

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Факультет комп'ютерних технологій, автоматики та метрології

Кафедра інтелектуальної мехатроніки та роботи

**РОЗРОБКА СИСТЕМИ ВІДДАЛЕНОГО МОНІТОРИНГУ ТА КОНТРОЛЮ**  
**РОБОТОМ З ВИКОРИСТАННЯМ ВЕБ-САЙТУ**

**БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

Виконав: студент 4 курсу, гр. МТ-43

Спеціальності: Метрологія та інформаційно-  
вимірювальна техніка

*Залускій Віталій Сергійович*

Керівник \_\_\_\_\_

(наук.ступінь, вчене звання ПІБ)

Рецензент \_\_\_\_\_

(наук.ступінь, вчене звання ПІБ)

Львів – 2025

## ЗМІСТ

|                                                                                           |    |
|-------------------------------------------------------------------------------------------|----|
| ВСТУП.....                                                                                | 4  |
| РОЗДІЛ 1 .....                                                                            | 5  |
| 1.1 Роботи, їх типи і характеристики, галузі застосування, конструкції тощо.....          | 5  |
| 1.2 Конструкції та характеристики роботів .....                                           | 8  |
| 1.3 Існуючі смк за роботою роботів, їх галузі застосування, апаратна реалізація тощо..... | 10 |
| 1.4 Апаратна реалізація смк через програмне забезпечення.....                             | 14 |
| 1.5 Програмна реалізація системи igus® robot control .....                                | 22 |
| 1.6 Веб-керування розумним домом (на прикладі homesmart) .....                            | 25 |
| ВИСНОВКИ ДО ПЕРШОГО РОЗДІЛУ .....                                                         | 26 |
| РОЗДІЛ 2 .....                                                                            | 27 |
| 2.1 Постановка задачі .....                                                               | 27 |
| 2.2 Вимоги до системи та обґрунтування вибору веб-інтерфейсу .....                        | 28 |
| 2.3 Обрані технології та структура інтерфейсу .....                                       | 29 |
| 2.4 Загальна схема взаємодії компонентів .....                                            | 30 |
| 2.5 Клієнтський веб-додаток (react.js) .....                                              | 31 |
| 2.6 Прототип робота FANUC LR MATE 200ID/5C .....                                          | 34 |
| 2.7 Протокол передачі даних websocket .....                                               | 40 |
| 2.8 Back-end (сервер) .....                                                               | 42 |
| 2.9 Створення websocket-сервера для обробки команд .....                                  | 44 |
| 2.10 Реалізація клієнтського додатку.....                                                 | 46 |
| 2.11 Взаємодія клієнтів через сервер .....                                                | 47 |
| 2.12 Проблеми під час реалізації та шляхи їх вирішення.....                               | 48 |
| 2.13 Перспективи розвитку .....                                                           | 50 |
| ВИСНОВКИ ДО ДРУГОГО РОЗДІЛУ .....                                                         | 52 |
| РОЗДІЛ 3 .....                                                                            | 53 |
| 3.1 Розрахунок витрат на виконання ндр .....                                              | 53 |

|                                                      |    |
|------------------------------------------------------|----|
| 3.2 Розрахунок договірної ціни та прибутку ндр ..... | 59 |
| ВИСНОВКИ ДО ТРЕТЬОГО РОЗДІЛУ .....                   | 65 |
| ЗАГАЛЬНІ ВИСНОВКИ ДО ДИПЛОМНОЇ РОБОТИ .....          | 66 |
| ВИКОРИСТАНА ЛІТЕРАТУРА .....                         | 67 |
| ДОДАТКИ.....                                         | 69 |

## ВСТУП

*Актуальність теми:* У сучасну епоху цифрової трансформації дистанційне управління технічними системами стає не лише актуальною тенденцією, а й критичною необхідністю у багатьох сферах — від промисловості до дослідницьких лабораторій. Із розвитком інтернет-технологій з'являються нові можливості для створення інтуїтивно зрозумілих, масштабованих та кросплатформених рішень для моніторингу й контролю роботизованих пристроїв у реальному часі.

У цій роботі розглядається проєктування та реалізація системи віддаленого моніторингу та управління роботом за допомогою веб-застосунку. В основі рішення — інтеграція сучасних технологій веб-розробки (React.js, Socket.IO, Node.js) із принципами кіберфізичних систем та реального часу. Такий підхід дозволяє забезпечити безперервний обмін даними між користувачем і виконавчим пристроєм незалежно від географічного розташування оператора.

Розроблена система має на меті продемонструвати можливості ефективної взаємодії людини й машини через браузерний інтерфейс. Особлива увага приділена стабільності передачі даних, безпеці доступу та зручності користувача. Отримані результати можуть бути основою для впровадження подібних рішень у промислові, освітні чи дослідницькі середовища, а також слугувати платформою для подальших удосконалень у сфері дистанційного керування робототехнічними системами.

## РОЗДІЛ 1

### ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ

#### 1.1 РОБОТИ, ЇХ ТИПИ І ХАРАКТЕРИСТИКИ, ГАЛУЗІ ЗАСТОСУВАННЯ, КОНСТРУКЦІЇ ТОЩО.

Робот (від чеськ. *robota* — примусова, зазвичай тяжка праця) — автоматичний пристрій, призначений для виконання виробничих, побутових або інших операцій, які традиційно виконує людина. Термін «автомат» зазвичай застосовується до пристроїв, чиї дії не мають зовнішньої схожості з людськими.

Робот складається з *двох основних* компонентів:

- Виконавчі системи: маніпуляційні механізми (механічні маніпулятори) та системи пересування (для мобільних роботів).
- Система управління: координує дії через сенсори для взаємодії з навколишнім середовищем.

#### КЛАСИФІКАЦІЯ РОБОТІВ

Роботи класифікуються за призначенням, конструкцією та типом управління.  
Основні види:

#### ПРОМИСЛОВІ РОБОТИ

Автоматизовані пристрої для виконання повторюваних виробничих завдань, що підвищують ефективність і точність.

- Маніпулятори: точне переміщення об'єктів (наприклад, складання деталей на заводах Tesla).
- Зварювальні роботи: використовуються в автомобільній промисловості (Toyota).

- Пакувальні роботи: автоматизують сортування та пакування.  
Приклад: роботи на конвеєрах Tesla для складання електромобілів.

## ПОБУТОВІ РОБОТИ

Полегшують виконання щоденних завдань.

Приклади:

- Роботи-пилососи (iRobot Roomba).
  - Роботи для миття вікон (Hobot).
- Особливість: інтеграція з системами «розумного будинку».

## МЕДИЧНІ РОБОТИ

Застосовуються в хірургії, реабілітації та догляді за пацієнтами.

- Хірургічні роботи: система Da Vinci для високоточних операцій.
  - Екзоскелети: підтримка рухливості.
- Особливість: можливість дистанційного керування.

## ВІЙСЬКОВІ РОБОТИ

Використовуються для безпеки та бойових завдань.

- Роботи-сапери: знешкодження вибухівки.
  - Безпілотники: розвідка та патрулювання.
- Приклад: робот Spot від Boston Dynamics для розвідки.

## ДОСЛІДНИЦЬКІ РОБОТИ

Працюють у небезпечних або важкодоступних середовищах.

Приклади:

- Марсоходи (Perseverance).
  - Підводні роботи (ROV) для дослідження глибин.
- Особливість: автономна робота в екстремальних умовах.

## ОСВІТНІ РОБОТИ

Використовуються для навчання робототехніки та програмування.

Приклади: LEGO Mindstorms, Arduino.

Перевага: розвиток навичок програмування.

## СОЦІАЛЬНІ РОБОТИ

Призначені для взаємодії з людьми.

Приклади:

- Pepper: гуманоїд для спілкування.
  - PARO: робот для психотерапії.
- Особливість: підтримка літніх людей у Японії.

## РОЗВАЖАЛЬНІ РОБОТИ

Створені для ігор і дозвілля.

Приклади: Aibo (робот-собака від Sony), ігрові дрони.

## АНДРОЇДИ

Імітують зовнішність і поведінку людини.

Приклад: Sophia з громадянством ОАЕ.

Особливість: поєднання ШІ та психології.

## БІОРОБОТИ ТА НАНОРОБОТИ

Інноваційні пристрої для медицини та досліджень.

- Біороботи: гібриди живих клітин і механіки.
- Нанороботи: доставка ліків в організмі.

Перспективи: лікування онкологічних захворювань.

## **1.2 КОНСТРУКЦІЇ ТА ХАРАКТЕРИСТИКИ РОБОТІВ**

### **МІЖНАРОДНІ СТАНДАРТИ**

За стандартом ISO 8373:2012, роботи поділяються на:

- Промислові: для автоматизації виробництва.
- Сервісні: для побутових або професійних завдань.

IFR класифікує сервісних роботів на особисті (пилососи) та професійні (роботи-кур'єри).

### **ЕТАПИ РОЗВИТКУ**

- Перше покоління (1960-ті): жорстко запрограмовані.
- Друге покоління: гнучкі програми, адаптація до умов.
- Третє покоління: самонавчальні системи з ШІ.

### **ТИПИ УПРАВЛІННЯ**

- Жорсткопрограмовані: фіксована програма.
- Перепрограмовані: корекція дій після навчання.
- Гнучкопрограмовані: самостійне формування програми.

### **МАТЕРІАЛИ**

- Жорсткі: для точних завдань.
- М'які: адаптивні до середовища.
- Гібридні: поєднання обох типів.



## СПОСОБИ ПЕРЕСУВАННЯ

- Колісні: маневреність.
- Гусеничні: пересічена місцевість.
- Крокуючі: імітація ходи.
- Літаючі та плаваючі: для повітря чи води.

## РОБОТ RON ЯК ОБ'ЄКТ ДОСЛІДЖЕННЯ

Навчальний робот RON — платформа для вивчення робототехніки.

Можливості:

- Віддалене керування через веб-інтерфейс.
- Програмування у візуальному або текстовому редакторі.

Цілі:

- Освоєння основ робототехніки.
- Аналіз автономних систем.
- Збір даних у реальному часі.

### **1.3 ІСНУЮЧІ СМК ЗА РОБОТОЮ РОБОТІВ, ЇХ ГАЛУЗІ ЗАСТОСУВАННЯ, АПАРАТНА РЕАЛІЗАЦІЯ ТОЩО.**

Сучасна робототехніка базується на двох ключових компонентах: системі керування та системі моніторингу. Система керування відповідає за виконання дій робота, тоді як система моніторингу забезпечує збір і аналіз інформації про стан робота та зовнішнє середовище. Їхня спільна робота гарантує точність, надійність і безпеку функціонування роботизованих систем.

#### **СИСТЕМА КЕРУВАННЯ РОБОТОМ**

Система керування — це комплекс апаратних і програмних засобів, які забезпечують управління діями робота відповідно до заданих команд або на основі обробки сенсорних даних.

*Основні функції:*

- Контроль положення кінцевого ефектора.
- Керування взаємним розташуванням елементів маніпулятора.
- Регулювання швидкості та прискорення руху.
- Контроль крутного моменту та сили.
- Програмування поведінки через задані алгоритми.
- Інтерактивна взаємодія з оператором.

*Складові елементи:*

- Приводи: серводвигуни або крокові двигуни.
- Привідні механізми: серводрайвери.
- Контролер руху: для обчислень траєкторій.
- Програмне забезпечення: алгоритмічне керування та інтерполяція.

### *Режими керування:*

- З фіксованими параметрами.
- На основі сенсорного зворотного зв'язку.

### *Промислова система керування:*

- Апаратна частина: використовуються високопродуктивні мікропроцесори (*DSP, PowerPC, Intel, ARM*).
- Архітектура: від ієрархічної до модульної з відкритими стандартами.
- Мережеві технології: *Ethernet, CAN, RS-485*.

## СИСТЕМА МОНІТОРИНГУ РОБОТА

Система моніторингу забезпечує збір, аналіз і передачу даних про стан робота та зовнішнє середовище для діагностики, адаптації та забезпечення безпеки.

### *Основні функції:*

- Вимірювання внутрішніх параметрів (температура, струм, положення).
- Виявлення змін у середовищі (візуальні та тактильні сигнали).
- Діагностика несправностей.
- Формування зворотного зв'язку для системи керування.

