

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»**

**ДОСЛІДЖЕННЯ АВТОМАТИЗОВАНОГО КЕРУВАННЯ
БІОМЕХАНІЧНОЇ СИСТЕМИ З ВИКОРИСТАННЯМ ARDUINO
НА БАЗІ МОДЕЛІ ПАЛЬЦЯ РУКИ РОБОТА**

**RESEARCH ON AUTOMATED CONTROL OF A
BIOMECHANICAL SYSTEM USING ARDUINO BASED ON A
ROBOT FINGER MODEL**

спеціальність 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

освітня програма «Комп'ютерна інженерія»
(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІМ-21
Мороз Богдан Сергійович

(підпис)

Керівник: к.т.н., доцент
Багнюк Наталія Володимирівна

(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

Морозу Богдану Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Дослідження автоматизованого керування біомеханічної системи з використанням Arduino на базі моделі пальця руки робота

Керівник роботи к.т.н., доцент Багнюк Наталія Володимирівна

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

Огляд досліджень та сучасних технологій моделювання систем керування роботизованими пристроями

Спеціалізовані програмні методи проектування систем керування

Розробка імітатора пальця на базі Arduino

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Огляд досліджень та сучасних технологій моделювання систем керування роботизованими пристроями</i>	<i>Багнюк Н.В., доцент</i>		
<i>Спеціалізовані програмні методи проектування систем керування</i>	<i>Багнюк Н.В., доцент</i>		
<i>Розробка імітатора пальця на базі Arduino</i>	<i>Багнюк Н.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>Огляд досліджень та сучасних технологій моделювання систем керування роботизованими пристроями</i>	До 20.08.2025 р.	
3.	<i>Спеціалізовані програмні методи проектування систем керування</i>	До 25.09.2025 р.	
4.	<i>Розробка імітатора пальця на базі Arduino</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)

Мороз Б.С.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Багнюк Н.В.

(прізвище, ініціали)

АНОТАЦІЯ

Мороз Б. С. Дослідження автоматизованого керування біомеханічної системи з використанням Arduino на базі моделі пальця руки робота. 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 огляду предметної області, тут розглядаються основні поняття про системи керування роботизованими пристроями, їх види та сфери їх використання, наведено багато практичних прикладів.

В другому розділі здійснено вибір та обґрунтування засобів розробки. Обрано засоби: Arduino та Matlab. Описано інтеграцію засобів розробки.

Третій розділ присвячено опису розробленого імітатора пальця людини на базі плати Arduino, установки та функціональні схеми, розробки практичного застосування, опис принципу роботи пристрою.

Ключові слова: протез, палець, Arduino, Matlab, система керування, смарт-протезування.

ANNOTATION

Moroz B. Research of automated control of a biomechanical system using Arduino based on a robot finger model. Manuscript.

Qualifying work of a Master's of EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

Qualification work consists of an introduction, three sections, conclusions, a references, three appendices.

The first section is devoted to an overview of the subject area, here the basic concepts of control systems for robotic devices, their types and areas of their use are considered, many practical examples are given.

In the second section, the selection and justification of development tools are made. The tools are selected: Arduino and Matlab. The integration of development tools is described.

The third section is devoted to a description of the developed human finger simulator based on the Arduino board, settings and functional diagrams, development of practical applications, description of the principle of operation of the device.

Keywords: prosthesis, finger, Arduino, Matlab, control system, smart prosthetics.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 ОГЛЯД ДОСЛІДЖЕНЬ ТА СУЧАСНИХ ТЕХНОЛОГІЙ МОДЕЛЮВАННЯ СИСТЕМ КЕРУВАННЯ РОБОТИЗОВАНИМИ ПРИСТРОЯМИ.....	11
1.1 Методи реалістичної реконструкції людського тіла та смарт- протезування.....	11
1.2 Огляд методів проектування імітаторів кінцівок людини.....	17
1.3 Перспективи застосування смарт-протезування на базі динамічних моделей біомеханіки	22
1.4 Особливості моделювання динаміки пальців людини.....	27
РОЗДІЛ 2 СПЕЦІАЛІЗОВАНІ ПРОГРАМНІ МЕТОДИ ПРОЕКТУВАННЯ СИСТЕМ КЕРУВАННЯ.....	32
2.1 Сучасні технології візуального проектування на базі Matlab	32
2.2 Характеристика мови програмування Matlab та програмно-апаратної платформи	38
2.3 Основні характеристики програмної платформи Arduino	39
2.4 Архітектура та основні компоненти платформи Arduino	41
2.5 Переваги та обмеження використання Arduino для смарт-проектування роботизованих пристроїв	42
2.6 Опис мови програмування Arduino та огляд середовища розробки.....	43
2.7 Підключення та налаштування Arduino-плати	44
2.8 Особливості програмування за допомогою Arduino та Matlab	47
РОЗДІЛ 3 РОЗРОБКА ІМІТАТОРА ПАЛЬЦЯ ЛЮДИНИ НА БАЗІ ARDUINO	49
3.1 Принципи розробки установки та функціональні вимоги.....	49
3.2 Опис установки та функціональна схема автоматизації імітатора пальця на базі Arduino	50
3.3 Вибір технічних засобів та виконавчих механізмів	52
3.4 Процес проектування демонстраційної установки.....	54

3.5 Обґрунтування вибору середовища програмування Arduino та розробка програмної частини установки	62
3.6 Розробка програмної частини установки	63
3.7 Обґрунтування вибору середовища MATLAB і пакету Control System Toolbox.....	64
3.8 Управління пристроями і реалізація програмно-функціонального зв'язку у середовищі MATLAB.....	66
3.9 Програмна реалізація управління пальцем на базі Arduino через середовище MATLAB	68
3.10 Дослідження способів практичного застосування установки у реабілітації.....	75
ВИСНОВКИ.....	81
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	82
ДОДАТКИ.....	86

ВСТУП

Актуальність теми даної кваліфікаційної роботи обґрунтовується наступним. У сучасних дослідженнях у галузі біомеханіки, робототехніки та біонічних протезів дедалі більшої уваги набувають системи, які імітують рухи людських кінцівок. Зокрема, демонстраційні установки та імітатори механізованих пальців, створені за допомогою доступних платформ, таких як Arduino, слугують ефективними науково-дослідними та навчальними інструментами. У даній роботі розробляється демонстраційна установка – імітатор роботизованого пальця руки людини на базі платформи Arduino. Система побудована за принципами біоміметичного моделювання, забезпечуючи відтворення ключових характеристик кінематики й динаміки природного руху людської кінцівки.

Основні складові установки включають розробку функціональної схеми автоматизації, що описує потоки сигналів і етапи керування, інтеграцію сенсорних і виконавчих компонентів, включно з датчиками сили та гнучкості, електричну схему під'єднання сенсорів і приводів до контролера, розробку програмного забезпечення для інтелектуального модуля керування, що реалізує алгоритми реакції на сенсорні дані та техніко-економічне обґрунтування, що підтверджує ефективність і доцільність використання системи в навчально-практичних цілях.

Таким чином, у даній кваліфікаційній роботі розроблено і досліджено демонстраційну установку – імітатор механізованого пальця робота-гуманоїда на базі платформи Arduino. Система побудована за принципами біоміметичного моделювання, забезпечуючи відтворення ключових характеристик кінематики й динаміки природного руху людської кінцівки, яка може мати застосування у відновленні дрібної моторики для людей після інсульту чи травм, яким особливо важливо тренувати згинання пальців і хватку. Імітатор роботизованого людського пальця на базі Arduino має потенційне використання як прототип тренажера хватальних рухів, у якій датчики допомагають контролювати силу

стиснення, що знижує ризик травм і поступово збільшує навантаження. Це є модельний прототип штучної кінцівки із функціональним смарт-керуванням, який поєднує сенсори зворотного зв'язку, мікроконтролер на базі Arduino та інтелектуальні алгоритми керування, що дозволить вирішити значну кількість проблем у сфері медичної реабілітації. Такі системи можуть адаптуватися до індивідуальних особливостей пацієнта, забезпечувати навчання рухів і навіть відновлювати втрачені функції за допомогою інтерактивних протоколів реабілітації.

Метою роботи є розробити та впровадити демонстраційну установку – імітатор механізованого пальця робота-гуманоїда на базі платформи Arduino. Основною метою дослідження є принципи розробки та інженерія демонстраційної установки для імітації рухів людського «смарт-пальця» на платформі Arduino, яка відтворює природні динамічні властивості біологічної руки, що є перспективним напрямом як для створення реабілітаційних технологій для осіб з обмеженими можливостями, так і для фундаментальних досліджень у сфері біомеханіки.

В контексті заявленої мети можна виділити наступні завдання проєкту:

- розробити структуру апаратної частини системи, функціональну схему автоматизації, що описує потоки сигналів і етапи керування, інтеграцію сенсорних і виконавчих компонентів, включно з датчиками сили чи гнучкості, електричну схему під'єднання сенсорів і приводів до контролера;

- розробити також програмне забезпечення для інтелектуального модуля керування, яке реалізує алгоритми реакції на сенсорні дані та показано приклад, що підтверджує ефективність і доцільність використання даної розробки для цілей медичної реабілітації;

- інтегрувати програмну логіку керування в установку;

- виконати тестування системи в реальних умовах, оцінити стабільність роботи, енергоефективність та адаптивність до змін зовнішніх умов, підготувати технічну документацію та методичні рекомендації щодо впровадження подібних систем та зробити висновки.

Завданням є створення установки імітатора «смарт-пальця» з керуванням на базі Arduino, що відтворює фізіологічну динаміку рухів людської кисті, відкриває нові можливості у сфері медичної реабілітації людей з порушеннями функціонування кінцівок та у дослідженні біомеханічних закономірностей роботи руки. Розробка демонстраційної установки імітатора біомеханічного «смарт-пальця» з керуванням на базі платформи Arduino здатна відтворювати природну динаміку рухів людської руки, що має важливе значення для вирішення завдань медичної реабілітації осіб з інвалідністю, а також для дослідження біологічних механізмів моторики.

Об'єктом дослідження є програмні засоби проектування автоматичного керування, розробка інтелектуального модуля керування, що реалізує алгоритми реакції на сенсорні дані.

Предметом дослідження є інтелектуальна система автоматичного керування.

Апробація результатів. Результати роботи представлені на Міжнародній науково-практичній конференції «Наука, освіта і суспільство в умовах змін: виклики та інновації», яка проходила 10 жовтня 2025 р., м. Кременчук [1].

РОЗДІЛ 1

ОГЛЯД ДОСЛІДЖЕНЬ ТА СУЧАСНИХ ТЕХНОЛОГІЙ МОДЕЛЮВАННЯ СИСТЕМ КЕРУВАННЯ РОБОТИЗОВАНИМИ ПРИСТРОЯМИ

1.1 Методи реалістичної реконструкції людського тіла та смарт-протезування

У сучасному науково-технічному середовищі зростає інтерес до розробки високоточної реконструкції людського тіла та впровадження інтелектуальних протезних систем, оскільки ці напрями відкривають нові горизонти в медицині, реабілітації, біоінженерії та індустрії віртуальної й доповненої реальності. Реалістичне відтворення фігури людини передбачає створення детальної тривимірної моделі на основі візуальних і сенсорних даних. Найпоширеніші підходи базуються на фотограмметрії з мультивидовою реконструкцією, де збір зображень з кількох ракурсів і порівняння пікселів дозволяє отримати анатомічно точні форми. Сучасні дослідження поєднують voxel-суперрезолюцію з глибинними неймережами, що значно підвищує деталізацію моделі. Параметричні системи на кшталт SMPL доповнюються неявними методами, які кодують геометрію в латентному просторі, завдяки чому моделі стають більш адаптивними до змінних анатомічних характеристик. Крім того, використання RGB-D сенсорів, таких як камери Microsoft Kinect або Intel RealSense, дає можливість реконструювати форму тіла в реальному часі, що є критично важливим для таких застосувань, як контроль пацієнта під час руху або інтерактивна віртуальна примірка [2].

Смарт-протезування перетворило традиційні механічні системи на високотехнологічні пристрої, що динамічно адаптуються до користувача. Керування протезами здійснюється за допомогою біосигналів: електроміографічних, електроенцефалографічних і навіть оптичних методик, як у випадку fNIRS, які дають змогу точніше інтерпретувати інтенції руху. Завдяки алгоритмам машинного навчання сучасні системи передбачають наступні рухи

кінцівки та коригують силу й напрямок протеза в реальному часі, що значно покращує ергономіку й зменшує відчуття «чужого тіла». Окремо варто зазначити зусилля в області тактильного зворотного зв'язку, коли через тактильні сенсори чи мікростимуляцію нервових закінчень користувачі можуть відчувати тиск і дотик ніби власної шкіри, що відкриває новий рівень взаємодії зі середовищем.

Практичне застосування цих технологій вже набуває масових рис: віртуальна підгонка протезів на основі 3D-сканування тіла значно скорочує час і витрати на виготовлення; екзоскелети (рис. 1.1), адаптовані до індивідуальних особливостей пацієнта, допомагають у реабілітації після інсультів чи травм; а автоматизоване створення анатомічно точних моделей служить надійною платформою для передопераційного планування у складних ортопедичних чи нейрохірургічних втручаннях [3].



Рисунок 1.1 – Екзоскелет «EksoNR» [4]

Інтеграція 3D-реконструкції з біомедичними системами відкриває шлях до справжньої персоналізації медицини (рис. 1.2). Тривимірні моделі тіла використовуються для проєктування індивідуалізованих імплантів, моделювання хірургічних втручань, моніторингу процесу реабілітації та створення цифрових аватарів пацієнтів для комплексного планування лікування.

Водночас поєднання параметричних моделей для кістково-суглобової структури з неявними представленнями м'яких тканин дозволяє досягати надзвичайно високої анатомічної достовірності та відкриває можливості для персоналізованих імплантів і протезів [5].



Рисунок 1.2 – 3D сканер тіла X-ONE SE [6]

Сенсорна інфраструктура інтелектуальних протезів включає широкий спектр датчиків тиску, температури та орієнтації (IMU), а також поверхневу електроміографію, яка вловлює навіть мінімальні м'язові скорочення. Центральне місце в сучасному пристрої посідає мікроконтролер із адаптивними алгоритмами: він обробляє вхідні сигнали, аналізує стиль руху та з часом

навчається розпізнавати індивідуальні особливості ходьби й навіть емоційні стани через варіації електроактивності. Зворотний тактильний зв'язок завдяки мікростимуляції нервових закінчень забезпечує глибше занурення у взаємодію зі світом.

