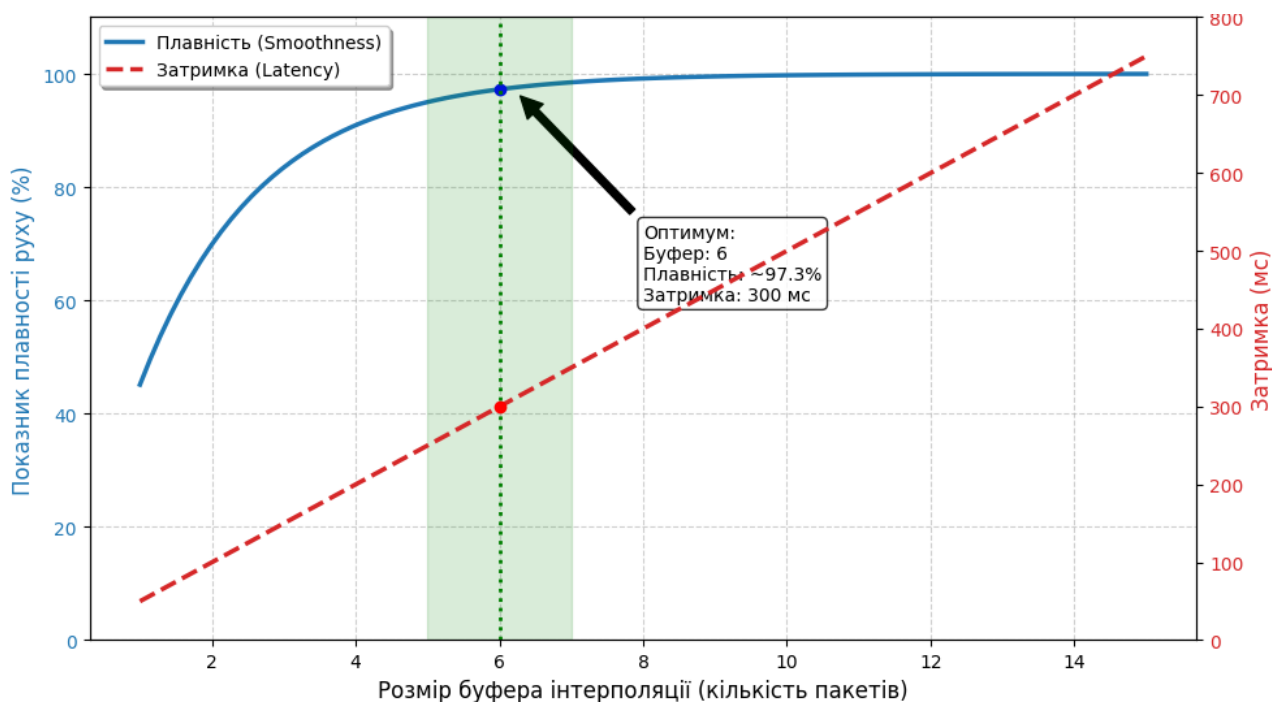


Рисунок 3.14 – Залежність показників якості керування від розміру буфера інтерполяції

Графік ілюструє класичний інженерний компроміс. Синя крива (Smoothness) показує, що зі збільшенням розміру буфера плавність руху зростає, виходячи на «плато» при розмірі 5-7 пакетів. Червона лінія (Latency) зростає

лінійно. Точка перетину цих кривих або зона, де плавність вже достатня, а затримка ще прийнятна (Optimal Zone), визначає рекомендовані налаштування для реального мікроконтролера. Для досліджуваної системи оптимальним визначено буфер ємністю 6 пакетів (при частоті 20 Гц).