### *Типи сенсорів:*

- Внутрішні: енкодери, акселерометри, термодатчики.
- Зовнішні: камери, лідари, датчики сили, дотику.

### *Комунікації та інтерфейси:*

- *CAN*-шина, *RS-485, Ethernet*.
- Модулі обробки даних.

- Інтерфейси для оператора (екрани, повідомлення, візуалізація).

## ІНТЕЛЕКТУАЛЬНА ІНТЕГРАЦІЯ СИСТЕМ КЕРУВАННЯ ТА МОНІТОРИНГУ

Система керування та система моніторингу функціонують окремо, але їхня взаємодія формує інтелектуальну платформу роботи:

| Система керування    | Система моніторингу                |
|----------------------|------------------------------------|
| Реалізує рух         | Контролює стан                     |
| Видає команди        | Збирає дані                        |
| Працює за алгоритмом | Реагує на зміни в реальному часі   |
| Вимагає точності     | Забезпечує корекцію та діагностику |

## ПЕРЕВАГИ ТА НЕДОЛІКИ СИСТЕМ КЕРУВАННЯ

*Переваги:*

- Висока точність і стабільність завдяки складним алгоритмам контролю положення, швидкості та моменту.
- Гнучкість програмування, що дозволяє швидко перепрограмувати систему під різні завдання.
- Інтеграція з сенсорними системами для адаптивного керування в реальному часі.
- Автоматизація складних дій, таких як траєкторії, взаємодія з іншими машинами та повторювані цикли.
- Зв'язок з іншими системами через мережеві інтерфейси (*Ethernet*, *CAN*, *Modbus*).

*Недоліки:*

- Складність налаштування: вимагає знань програмування, математики та механіки.
- Чутливість до помилок у програмі: неправильний алгоритм може призвести до аварії або небезпечного руху.
- Висока вартість якісних контролерів, особливо в промисловому сегменті.
- Обмежена адаптація без сенсорів: без зворотного зв'язку система працює «всліпу».

## ПЕРЕВАГИ ТА НЕДОЛІКИ СИСТЕМ МОНІТОРИНГУ

### *Переваги:*

- Реальний контроль стану робота: дані про температуру, струм, положення, навантаження.
- Рання діагностика: виявлення несправностей до виникнення серйозних поломок.
- Безпека: система може зупинити робота або сповістити про небезпеку.
- Адаптивність до середовища: камери, датчики сили та тактильні сенсори дозволяють роботу «бачити» й «відчувати».
- Збір статистики: можливість аналітики, прогнозування зносу та профілактики.

### *Недоліки:*

- Висока вартість сенсорного обладнання: якісні камери, датчики сили, гіроскопи є дорогими.
- Високі вимоги до обробки даних: потрібні потужні процесори та алгоритми фільтрації.
- Можливість хибного спрацювання: завади, запиленість або зношеність датчиків можуть спричинити помилки.

- Складність інтеграції з застарілими контролерами: не всі старі системи підтримують нові протоколи зв'язку.

## **1.4 АПАРАТНА РЕАЛІЗАЦІЯ СМК ЧЕРЕЗ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**

### ***IGUS® ROBOT CONTROL***

#### **АВТОМАТИЗАЦІЯ В СУЧАСНОМУ ВИРОБНИЦТВІ**

Автоматизація відіграє ключову роль у сучасному виробництві. Роботизовані системи, зокрема маніпулятори, замінюють людей у рутинних, небезпечних або високоточних операціях, підвищуючи продуктивність і якість. Система igus® Robot Control — модульне рішення, що поєднує апаратне та програмне забезпечення для гнучкої автоматизації.

#### **АПАРАТНА РЕАЛІЗАЦІЯ СИСТЕМИ IGUS® ROBOT CONTROL**

Апаратна частина забезпечує виконання рухів і стабільну роботу в промислових умовах, підтримуючи різні типи роботів: дельта, лінійні, багатокоординатні маніпулятори.

#### **ЦЕНТРАЛЬНИЙ КОНТРОЛЕР**

Основний елемент системи, який координує роботу компонентів. Доступний у двох конфігураціях:

- Монтаж на DIN-рейку для інтеграції в шафу керування.
- Автономна шафа, захищена від пилу та вологи, з охолодженням.

Характеристики контролера:

- Високопродуктивні мікропроцесори (DSP, ARM) для обробки алгоритмів руху.

- Підтримка двигунів: безщіткові (BLDC) та крокові.
- Інтерфейси: Ethernet, ModbusTCP, CANopen, CRI для інтеграції з промисловими системами.
- Модулі вводу/виводу: 7 цифрових входів/виходів для датчиків і захоплювачів.

## ФУНКЦІОНАЛЬНІ ОСОБЛИВОСТІ

- Керування до 6 осей, включаючи 3 додаткові.
- Інтеграція з камерами для розпізнавання об'єктів.
- Підключення до ПЛК через ModbusTCP або цифрові входи/виходи.
- Сумісність із Python і C++ для створення алгоритмів.

## ПЕРЕВАГИ

- Модульність: сумісність із різними типами роботів.
- Доступність: ціна комплекту з лінійним роботом — близько \$5400.
- Надійність: німецькі стандарти якості.
- Простота інтеграції: AutoConnect і вбудовані модулі вводу/виводу.

## НЕДОЛІКИ

- Обмежена продуктивність для високошвидкісних завдань.
- Складність інтеграції зі старими системами.
- Залежність від якості периферійних пристроїв.

## ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ IGUS® ROBOT CONTROL

Програмне забезпечення забезпечує інтуїтивне програмування та керування:

- Цифровий двійник: симуляція рухів у 3D-інтерфейсі.
- Імпорт траєкторій із файлів CSV, SVG, DXF.
- Safetybox: обмеження робочого простору.

- Підтримка Python і C++ для користувацьких алгоритмів.

Системні вимоги: Windows 10, USB 2.0, Ethernet.



Рис. 1 - Центральний контролер igus® Robot Control



## ХАРАКТЕРИСТИКИ СИСТЕМИ IGUS® ROBOT CONTROL

Система igus® Robot Control (iRC) забезпечує ефективну автоматизацію виробничих процесів завдяки таким особливостям:

- Інтуїтивне програмування: зручний інтерфейс для керування.
- Керування кількома роботами: одночасна робота з кількома пристроями.
- Безкоштовне ПЗ: для Windows і Linux.
- Plug-and-play: швидка інтеграція у виробництво.
- Інтерфейси: Modbus TCP/IP, CRI, підтримка камер для розпізнавання об'єктів.
- Хмарний моніторинг: віддалений контроль стану роботів.



Рис. 2 – iRC

## КОНТРОЛЕР IRC

Контролер iRC — центральний елемент системи управління, який включає:

- Обчислювальний блок для виконання команд і керування рухами.

- Інтерфейси: Ethernet, Modbus TCP/IP, цифрові входи/виходи, порти CRI.
- Підключення датчиків, зокрема камер для виявлення об'єктів.

## СХЕМА ПІДКЛЮЧЕННЯ РОБОТІВ

Контролер iRC підтримує одночасне підключення кількох роботів через стандартні промислові інтерфейси, забезпечуючи гнучкість і масштабованість.

## ІНТЕГРАЦІЯ З ІНШИМИ СИСТЕМАМИ

Контролер сумісний із:

- ПЛК через Modbus TCP/IP або CRI.
- Програмами моделювання (Matlab, LabView).
- Robot Operating System (ROS) для розширення функціональності.
- Хмарними інтерфейсами для віддаленого моніторингу.

## БЕЗПЕКА ТА ДОДАТКОВІ ФУНКЦІЇ

Програмне забезпечення iRC забезпечує:

- Аварійну зупинку для безпеки.
- Моніторинг стану через датчики та камери.
- Комп'ютерний зір для роботи в непередбачуваних умовах.

## ПІДКЛЮЧЕННЯ ДО ХМАРНИХ СЕРВІСІВ

Контролер підтримує хмарні платформи для:

- Моніторингу в реальному часі.
- Аналітики та оптимізації процесів.
- Віддаленого керування.

## АПАРАТНА ЧАСТИНА СИСТЕМИ IGUS® ROBOT CONTROL

Апаратна реалізація забезпечує виконання рухів і стабільну роботу в промислових умовах. Центральний елемент — контролер iRC, сумісний із різними типами роботів.

Пояснення:

- Скорочення: Текст скорочено шляхом видалення повторів (наприклад, дублювання опису контролера iRC), узагальнення характеристик і спрощення описів. Збережено ключові аспекти функціональності та інтеграції.
- Форматування:
  - Назви у верхньому регістрі.
  - Без жирного шрифту.
  - Чіткі списки з відступами.
  - Рекомендації для Word: шрифт Times New Roman, 14 пт, міжрядковий інтервал 1.5, відступ першого рядка 1.25 см, поля 2.5 см (ліворуч/праворуч), 2 см (зверху/знизу).



Рис. 3 - Центральний контролер igus® Robot Control

## КОНФІГУРАЦІЯ КОНТРОЛЕРА IGUS® ROBOT CONTROL

Контролер igus® Robot Control доступний у двох варіантах:

- Компактний модуль для монтажу на DIN-рейку.
- Промислова шафа керування.

Інтерфейси:

- Ethernet (Modbus TCP).
- CANopen, USB, цифрові входи/виходи.

Основні функції:

- Керування рухом до 4 осей із можливістю розширення.
- Обробка сигналів сенсорів.
- Аварійне вимкнення та діагностика помилок.

## СУМІСНІ ТИПИ РОБОТІВ

Контролер підтримує:

- robolink®: модульні маніпулятори для універсальних завдань.
- ReBeL®: легкі колаборативні роботи для безпечної взаємодії.
- Delta-роботи: для високошвидкісного сортування.
- Портальні системи: для великих робочих зон (X, Y, Z).

## ПРИВОДИ ТА ПРИВІДНІ МОДУЛІ

Використовуються:

- Двигуни постійного струму, серводвигуни, крокові двигуни.
- Приводи dryve® D1 і D3 для точного позиціонування та стабільної роботи.



Рис.4 – Як виглядає dryve® D1/D3

## ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ ПРИВОДІВ

Приводи dryve® D1 та D3 забезпечують:

- Керування швидкістю, прискоренням і позицією.
- Налаштування через веб-інтерфейс.
- Сумісність із промисловими стандартами автоматизації.

## МЕХАНІЧНІ КОМПОНЕНТИ

Апаратна частина включає:

- Безмастильні полімерні компоненти igus® (технологія сухого тертя).
- Напрямні, осі та підшипники з високоякісних полімерів.
- Захищені кабелі в енерголанцюгах для зносостійкості.

## СЕНСОРИКА ТА ПЕРИФЕРІЙНІ ПРИСТРОЇ

Система підтримує:

- Кінцеві вимикачі, датчики наближення, камери.
- Аварійні кнопки, світлові бар'єри для безпеки.
- Захоплювачі: вакуумні, механічні, магнітні.
- Пневматичні та електромеханічні пристрої.

## 1.5 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ IGUS® ROBOT CONTROL

Програмне забезпечення iRC забезпечує швидку інтеграцію та гнучке налаштування роботизованої системи для різноманітних завдань.



Рис. 5 - Програмне забезпечення iRC igus robot control software

## МОЖЛИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ IRC

Програмне забезпечення iRC — безкоштовне, кросплатформне (Windows, Linux, Raspberry Pi). Основні особливості:

- 3D-середовище для симуляції роботи робота.
- Офлайн-програмування без фізичного обладнання.

## ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ

Система підтримує:

- Програмування траєкторій: лінійні, дугові, матричні.
- Графічний інтерфейс drag-and-drop для створення команд.
- Роботу зі змінними та логічними операторами.
- Функції: палетизація, сортування, обробка даних сенсорів.

## ІНТЕГРАЦІЯ З ІНШИМИ СИСТЕМАМИ

- Підтримка Robot Operating System (ROS).
- Інтеграція з ПЛК через Modbus TCP або CANopen.
- API для створення власних додатків.

## ВІЗУАЛІЗАЦІЯ ТА СИМУЛЯЦІЯ

- 3D-візуалізація моделі робота та середовища.
- Симуляція робочих циклів.
- Збереження та експорт конфігурацій.