Попри значні досягнення, низка викликів і надалі грає ключову роль у впровадженні технологій у повсякденну практику. Реконструкція при частковому закритті тіла одягом, змінному освітленні чи під час руху пацієнта може давати похибки, особливо якщо анатомія ушкоджена або зменшена через ампутацію. Значні індивідуальні відмінності за формою, расовими та віковими ознаками вимагають більших вибірок для навчання моделей. Крім того, висока вартість просунутих протезів з активними приводами, нейроінтерфейсами та точними сенсорами обмежує їхню доступність, а складні етичні питання стосуються безпеки даних нейроінтерфейсів і питання ідентичності людини при значній заміні частин тіла.

Перспективи розвитку обіцяють ще цікавіші зміни. Інтеграція з AR/VR дозволить створювати реалістичні аватари для телемедицини, реабілітації та психотерапії, що особливо важливо при лікуванні фантомних болів. Регенеративне протезування поєднає смарт-системи з біоінженерією тканин, а нейрокомунікація безпосередньо з корою головного мозку дасть змогу отримувати наміри руху без потреби у м'язових сигналів. Таким чином, розвиток цих напрямів наближає нас до майбутнього, де персоналізована система відновлення та розширення можливостей людського тіла стане буденністю.

Крім згаданих підходів, фахівці все активніше інтегрують мультиспектральні сенсори та ультразвукові системи в процес 3D-сканування, що дозволяє не лише відтворювати контури тіла, а й одержувати інформацію про щільність і склад тканин. Поєднання фотограмметрії з томографічними знімками (КТ, МРТ) відкриває можливість створення віртуальних моделей з деталізацією кісткової та м'язової структури, що критично для передопераційного планування складних втручань (рис. 1.3). Завдяки цьому хірурги можуть «приміряти»

імпланти всередині цифрового пацієнта, імітуючи різні сценарії фіксації й оцінюючи навантаження на суглоби ще до першого розрізу скальпеля.



Рисунок 1.3 – Кабінет магнітно-резонансної томографії (МРТ) [7]

У галузі комп'ютерного зору активно експериментують з генеративними неймережами, зокрема з варіаційними автоенкодерами та GAN-моделями, що дозволяють відтворити візуальні деталі шкіри, зморшок та відбитків вен на долонях. Такі системи навчаються на величезних масивах даних, здобутих із відкритих медичних архівів та дослідницьких баз, і можуть генерувати текстури з реалістичною градацією кольору та глянцю. При цьому латентний простір моделі дозволяє коригувати вік, тон шкіри чи навіть змінювати статеві ознаки за запитом користувача.

Ідея digital twin – цифрового двійника пацієнта – прагне зробити моделі динамічними: аватар оновлюється під час кожного сеансу реабілітації чи після

курсу фізіотерапії, фіксуючи позитивні або небажані зміни в об'ємі м'язів, поставі й амплітуді рухів. Така інтерактивна модель здатна прогнозувати ризики повторної травми або появу ускладнень, підказуючи лікареві оптимальний протокол вправ і відпочинку.

У смарт-протезуванні останнім часом зростає інтерес до м'яких роботизованих елементів із пневматичними або гідравлічними приводами, що наближаються за своїми властивостями до натуральних м'язів. Вони легші й безпечніші, менше травмують залишкові тканини та дозволяють протезу плавніше реагувати на зміни навантаження.

Додатково розробляються системи енергетичної самодостатності, які використовують дуже цікаві рішення: наприклад, різоелементи, вмонтовані в підошву штучної стопи, перетворюють напругу під час ходьби на струм для підзарядки вбудованих акумуляторів.

Паралельно з розвитком «заліза» не відстають програмні платформи, відкриті фреймворки дозволяють швидко тестувати нові алгоритми машинного навчання, обмінюватися навчальними даними й одночасно підтримувати безпеку пацієнтських записів. Хмарні сервіси об'єднують інженерів, лікарів і пацієнтів у єдиному середовищі, де кожен крок прототипу можна верифікувати за участі кінцевого користувача.

Також важливо не забувати про соціальні й етичні аспекти: з одного боку, нейроінтерфейси та біометричні дані відкривають нові можливості для відстеження намірів людини, але з іншого – вимагають жорсткого контролю за приватністю та запобіганням несанкціонованого доступу. Глобальні ініціативи з розробки стандартизованих протоколів обміну даними й міжнародних норм безпеки вже сьогодні готують ґрунт для того, щоб майбутнє персоналізованої медицини стало не лише високотехнологічним, а й безпечним, етичним та доступним якомога більшій кількості людей.

1.2 Огляд методів проєктування імітаторів кінцівок людини

Імітатори кінцівок людини посідають важливе місце у сучасних дослідженнях біомеханіки та робототехніки. Вони використовуються не лише для компенсації функцій, втрачених унаслідок ампутацій, але й як науково-дослідні інструменти, що дозволяють моделювати рухи, перевіряти нові підходи до керування та тестувати конструктивні рішення у сфері протезування. На відміну від класичних протезів, такі системи здатні з більшою точністю відтворювати рухові функції й нерідко стають основою для створення високотехнологічних біонічних протезів.

Варто зазначити, що імітатори класифікують за різними критеріями. Функціональні (роботизовані) моделі здатні активно виконувати рухи, керовані користувачем або програмними алгоритмами. Пасивні імітатори не мають приводу та здебільшого застосовуються у візуалізаційних цілях або для реабілітаційних вправ. Окрему групу становлять механічні тренажери, призначені для фізіотерапії та дослідження моторики. Наприклад, у роботі *A Survey on Design and Control of Lower Extremity Exoskeletons* [8] наведено аналіз механізмів, що імітують людську ходу, де автори порівняли різні стратегії керування (PID, адаптивні, нейромережеві) та джерела енергії (електричні мотори, пневматика, гідравліка).

Традиційне ручне проєктування протезів передбачає зняття відбитків із культеподібного залишку за допомогою гіпсових або силіконових сумішей, створення моделі куксоприймача, багаторазові примірки та остаточне шліфування. Цей багатоступеневий процес потребує значних часових ресурсів і досвіду протезиста. Однак навіть при високій майстерності фахівця складно досягти ідеального прилягання, що призводить до натирання, незручностей під час ходи й порушення мікроциркуляції тканин під протезом. У зв'язку з цим з початку 2000-х активно впроваджуються цифрові технології, які кардинально змінюють підхід до створення імітаторів кінцівок.

Важливим етапом створення імітатора є біомеханічне моделювання. На цьому рівні визначається кількість ступенів свободи, типи суглобів, межі рухливості та необхідні моменти сили для адекватної імітації людських рухів. У дослідженні Zhang та ін. [9] представлено методику побудови кінематичних моделей на основі медичного сканування, що дозволяє формувати персоналізовану анатомічну геометрію для кожного користувача. Це має особливе значення у протезуванні, де точність анатомічної відповідності безпосередньо впливає на комфорт та ефективність використання пристрою.

Ключовим кроком стало використання 3D-сканерів для безконтактного моделювання поверхні культі. Високоточні лазерні або оптичні системи дозволяють з точністю до десятків мікрон зафіксувати форму залишку кінцівки без болісного знімання відбитків. Отриману тривимірну модель імпортують у CAD-програми (наприклад, Geomagic Freeform або Autodesk Fusion 360), де проводять віртуальне вирівнювання товщини стінок гільзи, аналіз точок тиску та елементів кріплення (рис. 1.4). У перспективі штучний інтелект здатен автоматизувати цей етап, пропонуючи оптимальні модифікації на основі баз даних реальних випадків і біомеханічних симуляцій [10].

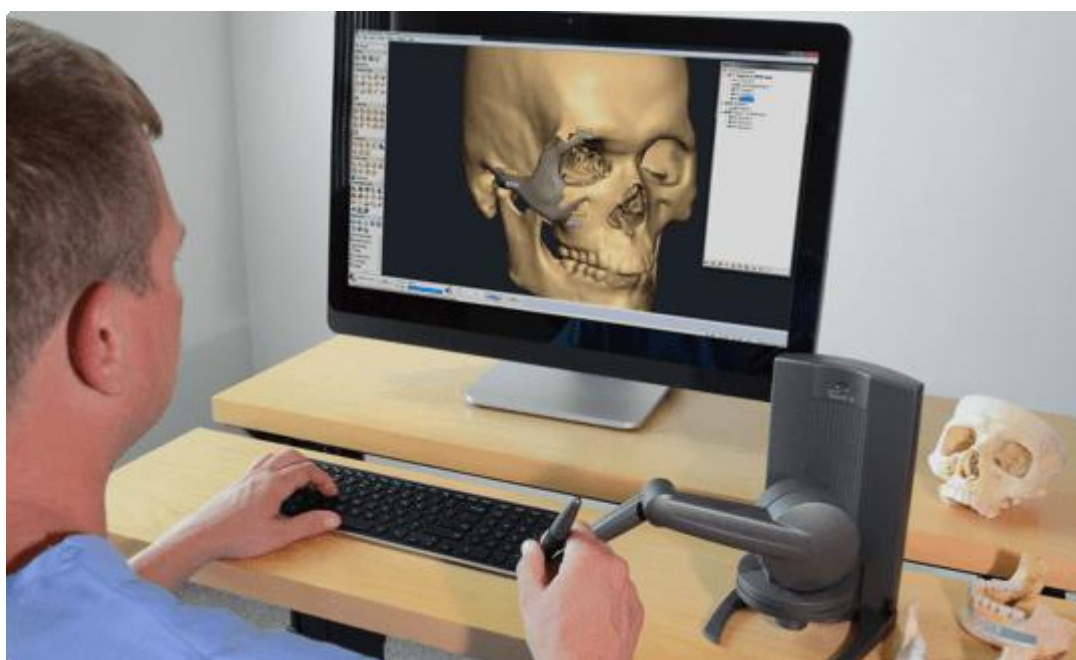


Рисунок 1.4 – Програма Geomagic Freeform [11]

Наступним етапом проєктування виступає використання систем CAD/CAE. За допомогою CAD-середовищ формується точна геометрична модель імітатора, яка враховує матеріали, розміри та конструктивні особливості. Подальший аналіз у CAE-системах, зокрема за допомогою методу скінченних елементів (FEA), дає змогу визначити міцність конструкції, перевірити її поведінку під дією навантажень та спрогнозувати зони можливих деформацій.

CAD/CAM-виробничий цикл, коли обробку готової моделі здійснюють на фрезерних верстатах із ЧПУ. Це значно скоротило час виготовлення остаточного протеза: від декількох тижнів до кількох днів. Автоматизація дозволяє підтримувати високу повторюваність розмірів і геометрії, що особливо важливо для серійного виготовлення лекальних форм для пацієнтів із подібними анатомічними характеристиками. Наприклад, дослідники з Лундського університету показали, що інтеграція CAM-технологій із клінічними даними зменшує кількість модифікацій після першої примірки майже на 40 %.

Окрему групу методів становлять біонічні алгоритми, що ґрунтуються на принципах імітації природної поведінки людських м'язів та сухожилів. Для створення більш реалістичних імітаторів застосовуються штучні м'язи з матеріалів із пам'яттю форми (SMA), пневматичні та гідравлічні актюатори, які реагують на електричні або теплові стимули, імітуючи скорочення м'язів. Такі підходи наближають механічну систему до природної біологічної.

Високофункціональні біонічні протези інтегрують в себе сенсори, мікроконтролери й приводи. Міоелектричне керування – найпоширеніший спосіб, коли сигнали з м'язів культі реєструються поверхневими електродами, фільтруються та перетворюються на команди для сервомоторів. Такі системи дозволяють відтворювати згинання, розгинання й навіть обертання кисті зі швидкістю, близькою до природної. Аналітичні огляди компанії Ottobock свідчать, що пацієнти із сучасними біонічними протезами завдяки інтуїтивному управлінню відновлюють до 80 % рівня діяльності здорової руки або ноги.

Ключова проблема біонічних систем – енергозабезпечення та маса пристрою. Сучасні літій-іонні батареї вбудовують у корпус протеза, однак їх

емності вистачає на 8-12 годин активного використання. Дослідники Технічного університету Мюнхена пропонують використовувати гібридні енергосистеми: міні-турбогенератори на основі паливних елементів із відновлюваних джерел водню, які заряджають акумулятор у процесі ходьби, а також енергію з реуправління серводвигунів під час гальмування кінцівки прикладами до землі.

Біоінспіровані підходи все частіше застосовують еволюційні алгоритми та цифрове скульптування за зразком природних структур. Імітація колагеново-еластинових волокон у м'яких частинах гільзи, «штучні м'язи» з полімерних волокон, здатних скорочуватися при нагріванні, та адаптивні матеріали з пам'яттю форми дозволяють досягати рівня комфорту, близького до натурального. Стаття групи в Університеті Каліфорнії в Берклі описує конструкцію динамічного каркаса, який підлаштовується під навантаження під час ходьби, забезпечуючи активну підтримку рухів і знижуючи вібрації до 30 %.

Матеріалознавчі інновації відіграють ключову роль: використання карбонових композитів, титанових сплавів із високою корозійною стійкістю, біосумісних полімерів РЕЕК і РЕКК, а також біорозкладних матеріалів для тимчасових протезів у дітей. Нині активно вивчають можливість імплантації 3D-друкованих гідрогелевих вставок із клітинними культурами, що сприяють регенерації м'яких тканин навколо культі та покращують інтеграцію з протезом.

Соціальні та етичні аспекти проектування імітаторів кінцівок не менш важливі за технічні. Доступність передових технологій у країнах із обмеженим бюджетом, підготовка кваліфікованих кадрів, підвищення обізнаності пацієнтів і психологічна підтримка мають стати складовими національних програм реабілітації. Показовим є досвід Польщі, де державне страхування покриває як стандартні, так і полегшені біонічні моделі, що дозволило скоротити час очікування протеза до одного місяця та збільшити охоплення потребуючих на 45 %.