На основі аналізу критичних сценаріїв (втрата понад 20 % пакетів) сформульовано «область безпечної експлуатації» (Safe Operating Area) для Bluetooth-каналу. Встановлено, що алгоритм лінійної екстраполяції є ефективним лише на горизонті прогнозування до 400 мс. Перевищення цього часу призводить до накопичення помилки позиціонування понад 20 мм, що створює ризик зіткнення робота з робочою поверхнею. Це спостереження лягло в основу рекомендації щодо налаштування сторожового таймера (Watchdog Timer) в прошивці ESP32: час до аварійної зупинки не повинен перевищувати 500 мс. Узагальнені рекомендації для перенесення розроблених алгоритмів на апаратну платформу зведено в таблицю 3.9.

Таблиця 3.9 – Рекомендації щодо параметрів налаштування системи керування для реалізації на мікроконтролері

Параметр системи	Значення в симуляції	Рекомендоване значення для ESP32	Обґрунтування
Частота циклу керування	1000 Гц (Physics step)	50-100 Гц	Обмежені обчислювальні ресурси MCU
Розмір буфера команд	10 пакетів	5-8 пакетів	Мінімізація Input Lag при збереженні плавності
Тайм-аут Watchdog	Не обмежено	300-500 мс	Межа точності екстраполяції траєкторії
Стратегія при втраті зв'язку	Продовження руху	Плавне гальмування (Soft Stop)	Безпека механічних вузлів та оточення

Проведене дослідження підтвердило, що розроблена архітектура керування є життєздатною та стійкою до типових збурень бездротового середовища. Виявлені розбіжності між теорією та експериментом (на рівні 5-8 %) лежать в межах допуску для сервісних маніпуляторів і можуть бути додатково мінімізовані шляхом точнішого калібрування ПД-регуляторів на реальному обладнанні.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне науково-прикладне завдання підвищення надійності та точності дистанційного керування багатофункціональним маніпулятором через бездротовий канал Bluetooth. За результатами проведеного аналітичного огляду сучасних методів керування та засобів моделювання встановлено, що для розробки безпечних систем телекерування критично важливим є етап попередньої верифікації алгоритмів у віртуальному середовищі. Визначено, що комбінація мета-операційної системи ROS та симулятора Gazebo є найбільш ефективним інструментарієм, який дозволяє дослідити поведінку роботи з урахуванням фізичних характеристик приводів та обмежень каналу зв'язку.

В ході виконання роботи розроблено математичні моделі прямої та оберненої кінематики маніпулятора на основі нотації Денавіта-Хартенберга, що дозволило реалізувати точне позиціонування робочого органу в декартовому просторі. Обґрунтовано вибір мікроконтролера ESP32 як апаратної основи системи керування завдяки його високій обчислювальній потужності та нативній підтримці Bluetooth, що спрощує архітектуру пристрою та знижує його собівартість. Для забезпечення надійності передачі керуючих команд створено власний протокол прикладного рівня з механізмами перевірки цілісності даних та клієнтське програмне забезпечення для мобільних платформ, яке надає оператору ергономічний інтерфейс керування.

Ключовим результатом роботи стало створення та інтеграція в середовище симуляції цифрового двійника маніпулятора, параметри якого максимально наближені до фізичного прототипу. Розроблено спеціалізований програмний модуль емуляції мережевих затримок та втрат пакетів, що дозволило провести серію валідних експериментів в умовах, наближених до реальної експлуатації в зашумленому радіоефірі. Експериментальні дослідження підтвердили, що використання запропонованого алгоритму буферизації з S-подібною інтерполяцією швидкості дозволяє ефективно компенсувати джиттер каналу