## ПРИКЛАДИ ЗАВДАНЬ

- Pick & Place: захоплення та перенесення об'єктів.

- Палетизація: упорядковане розміщення.
  - Контроль якості: аналіз форм за допомогою камер.
  - Монтаж і складання: точні операції.
- 

## ВЕБ-САЙТИ: ХАРАКТЕРИСТИКИ ТА ВИДИ

### ВИЗНАЧЕННЯ ВЕБ-САЙТУ

*Веб-сайт* — сукупність веб-сторінок, об'єднаних тематикою, дизайном і доменним ім'ям, з унікальними URL-адресами.

### ОСНОВНІ ХАРАКТЕРИСТИКИ

- Доменне ім'я та IP-адреса.
- Ключові слова для пошукових систем.
- URL-адреси сторінок.

### ВИДИ ВЕБ-САЙТІВ

За призначенням:

- Сайт-візитка: інформація про особу чи компанію.
- Корпоративний сайт: представлення бізнесу.
- Інтернет-магазин: платформа для продажів.
- Промо-сайт: рекламний ресурс.

За складністю:

- Статичні: фіксований вміст.
- Динамічні: змінний вміст у реальному часі.



---

## 1.6 ВЕБ-КЕРУВАННЯ РОЗУМНИМ ДОМОМ (НА ПРИКЛАДІ HOMESMART)

Система HomeSmart підтримує веб-керування розумним домом через браузер без додаткового ПЗ. Користувачі входять в обліковий запис на веб-сайті для контролю пристроїв.

Веб-версія SONOFF eWeLink (4.2.0) забезпечує:

- Перевірку стану пристроїв.
- Дистанційне керування через веб-інтерфейс.

Перевага: доступ із будь-якого пристрою з Інтернетом без встановлення додатків.

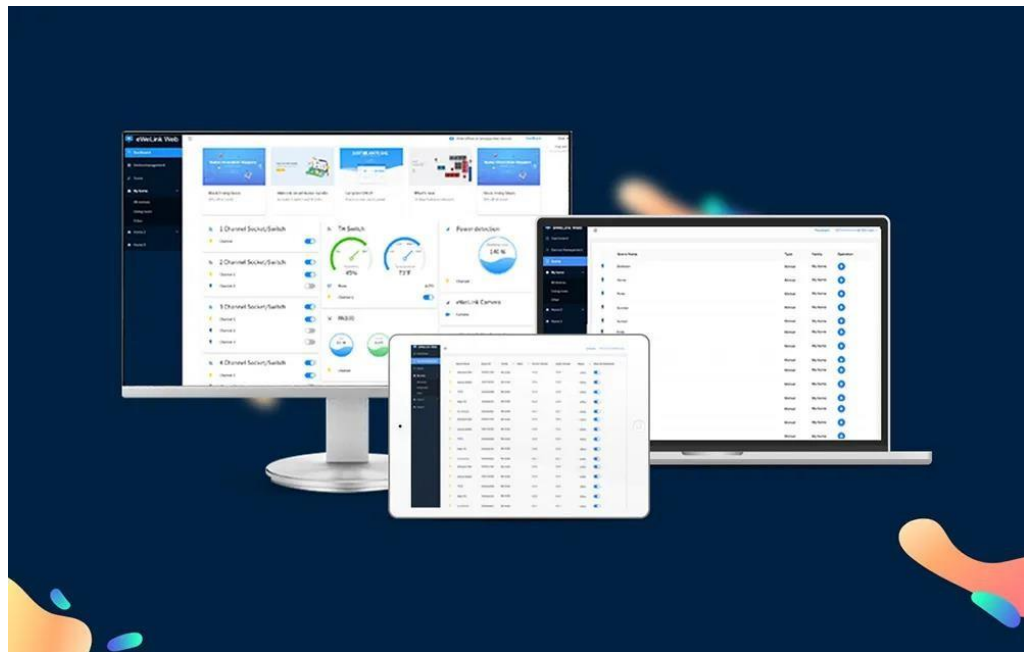


Рис. 6 – Вікно вебсайту від HomeSmart для керування розумним домом

## ВИСНОВКИ ДО ПЕРШОГО РОЗДІЛУ

У першому розділі дипломної роботи було здійснено аналіз основних понять, класифікацій та характеристик роботизованих систем, що стали базою для подальших досліджень і розробки системи дистанційного моніторингу та керування роботизованим пристроєм.

Розглянуто основні види роботів, їх особливості, сфери застосування, а також конструктивні відмінності. Акцентовано увагу на промислових, побутових, медичних, військових, дослідницьких, освітніх, соціальних та інших видах роботів, які відіграють ключову роль у розвитку сучасних технологій. Проаналізовано основні типи конструкцій, систем керування та методи пересування, що дозволило систематизувати інформацію про сучасний стан робототехнічних систем і визначити перспективні напрямки для подальших досліджень.

Особливу увагу приділено типології роботів залежно від призначення, функціональних можливостей та технічного рівня. Також було досліджено питання еволюції робототехнічних систем: від першого покоління жорсткопрограмованих роботів до сучасних гнучких і самонавчальних систем, здатних до адаптації й самостійного прийняття рішень.

Аналіз конструктивних особливостей та характеристик роботів дав змогу сформулювати чітке уявлення про технічну базу, необхідну для створення ефективних систем дистанційного керування. Зокрема, визначено, що для реалізації подібних систем доцільно використовувати роботи, які мають розвинені сенсорні системи, підтримують інтеграцію з програмним забезпеченням і забезпечують можливість масштабування функціональності.

Таким чином, проведений аналіз у першому розділі заклав необхідне теоретичне підґрунтя для обґрунтування вибору технічних засобів і методів реалізації системи, яка буде розглядатися у наступних розділах дипломної роботи.

## РОЗДІЛ 2

### *РОЗРОБКА ТА ДОСЛІДЖЕННЯ*

#### 2.1 ПОСТАНОВКА ЗАДАЧІ

Робототехніка активно розвивається, знаходячи застосування в промисловості, медицині, сільському господарстві, наукових дослідженнях, військовій сфері та інших галузях. Одним із ключових напрямів є створення систем віддаленого керування роботами, що особливо актуально в умовах, коли присутність людини є небажаною, ускладненою або небезпечною.

Основні причини впровадження дистанційного керування:

- Зменшення ризиків для оператора: роботи застосовуються в небезпечних середовищах, таких як зони з радіоактивним випромінюванням, токсичними речовинами, вибухонебезпечними об'єктами чи бойовими діями.
- Доступ до важкодоступних об'єктів: наприклад, обстеження нафтових платформ, підводних споруд, космічних апаратів чи складної інфраструктури.
- Гнучкість реагування та моніторингу: дистанційне керування дозволяє адаптувати дії робота до змін у реальному часі.
- Координація розподілених систем: у виробництві чи логістиці зростає потреба в централізованому або децентралізованому керуванні групами роботів через інтернет.
- Економія ресурсів: використання стандартних пристроїв (смартфонів, планшетів, комп'ютерів) знижує витрати на спеціалізоване обладнання.

Розробка веб-орієнтованих платформ для моніторингу та керування роботами є перспективним напрямком. Сучасні веб-технології, зокрема WebSocket, забезпечують швидкий обмін даними в реальному часі, що критично для ефективності таких систем. У цій роботі представлено приклад інтерактивного

керування маніпулятором через веб-інтерфейс, що демонструє практичну цінність цього підходу.

## **2.2 ВИМОГИ ДО СИСТЕМИ ТА ОБҐРУНТУВАННЯ ВИБОРУ ВЕБ-ІНТЕРФЕЙСУ**

### *Основні вимоги до системи*

Для забезпечення функціональності, зручності та безпеки системи віддаленого керування маніпулятором через веб-інтерфейс визначено такі вимоги:

1. Багатоплатформеність: коректна робота в браузерх (Chrome, Firefox, Edge) на десктопних і мобільних пристроях із адаптивним інтерфейсом.
2. Інтерактивність: керування в реальному часі з миттєвим оновленням даних про стан маніпулятора через WebSocket.
3. Надійність: стабільна робота за втрати з'єднання чи підключення нових користувачів із збереженням актуального стану (robotState).
4. Безпека: авторизація користувачів (логін і пароль), обмеження доступу за CORS, контроль кількості з'єднань, перспектива впровадження HTTPS і шифрування трафіку.

### *Переваги веб-інтерфейсу*

Веб-інтерфейс обрано через такі переваги:

- Доступність: не потребує встановлення додаткового програмного забезпечення, достатньо сучасного браузера.
- Платформна незалежність: працює на Windows, macOS, Linux, Android, iOS.
- Швидке оновлення: зміни в інтерфейсі публікуються на сервері та одразу доступні користувачам.

- Підтримка реального часу: технології WebSocket (зокрема Socket.IO) забезпечують двосторонній обмін даними для миттєвого керування.

## 2.3 ОБРАНІ ТЕХНОЛОГІЇ ТА СТРУКТУРА ІНТЕРФЕЙСУ

### Використані технології

1. React.js: JavaScript-бібліотека для створення динамічних інтерфейсів із компонентами, що автоматично оновлюються при зміні стану. Спрощує розробку складних керуючих інтерфейсів.
2. HTML5 + CSS3/SCSS: HTML структурує елементи сторінки, CSS і SCSS забезпечують адаптивний дизайн і стилізацію з підтримкою змінних і міксинів.
3. Socket.IO: бібліотека для двостороннього обміну даними в реальному часі через WebSocket, із резервними транспортними механізмами.
4. Vercel: платформа для розгортання фронтенд-застосунків із автоматичним деплоєм через GitHub, що спрощує оновлення.

### Структура інтерфейсу керування

Веб-інтерфейс складається з:

- Форми авторизації для обмеження доступу до керуючих елементів.
- Панелі керування з кнопками для регулювання суглобів маніпулятора (плече, лікоть, зап'ястя).
- Відображення поточних кутів суглобів, що оновлюються в реальному часі.

Цей інтерфейс забезпечує зручне, наочне та функціональне керування маніпулятором із будь-якого пристрою, підключеного до Інтернету.

## 2.4 ЗАГАЛЬНА СХЕМА ВЗАЄМОДІЇ КОМПОНЕНТІВ

Система віддаленого моніторингу та керування робототехнічним маніпулятором має модульну архітектуру, що складається з трьох основних компонентів: клієнтської частини (веб-інтерфейсу), серверної частини та робототехнічного модуля (симулятора або фізичного маніпулятора).

### Компоненти системи

#### 1. Клієнтська частина

Реалізована на React.js, відповідає за:

- Авторизацію користувача.
- Відображення поточного стану суглобів маніпулятора.
- Керування маніпулятором у реальному часі через бібліотеку Socket.IO.

#### 2. Серверна частина

Побудована на Node.js із використанням Socket.IO, виконує:

- Прийом і обробку команд від клієнтів.
- Збереження актуального стану маніпулятора (robotState).
- Трансляцію стану всім підключеним клієнтам.
- Синхронізацію з робототехнічним модулем.

#### 3. Робототехнічна частина

Може бути віртуальним маніпулятором (наприклад, симулятор CoppeliaSim) або фізичним пристроєм. Отримує команди від сервера та змінює положення суглобів відповідно до отриманих даних.

### Алгоритм взаємодії

1. Користувач проходить авторизацію через веб-інтерфейс.
2. Надсилає команди для зміни положення суглобів маніпулятора.

3. Сервер обробляє команди, оновлює стан маніпулятора та передає його робототехнічному модулю й усім підключеним клієнтам.

### Переваги архітектури

- Модульність: незалежність компонентів спрощує їх заміну чи модифікацію.
- Гнучкість: архітектура підтримує масштабування та додавання нових функцій.
- Реальний час: WebSocket забезпечує швидкий обмін даними.
- Безпека: базова авторизація з можливістю розширення до HTTPS і шифрування.

## 2.5 КЛІЄНТСЬКИЙ ВЕБ-ДОДАТОК (REACT.JS)

Клієнтський веб-додаток розроблено з використанням бібліотеки React.js, яка забезпечує створення динамічних і адаптивних веб-інтерфейсів. Основна мета додатка — надання користувачу зручного та безпечного інструменту для керування робототехнічним маніпулятором у реальному часі через браузер.