Перспективні напрямки включають інтеграцію нейроінтерфейсів для зворотного зв'язку: чутливі датчики тиску й температури передаватимуть

сигнали в кору головного мозку через імплантовані електроди, відтворюючи відчуття дотику. Проєкт Neuralink демонструє, що штучні сенсорні шляхи можна поєднати з екзоскелетними системами, створюючи гібридні структури для часткової або повної заміни кінцівок. Крім того, розвиток штучного інтелекту дозволить «навчати» протез під індивідуальну кінематику кожного пацієнта, постійно коригуючи алгоритми керування в процесі реального використання.

Не менш важливим аспектом є сенсорна інтеграція. Сучасні імітатори оснащуються датчиками тиску, сили, деформації, інерційними вимірювальними модулями (IMU), а також електроміографічними сенсорами (EMG), що зчитують активність м'язів користувача. Це дозволяє реалізувати зворотний зв'язок і підвищити точність відтворення рухів. У роботі Jafarzadeh та ін. (2023) описано використання нейромереж для обробки EMG-сигналів, що забезпечує адаптивне керування протезом у реальному часі [12].

Розвиток імітаторів кінцівок демонструють сучасні проєкти, такі як Shadow Hand (рис. 1.5), яка має 20 ступенів свободи і використовується для досліджень у галузі захоплення предметів, або CyberLegs – європейський проєкт активного протезування нижньої кінцівки з інтеграцією інерційних сенсорів і приводів у колінному та тазостегновому суглобах.



Рисунок 1.5 – Приклад проєкту Shadow Hand [13]

Прикладом доступного рішення є Hero Arm від Open Bionics, який поєднує 3D-друковані компоненти та EMG-керування, роблячи біонічні технології більш масовими.

Перспективи розвитку галузі полягають у впровадженні інтелектуальних алгоритмів, які дозволять імітаторам навчатися від рухів користувача, застосуванні біосумісних матеріалів і технологій 4D-друку, що забезпечать зміну форми конструкції під дією зовнішніх стимулів, а також у розвитку нейроінтерфейсів, здатних забезпечити прямий зв'язок мозку з пристроєм.

Таким чином, методи проектування імітаторів кінцівок людини формуються на стику інженерії, біомеханіки, електроніки та штучного інтелекту. Їхній розвиток забезпечує не лише підвищення функціональності протезів, а й створює нові можливості для реабілітації, біонічної інтеграції (рис. 1.6) та дослідження складних процесів рухової діяльності.



Рисунок 1.6 – Біонічний протез руки [14]

1.3 Перспективи застосування смарт-протезування на базі динамічних моделей біомеханіки

Останні десятиліття розвиток протезування демонструє чіткий перехід від простих механічних конструкцій до інтелектуальних систем, що базуються на знаннях із біомеханіки, сенсорних технологіях та адаптивних методах керування. Смарт-протези, які проектуються на основі динамічних моделей біомеханіки, відкривають нові можливості у сфері реабілітації та відновлення рухових функцій. Вони дозволяють досягати більшої природності рухів, забезпечують

енергоефективність, адаптивність до змінних умов середовища та високий рівень персоналізації під конкретного користувача.

Ключову роль відіграє багатосенсорна архітектура: поверхневі міоелектричні датчики реєструють активність м'язів культі, інерційні вимірювальні модулі фіксують прискорення й кути нахилу сегментів тіла, тензорезистивні платформи в гільзі оцінюють розподіл тиску. Усі ці сигнали слугують вхідними даними для динамічної моделі, яка в режимі реального часу обчислює оптимальні команди для приводів та адаптивних елементів конструкції.

Одним із напрямів розвитку є використання електроміографічних (ЕМГ) сигналів для керування рухами протезів. Наприклад, протези компаній Open Bionics, Ottobock та Coapt використовують ЕМГ-сенсори, які зчитують активність м'язів на залишковій кінцівці, перетворюючи її в сигнали керування електродвигунами. Такі рішення забезпечують користувачу можливість виконувати базові маніпуляції – стискання, розтискання, обертання кисті – без затримки й з високою точністю. Однак недоліком цих систем є складність налаштування, необхідність адаптації користувача та обмежена кількість ступенів свободи у порівнянні з природною рукою [15].

Інший підхід полягає у використанні нейрокерованих протезів, які отримують сигнали безпосередньо з нервової системи через імплантовані електроди (наприклад, проекти DARPA Revolutionizing Prosthetics та Modular Prosthetic Limb (MPL) від Johns Hopkins University). Такі протези забезпечують високий рівень точності рухів та навіть тактильний зворотний зв'язок, що дозволяє користувачу відчувати силу стиснення об'єкта. Основним викликом залишається інвазивність імплантації, висока вартість і потреба у тривалому навчанні користувача.

Серед комерційних і доступних рішень варто відзначити DEKA Arm System (відомий як «Luke Arm»), який дозволяє виконувати понад 10 різних рухів завдяки системі багатоканального керування та гіроскопічним сенсорам. Інший приклад – BrainCo Dexus, який застосовує алгоритми машинного

навчання для розпізнавання шаблонів м'язової активності, адаптуючись під конкретного користувача. Такі рішення демонструють, що інтеграція штучного інтелекту стає ключовим фактором у розвитку протезування – системи вчаться самостійно коригувати рухи, підлаштовуючись під фізіологічні особливості людини.

На концептуальному рівні всі смарт-протези базуються на трьох технологічних складових: сенсорній системі (ЕМГ, тиск, прискорення, температура), системі керування (мікроконтролери, нейронні мережі, нечітка логіка) та актуаторній частині (серводвигуни, лінійні приводи). Оптимальне поєднання цих компонентів дозволяє створювати біомеханічно адекватні рішення, здатні до адаптації у реальному часі.

Порівнюючи різні типи смарт-протезів, можна зазначити, що біоелектричні протези є більш доступними та зручними для користувача, однак мають обмежену точність та швидкість реакції. Нейрокеровані протези демонструють найвищий рівень функціональності, проте їхня складність та інвазивність обмежують масове впровадження. Гібридні моделі, які поєднують ЕМГ-аналіз із машинним навчанням і сенсорними системами, вважаються перспективним компромісом між функціональністю, комфортом і ціною.

Використання динамічних моделей у проектуванні протезів має ключове значення, адже саме вони дають змогу описати рухи кінцівки з урахуванням сил, моментів інерції, демпфування, еластичності матеріалів і взаємодії з тілом людини. Такі моделі дозволяють прогнозувати та контролювати поведінку протеза у різних фазах руху, інтегрувати систему зворотного зв'язку від сенсорів і реалізовувати адаптивне керування змінною жорсткістю залежно від умов навантаження або швидкості руху. У дослідженні Rogers-Bradley та співавт [16] показано, що мікропроцесорно-контрольований протез із можливістю зміни жорсткості підвищує ефективність відштовхування під час ходьби та знижує навантаження на протилежну кінцівку, у порівнянні зі стандартними пасивними системами. Подібні результати підтверджують і інші дослідження, зокрема робота Varattini та колег [17], де динамічне моделювання бігового «лезового»

протеза допомогло оптимізувати його конструкцію та зменшити масу без втрати стабільності. Driessen та співавтори [18] запропонували гібридну динамічну модель «людина-протез», яка спрощує налаштування систем управління та придатна для онлайн-контролю.

Ключовим напрямом розвитку смарт-протезів є інтеграція сенсорних систем. Використання датчиків сили, тиску, акселерометрів і гіроскопів дозволяє оцінювати взаємодію протеза з довкіллям і тілом користувача. Це дає змогу автоматично коригувати режими роботи пристрою, визначати зміщення чи некоректну посадку та миттєво компенсувати відхилення. Застосування сенсорів також сприяє розвитку адаптивних алгоритмів, які навчаються на основі даних користувача. У роботі Manz та співавтори [19] продемонстровано, що вбудовані сенсори дозволяють у реальному часі оцінювати кінематичні та кінетичні змінні під час ходьби, що забезпечує точне коригування рухів.

Інтеграція штучного інтелекту та алгоритмів машинного навчання у протезування відкриває ще ширші перспективи. Алгоритми можуть прогнозувати рухи користувача та коригувати роботу протеза у реальному часі. У своїй оглядовій статті Jyothish і Mishra (2024) систематизували сучасні підходи до побудови протезів, зокрема використання м'яких приводів (SMA, EAP) та новітніх стратегій керування [20].

Особлива увага у сучасних дослідженнях приділяється персоналізації. Смарт-протези можуть навчатися на даних користувача, автоматично оптимізуючи параметри роботи – від регулювання жорсткості до корекції реакцій на зовнішні подразники. Завдяки розвитку IoT-технологій і хмарних платформ можливе дистанційне оновлення програмного забезпечення та налаштування пристроїв. Перспективним напрямом є створення протезів із функцією самонавчання, які поступово накопичують досвід взаємодії з користувачем, що дозволяє підвищити комфорт та ефективність використання.

Разом із тим, існують і певні виклики. Використання складних динамічних моделей вимагає значних обчислювальних ресурсів, що є обмеженням для вбудованих систем із невеликою енергетичною автономністю. Додатково,

сенсорні дані часто містять шум і потребують фільтрації, що може створювати затримки. Також важливими є питання біоадаптації, адже з часом взаємодія протеза з тканинами змінюється, що впливає на комфорт користувача. Надійність і безпека системи мають залишатися ключовими критеріями, щоб запобігати небезпечним ситуаціям у критичних режимах роботи.

Серед реальних прикладів можна виділити Michelangelo Hand – електронний протез кисті з кількома хватовими режимами, який керується електроміографічними сигналами.

Практична реалізація таких рішень вимагає високопродуктивної обчислювальної платформи – вбудованих мікроконтролерів із можливістю апаратного прискорення матричних операцій або гібридних систем із частковою обробкою в «хмарі». Розподілена архітектура забезпечує необхідну швидкодію, зберігаючи енергоефективність і компактність протеза.

У перспективі розвиток смарт-протезування кінцівок рук рухається в напрямі інтеграції біонічних матеріалів, використання бездротових нейроінтерфейсів, м'яких роботизованих структур і штучного інтелекту, що здатен імітувати нейромоторний контроль людини. Такі системи поступово наближають межу між біологічним та технічним, створюючи умови для появи протезів, які не лише відновлюють, а й розширюють людські можливості.

Отже, смарт-протези на базі динамічних моделей біомеханіки поєднують сучасні досягнення в галузі матеріалознавства, електроніки та штучного інтелекту. Вони мають потенціал не лише покращувати якість життя людей із втраченою кінцівкою, але й сприяти розвитку біомеханічних досліджень і створенню нових поколінь біонічних пристроїв.

Розглянемо основні переваги та недоліки описаних технологій, які представлені в таблиці 1.1.

Таблиця 1.1 – Основні переваги та недоліки існуючих рішень та технологій, на яких вони базуються

Тип технології	Принцип роботи	Приклади реалізації	Переваги	Недоліки
Електроміографічне (ЕМГ) керування	Зчитування електричних сигналів м'язів і перетворення їх у команди керування двигунами	Open Bionics Hero Arm, Ottobock MyoHand, Coapt Complete Control	Відносна доступність; безінвазивність; висока точність базових рухів	Потребує калібрування; обмежена кількість жестів; чутливість до шумів ЕМГ
Нейроінтерфейси (імплантовані електроди)	Безпосереднє зчитування сигналів з нервових закінчень або кори головного мозку	DARPA MPL, Johns Hopkins Revolutionizing Prosthetics	Висока точність; можливість відчуття дотику; природна координація рухів	Інвазивність; складна імплантація; висока вартість
Машинне навчання та ШІ-керування	Аналіз шаблонів м'язової активності й адаптація алгоритмів керування під користувача	BrainCo Dexus, Esper Bionics Arm	Самонавчання; індивідуальна адаптація; покращення плавності рухів	Вимагає тренування моделі; залежність від обчислювальних ресурсів
Гібридні системи (ЕМГ + сенсори + ШІ)	Поєднання декількох методів: ЕМГ-аналіз, сенсорика тиску, машинне навчання	Bebionic, Michelangelo Hand, DEKA Arm (Luke Arm)	Висока точність; адаптивність; багатофункціональність	Висока складність реалізації; потреба в енергоспоживанні
М'яка робототехніка (soft robotics)	Використання гнучких матеріалів і пневматичних приводів, що імітують роботу м'язів	SoftHand Pro, Pneumatic Hand Project (MIT)	Природність рухів; безпечна взаємодія з об'єктами; енергоефективність	Низька точність при складних рухах; обмежена сила захоплення

1.4 Особливості моделювання динаміки пальців людини

Моделювання рухів пальців людини є складним завданням у прикладній біомеханіці та робототехніці через багатоланкову структуру пальців, високу рухливість і складну організацію м'язово-сухожильного апарату. Створення кінематичних та динамічних моделей дає змогу аналізувати природні рухи,

розробляти роботизовані протези та вдосконалювати алгоритми керування біомеханічними системами.

Кінематична структура пальця включає три фаланги – проксимальну, середню та дистальну, які з'єднані міжфаланговими і п'ястно-фаланговими суглобами. Кожен суглоб забезпечує рух у певних площинах. У більшості моделей палець подають як триланковий механізм із трьома ступенями вільності. Проте більш складні підходи враховують біомеханічні обмеження, зокрема синхронність руху середньої та дистальної фаланги, що суттєво впливає на можливі траєкторії руху (рис. 1.7).