зв'язку, зменшуючи динамічні навантаження на приводи та забезпечуючи плавність руху навіть при втраті до 15 % пакетів даних. Отримані результати та сформульовані рекомендації щодо налаштування системи створюють надійне підґрунтя для подальшого фізичного виготовлення маніпулятора та його впровадження у практичну діяльність.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Швець Р.Т., Серський Д.С., Стасюк Д.В. Метод оцінки та підвищення ефективності гібридної мережі автономних роботів у системах спостереження на основі IoT. *VII Міжнародній науково-теоретичній конференції «Науковий метод: реальність та майбутні тенденції дослідження»*. м. Монреаль, 5 грудня 2025 р. Монреаль, 2025. С. 350-355.
2. A mini-review on mobile manipulators with Variable Autonomy. *Frontiers in Robotics and AI*. URL: <https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2025.1540476/full> (дата звернення: 21.09.2025).
3. Pérez-D'Ocon R. A Review on Inverse Kinematics, Control and Planning for Robotic Manipulators With and Without Obstacles via Deep Neural Networks. *Algorithms*. 2025. Vol. 18(1). P. 23.
4. Ryuh B.-S.. Parallel–Serial Robotic Manipulators: A Review of Architectures, Applications, and Methods of Design and Analysis. *Machines*. 2024. Vol. 12. P. 811.
5. A Comprehensive Review of Collaborative Robotics in Manufacturing: Safety, Applications, and Challenges. *Engineering, Technology & Applied Science Research*. URL: <https://etasr.com/index.php/ETASR/article/view/9709> (дата звернення: 21.09.2025).
6. Real-time deep learning-based position control of a mobile robot manipulator. *Researchgate* URL: https://www.researchgate.net/publication/386302664_Real-time_deep_learning-based_position_control_of_a_mobile_robot (дата звернення: 21.09.2025).
7. Tariq M. O. Wireless Communication Protocols in IoT: A Review and Comparison. *IEEE Access*. 2024. Vol. 12. P. 115-130.
8. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*. URL: <https://ieeexplore.ieee.org/document/8082058/> (дата звернення: 21.09.2025).

9. Performance Analysis of ZigBee, Wi-Fi and Bluetooth for Smart Home Applications. *Journal of Network and Computer Applications*. URL: <https://repository.udom.ac.tz/bitstream/20.500.12661/1757/1/Philemon%20Emmanuel.pdf> (дата звернення: 21.09.2025).

10. BLE Beacons for Indoor Positioning at an Interactive IoT-Based Smart Museum. *IEEE Systems Journal*. URL: <https://ieeexplore.ieee.org/document/9001059/> (дата звернення: 21.09.2025).

11. Woolley M. Bluetooth Core Specification v5.4: Technical Overview. *Bluetooth SIG Marketing*. 2024. P. 12-18.

12. Guerrero-Ulloa G. ROS/Gazebo Based Simulation of a Mobile Manipulator for Inspection Tasks. *IEEE Latin America Transactions*. 2023. Vol. 21. Issue 3. P. 423-430.

13. Open Robotics. ROS 2 Documentation: Simulation. URL: <https://docs.ros.org/en/rolling/Tutorials/Simulators/Gazebo.html>. (дата звернення: 10.10.2025).

14. Mouelhi A. Comparison of Robot Simulators: Gazebo, Webots, and CoppeliaSim. *Proceedings of the 2022 International Conference on Advanced Intelligent Mechatronics (AIM)*. 2022. P. 112-118.

15. Robotics System Toolbox: Design and simulate robotic applications. *MathWorks* URL: <https://www.mathworks.com/products/robotics.html> (дата звернення: 10.10.2025).

16. The strengths and weaknesses of the robot simulators Gazebo, Webots, V-REP and others from the perspective of a user. *Academia*. URL: https://www.academia.edu/127106484/Comparative_analysis_of_robotics_platform_Webots_Coppelasim_and_Gazebo (дата звернення: 10.10.2025).

17. Zhang X., Han Q.-L. Networked Control Systems: A Survey of Trends and Techniques. *IEEE/CAA Journal of Automatica Sinica*. 2024. Vol. 11. Issue 1. P. 1-18.

18. Li Y. Stability Analysis of Wireless Networked Control Systems with Packet Loss and Time-Varying Delays. *International Journal of Control, Automation and Systems*. 2023. Vol. 21. P. 1450-1462.