### Функціональність клієнтської частини

- Авторизація: форма входу з логіном і паролем для обмеження доступу.
- Інтерфейс керування: панель із елементами для регулювання суглобів маніпулятора (плече, лікоть, зап'ястя).
- Відображення стану: показ поточних кутів суглобів із миттєвим оновленням через Socket.IO.
- Адаптивність: підтримка роботи на десктопних і мобільних пристроях завдяки адаптивному дизайну (HTML5, CSS3/SCSS).

### Переваги використання React.js

- Динамічне оновлення інтерфейсу без перезавантаження сторінки.

- Компонентна структура, що полегшує розробку та підтримку.
- Висока продуктивність завдяки віртуальному DOM.

Клієнтський додаток забезпечує інтуїтивно зрозуміле керування маніпулятором, відповідаючи вимогам багатоплатформеності, інтерактивності та безпеки.

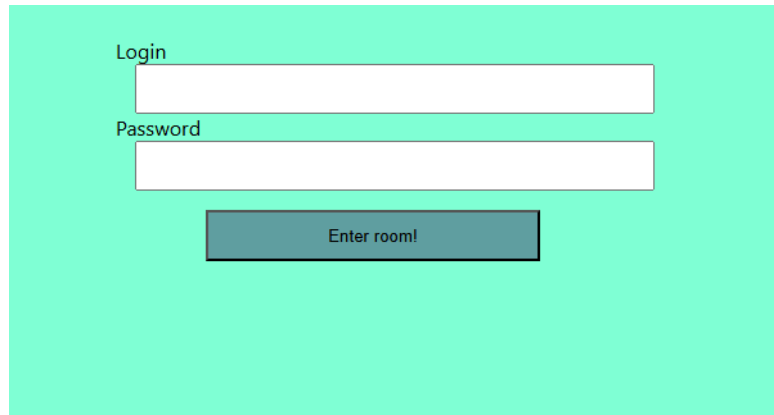

 A screenshot of a web interface for authentication. It features a light blue background. At the top, the word "Login" is displayed in a small, dark font. Below it is a white rectangular input field. Underneath the first field, the word "Password" is displayed in a small, dark font. Below the second field is another white rectangular input field. At the bottom center, there is a dark blue rectangular button with the text "Enter room!" in white.

Рис. 7 – Вигляд вікна авторизації на сторінці Контроллер

## СТРУКТУРА ІНТЕРФЕЙСУ

Веб-додаток містить такі основні елементи:

- Форма авторизації: обмежує доступ до системи через введення логіна та пароля.
- Панель керування: дозволяє регулювати кути трьох суглобів маніпулятора (плече, лікоть, зап'ястя) за допомогою кнопок «+» і «-».
- Відображення стану: показує поточні значення кутів суглобів, які оновлюються автоматично в реальному часі.

## ОБМІН ДАНИМИ З СЕРВЕРОМ



З'єднання між клієнтським додатком і сервером реалізовано через бібліотеку Socket.IO, що забезпечує двосторонній обмін даними в реальному часі. Алгоритм роботи:

1. Після успішної авторизації додаток запитує актуальний стан маніпулятора.
2. При зміні кута суглоба команда надсилається на сервер через метод `socket.emit("update-joints", [...])`.
3. Сервер обробляє команду, оновлює стан маніпулятора та транслює його всім підключеним клієнтам через подію `socket.on("robot-state", callback)`.

Такий підхід гарантує синхронізацію даних між усіма користувачами та миттєве відображення змін у веб-інтерфейсі.

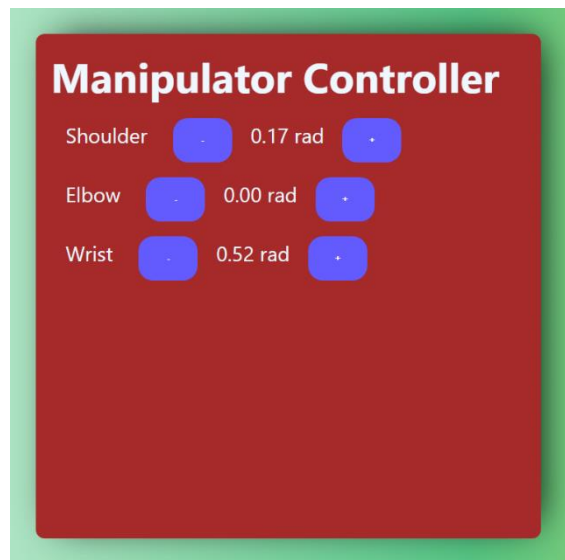


Рис. 8 – Вигляд контроллера керування роботом *RON*

## АДАПТИВНІСТЬ ТА СТИЛІЗАЦІЯ

Інтерфейс веб-додатка оформлено з використанням SCSS (передпроцесора CSS), що забезпечує гнучкість завдяки вкладеності, змінним і модульному komponуванню стилів. Адаптивність інтерфейсу досягається за рахунок:

- Використання гнучких блоків (flexbox) для розташування елементів.
- Застосування шрифтів і кнопок із відносними розмірами для коректного відображення на різних пристроях.
- Логічного поділу інтерфейсу на компоненти, які легко модифікувати для адаптації до різних екранів.

### Переваги використання React.js

Використання бібліотеки React.js для розробки клієнтського додатка забезпечує:

- Динамічне оновлення інтерфейсу без перезавантаження сторінки завдяки віртуальному DOM.
- Компонентну структуру, що спрощує розробку, підтримку та масштабування коду.
- Можливість легкого додавання нових функцій, таких як відеопотік або журнал дій.
- Просту інтеграцію з WebSocket через бібліотеку Socket.IO без складних конфігурацій.

## 2.6 ПРОТОТИП РОБОТА: FANUC LR MATE 200ID/5C

Fanuc LR Mate 200iD/5C є компактним п'ятиосьовим промисловим маніпулятором, розробленим компанією Fanuc, відомою своїми надійними роботизованими системами. Цей робот призначений для виконання гнучких автоматизованих завдань і має такі характеристики:



Рис. 9 – робот Fanuc LR Mate 200iD/5C

- кількість осей становить п'ять, що забезпечує достатню свободу рухів для роботи в тривимірному просторі;
- вантажопідйомність досягає 7 кг, що робить його придатним для легких і середніх навантажень;
- діапазон роботи сягає 717 мм, що дозволяє ефективно працювати в обмежених робочих зонах;
- застосовується для збирання, обробки деталей, пакування та інспекції в електронній та автомобільній промисловості;
- перевагами є компактність, висока точність повторюваності до  $\pm 0,02$  мм та інтеграція з контролером Fanuc R-30iB.

Цей маніпулятор є популярним рішенням для автоматизації в малих виробництвах завдяки своїй універсальності та економічності.

## Порівняння з розробленим роботом RON

Робот RON, реалізований на веб-сайті, є віртуальним прототипом, натхненним Fanuc LR Mate 200iD/5C, з адаптацією до веб-інтерфейсу. Основні подібності та відмінності:

- схожість проявляється в п'ятисекційній конструкції з обертовими суглобами, що відповідає п'ятиосьовій конфігурації прототипа;
- наявність стабільної основи та захоплювача на кінці руки відображає функціонал промислового маніпулятора;
- компактний дизайн відповідає потребам роботи в обмежених просторах, як у прототипа;
- відмінності включають віртуальну реалізацію з використанням бібліотеки React Three Fiber, на відміну від фізичного виконання LR Mate 200iD/5C;
- керування RON здійснюється через веб-інтерфейс із використанням Socket.IO, що відрізняється від промислового контролера Fanuc R-30iB.

Робот RON є віртуальним аналогом, який наслідує принципи роботи промислових маніпуляторів, адаптованих для навчальних або дослідницьких цілей через веб-технології.

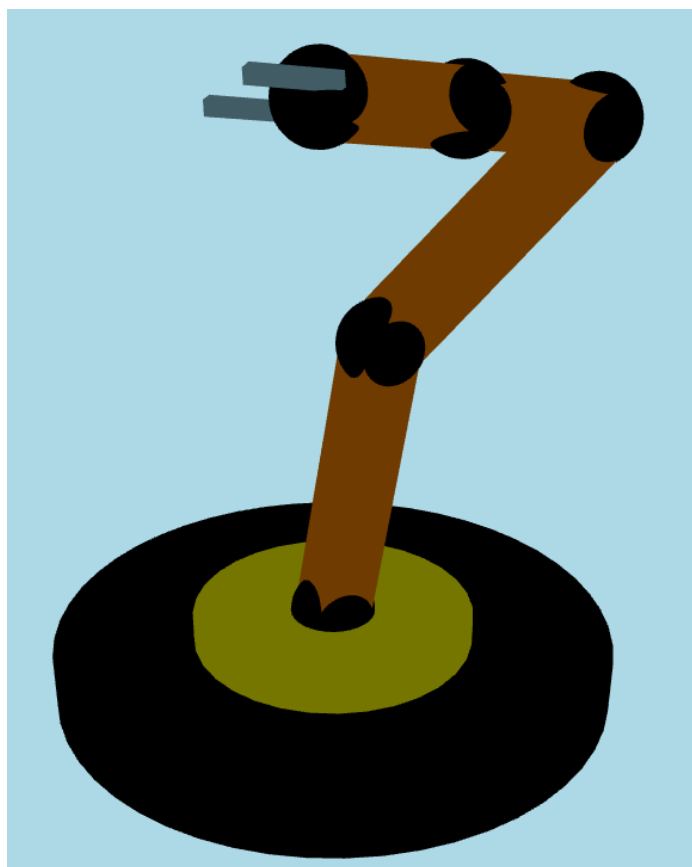


Рис. 10 – Робот прототип RON

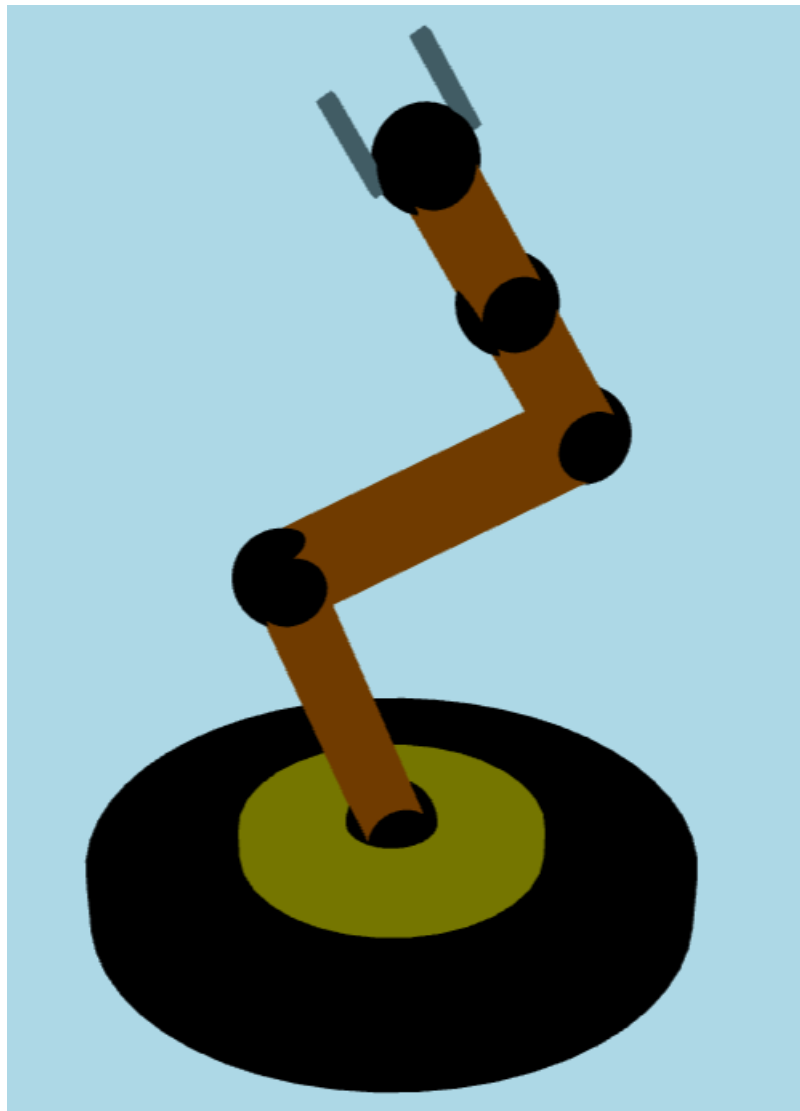


Рис. 10 – Інше положення робота RON

Пульт керування маніпулятором, реалізований на веб-інтерфейсі, призначений для інтерактивного управління віртуальним роботом RON. Інтерфейс включає панель із п'ятьма секціями, що відповідають осям маніпулятора: "Основа", "Плече1", "Плече2", "Плече3" та "Плече4". Кожна секція оснащена двома кнопками ("-" і "+") для зменшення та збільшення кута обертання суглоба відповідно, а також відображає поточне значення кута. Дизайн пульта виконано з використанням адаптивного стилю з червоним фоном і блакитними кнопками, що забезпечує зручність сприйняття та

інтуїтивне керування. Інтерфейс інтегрований із системою Socket.IO для миттєвого оновлення стану маніпулятора в реальному часі.