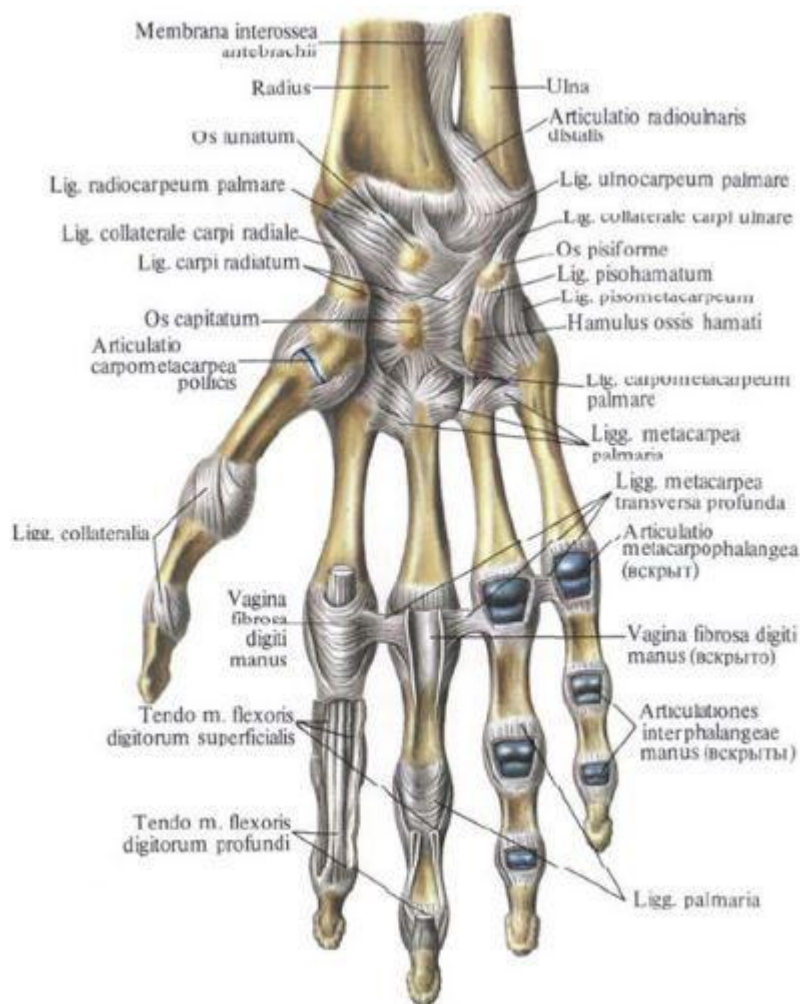


Рисунок 1.7 – Анатомічна будова кистки людини [21]

Особливе місце посідають моделі, що поєднують динамічні рівняння з експериментальними даними електроміографії (ЕМГ). Це дозволило забезпечити

природне керування роботизованою кистю, адекватно реагуючи на інтенсивність скорочення м'язів користувача (рис. 1.8). Подібний підхід реалізований у моделях, де поєднання ЕМГ-аналізу з нейронними мережами дозволило прогнозувати динамічну поведінку пальців у режимі реального часу.

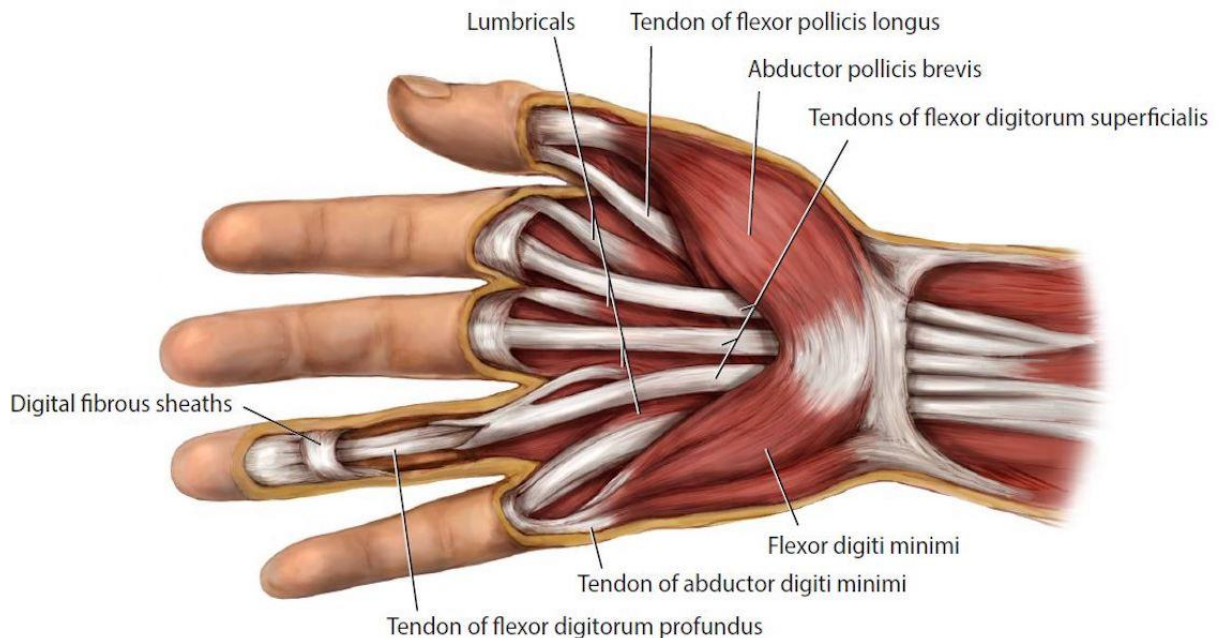


Рисунок 1.8 – Будова сухожиль руки людини [22]

Під час виконання хватальних дій необхідно враховувати контактні взаємодії з об'єктами. Тут важливу роль відіграють сили тертя, реакції у точках дотику та мікропереміщення. Для опису цих процесів використовують методи контактної механіки та скінченно-елементного аналізу, які дозволяють моделювати розподіл тиску на поверхні шкіри, що є важливим у проектуванні протезів та роботизованих маніпуляторів.

У моделюванні контактних взаємодій під час хвату ключову роль відіграють методи скінченно-елементного аналізу (Finite Element Analysis, FEA). Наприклад, Wolterink et al. [23] створили 3D-модель пальця з використанням FEA для аналізу розподілу напружень у шкірі та м'яких тканинах при контакті з предметом. Це дослідження мало практичне застосування в оптимізації форми та матеріалів протезів пальців та використали FEA для оцінки деформацій

біонічних сухожиль і визначення оптимальної геометрії приводів, що покращує відчуття сили захоплення.

Ще одним перспективним напрямом є використання методів оптимізації для керування надлишковими ступенями свободи пальців. Моделі, запропоновані Ringward та інші [24], застосовують критерій мінімізації енерговитрат, що дозволяє знаходити найраціональніші комбінації м'язових зусиль при виконанні однієї дії. У роботі реалізовано оптимізаційний підхід для налаштування параметрів м'язових моделей у реальному часі – цей метод успішно використовується у симуляторі OpenSim для дослідження рухів пальців під навантаженням.

У сучасних підходах значна увага приділяється інтеграції нейромоторного контролю. Для цього в моделі включають електроміографічні (ЕМГ) сигнали (рис. 1.9), які відображають команди від центральної нервової системи. Завдяки цьому математичні моделі наближаються до реальних умов функціонування людської руки та стають більш придатними для використання у протезуванні.

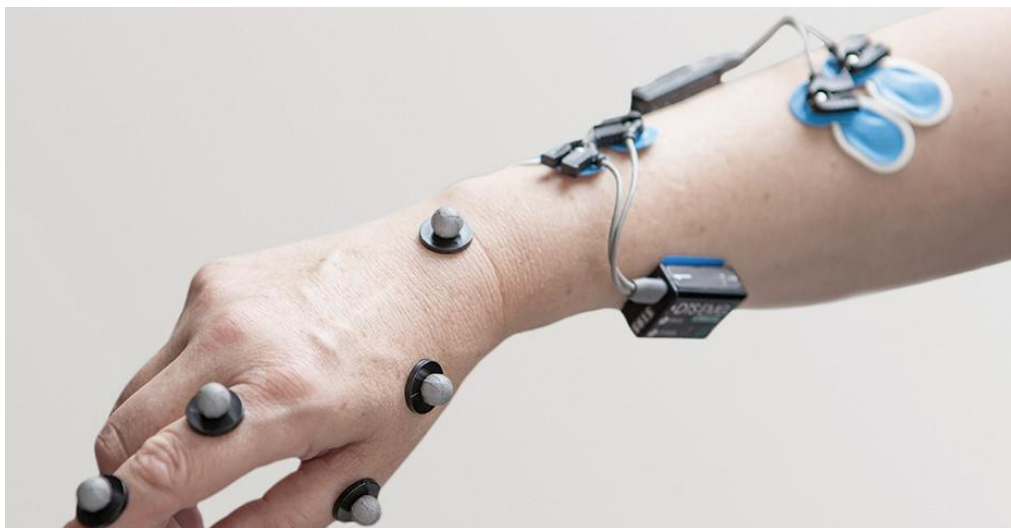


Рисунок 1.9 – Електроміографія верхніх кінцівок [25]

Основні виклики моделювання динаміки пальців пов'язані з нелінійністю системи, індивідуальними відмінностями анатомії, необхідністю роботи моделей у режимі реального часу та складністю поєднання біомеханічних і нейрофізіологічних факторів. У практичних застосуваннях часто

використовують гібридні підходи, які поєднують спрощені кінематичні схеми з алгоритмами машинного навчання, що навчаються на експериментальних даних.

Розглянемо основні переваги та недоліки методів моделювання пальців людини в таблиці 1.2.

Таблиця 1.2 – Основні методи моделювання пальців людини

Методи	Переваги	Недоліки	Типові галузі використання
Лагранжа другого роду	Точне врахування кінематичних зв'язків і потенціальних енергій	Висока обчислювальна складність	Аналіз плавних рухів, біомеханічні симуляції
Ньютона–Ейлера	Простота реалізації; придатність для реального часу	Менша точність для складних систем	Протези з адаптивним контролем
Скінченно-елементний аналіз (FEA)	Висока точність у дослідженні напружень і деформацій	Вимогливість до ресурсів; складність моделювання м'яких тканин	Оптимізація матеріалів і форм протезів
ЕМГ-моделі з нейромережами	Реалістичне керування; адаптація до користувача	Необхідність попереднього навчання	Біонічні кисті, нейроінтерфейси
Оптимізаційні моделі (енергетичні, стабілізаційні)	Енергоефективність; природність рухів	Складність визначення критеріїв оптимізації	Робототехніка, реабілітаційні системи

Сучасні тенденції свідчать про перехід до гібридних моделей, що поєднують динамічні рівняння, скінченно-елементні підходи та методи машинного навчання. Такі системи дозволяють створювати віртуальні «цифрові двійники» людських пальців, що адаптуються до індивідуальної анатомії користувача. У майбутньому ці підходи сприятимуть розробці біонічних протезів із природним керуванням і відчуттям дотику, максимально наближених до функціональних можливостей людської руки.

РОЗДІЛ 2

СПЕЦІАЛІЗОВАНІ ПРОГРАМНІ МЕТОДИ ПРОЕКТУВАННЯ СИСТЕМ КЕРУВАННЯ

Розробка сучасних систем керування вимагає використання спеціальних програмних інструментів. Такі інструменти дозволяють не тільки автоматизувати рутинні розрахунки, а й виконувати симуляції, аналіз стабільності, оптимізування параметрів та тестування алгоритмів у різних режимах. Використання програмних методів значно скорочує час розробки, підвищує точність результатів і дає змогу створювати складні системи з урахуванням реальних фізичних обмежень.

2.1 Сучасні технології візуального проектування на базі Matlab

MATLAB є однією з найпоширеніших платформ для моделювання, аналізу й проектування складних технічних систем. Однією з його ключових переваг є розвинуті інструменти для візуального проектування, які забезпечують можливість працювати з моделями на інтуїтивному графічному рівні, зменшуючи потребу у програмуванні низького рівня. Це робить MATLAB ефективним середовищем для інженерів, дослідників та студентів, дозволяючи швидко створювати й перевіряти складні схеми, моделі та алгоритми [26].

Інтегрований App Designer відкриває шлях до створення складних графічних інтерфейсів без потреби у знанні зовнішніх фреймворків: компоненти UI розміщуються перетягуванням, а генерований код MATLAB управляє подіями користувацької взаємодії. Це дозволяє розробляти повноцінні десктоп-додатки з табличними переглядачами, повзунками, кнопками й інтегрованими полотнами графіків, які можна з легкістю розповсюджувати серед колег чи клієнтів. Паралельно MATLAB Web App Server дає можливість публікувати такі програми у вигляді веб-інтерфейсів, доступних прямо з браузера, що особливо

корисно для віддаленої інженерної співпраці й декількох користувачів, які працюють із однаковим набором інструментів [27].

Основою візуального проєктування у MATLAB є середовище Simulink, яке забезпечує побудову моделей у вигляді блок-схем. Кожен блок представляє певний елемент системи – математичну операцію, фізичний процес, сенсор чи контролер. Використовуючи бібліотеки готових компонентів, користувач може з'єднати їх у єдину систему, створюючи симуляційні моделі, що відображають як прості динамічні об'єкти, так і багатокomпонентні кіберфізичні системи.

Сучасний розвиток MATLAB дав змогу інтегрувати Simulink із розширеними інструментами: Simscape для моделювання фізичних процесів (електричних, механічних, гідравлічних), Stateflow для розробки систем із дискретною логікою та подієвою поведінкою, а також спеціалізованими модулями для обробки сигналів, управління, біомеханіки та машинного навчання. Це дозволяє створювати комплексні моделі, які враховують не лише математичні, але й реальні фізичні властивості системи.

Особливе значення має можливість автоматичної генерації коду на основі побудованих моделей. MATLAB/Simulink дозволяє отримати код мовою C або HDL (для апаратних реалізацій), що значно спрощує процес переходу від прототипування до практичної реалізації. Такий підхід активно використовується у галузях робототехніки, управління складними динамічними системами, а також у сфері проєктування біонічних пристроїв.

Візуальне моделювання у MATLAB підтримує інтерактивну перевірку алгоритмів та можливість тестування в режимі Hardware-in-the-Loop (HIL). Це означає, що розробник може випробовувати модель на реальному обладнанні, використовуючи її віртуальний аналог, що дозволяє виявляти помилки на ранніх етапах і підвищує надійність проєктованих систем.

Інтеграція MATLAB із Arduino може здійснюватися у двох основних режимах. Перший – це пряме з'єднання, коли MATLAB надсилає команди на плату Arduino через USB або Wi-Fi, дозволяючи змінювати параметри системи у процесі роботи та отримувати дані з сенсорів у реальному часі. Другий режим

передбачає автоматичну генерацію коду на основі побудованої у Simulink моделі – після компіляції програма завантажується на мікроконтролер і може працювати автономно. Такий підхід дозволяє переходити від моделювання до реальної реалізації без суттєвих змін у структурі системи, що є надзвичайно корисним у біомеханічних проєктах, де потрібна точна відповідність моделі та фізичного пристрою [28].

У контексті керування смарт-пальцем людини MATLAB і Arduino використовуються як єдина інтегрована система для реалізації повного циклу – від зчитування біосигналів до виконання руху актуаторів. Наприклад, дані з ЕМГ-датчиків, що фіксують електричну активність м'язів, надходять у MATLAB, де вони фільтруються та аналізуються за допомогою спеціальних алгоритмів або нейронних мереж. Після обробки формується керуючий сигнал, який через MATLAB передається на Arduino, що у свою чергу генерує PWM-сигнали для керування сервоприводами. У результаті досягається плавне й координоване згинання або розгинання пальців, що дозволяє точно імітувати природний рух руки.