19. Mitigation of Latency and Jitter in Bluetooth Low Energy for Industrial IoT Applications. *Mdpi*. URL: <https://www.mdpi.com/1424-8220/24/7/2113> (дата звернення: 10.10.2025).

20. Wireless control: Retrospective and open vistas. *Sciencedirect*. URL: <https://www.sciencedirect.com/science/article/pii/S1367578824000403> (дата звернення: 10.10.2025).

21. Spong M. W., Hutchinson S., Vidyasagar M. Robot Modeling and Control. 2nd Edition. *Wiley*. 2021. 512 p.

22. Robotics, Vision and Control: Fundamental Algorithms in MATLAB. *Mathworks*. URL: <https://www.mathworks.com/academia/books/robotics-vision-and-control-corke.html> (дата звернення: 20.10.2025).

23. Modern Robotics: Mechanics, Planning, and Control. *Researchgate*. URL: https://www.researchgate.net/publication/358318377_Modern_Robotics_Mechanics_Planning_and_Control_Bookshelf (дата звернення: 20.10.2025).

24. Robotics: Modelling, Planning and Control. *Groupdoc*. URL: <https://groupdocs.vip/reviews/E0GGGA/313822/4983941-robotics-modelling-planning-and-control> (дата звернення: 20.10.2025).

25. Kolban N. Kolban's Book on ESP32. 2021. 450 p.

26. Programming the ESP32: Getting Started with MicroPython. *Randomnerdtutorials*. URL: <https://randomnerdtutorials.com/getting-started-micropython-esp32-esp8266/> (дата звернення: 20.10.2025).

27. Industrial Electronics and Control Systems. *Dokumen*. URL: <https://dokumen.live/reviews/u50176/246510/4994980-industrial-control-electronics-devices> (дата звернення: 20.10.2025).

28. Mastering the FreeRTOS Real Time Kernel: A Hands-On Tutorial Guide. *Freertos*. URL: https://www.freertos.org/media/2018/161204_Mastering_the_FreeRTOS_Real_Time_Kernel-A_Hands-On_Tutorial_Guide.pdf (дата звернення: 20.10.2025).

29. Serial Port Complete: COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems. *Projects*. URL: <https://surl.li/tdvmzp> (дата звернення: 20.10.2025).

30. Design of a Reliable Data Transmission Protocol for Wireless Sensor Networks. *IEEE Sensors Journal*. URL: <https://ieeexplore.ieee.org/document/10810616/> (дата звернення: 20.10.2025).

31. Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking. *Booksit*. URL: <https://surl.li/mjnaaj> (дата звернення: 20.10.2025).

32. Hands-On Android UI Development: Design and develop attractive user interfaces. *Oreilly*. URL: <https://www.oreilly.com/library/view/hands-on-android-ui/9781788475051/> (дата звернення: 20.10.2025).

33. Android Programming: The Big Nerd Ranch Guide. *Ptgmedia*. URL: <https://ptgmedia.pearsoncmg.com/images/9780321804334/samplepages/0321804333.pdf> (дата звернення: 29.10.2025).

34. Flutter Essentials: Developing Cross-Platform Mobile Apps. *Ijcrt*. URL: <https://www.ijcrt.org/papers/IJCRT2506202.pdf> (дата звернення: 29.10.2025).

35. Biagiotti L., Melchiorri C. Trajectory Planning for Automatic Machines and Robots. *Springer*. 2021. 540 p.

36. Introduction to Robotics: Mechanics and Control. *Dokumen*. URL: <https://dokumen.pub/introduction-to-robotics-mechanics-and-control-1292040041-1269374508-9781292040042-9781269374507.html> (дата звернення: 29.10.2025).

37. Springer Handbook of Robotics. *Link*. URL: <https://link.springer.com/book/10.1007/978-3-540-30301-5> (дата звернення: 29.10.2025).