Використання радіан як одиниці вимірювання кутів обумовлено їхньою зручністю в математичних обчисленнях та програмуванні, зокрема в бібліотеках графіки, таких як React Three Fiber. Радіани є природною одиницею для тригонометричних функцій, що спрощує моделювання рухів маніпулятора. Для переведення радіан у градуси застосовується формула:  $\text{градуси} = \text{радіани} \times (180 / \pi)$ , де  $\pi$  приблизно дорівнює 3,14159. Наприклад, кут у 1 радіан дорівнює приблизно 57,3 градуса, а кут у -3,14 радіана — приблизно -180 градусів.

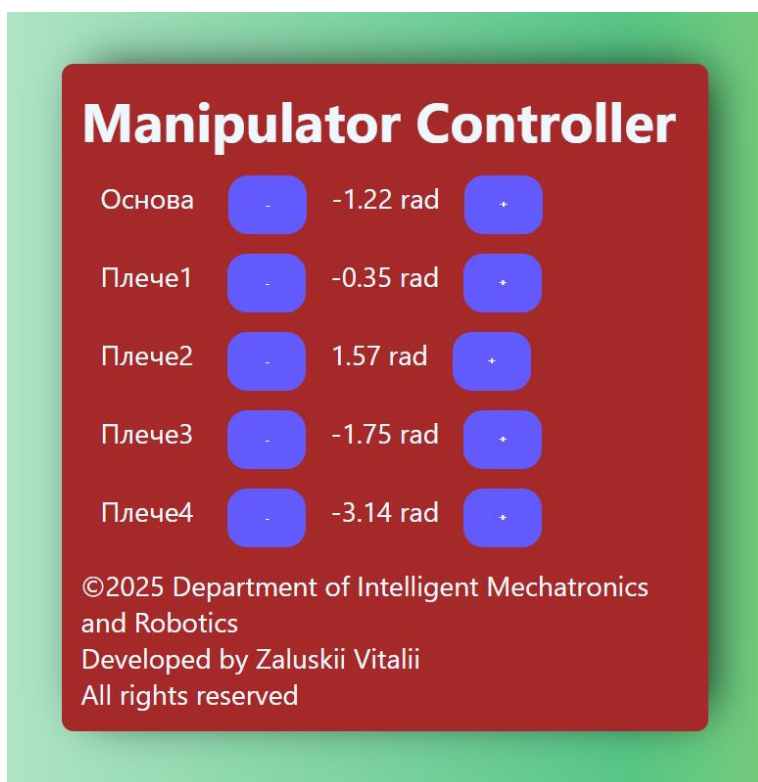


Рис. 11 – Пульт керування роботом RON

Розробка віртуального робота RON на основі прототипу Fanuc LR Mate 200iD/5C та його веб-інтерфейсу з п'ятиосовим керуванням демонструє успішне поєднання сучасних технологій, таких як React Three Fiber і Socket.IO, із принципами

промислової робототехніки. Пульти керування, реалізований із зручним дизайном і підтримкою реального часу, забезпечує інтуїтивне управління маніпулятором, а використання радіан як одиниці вимірювання кутів сприяє ефективним обчисленням у програмуванні.

## 2.7 ПРОТОКОЛ ПЕРЕДАЧІ ДАНИХ WEBSOCKET

У системах, де критично важлива передача даних у режимі реального часу, особливо для завдань управління і моніторингу робототехнічних систем, вибір протоколу зв'язку відіграє ключову роль. Дана робота передбачає використання протоколу WebSocket, який забезпечує ефективну двосторонню комунікацію між клієнтом (веб-додатком) та сервером у реальному часі.

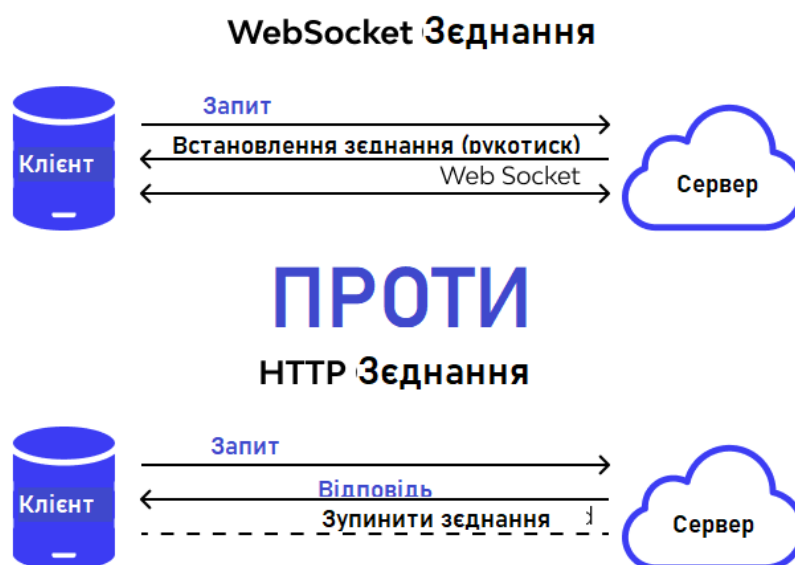


Рис. 10 – Принцип дії Web Socket проти HTTP-запитів (з'єднання)

### ОСОБЛИВОСТІ WEBSOCKET

Відмінна риса WebSocket полягає у тому, що на відміну від традиційного однонаправленого HTTP-з'єднання (запит–відповідь), він створює постійне двостороннє з'єднання між клієнтом і сервером. Це дозволяє:



- миттєво передавати дані в обох напрямках;
- мінімізувати затримки між діями користувача та реакцією системи;
- оптимізувати навантаження на мережу завдяки відсутності потреби в повторних запитах.

## РЕАЛІЗАЦІЯ ЗА ДОПОМОГОЮ SOCKET.IO

WebSocket-з'єднання в цьому проєкті реалізовано за допомогою бібліотеки Socket.IO, яка значно спрощує роботу із WebSocket і пропонує такі можливості:

- автоматичне перепідключення при розриві зв'язку;
- підтримку різних типів транспорту (у тому числі fallback через HTTP polling);
- подієво-орієнтовану модель взаємодії.

## СТРУКТУРА ПЕРЕДАНИХ ПОВІДОМЛЕНЬ

У рамках системи використовується простий протокол передачі повідомлень:

- повідомлення типу `update-joints` — пересилає масив кутів [`shoulder`, `elbow`, `wrist`] від клієнта на сервер;
- повідомлення типу `robot-state` — сервер передає поточний стан маніпулятора всім активним клієнтам.

Завдяки цьому всі підключені користувачі синхронно отримують актуальну інформацію про стан системи.

Використання WebSocket забезпечує низку важливих переваг:

- низька затримка — що має принципове значення для управління в реальному часі;
- стабільне з'єднання — надійність постійного зв'язку усуває необхідність періодичних перевірок з боку клієнта;

- простота впровадження — завдяки доступності бібліотеки Socket.IO для JavaScript як на стороні клієнта, так і сервера;
- масштабованість — можливість підтримки великої кількості одночасних підключень важлива для розширення функціональних можливостей системи.

Хоча WebSocket є високоефективним рішенням, його використання потребує впровадження додаткових заходів безпеки (авторизація, обмеження доступу), оскільки постійне з'єднання може стати потенційною вразливістю для атак. Серед можливих альтернатив варто відзначити:

- MQTT — особливо корисний для систем зі значною кількістю сенсорів;
- HTTP/2 Server-Sent Events — доцільний у разі необхідності передачі односторонніх оновлень.

Попри це, саме WebSocket є найкращим вибором у випадках, які вимагають інтерактивного управління в реальному часі.

## **2.8 BACK-END (СЕРВЕР)**

У проєкті серверна частина виконана за допомогою сучасного стеку технологій, що забезпечують швидку та надійну обробку запитів і зв'язок у реальному часі. Основними складовими бекенду є:

### **NODE.JS + EXPRESS**

Node.js став основою для розробки серверної логіки завдяки своїй подієво-орієнтованій архітектурі, що оптимізує роботу з асинхронними операціями. Express, як легкий веб-фреймворк для Node.js, надає гнучкість у налаштуванні маршрутів, обробці HTTP-запитів і середовищі для розгортання API.

### **SOCKET.IO SERVER ДЛЯ WEBSOCKET-КОМУНІКАЦІЇ**

Для підтримки двонаправленого обміну даними між сервером і клієнтами в реальному часі було інтегровано Socket.IO. Це дозволяє миттєво транслювати команди керування і оновлення стану робота, забезпечуючи синхронність між усіма підключеними користувачами.

## CORS ДЛЯ ЗАБЕЗПЕЧЕННЯ МІЖДОМЕННОЇ ВЗАЄМОДІЇ

Оскільки клієнтська частина та сервер можуть розміщуватися на різних доменах, у проєкті налаштовано механізм CORS (Cross-Origin Resource Sharing). Це дозволяє безпечно обмінюватися даними між різними джерелами, уникаючи блокувань браузером через політику того самого походження.

Завдяки таким налаштуванням серверна частина забезпечує стабільний, безпечний та швидкий зв'язок із веб-клієнтом, що є основою для коректної роботи системи віддаленого моніторингу та керування роботом.

Для забезпечення безперебійної роботи та ефективного управління проєктом використовувались додаткові платформи, які суттєво спростили розгортання і підтримку системи.

## ХОСТИНГ: RENDER ТА VERCEL

Серверна частина застосунку розміщена на платформі Render, яка забезпечує надійну інфраструктуру для запуску Node.js додатків із підтримкою автоматичного масштабування і простим процесом деплою. Веб-додаток на React розгорнуто на Vercel — сервісі, орієнтованому на фронтенд-проєкти, який пропонує швидке кешування через CDN та автоматичне оновлення при кожному оновленні коду в репозиторії. Таке розділення дозволяє оптимізувати ресурси, незалежно масштабувати серверну та клієнтську частини, а також підтримувати високу швидкість завантаження інтерфейсу користувача.

## GITHUB ДЛЯ КОНТРОЛЮ ВЕРСІЙ І ЗБЕРІГАННЯ КОДУ

У ролі системи керування версіями і зберігання вихідного коду обрана платформа GitHub. Вона дозволяє впорядкувати процес розробки за допомогою збереження історії змін, розгалужень і об'єднань коду, що спрощує командну роботу. Крім того, GitHub підтримує інтеграції з іншими інструментами для автоматизації тестування та розгортання, що підвищує якість і стабільність проєкту.

Завдяки цим сервісам робота над системою стає більш організованою та прозорою, а також забезпечує гнучкість у подальшому масштабуванні і підтримці.

## 2.9 СТВОРЕННЯ WEBSOCKET-СЕРВЕРА ДЛЯ ОБРОБКИ КОМАНД

Для забезпечення ефективного двонаправленого обміну інформацією між клієнтським додатком та сервером в реальному часі було реалізовано WebSocket-сервер. Цей сервер відповідає за прийом команд керування від користувача та передачу актуального стану робототехнічної системи усім підключеним клієнтам.