Для задач, що потребують високої обчислювальної продуктивності, MATLAB пропонує Parallel Computing Toolbox і GPU Coder: просте перетворення циклів for на конструкції parfor або gpuArray дозволяє задіяти багатоядерні процесори, обчислювальні кластери та графічні процесори NVIDIA без зміни алгоритмічної частини коду. Тим часом векторизація операцій автоматично прискорює виконання критичних обчислень через виклики оптимізованих бібліотек BLAS/LAPACK, а MATLAB Coder генерує переносимий C/C++ код для вбудованих систем, що суттєво спрощує інтеграцію з іншими мовами програмування та апаратними платформами [29].

Сучасні труболіси MathWorks – Image Processing Toolbox, Computer Vision Toolbox, Robotics System Toolbox, Deep Learning Toolbox – розширюють можливості візуалізації та проєктування: від обробки зображень і відео в режимі реального часу до побудови графічних дашбордів для моніторингу трендів у нейромережах або тестування алгоритмів розпізнавання об'єктів. Завдяки

підтримці популярних форматів обміну даними та інтеграції з Python, Java, ROS та TCP/IP-інтерфейсами MATLAB служить центром об'єднання різномірних модулів у єдине рішення для інтернету речей та індустрії 4.0 [30].

Нарешті, перспективними напрямками стає застосування доповненої (AR) та віртуальної (VR) реальності через MATLAB Support Package for Unity, а також поява цифрових двійників із телеметрією в реальному часі – Web App Server дозволяє впроваджувати їх у веб-середовище, а Simulink Cloud дозволяє запускати симуляції на віддалених серверах. Усе це створює екосистему, де цикл від ідеї до готового промислового рішення скорочується від місяців до днів, а висока ступінь автоматизації гарантує повторюваність, масштабованість і зрозумілість всіх етапів розробки.

Сучасні технології MATLAB також активно інтегруються з методами штучного інтелекту та глибинного навчання. Завдяки цьому візуальне моделювання перестає бути лише інженерним інструментом і стає платформою для створення «розумних» систем, які здатні адаптуватися до змін середовища. Наприклад, у біомеханіці MATLAB використовується для побудови моделей руху, тестування алгоритмів керування протезами та інтеграції сенсорних сигналів у єдину систему керування.

Технологія застосування MATLAB для керування смарт-пальцем базується на інтегрованому підході до моделювання, аналізу та апаратної реалізації біомеханічної системи. Основна ідея полягає у створенні математичної моделі пальця людини, її симуляції у середовищі MATLAB/Simulink, подальшій генерації керуючих сигналів та переданні їх на мікроконтролер Arduino, який безпосередньо керує виконавчими механізмами. Такий підхід забезпечує гнучкість, точність та можливість оперативного налаштування системи без необхідності змінювати апаратну частину.

Першим етапом є моделювання рухів пальця у середовищі MATLAB/Simulink. Використовуючи інструменти Simscape Multibody, створюється динамічна модель з урахуванням фалангової структури, мас, інерційних характеристик і кінематичних обмежень суглобів. Кожна фаланга

моделюється як ланка з одним ступенем вільності, а з'єднання між ними – як шарніри з відповідними моментами опору. На цьому етапі формується віртуальний аналог пальця, який реагує на вхідні сигнали (моменти сил або кути обертання) подібно до реального біомеханічного об'єкта.

Далі система доповнюється блоками керування, які формують логіку взаємодії між датчиками та приводами. MATLAB дозволяє реалізувати різні типи регуляторів – пропорційно-інтегрально-диференціальні (PID), нечіткі регулятори або моделі на основі машинного навчання. Ці алгоритми керування оптимізуються під конкретні завдання: наприклад, досягнення плавного руху, компенсації інерційних ефектів або стабільного утримання заданого положення.

Наступним кроком є підключення Arduino до MATLAB через пакет MATLAB Support Package for Arduino Hardware. Це забезпечує комунікацію між моделлю, побудованою у MATLAB, та реальною апаратною частиною системи. Завдяки цьому MATLAB може зчитувати дані з датчиків (наприклад, потенціометрів, датчиків згину, тензометрів чи ЕМГ-датчиків) і в реальному часі обчислювати відповідні керуючі сигнали, які надсилаються на Arduino. Arduino, у свою чергу, генерує PWM-сигнали (широкоімпульсну модуляцію), які керують серводвигунами або лінійними актуаторами, що забезпечують рух пальців.

У випадку використання ЕМГ-сигналів MATLAB виконує їхню попередню обробку – фільтрацію шумів, нормалізацію амплітуди, екстракцію особливостей (наприклад, середньоквадратичного значення або частоти сигналу) – та перетворює їх у керуючі команди. Це дозволяє смарт-пальцю реагувати на реальні біоелектричні сигнали користувача, що робить систему максимально природною.

Однією з ключових переваг технології є можливість Hardware-in-the-Loop (HIL) симуляції, коли MATLAB-симуляція та фізичний пристрій працюють одночасно. Це дає змогу протестувати поведінку смарт-пальця ще до його повної реалізації, змінювати параметри в реальному часі та аналізувати точність виконання рухів. MATLAB також дозволяє візуалізувати процеси – відображати

траєкторії рухів пальців, кути згинання суглобів або напруження у сухожиллях у вигляді 3D-анімацій.

Після налагодження системи MATLAB може автоматично генерувати код C/C++ для Arduino на основі побудованої моделі у Simulink. Цей код компілюється і завантажується на мікроконтролер, що дозволяє смарт-пальцю працювати автономно без постійного підключення до комп'ютера.

Таким чином, технологія застосування MATLAB для керування смарт-пальцем передбачає повний цикл:

- моделювання біомеханічної системи пальця у середовищі MATLAB/Simulink;
- створення та оптимізацію регуляторів;
- підключення сенсорів і виконавчих механізмів через Arduino;
- симуляцію рухів у режимі реального часу;
- автоматичну генерацію коду для автономної роботи системи.

Завдяки такій інтеграції MATLAB і Arduino утворюють потужну платформу для досліджень у галузі біомеханіки та смарт-протезування. MATLAB забезпечує високий рівень математичного та візуального моделювання, тоді як Arduino – гнучке й доступне апаратне виконання, що робить цю комбінацію оптимальною для створення роботизованих пристроїв нового покоління, таких як смарт-палець людини.

Отже, сучасні технології візуального проєктування на базі MATLAB забезпечують універсальний підхід до створення й перевірки інженерних систем. Їхнє використання дозволяє скоротити час розробки, знизити ймовірність помилок та створити основу для швидкої інтеграції розроблених рішень у практичні застосування. Завдяки поєднанню візуального середовища, математичної строгості та широких можливостей для інтеграції, MATLAB залишається одним із провідних інструментів для досліджень та розробки в інженерії, біомеханіці та інформаційних технологіях.

2.2 Характеристика мови програмування Matlab та програмно-апаратної платформи

Matlab – це високорівнева мова програмування та інтегроване середовище для наукових розрахунків, моделювання, аналізу даних та розробки систем керування. Вона надає широкий спектр функцій для чисельних обчислень, лінійної алгебри, обробки сигналів, статистики, оптимізації та візуалізації результатів. Однією з головних переваг Matlab є його інтеграція з Simulink – платформою для моделювання динамічних систем та створення блок-схем алгоритмів керування, що дозволяє швидко тестувати і відлагоджувати моделі у графічному середовищі.

Програмно-апаратна платформа Matlab/Simulink підтримує взаємодію з різними мікроконтролерами, зокрема Arduino, що дозволяє реалізувати алгоритми керування на реальному обладнанні в реальному часі. Використання цієї платформи дає змогу здійснювати апаратно-програмне тестування систем, отримувати зворотний зв'язок від сенсорів і перевіряти роботу виконавчих механізмів безпосередньо під час розробки. Matlab також підтримує бібліотеки для інтеграції з зовнішніми пристроями, інструменти для обробки сигналів від датчиків, а також можливість створювати адаптивні алгоритми керування на основі експериментальних даних [31].

Завдяки своїй універсальності, Matlab дозволяє інженерам і дослідникам швидко прототипувати, моделювати та оптимізувати складні біомеханічні системи, забезпечуючи високу точність обчислень та ефективну реалізацію алгоритмів керування, що порівняно в таблиці 2.1.

Таблиця 2.1 – Основні переваги та недоліки Matlab у проектуванні біомеханічних систем

Переваги	Недоліки
Широкий набір бібліотек для чисельних обчислень та обробки сигналів	Висока вартість ліцензії для повнофункціональної версії

Продовження таблиці 2.1

Переваги	Недоліки
Інтеграція з Simulink для графічного моделювання систем	Вимагає високих апаратних ресурсів при складних симуляціях
Можливість підключення мікроконтролерів (Arduino, Raspberry Pi)	Не завжди оптимальна швидкість виконання коду у порівнянні з низькорівневими мовами
Підтримка роботи з сенсорними та виконавчими модулями	Обмежена переносимість моделей на інші програмні середовища без адаптації
Зручні інструменти для візуалізації та аналізу результатів	Потрібні базові знання MATLAB/Simulink для ефективною роботи

2.3 Основні характеристики програмної платформи Arduino

Arduino – це відкрита апаратно-програмна платформа, призначена для швидкого прототипування електронних пристроїв та систем керування. Вона складається з серії мікроконтролерних плат, які легко програмуються за допомогою спеціального середовища Arduino IDE, та підтримує підключення різноманітних датчиків і виконавчих механізмів. Завдяки своїй простоті та доступності, Arduino широко використовується у робототехніці, автоматизації, навчанні та дослідницьких проєктах.

Однією з ключових переваг платформи є відкритий характер апаратної та програмної частини, що дозволяє розробникам створювати власні модулі, розширення та бібліотеки. Arduino підтримує широкий спектр мов програмування, включаючи C/C++, і має готові бібліотеки для роботи з сенсорами, сервомоторами, двигунами, комунікаційними інтерфейсами (I2C, SPI, UART) та бездротовими модулями (Wi-Fi, Bluetooth).

Мікроконтролерні плати Arduino відрізняються різним об'ємом оперативної пам'яті, кількістю цифрових та аналогових входів/виходів, потужністю обробки та розміром фізичної плати. Платформа дозволяє реалізовувати алгоритми керування в реальному часі, збирати дані від сенсорів,

передавати їх на зовнішні пристрої та виконувати адаптивні дії відповідно до отриманої інформації.

Завдяки гнучкості та універсальності Arduino стала стандартним інструментом у розробці прототипів роботизованих систем, включаючи біомеханічні моделі пальців та роботизовані маніпулятори. Вона дозволяє швидко перевіряти алгоритми керування та інтегрувати апаратні компоненти з програмним забезпеченням, що значно скорочує час розробки та тестування систем. Розглянемо основні характеристики плат в таблиці 2.2.

Таблиця 2.2 – Основні характеристики популярних плат Arduino

Плата	Мікроконтролер	Частота роботи	Пам'ять Flash	Оперативна пам'ять (RAM)	Цифрові входи/виходи	Аналогові входи	Особливості
Arduino Uno	ATmega328P	16 МГц	32 КБ	2 КБ	14	6	Найпопулярніша плата, ідеальна для початківців
Arduino Mega 2560	ATmega2560	16 МГц	256 КБ	8 КБ	54	16	Велика кількість входів/виходів, підходить для складних проектів
Arduino Nano	ATmega328P	16 МГц	32 КБ	2 КБ	14	8	Компактна плата, зручна для прототипування у малих пристроях
Arduino Due	ATSAM3X8E	84 МГц	512 КБ	96 КБ	54	12	Підтримка 32-бітних обчислень, високопродуктивна плата для складних систем
Arduino Leonardo	ATmega32u4	16 МГц	32 КБ	2,5 КБ	20	12	Має можливість емуляції USB-пристрою (клавіатура, миша)
Arduino MKR WiFi 1010	SAMD21	48 МГц	256 КБ	32 КБ	22	7	Вбудований Wi-Fi і Bluetooth, підходить для IoT-проектів

2.4 Архітектура та основні компоненти платформи Arduino

Платформа Arduino побудована на базі мікроконтролера, який є центральним елементом апаратної частини та відповідає за обробку даних, виконання алгоритмів керування і взаємодію з підключеними пристроями. Архітектура плат включає кілька основних компонентів:

- мікроконтролер – головний обчислювальний елемент, що виконує програмний код. Найпоширенішими є ATmega328P (Arduino Uno, Nano), ATmega2560 (Mega) та ARM-базовані процесори (Arduino Due, MKR). Мікроконтролер забезпечує роботу цифрових та аналогових входів/виходів, а також обробку сигналів від датчиків і керування виконавчими пристроями;

- цифрові входи/виходи (Digital I/O) – контакти для підключення сенсорів, світлодіодів, кнопок та виконавчих пристроїв. Вони можуть працювати як на вході (зчитування стану), так і на виході (управління сигналом);

- аналогові входи (Analog Input) – використовуються для підключення датчиків, що формують змінний сигнал (напруга від 0 до 5 В або іншого діапазону залежно від плати). Ці входи дозволяють отримувати більш точну інформацію про фізичні величини (температуру, силу, тиск);

- шини комунікації – плати підтримують стандарти I2C, SPI та UART для взаємодії з іншими мікроконтролерами, датчиками або периферійними пристроями. Це дозволяє створювати складні системи з великою кількістю компонентів;

- порт живлення – забезпечує подачу електроживлення на плату та підключені модулі. Arduino може житися через USB або зовнішній адаптер (5-12 В);

- програмне середовище Arduino IDE – забезпечує розробку, компіляцію та завантаження програм у мікроконтролер. IDE підтримує бібліотеки для роботи з сенсорами, сервоприводами, двигунами та комунікаційними модулями, що спрощує створення складних проектів;

– додаткові модулі та щити (Shields) – дозволяють швидко розширювати функціональність плати, додаючи Wi-Fi, Bluetooth, Ethernet, дисплеї, моторні драйвери та інші периферійні пристрої.