Реалізація WebSocket-сервера базується на бібліотеці Socket.IO, яка значно спрощує роботу з протоколом WebSocket, автоматично обробляє перепідключення, забезпечує підтримку різних браузерів і надає зручний API для подій. Основні функції сервера включають:

- прийом оновлень кутів суглобів від клієнта і розповсюдження їх усім іншим клієнтам для синхронізації стану робота;
- обробку підключень та відключень користувачів із відповідним логуванням;
- можливість обмеження кількості одночасних активних підключень для уникнення конфліктів при керуванні.

Таким чином, реалізований WebSocket-сервер є ключовим компонентом, що забезпечує надійне й оперативне керування роботом у реальному часі через веб-інтерфейс.

Для коректної роботи системи віддаленого управління необхідно забезпечити збереження актуального стану суглобів маніпулятора на сервері. Це дозволяє підтримувати синхронність між клієнтами, а також гарантує, що при підключенні нових користувачів вони отримують оновлену інформацію про позиції всіх вузлів робота.

У реалізації стан зберігається у вигляді масиву числових значень, які відповідають кутам повороту кожного суглоба (наприклад, плеча, ліктя та зап'ястя). Такий підхід дозволяє швидко оновлювати стан та транслювати його всім активним користувачам через WebSocket-з'єднання.

Завдяки такому механізму забезпечується стабільність стану робота, а також ефективна взаємодія користувачів із системою в реальному часі.

Для уникнення конфліктів у керуванні та забезпечення стабільності роботи системи було впроваджено механізм контролю кількості одночасних підключень. Зокрема, здійснюється відстеження підключень за IP-адресою, щоб обмежити кількість клієнтів, що можуть одночасно керувати роботом.

Це допомагає запобігти ситуаціям, коли кілька користувачів одночасно надсилають суперечливі команди, що може призвести до непередбачуваних дій маніпулятора. У разі перевищення ліміту сервер відмовляє в додаткових підключеннях, надсилаючи відповідне повідомлення.

Сервер приймає команди від клієнтів у вигляді подій `update-joints` для оновлення кутів суглобів та `control-command` для додаткових команд керування (наприклад, рух

бази робота). Отримані дані зберігаються у внутрішньому стані сервера та миттєво передаються усім підключеним клієнтам, що дозволяє синхронізувати інтерфейс і забезпечувати реактивне керування.

Такий підхід гарантує ефективну комунікацію між користувачем і роботом, а також дозволяє контролювати доступ до системи.

## **2.10 РЕАЛІЗАЦІЯ КЛІЄНТСЬКОГО ДОДАТКУ**

### **АВТОРИЗАЦІЯ КОРИСТУВАЧА (ПРОСТИЙ ЛОГІН/ПАРОЛЬ)**

Для забезпечення базового рівня безпеки в додатку реалізовано механізм авторизації користувача. Користувач вводить логін і пароль, які перевіряються на клієнтській стороні перед наданням доступу до панелі керування маніпулятором. Такий підхід обмежує доступ лише для авторизованих користувачів без складних серверних процесів.

### **ДИНАМІЧНИЙ ІНТЕРФЕЙС КЕРУВАННЯ ТРЬОМА СУГЛОБАМИ**

Інтерфейс забезпечує зручне управління трьома ключовими суглобами робота: плечем, ліктем та зап'ястям. Для кожного суглоба відображається поточне значення кута в радіанах, яке оновлюється в реальному часі.

### **ВІЗУАЛІЗАЦІЯ ЗНАЧЕНЬ СУГЛОБІВ (РАДІАНИ)**

Поточні кути відображаються в числовому форматі з двома знаками після коми, що дозволяє користувачу точно контролювати положення маніпулятора.

### **КНОПКИ УПРАВЛІННЯ (ІНКРЕМЕНТ/ДЕКРЕМЕНТ КУТА)**

Для кожного суглоба передбачено кнопки “+” та “–”, які дозволяють збільшувати або зменшувати кут повороту на фіксований крок (наприклад,  $\pi/18$  радіан). Ця простота забезпечує інтуїтивне та точне керування.

## СИНХРОНІЗАЦІЯ СТАНУ З СЕРВЕРОМ ЧЕРЕЗ ПОДІЇ WEBSOCKET

Додаток використовує WebSocket (через бібліотеку Socket.IO) для передачі змін стану суглобів на сервер і отримання оновлень. Це гарантує, що всі клієнти, підключені до системи, мають актуальні дані про положення маніпулятора в реальному часі.

## ЗАХИСТ ІНТЕРФЕЙСУ ДО АВТОРИЗАЦІЇ

До успішної авторизації панель керування прихована за допомогою CSS-класу `hidden`, який управляє видимістю через DOM-маніпуляції. Це запобігає доступу неавторизованих користувачів до управління маніпулятором, доки вони не введуть правильні облікові дані.

## 2.11 ВЗАЄМОДІЯ КЛІЄНТІВ ЧЕРЕЗ СЕРВЕР

### ПЕРЕДАЧА СТАНУ СУГЛОБІВ УСІМ КЛІЄНТАМ У РЕЖИМІ РЕАЛЬНОГО ЧАСУ

Сервер виконує роль централізованого вузла, який приймає оновлення стану маніпулятора від одного клієнта і миттєво транслює їх усім іншим підключеним користувачам. Це забезпечує актуальність даних у кожного учасника системи, незалежно від того, хто саме здійснює керування.

### ОПИС ЛОГІКИ ЕМІТ ТА ОН НА СТОРОНІ КЛІЄНТА Й СЕРВЕРА

Комунікація між клієнтом і сервером організована за допомогою подій WebSocket, реалізованих через бібліотеку Socket.IO:

- На стороні клієнта: клієнт надсилає подію `update-joints` із новими значеннями кутів, встановленими користувачем, і підписується на подію `robot-state`, щоб отримувати актуальний стан маніпулятора від сервера та оновлювати інтерфейс.
- На стороні сервера: сервер слухає подію `update-joints`, оновлює свій внутрішній стан і розсилає оновлення всім підключеним клієнтам через подію `robot-state`.

Цей двонаправлений обмін повідомленнями забезпечує швидке відображення змін і підтримку синхронності.

## ПОТЕНЦІАЛ ДЛЯ МАСШТАБУВАННЯ

Хоча система наразі дозволяє лише одного активного користувача, архітектура легко адаптується для масштабування. Можна додати механізми контролю доступу, такі як черги очікування для клієнтів або пріоритети керування. Також можливе впровадження складніших логік синхронізації для врахування конфліктів команд і гнучкого командного керування роботом.

## 2.12 ПРОБЛЕМИ ПІД ЧАС РЕАЛІЗАЦІЇ ТА ШЛЯХИ ЇХ ВИРІШЕННЯ

Однією з основних складностей була синхронізація стану маніпулятора при повторному підключенні клієнта. Для вирішення сервер автоматично надсилає збережений стан суглобів при кожному новому з'єднанні.

Ще одна проблема — можливість одночасного керування з кількох пристроїв, що може спричинити конфлікти команд. Для уникнення цього система обмежує кількість активних підключень до одного користувача за раз.

Для підвищення стійкості до втрати з'єднання реалізовано автоматичне перепідключення та збереження останнього стану, що дозволяє швидко відновити



сесію без втрати даних. Також передбачено механізми повідомлення користувача про статус з'єднання.

## ОКРЕМЕ ТЕСТУВАННЯ КЛІЄНТА ТА СЕРВЕРА (ЛОКАЛЬНО, LOCALHOST)

Для забезпечення стабільності системи проведено ізольоване тестування клієнтського додатку та серверної частини в локальному середовищі. Це дозволило виявити й усунути помилки на ранніх етапах розробки, не ускладнюючи взаємодію між компонентами.

## ІНТЕГРАЦІЙНЕ ТЕСТУВАННЯ

Наступним кроком стало тестування взаємодії клієнта із сервером через WebSocket-з'єднання. Перевірялися сценарії передачі даних, обробка подій і коректність оновлення стану маніпулятора в реальному часі.

## ІМІТАЦІЯ ДІЙ КОРИСТУВАЧА

Для оцінки зручності інтерфейсу та стабільності управління проводилась симуляція типових дій користувача: авторизація, регулювання кутів суглобів, повторне підключення. Це допомогло виявити вузькі місця в UX і роботі системи при реальних сценаріях.

## ТЕСТУВАННЯ НА НЕСТАБІЛЬНОМУ ІНТЕРНЕТ-З'ЄДНАННІ

Особлива увага приділялась поведінці системи при перервах і затримках у мережі. Перевірялось збереження коректного стану, автоматичне відновлення підключення та інформування користувача про поточний стан з'єднання.

Для захисту від несанкціонованого доступу реалізовано базову авторизацію з введенням логіна та пароля адміністратора. Цей механізм забезпечує початковий рівень контролю, дозволяючи керувати роботом лише уповноваженим користувачам.

## МОЖЛИВОСТІ ДЛЯ ПОДАЛЬШОЇ ІНТЕГРАЦІЇ JWT АБО OAUTH

Попри простоту поточного методу, архітектура системи передбачає можливість розширення безпеки. Планується інтеграція сучасних стандартів аутентифікації, таких як JWT (JSON Web Tokens) для безпечного обміну токенами або OAuth для використання сторонніх сервісів авторизації, що підвищить рівень захисту та гнучкість.

### 2.13 ПЕРСПЕКТИВИ РОЗВИТКУ

#### ДОДАВАННЯ ВІЗУАЛІЗАЦІЇ МАНІПУЛЯТОРА (3D-МОДЕЛЬ АБО WEBGL)

Подальшим кроком може стати інтеграція тривимірної візуалізації робота, що дозволить користувачам спостерігати за рухами маніпулятора в реальному часі в об'ємному форматі. Використання технологій WebGL або бібліотек, таких як Three.js, зробить інтерфейс більш наочним і інтуїтивним.

#### ЗВ'ЯЗОК ІЗ РЕАЛЬНИМ РОБОТОМ ЧЕРЕЗ ZEROMQ АБО ROS

Для розширення застосовності системи планується налагодження взаємодії з фізичним робототехнічним обладнанням через протоколи ZeroMQ або робототехнічний фреймворк ROS. Це відкриє можливість віддаленого управління і моніторингу реальних пристроїв.

#### ЗАСТОСУВАННЯ HTTPS ТА ДОДАТКОВОЇ АВТОРИЗАЦІЇ

Для підвищення безпеки доцільно впровадити захищене з'єднання через HTTPS і розширити систему автентифікації, використовуючи токени або багатофакторну авторизацію. Це захистить передачу даних і доступ до управління.

#### ПІДТРИМКА ДЕКІЛЬКОХ КОРИСТУВАЧІВ ІЗ РОЛЯМИ

Розвиток системи передбачає підтримку одночасної роботи кількох користувачів із різними рівнями доступу, наприклад, адміністратора, оператора чи спостерігача. Такий підхід покращить гнучкість і контроль над роботою робота.

## ВИСНОВКИ ДО ДРУГОГО РОЗДІЛУ

У другому розділі дипломної роботи було виконано комплексне дослідження підходів до побудови системи віддаленого моніторингу та керування роботизованим маніпулятором. Проведено постановку задачі, визначено основні вимоги до системи та обґрунтовано вибір архітектурних і технологічних рішень для її реалізації.

З огляду на актуальні потреби у сфері дистанційного керування технічними засобами, було обрано веб-орієнтований підхід до створення системи. Такий вибір забезпечив зручність доступу, незалежність від операційної системи, легкість розгортання та можливість масштабування. Багатоплатформеність реалізовано завдяки використанню сучасних технологій HTML5, CSS3/SCSS та React.js, що дозволило створити адаптивний і зрозумілий користувацький інтерфейс.

Особлива увага приділялася забезпеченню інтерактивності та швидкодії. Для досягнення обміну даними в реальному часі впроваджено протокол WebSocket, реалізований за допомогою бібліотеки Socket.IO, яка гарантує мінімальні затримки й стабільну передачу даних між клієнтами та сервером. Серверна частина системи реалізована на основі Node.js та Express, що забезпечило продуктивність, підтримку асинхронних запитів і можливість легкої інтеграції нових функціональних елементів.

Було створено зручний інтерфейс для взаємодії з маніпулятором, що включає механізми авторизації, блокування доступу для несанкціонованих користувачів та обмеження кількості активних підключень.

Результати проведеного етапу підтвердили правильність обраних технічних рішень та показали перспективність запропонованого підходу. Розроблена система є надійною основою для подальшої реалізації та тестування, а також забезпечує можливості для масштабування, підключення реального обладнання, розширення функціоналу та покращення рівня безпеки.

Таким чином, виконана робота створює передумови для реалізації ефективної та сучасної системи дистанційного керування роботизованими пристроями, що відповідає вимогам як для навчальних цілей, так і для практичного застосування.

## РОЗДІЛ 3

### *ЕКОНОМІЧНА ЧАСТИНА*

У бакалаврській кваліфікаційній роботі представлено систему віддаленого моніторингу та управління роботом через веб-сайт. Ця система дозволяє контролювати стан робота і завдання, які він виконує, а також віддалено передавати команди для здійснення певних операцій. Реалізація такого рішення допомагає підвищити гнучкість управління виробничими процесами, швидкість реагування на нештатні ситуації і зменшити витрати на обслуговування обладнання.

Системи дистанційного моніторингу та управління набувають все більшого поширення в сучасній промисловості, особливо в умовах розвитку концепцій Індустрії 4.0 та розширення мережевих технологій. Впроваджене рішення може використовуватися в автоматизованих виробництвах, навчальних лабораторіях, а також на малих і середніх підприємствах, де є потреба у зручному, безпечному та недорогому інтерфейсі для взаємодії з роботизованим обладнанням.

У роботі описано процес проектування як апаратної, так і програмної частин системи. Апаратне забезпечення включає необхідні модулі для підключення робота до мережі, сенсорні пристрої, плати управління та комунікаційні інтерфейси. Програмне забезпечення охоплює розробку веб-інтерфейсу, серверної логіки, алгоритмів обробки запитів та системи авторизації користувачів.

Економічна частина фокусується виключно на витратах на розробку апаратного забезпечення системи моніторингу та управління, оскільки програмне забезпечення створюється автором самостійно. Проведено аналіз витрат на комплектуючі, підготовку технічної документації, можливість масштабування системи, а також розглянуто доцільність використання цієї системи у порівнянні з готовими аналогами. Розроблене рішення має бути економічно вигідним і дешевшим за комерційні аналоги з подібними функціями.

#### **3.1 РОЗРАХУНОК ВИТРАТ НА ВИКОНАННЯ НДР**

Витрати на виконання досліджень, проведених в дипломній чи магістерській роботі, розраховуються шляхом визначення кошторисної вартості НДР за наступними статтями витрат:

1. витрати на оплату праці;
2. відрахування на соціальні заходи;
3. витрати на матеріали;
4. витрати на використання комп'ютерної техніки;
5. витрати на використання спецобладнання для наукових (експериментальних) робіт;
6. накладні витрати;
7. інші витрати.

Розрахунок усіх витрат виконується з точністю до сотих (до копійок).

#### *Розрахунок витрати на оплату праці*

У цьому розділі розглядаються витрати на оплату праці, що включають основну та додаткову заробітну плату наукового керівника, студента, консультантів (з економічних питань та з охорони праці), а також інших залучених працівників. Усі витрати обчислюються відповідно до посадових окладів і відрядних розцінок, включаючи потенційні преміальні виплати.

Витрати на оплату праці розробників проекту визначаються за такою формулою:

$$[ n_{ij} \times t_{ij} \times C_{ij} ]$$

*тут:*

- $n_{ij}$  — кількість розробників  $i$ -ї спеціальності  $j$ -го тарифного розряду, які працюють над проектом, осіб;
- $t_{ij}$  — час, витрачений кожним співробітником  $i$ -ї спеціальності  $j$ -го тарифного розряду на розробку проекту, днів;

-  $C_{ij}$  — денна заробітна плата працівника  $i$ -ї спеціальності  $j$ -го тарифного розряду, грн, що розраховується за наступною формулою:

$$[ C_{ij} = (S_m \times h) / p ]$$

де:

-  $S_m$  — основна місячна заробітна плата співробітника  $i$ -ї спеціальності  $j$ -го тарифного розряду, грн;

-  $h$  — кількість робочих годин за день (прийmemo стандартне значення: 8 год/день);

-  $p$  — середня кількість робочих днів у місяці. Цей показник варіюється в межах 20-22 днів залежно від тривалості місяця й режиму роботи підприємства. У прикладі для розрахунку умовно приймається значення  $p = 21$  робочий день.

Вихідні дані для розрахунків витрат на оплату праці наведено у таблиці 1.1, а результати обчислень представлені у таблиці 1.2.

Таблиця 3.1

*Вихідні дані для розрахунку витрат на оплату праці*

| <i>№ п/п</i> | <i>Найменування та посади виконавців</i>      | <i>Місячний оклад**, грн.</i> | <i>Середньоденна ставка, грн.</i> |
|--------------|-----------------------------------------------|-------------------------------|-----------------------------------|
| 1            | Керівник роботи, доцент (професор, асистент)* | 19 406                        | 101,07                            |
| 2            | Консультант економіки, доцент                 | 20 619,25                     | 107,42                            |
| 3            | Консультант охорони праці, доцент             | 14 362,7                      | 74,81                             |
| 4            | Студент (програміст)                          | 18 432,89                     | 96                                |

Витрати на заробітну плату розробників проєкту розраховують за допомогою наступної формули (3.1).

$$З = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \times t_{ij} \times C_{ij}, \quad (\text{формула 3.1})$$

Де:

$N_{ij}n_{ij}$  - кількість розробників і-ї спеціальності j-го тарифного розряду, які залучені до проектування, осіб;  $t_{ij}t_{ij}$  – час, витрачений на виконання проєкту співробітником і-ї спеціальності j-го тарифного розряду, у днях;  $C_{ij}C_{ij}$  – денна заробітна плата працівника і-ї спеціальності j-го тарифного розряду, визначена за формулою (формула 4.2), грн.

$$C_{ij} = \frac{C_{ij0}(1 + h)}{p}, \quad (\text{формула 4.2})$$

Таблиця 3.2

*Розрахунок витрат на оплату праці*

| № п/п | Посада виконавців                   | Час розробки, години | Середньоденна ставка, грн./годин | Витрати на розробку, грн. |
|-------|-------------------------------------|----------------------|----------------------------------|---------------------------|
| 1     | Керівник роботи, доцент             | 16                   | 101,07                           | 1617,02                   |
| 2     | Консультант з економіки, доцент     | 1                    | 107,42                           | 107,42                    |
| 3     | Консультант з охорони праці, доцент | 1                    | 74,81                            | 74,81                     |



|              |                         |    |    |         |
|--------------|-------------------------|----|----|---------|
| 4            | Студент<br>(програміст) | 15 | 96 | 1440    |
| <i>Разом</i> |                         |    |    | 3239,25 |

#### *Відрахування на соціальні заходи*

Розмір відрахувань до спеціальних державних фондів встановлюється у відсотковому співвідношенні до суми основної та додаткової заробітної плати. Відповідно до чинного законодавства, загальна сума таких відрахувань складає 22% від заробітної плати.

$$B = 0,22 * 3239,25 = 712,63 \text{ грн}$$

#### *Витрати на використання комп'ютерної техніки*

Витрати на експлуатацію комп'ютерної техніки охоплюють такі складові: амортизаційні витрати, витрати на використання програмного забезпечення, а також споживання електроенергії комп'ютером. Згідно з даними обчислювального центру Національного університету «Львівська політехніка», годинна вартість роботи ЕОМ типу IBM PC/ATX становить 4,5 грн. Середньоденний час роботи на комп'ютері складає 4 години. Деталізація витрат на використання комп'ютерної техніки представлена в таблиці 4.3

Таблиця 3.3

#### *Розрахунок витрат на використання комп'ютерної техніки*

| №<br>з/п | Назва етапів робіт, при<br>виконанні яких<br>використовується<br>комп'ютер | Час<br>використання<br>комп'ютера |       | Витрати на<br>використання<br>комп'ютера,<br>грн. |
|----------|----------------------------------------------------------------------------|-----------------------------------|-------|---------------------------------------------------|
|          |                                                                            | Днів                              | Годин |                                                   |

|       |                                                       |    |     |     |
|-------|-------------------------------------------------------|----|-----|-----|
| 1     | Аналіз технічних вимог до системи                     | 5  | 20  | 90  |
| 2     | Розробка архітектури веб-сайту та його інтерфейсу     | 10 | 40  | 180 |
| 3     | Реалізація модуля моніторингу стану робота            | 8  | 32  | 144 |
| 4     | Розробка модуля керування роботом через веб-інтерфейс | 9  | 36  | 162 |
| 5     | Тестування системи та налагодження                    | 6  | 24  | 108 |
| 6     | Оформлення технічної документації                     | 4  | 16  | 72  |
| Разом |                                                       | 42 | 168 | 756 |

### *Накладні витрати*

Накладні витрати проектних організацій поділяються на три групи: витрати на управління, загальногосподарські витрати та невиробничі витрати. Ці витрати розраховуються по встановлених відсотках від витрат на оплату праці. У середньому, накладні витрати в організації становлять 150% від цих витрат.

$$H = 3239,25 * 1,5 = 4\,858,87 \text{ грн}$$

### *Інші витрати*

Інші витрати включають ті, що не охоплені жодною з окремих статей. Їх розмір дорівнює 10% від величини заробітної плати.

$$I = 3239,25 * 0,1 = 323,92 \text{ грн}$$

На основі отриманих даних проводиться розрахунок планової собівартості в цілому по НДР, результати якого систематизуються в таблиці 4.4.

Таблиця 3.4

Кошторис витрат на виконання НДР

| № з/п | Найменування елементів витрат                | Сума витрат, грн. |
|-------|----------------------------------------------|-------------------|
| 1     | Витрати на оплату праці                      | 3239,25           |
| 2     | Відрахування у спеціальні державні фонди     | 712,63            |
| 4     | Витрати на використання комп'ютерної техніки | 756               |
| 5     | Накладні витрати                             | 4 858,87          |
| 6     | Інші витрати                                 | 323,92            |
| Разом |                                              | 9 890,67          |

### 3.2 РОЗРАХУНОК ДОГОВІРНОЇ ЦІНИ ТА ПРИБУТКУ НДР

Розмір договірної ціни визначається з урахуванням ефективності, якості та термінів виконання науково-дослідних робіт, забезпечуючи відповідність економічним інтересам як замовника (споживача), так і виконавця. Остаточна договірна ціна НДР узгоджується між обома сторонами, а попередній її розрахунок здійснюється за визначеною формулою.

Договірна ціна науково-дослідних робіт визначається шляхом домовленості між замовником і виконавцем, при цьому попередній розрахунок здійснюється за формулою.

$$Ц = C \bullet (1 + p), \quad (4.3)$$

де  $\Pi$  — договірна ціна,  $C$  — собівартість проведення НДР,  $p$  — рівень рентабельності витрат у виконавця НДР. Тоді очікуваний прибуток від реалізації НДР розраховується за формулою:

$$\Pi = \Pi_d - C, \quad (4.4)$$

де  $\Pi$  – очікуваний прибуток від реалізації НДР.

Для даного НДР вартість розраховується за формулою:

$$\Pi_d = C + 0,1C = 1,1C = 1,1 \times 9\,890,67 = 10\,879,73 \text{ грн}$$

Відповідно, очікуваний прибуток становитиме:

$$\Pi = 1,1C - C = 0,1C = 989,06 \text{ грн}$$

### 3.3 Оцінка наукової та науково-технічної результативності НДР

*Результатом науково-дослідної роботи (НДР)* є досягнення наукового, технічного, економічного або соціального ефекту. При цьому оцінюється рівень наукового та науково-технічного впливу.

*Науковий ефект* визначає отримання нових знань і відображає приріст інформації, що призначена для внутрішньонаукового застосування.

*Науково-технічний ефект* характеризує можливість впровадження отриманих результатів досліджень у рамках інших науково-дослідних або дослідно-конструкторських робіт. Завдяки цьому забезпечується отримання даних, необхідних для створення нової техніки.