Архітектура Arduino поєднує простоту, гнучкість та відкритість, що робить платформу ідеальною для прототипування біомеханічних систем, роботизованих маніпуляторів і сенсорних пристроїв. Вона дозволяє поєднувати апаратне забезпечення та алгоритми керування в єдину інтегровану систему, забезпечуючи роботу у реальному часі та легку масштабованість проєктів.

2.5 Переваги та обмеження використання Arduino для смарт-проектування роботизованих пристроїв

Платформа Arduino завдяки своїй гнучкості, відкритості та доступності стала однією з найпопулярніших у розробці роботизованих пристроїв та смарт-систем. Основні переваги та обмеження використання Arduino представлені в таблиці 2.3.

Таблиця 2.3 – Переваги та обмеження використання Arduino

Категорія	Переваги	Обмеження
Простота	Легко навчитися та програмувати; готові бібліотеки для сенсорів і моторів	Обмежена багатозадачність; однопоточне виконання коду
Гнучкість	Відкрита апаратна та програмна платформа; можна модифікувати та розширювати	Потрібні додаткові модулі для складних комунікацій та живлення
Сумісність	Підтримка великої кількості датчиків, моторів, дисплеїв та Shields	Аналогові входи мають обмежену точність і швидкість
Масштабованість	Можна об'єднувати кілька плат, підключати до IoT та хмарних сервісів	Обмежена обчислювальна потужність і пам'ять
Вартість	Дешева та доступна апаратна частина	Для потужних виконавчих пристроїв потрібні зовнішні драйвери
Прототипування	Ідеальна для навчальних та експериментальних проєктів	Не завжди підходить для промислових систем із високими вимогами

2.6 Опис мови програмування Arduino та огляд середовища розробки

Мова програмування Arduino є спрощеною реалізацією C/C++ із набором спеціалізованих бібліотек, що дозволяють швидко і зручно взаємодіяти з апаратною частиною мікроконтролера. Основна мета мови Arduino – забезпечити простий доступ до сенсорів, виконавчих пристроїв та периферії, зменшуючи складність низькорівневого програмування мікроконтролерів.

Програмування Arduino базується на двох основних функціях: `setup()` і `loop()`. Функція `setup()` використовується для ініціалізації всіх апаратних компонентів і виконується лише один раз на старті програми. Функція `loop()` містить основний код роботи пристрою і виконується циклічно протягом всього часу роботи системи. Така структура дозволяє легко створювати алгоритми управління роботизованими компонентами в реальному часі.

Середовище розробки Arduino IDE (Integrated Development Environment) забезпечує користувачеві:

- зручний редактор коду з підсвіткою синтаксису;
- можливість компіляції та завантаження програм на плату одним натисканням кнопки;
- доступ до великої кількості готових бібліотек для роботи з сенсорами, моторами, сервоприводами, дисплеями та комунікаційними модулями;
- монітор послідовного порту для відлагодження та виведення даних з мікроконтролера.

Мова Arduino підтримує роботу з цифровими та аналоговими входами і виходами, PWM-сигналами, серійною передачею даних, а також інтеграцію з бездротовими модулями (Wi-Fi, Bluetooth). Завдяки цьому, платформа дозволяє реалізовувати як прості експериментальні проекти, так і складні смарт-системи, включно з роботизованими маніпуляторами та біомеханічними протезами.

Надзвичайною перевагою Arduino є велика спільнота користувачів та розробників, яка підтримує численні ресурси, приклади проектів, документацію

та форуми. Це робить платформу особливо зручною для навчання, прототипування та дослідницьких робіт у сфері робототехніки та автоматизації.

2.7 Підключення та налаштування Arduino-плати

Arduino Uno – це базова модель апаратної платформи Arduino (рис. 2.1), яка найчастіше використовується у навчальних, дослідницьких та інженерних проектах. Плата побудована на мікроконтролері ATmega328P (рис. 2.2) і має оптимальне співвідношення між функціональністю та простотою використання.



Рисунок 2.1 – Зовнішній вигляд плати Arduino UNO [32]



Рисунок 2.2 – Елементи плати Arduino Uno [33]

Підключення Arduino Uno (рис. 2.3) здійснюється через стандартний USB-кабель типу A-B, який забезпечує не лише передачу даних, а й живлення пристрою (5 В). Після підключення операційна система автоматично розпізнає пристрій та встановлює необхідні драйвери. Якщо цього не відбувається, користувач може вручну встановити драйвери з офіційного сайту.

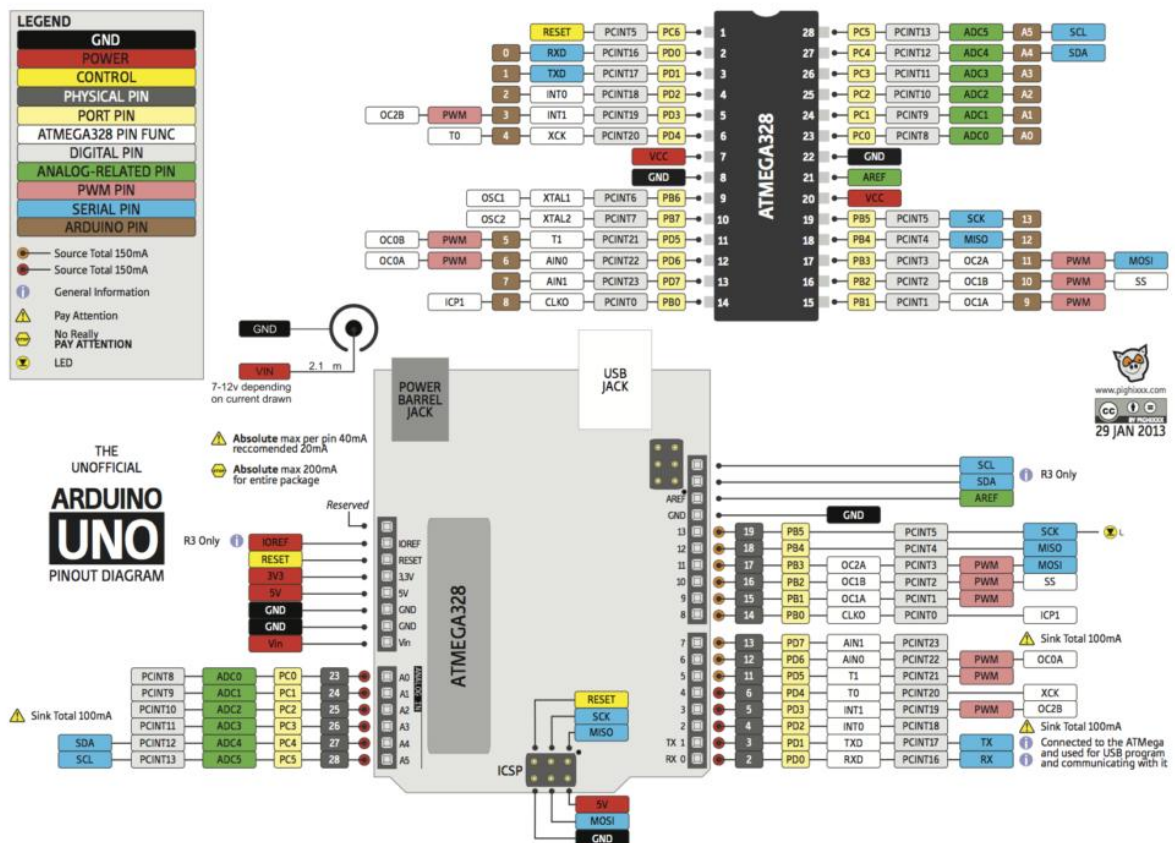


Рисунок 2.3 – Розпіновка плати на базі мікроконтролера ATMEGA 328 [34]

Після встановлення Arduino IDE потрібно виконати кілька базових кроків для підготовки плати до роботи:

Відкрити Arduino IDE (рис. 2.4); у меню вибрати Tools, потім Board, потім вибрати модель Arduino Uno; у меню Tools, потім Port обрати відповідний COM-порт, до якого підключено плату; перевірити правильність налаштувань у вкладці Processor (ATmega328P); натиснути Verify для перевірки коду, або Upload для завантаження програми на плату.

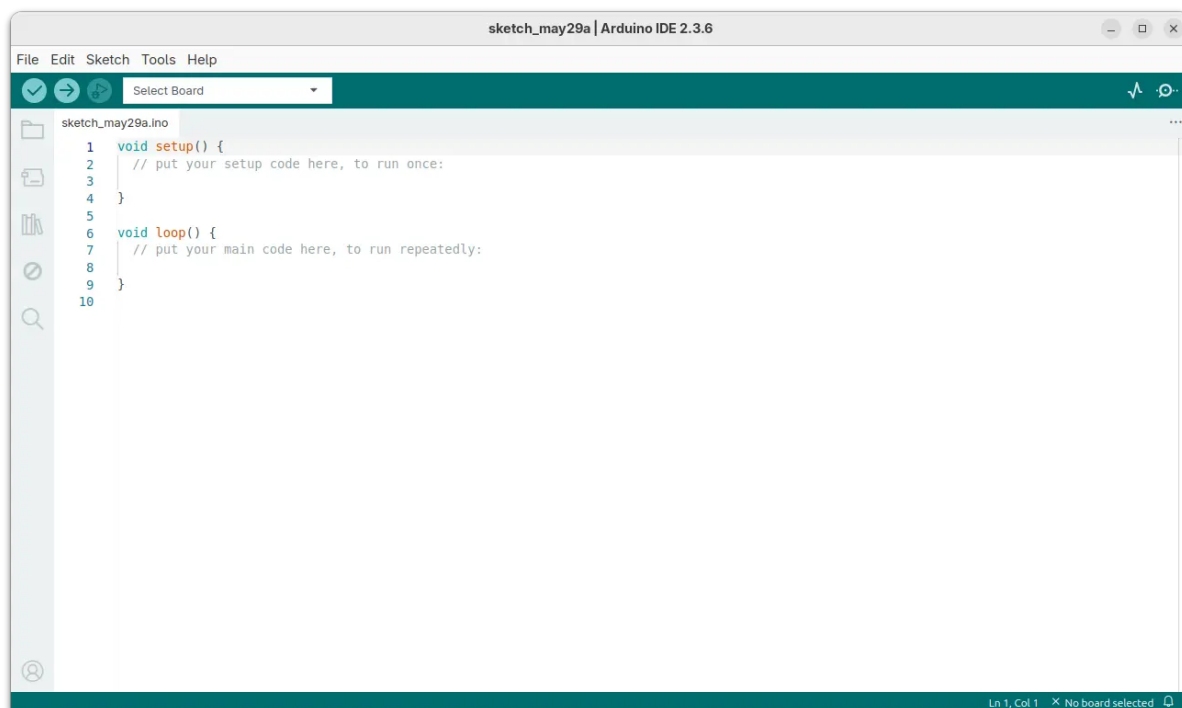


Рисунок 2.4 – Середовище розробки Arduino IDE [35]

Для перевірки працездатності плати зазвичай використовується стандартний приклад програми Blink, яка змушує мигати вбудований світлодіод (на піні 13). Це підтверджує, що програмування та передача коду на плату виконані успішно. Основні характеристики плати розглянуто в таблиці 2.4.

Таблиця 2.4 – Основні технічні характеристики Arduino Uno

Характеристика	Значення
Мікроконтролер	ATmega328P
Робоча напруга	5 В
Вхідна напруга (рекомендована)	7-12 В
Кількість цифрових входів/виходів	14 (6 з них – PWM)
Кількість аналогових входів	6
Тактова частота	16 МГц
Пам'ять Flash	32 КБ (з яких 0,5 КБ використовується завантажувачем)
ОЗП (SRAM)	2 КБ
Енергонезалежна пам'ять (EEPROM)	1 КБ
Інтерфейси зв'язку	UART, SPI, I ² C
Порти живлення	USB або зовнішнє джерело
Вбудований світлодіод	Пін 13

Процес підключення та налаштування плати Arduino Uno є інтуїтивно зрозумілим навіть для початківців. Завдяки простому інтерфейсу Arduino IDE та стандартним бібліотекам користувач може швидко розпочати розробку власних проектів у галузі автоматизації, робототехніки та IoT.

2.8 Особливості програмування за допомогою Arduino та Matlab

Програмування з використанням платформи Arduino та середовища MATLAB є сучасним підходом до створення інтелектуальних систем керування, що поєднує простоту роботи з апаратною частиною та потужні аналітичні можливості високорівневого програмного середовища. Такий підхід активно використовується в наукових дослідженнях, освітніх проектах та розробці роботизованих систем, де важливо забезпечити ефективну взаємодію між датчиками, виконавчими механізмами та алгоритмами обробки даних.

Середовище Arduino IDE базується на мові програмування, подібній до C/C++, і дозволяє безпосередньо працювати з апаратними ресурсами мікроконтролера. Програми, створені в цьому середовищі, мають структуру, що включає дві основні функції – `setup()` і `loop()`. Перша виконується один раз після запуску пристрою та використовується для ініціалізації портів, змінних і бібліотек, тоді як друга містить основний цикл, який виконується безперервно, реалізуючи логіку роботи системи. Цей підхід забезпечує високу швидкодію та точність виконання, що є важливим у системах реального часу. Однак програмування на низькому рівні потребує більш глибоких знань про апаратну структуру плати та може бути складним для реалізації складних алгоритмів аналізу чи моделювання.

На відміну від Arduino IDE, середовище MATLAB орієнтоване на високорівневе програмування та наукові обчислення. Завдяки пакету MATLAB Support Package for Arduino Hardware, користувач може безпосередньо підключити плату Arduino до MATLAB і взаємодіяти з нею без написання складного коду. Наприклад, можна зчитувати аналогові дані з датчиків або

керувати світлодіодами, двигунами чи сервоприводами за допомогою кількох простих команд. MATLAB дозволяє не лише керувати пристроями, а й одразу обробляти отримані дані – фільтрувати, аналізувати, будувати графіки, виконувати математичні розрахунки або створювати візуальні інтерфейси користувача.

Крім того, MATLAB підтримує інтеграцію з середовищем Simulink, що відкриває широкі можливості для моделювання процесів керування в реальному часі. Це особливо важливо під час розробки роботизованих систем або біомеханічних моделей, де необхідно дослідити динаміку рухів, поведінку системи під дією різних збурень або протестувати алгоритми керування безпосередньо на апаратному рівні. У Simulink можна створити блок-схему, яка з'єднує математичну модель з реальним пристроєм, дозволяючи проводити експерименти без ручного програмування мікроконтролера.

Використання MATLAB у поєднанні з Arduino також дозволяє реалізовувати концепції адаптивного керування, нечіткої логіки та нейронних мереж, що є особливо актуальним для сучасних систем автоматизації, смарт-протезування та робототехніки. MATLAB надає інструменти для створення складних моделей, які враховують нелінійні зв'язки, інерційні властивості та зовнішні збурення, тоді як Arduino забезпечує фізичну реалізацію цих алгоритмів на реальних пристроях.

Отже, поєднання Arduino та MATLAB створює потужну платформу для розробки, моделювання та тестування інтелектуальних систем керування. Arduino відповідає за збирання даних і взаємодію з реальним середовищем, тоді як MATLAB виконує роль обчислювального ядра, яке забезпечує аналітичну обробку інформації, оптимізацію алгоритмів і візуалізацію результатів. Така інтеграція забезпечує ефективність, гнучкість і високу точність у процесі створення роботизованих і біомеханічних систем нового покоління.

РОЗДІЛ 3

РОЗРОБКА ІМІТАТОРА ПАЛЬЦЯ ЛЮДИНИ НА БАЗІ ARDUINO

3.1 Принципи розробки установки та функціональні вимоги

Сучасні тенденції у біомедичній інженерії свідчать про активне використання відкритих платформ і рішень з відкритим вихідним кодом (Open Source) для створення прототипів штучних кінцівок людини. Прикладом таких рішень є проекти класу OpenSource Humanoid Robot, серед яких відомий 3D PRINTED LIFE SIZE ROBOT InMoov Humanoid Robot, що об'єднує 3D-друк та робототехніку. Подібні ініціативи активно впроваджуються в освітніх та науково-дослідних лабораторіях у різних країнах світу.

В рамках кваліфікаційної роботи створено демонстраційну установку штучної кінцівки, зокрема – штучного пальця руки. Цей проект ставить за мету не лише повторення функціоналу open-source рішень, а й їх вдосконалення, спрямоване на оптимізацію механіки та розширення можливостей керування.

Розробка штучного пальця передбачає інтеграцію 3D-друку, електронних компонентів та алгоритмів керування:

Механічна частина проекту передбачає створення 3D-моделі пальця з урахуванням біомеханічних характеристик людської руки (сегментація фаланг, наявність шарнірних з'єднань). Модель друкується на 3D-принтері з біосумісних або міцних полімерних матеріалів.

Актuatorи забезпечені встановленням мініатюрних сервоприводів для забезпечення згинання та розгинання кожної фаланги. Передбачається незалежне керування кожним сегментом пальця.

Сенсори реалізуються через використання датчиків сили та положення (наприклад, на основі потенціометрів або тензодатчиків), що дають змогу відслідковувати рівень зусилля й точність руху.

Система керування установкою забезпечується наступним чином: Здійснюється розробка програми на базі Arduino з метою реалізації

індивідуального управління приводами кожної фаланги пальця. Планується інтеграція з планшетним ПК для візуалізації рухів і налаштування режимів.

Інтерфейс взаємодії передбачає можливість використання сенсорних систем, зокрема Microsoft Kinect або електроміографічних датчиків (ЕМГ), для відтворення рухів пацієнта у вигляді керованого «аватара».

Розроблений прототип штучного пальця демонструє принципи індивідуального керування сегментами штучної кінцівки людини. Це дозволить:

- дослідити можливості оптимізації керування на основі алгоритмів (наприклад, PID або LQR-регуляторів);
- відпрацювати технологію 3D-моделювання та друку біонічних структур;
- створити експериментальну платформу для подальших досліджень у напрямку протезування та реабілітації;
- реалізувати концепцію адаптивного протезу, здатного імітувати як базові рухи (згинання/розгинання), так і складніші функції, пов'язані з маніпуляцією предметами.

Таким чином, дана розробка є кроком до створення повнофункціональних протезів верхніх кінцівок, у яких поєднуються механічна точність, електронна адаптивність та інтеграція з біосенсорними інтерфейсами, що відкриває нові можливості для сучасної реабілітаційної інженерії.

3.2 Опис установки та функціональна схема автоматизації імітатора пальця на базі Arduino

У даному параграфі описано функціональну схему автоматизації імітатора пальця на базі Arduino, яка показує взаємодію основних елементів системи керування, такі як датчики, платформа Arduino, програмні алгоритми та відповідні виконавчі механізми. Завдання проєкту включає аналіз різних моделей пальців, кисті або руки людини, які використовуються для імітації руху кінцівки людини у роботі-гуманоїді. Розробка системи механізованого штучного пальця і руки, здатної імітувати людську руку, здатна бути платформою для

відповідних експериментів у навчальних цілях з подальшим застосуванням у вирішенні проблем медичної реабілітації людей з інвалідністю.

Модель пальця з інтелектуальним керуванням створюється на базі та за прикладом проекту відкритого коду (OpenSource Humanoid Robot), за прикладом проекту 3D PRINTED LIFE SIZE ROBOT, InMoov.ii Humanoid Robot реалізованого у Novia University of Applied Science (Фінляндія). 3D-модель пальця видруковується за допомогою 3D-принтера та додаються інші частини установки для забезпечення її функціонування. Установка вимагає встановлення поворотних приводів (сервоприводів), пристрою комутації з датчиком руху, платформа для керування Arduino.

Узагальнена логіка роботи описується такою функціональною схемою (рис. 3.1).

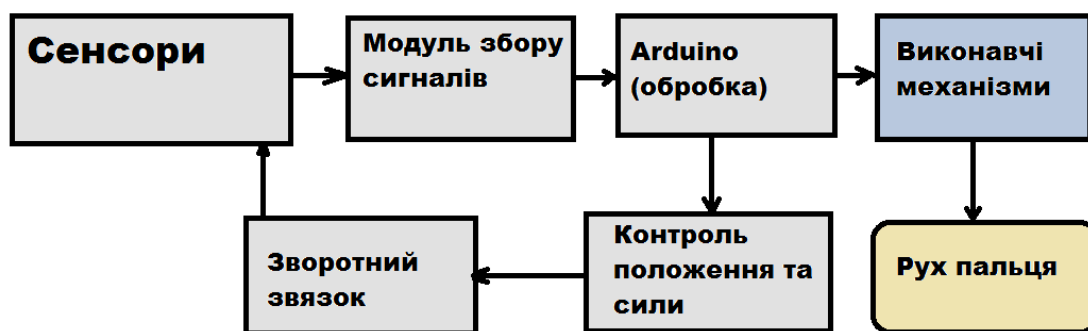


Рисунок 3.1 – Функціональна схема автоматизації імітатора пальця на базі Arduino

Функціональна схема автоматизації імітатора пальця на базі Arduino має наступні блоки:

- 1) сенсорний рівень (Input) включає:
 - датчики кута згину (flex-сенсори/потенціометри);
 - датчики сили натискання (FSR-сенсори);
 - опціонально – EMG-датчики (для зчитування м'язових сигналів);
- 2) контролер (Processing): Arduino UNO;
- 3) модуль обробки даних включає функції:

- фільтрації сигналів;
 - нормалізації значень;
 - алгоритми керування (PID, LQR або логіка на основі порогів);
- 4) виконавчий рівень (Output) включає:
- серводвигуни / мікродвигуни для згину і розгинання пальця;
 - приводи зворотного зв'язку;
- 5) людино-машинний інтерфейс (HMI):
- панель керування (кнопки, потенціометри, джойстик);
 - LCD/OLED дисплей для відображення параметрів;
 - Bluetooth/Wi-Fi модуль (ESP8266/HC-05) для бездротового керування;
- 6) зворотний зв'язок (Feedback) базується на:
- передачі даних від сенсорів у реальному часі на Arduino;
 - візуалізації рухів на дисплеї або ПК;
 - автоматичній корекції положення пальця.

3.3 Вибір технічних засобів та виконавчих механізмів

Основними компонентами роботизованого пальця та рукавички як елемента пристрою керування є сам механізований палець, сервопривод, платформа Arduino, рукавичка та датчик гнучкості.

Рукавичка облаштовується датчиком (сенсором) гнучкості: змінним резистором, який змінює значення опору при згині. Він прикріплений до однієї сторони дільника напруги, а резистор постійного значення поміщений з іншого боку. Arduino зчитує зміну напруги, коли датчик зігнутий і запускає сервопривод рухатися відповідно сенсору. Сервопривод тягне струну, яка діє як сухожилля, дозволяючи пальцю рухатися.

Палець є частиною проєкту з відкритим кодом під назвою InMoov, що містить частини тіла робота. Друк частини пальця виконувався зі стандартним дозволом, щоб уникнути небажаного тертя (рис. 3.2).

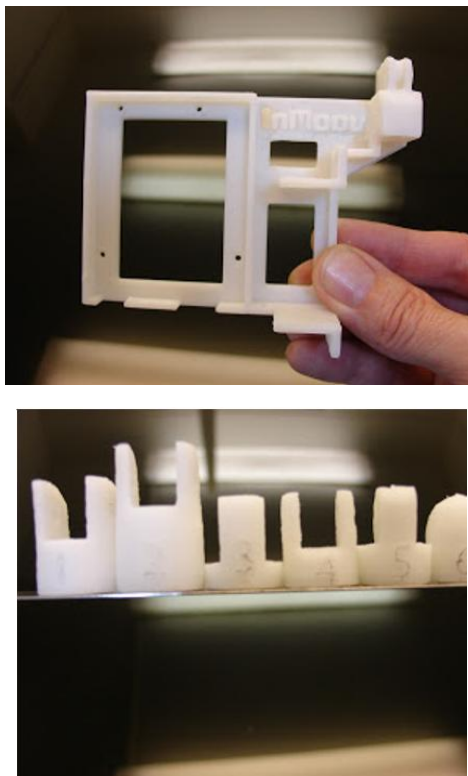


Рисунок 3.2 – Роздрукування деталей пальця

Гнучкі датчики всієї руки потребують розробки відповідної схеми, таким чином, щоб вони були сумісні з платформою Arduino.

Дільник напруги має наступні особливості: відповідні датчики гнучкості є змінними резисторами, а в парі з резисторами статичного значення, зміну опору (у цьому випадку згинання датчика) можна відчутти через зміну напруги між резисторами. Це можна виміряти з використанням Arduino через його аналогові входи. Розроблена схема має наступний вигляд: червоний колір відображає позитивний полюс напруги, чорний – негативний, синій – комутатор підключений до Arduino (рис. 3.3).

Резистори мають ємність 22К. Основний кабель GND (заземлення), який під'єднується до всіх окремих проводів GND від датчиків, підключається до GND Arduino. Напруга +5 В від Arduino подається до основної мережі, а кабель позитивної напруги і кожен синій кабель підключається до окремого аналогового входу.

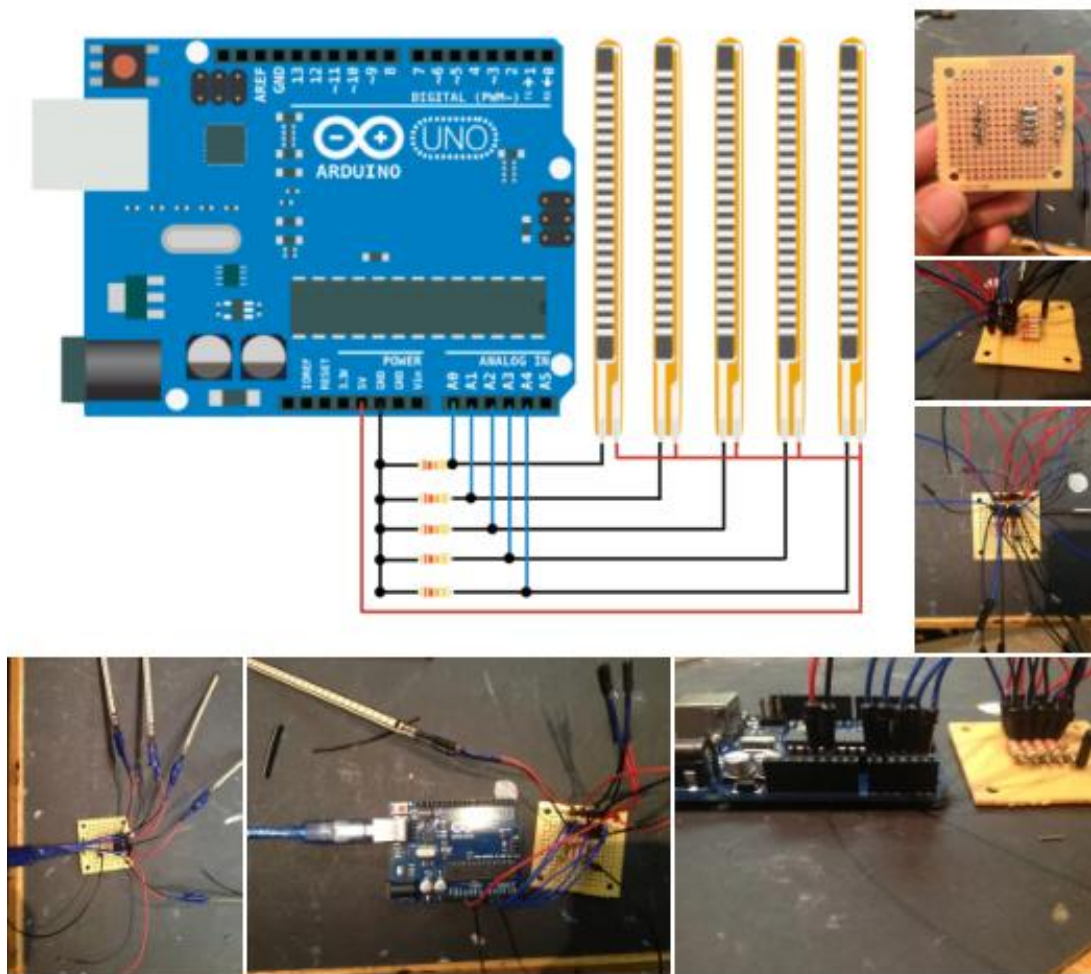


Рисунок 3.3 – Схема з'єднань

Виконується припаювання схеми на невелику друковану плату від RadioShack, таку, яку можна легко встановити на рукавичку. Також виконано припаювання кабелів до датчиків з використанням термоусадки, щоб не було короткого замикання. Також необхідно ізолювати місце, де знаходяться кабелі під'єднані до датчиків ізоляційною стрічкою для стабілізації датчиків. Біля дна, де проводи прикріплені, датчики трохи слабші, і стрічка гарантує, що вони не згинаються занадто далеко і не пошкодяться.