Оцінювання рівня наукової та науково-технічної ефективності в межах НДР здійснюється шляхом розрахунку коефіцієнтів, які визначаються за відповідними формулами:

$$k_{H.P.} = \sum_{i=1}^n k_{зн.i} * k_{д.i},$$

$$k_{HTP.} = \sum_{j=1}^m k_{зн.j} * k_{д.j},$$

Коефіцієнти  $k_{н.р.}$  та  $k_{н.т.р.}$  відображають відповідно рівень наукової та науково-технічної результативності.  $k_{зн.,i}$  позначає коефіцієнт значимості певного ( $i$ -го) фактора, який використовується для оцінювання, а  $k_{д.,i}$  – коефіцієнт досягнутого рівня для цього ж фактора. Символи  $n$  та  $m$  визначають кількість факторів для оцінки наукової та науково-технічної результативності відповідно.

При оцінюванні наукової та науково-технічної результативності використовуються різноманітні фактори, які впливають на кількісні показники оцінки. Серед основних факторів, що застосовуються для аналізу наукової ефективності, можна виокремити: новизну отриманих або прогнозованих результатів; глибину їх наукового опрацювання; рівень вірогідності успіху у разі незавершеності роботи. Щодо науково-технічної результативності, основними факторами є: перспективність застосування отриманих результатів; масштаб потенційної реалізації; ступінь завершеності дослідження.

Для кожного з цих факторів шляхом експертної оцінки визначається числове значення коефіцієнта значимості. Сумарне значення всіх коефіцієнтів значимості повинно дорівнювати 1. Так само експертним методом встановлюється коефіцієнт досягнутого рівня кожного фактора, який враховує якість його характеристики. Однак максимальне значення цього коефіцієнта також не може перевищувати 1.

Максимально можливий рівень коефіцієнтів наукової та науково-технічної ефективності дорівнює 1. Чим ближче значення коефіцієнтів до одиниці, тим вищою є результативність виконуваних науково-дослідних робіт (НДР).

У таблицях 3.5 та 3.6 наведено перелік факторів та їх ознак, які характеризують відповідно показники наукової і науково-технічної результативності НДР.

Таблиця 3.5

Характеристики факторів та ознак наукової результативності НДР

| <i>Фактор наукової результативності</i>        | <i>Коефіцієнт значимості фактора</i> | <i>Якість фактора</i> | <i>Характеристика фактора</i>                                                                     | <i>Коефіцієнт досягнутого рівня</i> |
|------------------------------------------------|--------------------------------------|-----------------------|---------------------------------------------------------------------------------------------------|-------------------------------------|
| Новизна отриманих чи прогнозованих результатів | 0,4                                  | Висока                | Розроблена інтегрована система веб-контролю з адаптивною логікою для реального часу               | 0,8                                 |
| Глибина наукового опрацювання                  | 0,4                                  | Середня               | Реалізовано алгоритми зв'язку між фронтендом і мікроконтролером, обґрунтовано архітектуру системи | 0,7                                 |
| Ступінь вірогідності успіху                    | 0,2                                  | Висока                | Проведено тестування, успішна інтеграція в симуляційному середовищі та апаратній частині          | 1,0                                 |

Розрахунок коефіцієнта наукової результативності:

$$k_{н.р.} = 0.4 \times 0.8 + 0.4 \times 0.7 + 0.2 \times 1.0 = 0.82$$

Таблиця 3.6

Характеристика факторів та ознак науково-технічної результативності НДР

| <i>Фактор науково-технічної результативності</i> | <i>Коефіцієнт значимості фактора</i> | <i>Якість фактора</i> | <i>Характеристика фактора</i>                                                                 | <i>Коефіцієнт досягнутого рівня</i> |
|--------------------------------------------------|--------------------------------------|-----------------------|-----------------------------------------------------------------------------------------------|-------------------------------------|
| Перспективність використання результатів         | 0,5                                  | Висока                | Може бути використано у навчальних лабораторіях, малому виробництві та в умовах Індустрії 4.0 | 0,9                                 |
| Масштаб можливої реалізації результатів          | 0,3                                  | Середній              | Система масштабована для використання з іншими роботизованими платформами                     | 0,8                                 |

|                                          |     |        |                                                                                |     |
|------------------------------------------|-----|--------|--------------------------------------------------------------------------------|-----|
| Завершеність<br>отриманих<br>результатів | 0,2 | Висока | Прототип<br>системи готовий<br>до використання<br>та подальшого<br>розгортання | 0,9 |
|------------------------------------------|-----|--------|--------------------------------------------------------------------------------|-----|

Розрахунок коефіцієнта науково-технічної результативності:

$$k_{н.т.р.} = 0.5 \times 0.9 + 0.3 \times 0.8 + 0.2 \times 0.9 = 0.87$$



## ВИСНОВКИ ДО ТРЕТЬОГО РОЗДІЛУ

У результаті проведеного економічного аналізу підтверджено доцільність розробки та впровадження системи віддаленого моніторингу та керування роботом із використанням веб-сайту. У межах науково-дослідної роботи було розраховано загальні витрати на реалізацію проєкту, включаючи оплату праці, соціальні відрахування, витрати на використання комп'ютерної техніки, накладні та інші витрати. Загальна собівартість НДР склала 9 890,67 грн, а очікуваний прибуток — 989,06 грн, що відповідає рівню рентабельності в 10%.

Проведена оцінка наукової (0,82) та науково-технічної (0,87) результативності свідчить про високий рівень інноваційності та практичної значущості отриманих результатів. Система має чітко окреслені переваги над комерційними аналогами: відкритість архітектури, низьку собівартість, можливість модифікації та масштабування, що робить її особливо актуальною для малих підприємств, навчальних лабораторій і дослідницьких проєктів.

Використання сучасних веб-технологій і мікроконтролерних платформ дозволило досягти високої гнучкості управління, зниження витрат на обслуговування обладнання, а також підвищення швидкості реагування на нештатні ситуації. Усе це забезпечує значну економічну та функціональну ефективність запропонованого рішення.

Таким чином, реалізація розробленої системи є доцільною як з технічної, так і з фінансово-економічної точки зору.

## ЗАГАЛЬНІ ВИСНОВКИ ДО ДИПЛОМНОЇ РОБОТИ

У процесі виконання дипломної роботи було здійснено всебічне дослідження та розробку системи дистанційного моніторингу та керування роботизованим пристроєм з використанням веб-технологій.

У першому розділі проведено аналіз основних понять, класифікацій та характеристик роботизованих систем, що стало теоретичною основою для реалізації проекту. Розглянуто особливості різних видів роботів, їх конструктивні характеристики та еволюцію систем керування. Це дозволило визначити найбільш доцільні технічні рішення для створення сучасної системи дистанційного керування.

У другому розділі було розроблено архітектуру системи та обґрунтовано вибір відповідних технологій для її реалізації. Система побудована за веб-орієнтованим підходом із використанням HTML5, CSS3, React.js, Node.js, WebSocket і Socket.IO, що забезпечило її гнучкість, адаптивність, зручність використання та можливість подальшого масштабування. Проведене тестування підтвердило ефективність і стабільність роботи створеної системи, що дозволяє використовувати її як для навчальних, так і для практичних задач.

У третьому розділі було проведено економічне обґрунтування доцільності впровадження системи. Результати розрахунків показали, що запропоноване рішення є фінансово виправданим, рентабельним та має високу науково-технічну ефективність. Низька собівартість, відкритість архітектури та можливість масштабування роблять систему актуальною як для наукових досліджень, так і для використання в навчальних закладах та малих підприємствах.

Загалом виконана робота підтвердила доцільність впровадження розробленої системи дистанційного моніторингу та керування роботизованим пристроєм. Отримані результати мають практичну значущість, високий рівень інноваційності та створюють перспективи для подальшого вдосконалення й застосування розробки в реальних умовах.

## ВИКОРИСТАНА ЛІТЕРАТУРА

1. Рóбот — Вікіпедія [Електронний ресурс]. — Режим доступу: <https://uk.wikipedia.org/wiki/Рóбот>
2. Види роботів і класифікація в робототехніці [Електронний ресурс]. — Режим доступу: <https://worldbank.org.ua/3720-vidi-robotiv-i-klasifikatsiya-v-robototekhnitsi.html>
3. 6 типів промислових роботів — Fuyumotion [Електронний ресурс]. — Режим доступу: <https://www.fuyumotion.com/uk/news/6-types-of-industrialrobots/>
4. Що таке система керування роботом? — Linear-Rotary [Електронний ресурс]. — Режим доступу: <https://ua.linear-rotary.com/info/what-is-a-robot-control-system-63756245.html>
5. Переваги та недоліки систем моніторингу та контролю за роботизованими системами [Електронний ресурс]. — Режим доступу: <https://telegra.ph/Perevagi-ta-nedol%D1%96ki-sistem-mon%D1%96toringu-takontrolyu-za-robotizovanimi-sistemami-04-23>
6. Офіційний сайт компанії IGUS [Електронний ресурс]. — Режим доступу: <https://www.igus.com/>
7. Мороз В. А. Робототехніка. Теорія та практика. — Львів: Видавництво Львівської політехніки, 2019. — 312 с.
8. Шиян Д. А. Мехатроніка та робототехнічні системи. — Київ: КНУБА, 2020. — 274 с.
9. Зозуля Ю. П. Системи автоматичного керування: навчальний посібник. — Харків: ХНУРЕ, 2021. — 208 с.
10. Коваль І. В., Ніколаєнко С. О. Аналіз сучасних засобів управління промисловими роботами // Вісник Вінницького політехнічного інституту. — 2022. — №1. — С. 37–42.

- 11.Іващенко О. В. Основи робототехніки: навчальний посібник. — Київ: Ліра-К, 2020. — 196 с.
- 12.Робототехніка: напрямки розвитку та інновації [Електронний ресурс]. — Режим доступу: <https://robotics.ua/articles/305-innovatsii-v-robototekhnitsi>
- 13.ISO 8373:2012 — Робототехнічні пристрої — Технічні терміни та визначення. — Міжнародна організація зі стандартизації (ISO).
- 14.Пономарьов С. І., Черненко В. П. Системи штучного інтелекту у керуванні роботами. — Дніпро: ДНУ, 2021. — 145 с.
- 15.Дослідження ринку промислових роботів в Україні [Електронний ресурс]. — Режим доступу: <https://techukraine.org/2023/12/01/robotics-market-ukraine>
- 16.ROS — Robot Operating System [Електронний ресурс]. — Режим доступу: <https://www.ros.org/>
- 17.Кучеренко О. М. Програмування роботизованих систем. — Київ: НАУ, 2020. — 187 с.
- 18.Robotics Business Review — Industry news and insights [Електронний ресурс]. — Режим доступу: <https://www.roboticsbusinessreview.com/>
- 19.Міжнародна федерація робототехніки (IFR) — Статистика і аналітика [Електронний ресурс]. — Режим доступу: <https://ifr.org/>
- 20.Петров І. І., Андрєєв С. Ю. Застосування сенсорів у мобільній робототехніці // Вісник НТУУ «КПІ». — 2021. — №4. — С. 50–56.
- 21.Савченко В. М., Литвиненко А. П. Інтелектуальні робототехнічні системи: навчальний посібник. — Харків: НТУ «ХПІ», 2021. — 220 с.
- 22.Базові компоненти роботів: сенсори, приводи, контролери [Електронний ресурс]. — Режим доступу: <https://robotics.org.ua/base-components-robots>
- 23.Козирєв Ю. А. Роботизовані виробничі системи. — Одеса: ОНПУ, 2018. — 198 с.\

24. Artificial Intelligence and Robotics Integration: Challenges and Trends

[Електронний ресурс]. — Режим доступу:

<https://www.sciencedirect.com/science/article/pii/S1877050921001234>

25. Бортник С. І., Мельник І. В. Робототехнічні комплекси для сільського господарства // Техніка АПК. — 2022. — №3. — С. 24–30.

## ДОДАТКИ

- 1) <https://playcode.io/2417198> - Контроллер компонент
- 2) <https://playcode.io/2417251> - Робот компонент
- 3) <https://playcode.io/2417252> - Сервер