3.4 Процес проєктування демонстраційної установки

Процес проєктування демонстраційної установки імітатора-аватара пальця робота-гуманоїда здійснюється за нижчеподаними схемами і ілюстраціями. На

рисунках нижче показано вигляд пальця робота-гуманоїда та модуль керування у вигляді рукавиці з сенсором. Подано функціонал у вигляді згинання вказівного пальця та керований згин через «керуючу рукавицю» (рис. 3.4).



Рисунок 3.4 – Проектування пальця. Конструювання деталей

На наступному рисунку подано розробку пальців та показана операція згину (рис. 3.5).

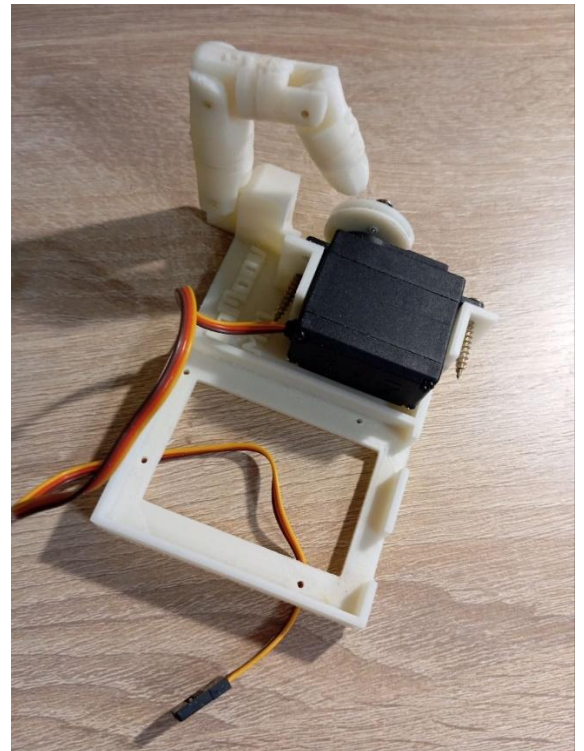
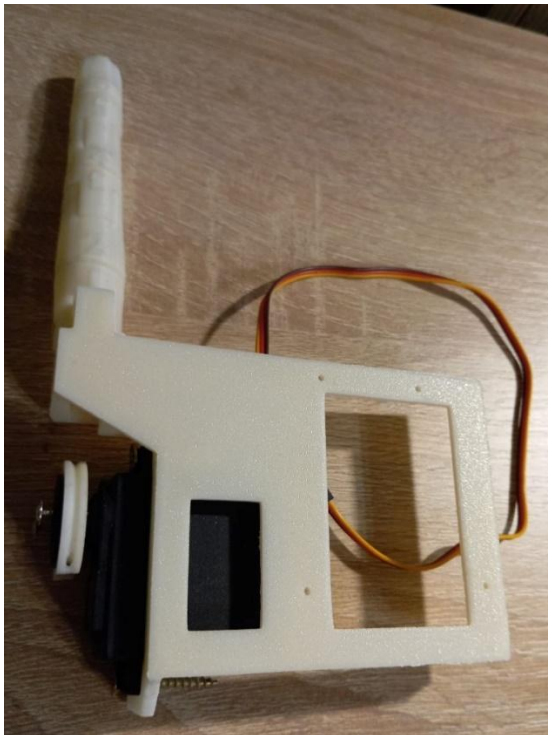


Рисунок 3.5 – Розробка пальців. Операція згину

Наступним кроком є встановлення сервоприводу і системи керування на базі Arduino та встановлення і підшивання сенсора (рис. 3.6-3.7).

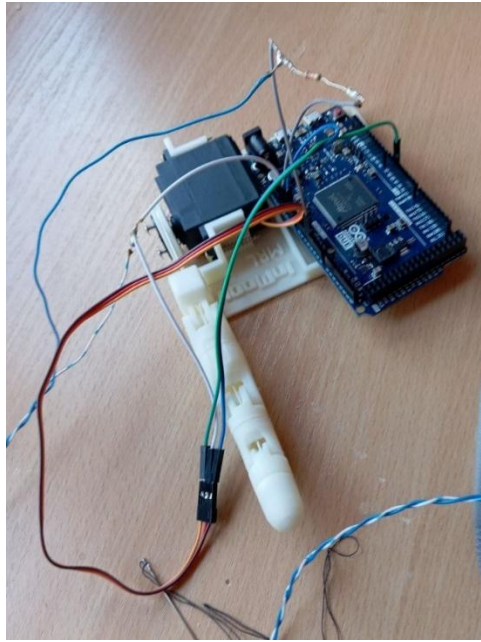


Рисунок 3.6 – Встановлення сервоприводу і системи керування на базі Arduino

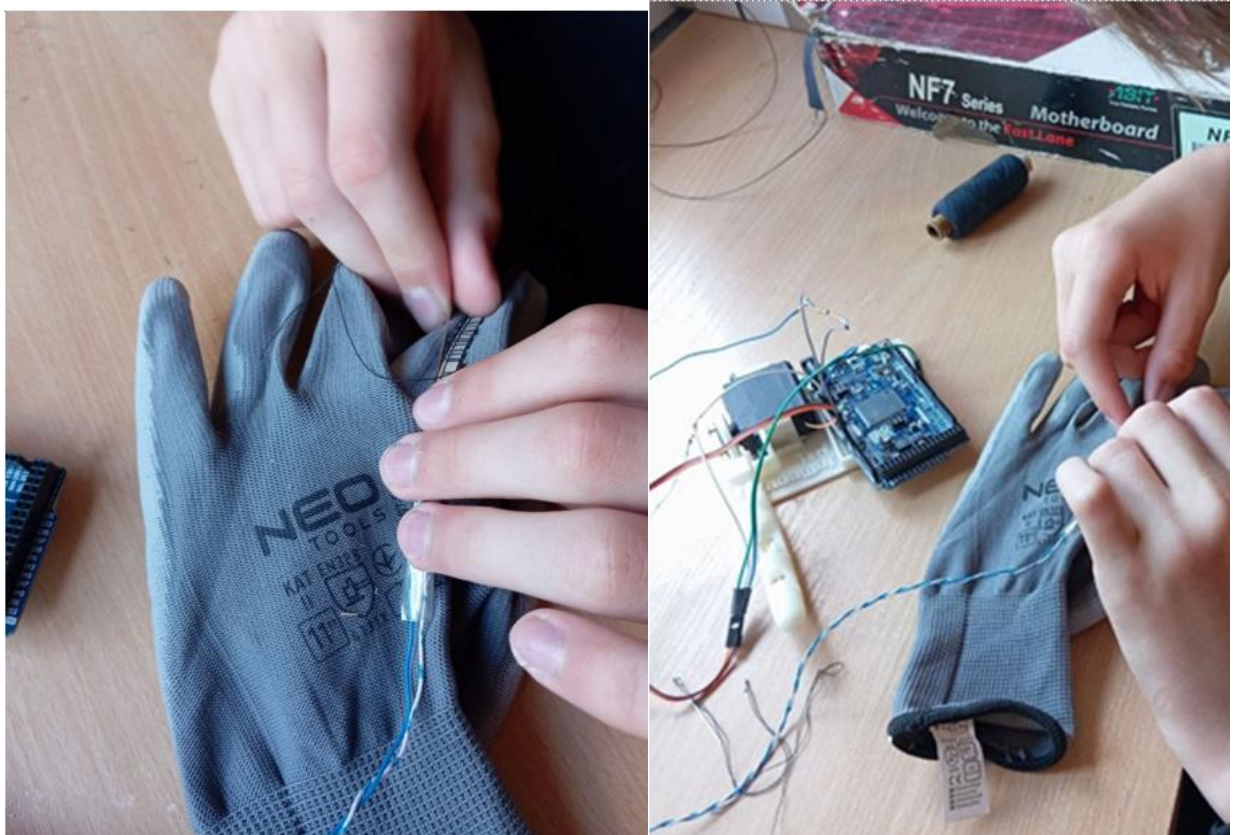


Рисунок 3.7 – Встановлення і підшивання сенсора

Основними компонентами роботизованого пальця та рукавички як елемента пристрою керування є сам механізований палець, сервопривод, платформа Arduino, рукавичка та датчик гнучкості.

Рукавичка облаштовується датчиком (сенсором) гнучкості: змінним резистором, який змінює значення опору при згині. Він прикріплений до однієї сторони дільника напруги, а резистор постійного значення поміщений з іншого боку. Arduino зчитує зміну напруги, коли датчик зігнутий і запускає сервопривод рухатися відповідно сенсору. Сервопривод тягне струну, яка діє як сухожилля, дозволяючи пальцю рухатися. Реалізація процесу створення механізованого пальця робота-гуманоїда здійснюється згідно наступних етапів:

Етап 1. Підготовка матеріалів і збірка конструкції.

Перелік необхідних матеріалів наступний:

- 1 сервопривод MG946R (або еквівалент);
- 4,5-дюймовий гнучкий датчик;
- 1 установка Arduino Uno або еквівалент;
- 1 резистор ємністю 22k;
- 1 батарея 6,0-7,2 В (для сервоприводу);
- акумулятор, перемички макетної плати/провід підключення;
- 1 рукавичка для лівої руки;
- гвинти, шурупи, рибальська струна (повинна мати високу міцність на розрив), гарячий клей;
- наждачний папір (~220 зернистість);
- голка і нитка;
- електрична дріль;
- паяльник;
- 3D принтер.

Етап 2. Розрукування на 3D-принтері частин механізованого смарт-пальця (рис. 3.1) та створення робочої схеми гнучких сенсорів.

Гнучкий датчик пальця потребує розробки відповідної схеми, таким чином, щоб він був сумісний з платформою Arduino. При цьому дільник напруги має наступні особливості: відповідні датчики гнучкості є змінними резисторами, а в парі з резисторами статичного значення, зміну опору (у цьому випадку згинання датчика) можна відчувати через зміну напруги між резисторами. Це можна виміряти з використанням Arduino через його аналогові входи. Резистори мають ємність 22К. Основний кабель GND (заземлення), який під'єднується до всіх окремих проводів GND від датчиків, підключається до GND Arduino. Напруга +5 В від Arduino подається до основної мережі, а кабель позитивної напруги, і кожен синій кабель підключається до окремого аналогового входу.

Виконується припаювання схеми на невелику друковану плату від RadioShack, таку, яку можна легко встановити на рукавичку. Також виконано припаювання кабелів до датчика, з використанням термоусадки, щоб не було короткого замикання.

Після цього необхідно ізолювати місце, де знаходяться кабелі під'єднані до датчиків ізоляційною стрічкою для стабілізації датчиків. Біля дна, де проводи прикріплені, датчики трохи слабші, і стрічка гарантує, що вони не згинаються занадто далеко і не пошкодяться.

Етап 3. Збірка механізованого пальця.

На даному етапі здійснюється спочатку кінцева підшивка рукавички. Після цього встановлюється датчик та відповідна схема на рукавичці. Спочатку просвердлюється маленький отвір у пластичній основі датчика (у верхній частині, де закінчується резистивний матеріал), при цьому слід слідкувати за тим, щоб не зачепити резистивний матеріал.

Далі надягається рукавичка і щільно притягується до руки. На пальці олівцем або ручкою помічаються невеликі лінії на вершині кожного суглоба чи кістки. Це підкаже точні місця вшивання датчика.

Після цього підшивається кожен наконечник датчика до області пальця трохи вище нігтя (використано щойно просвердлений отвір). Потім обмотується нитка навколо датчика над обома суглобами пальця. Після того, як датчик буде

на місці та добре ковзне під петлями нитки, щільно пришивається друкована плата до частини зап'ястя рукавички. Далі відбувається збірка механізованого пальця. Збираючи палець, необхідно переконатися, що деталі правильно орієнтовані перед склеюванням. Також треба переконатися в тому, щоб 3-міліметрові гвинти діючі як шарнірні штифти, не спричиняли тертя. Це можна зробити за допомогою гарячого клею на зовнішній стороні пальців руки. Потрібно зачекати перш ніж встановлювати струну в палець для того, щоб переконатися спершу, що сервопривод працює коректно. На даному етапі перевіряється сервопривод, який повинен бути встановлений на пальці. Щоб підключити його до джерела живлення та Arduino, слід використати маленьку безпаяну макетну плату. Підключається позитивний провід сервоприводу (червоний) до однієї рейки на макетній платі, а кабель негативного полюсу до іншої рейки. При цьому негативну шину на макетній платі потрібно підключити до Arduino через інше заземлення, при цьому усі заземлення в ланцюзі повинні бути підключені, щоб все працювало коректно.

Далі здійснюється розробка та завантаження відповідної програми на базі Arduino. При цьому потрібно переконатися, що всі підключення до рукавички та сервоприводу правильні. Потрібно одягнути рукавичку і ввімкнути Arduino. Сервопривод повинен обертатися залежно від того, наскільки зігнутий палець. На Arduino потрібно перевірити і відкалібрувати вхідні значення датчика, налаштувати діапазон у програмі так, щоб вона працювала найкраще. Якщо сервопривод не працює належним чином, перевіряємо чи всі підключення правильні.

Далі виконується додавання струни що імітує сухожилля. Потрібно використати гарячий клей для прикріплення ниток до сервоприводів. Гарячий клей краще використовується під час калібрування пальця. Щоб відкалібрувати кожне кільце сервоприводу, так щоб воно згинало та розслабляло палець, коли це потрібно на основі запиту, спочатку треба підключити Arduino та акумулятор сервоприводу та запустити програму. Далі потрібно надягнути рукавичку і зігнути палець, який відповідає сервоприводу, над яким іде робота. Слід

відрегулювати шків сервоприводу таким чином, що отвір знаходиться найближче до пальця, далі потрібно натягнути «розслаблюючу» струну цього пальця так сильно, як тільки можливо, не згинаючи палець. Після чого потрібно просунути його через найближчий отвір кільця і приклеїти на місце. Потім потрібно випрямити палець, протягнути і закріпити іншу нитку в іншому отворі. Повторити цей процес з кожним пальцем. Важливо, щоб кожна струна була натягнута. Готова установка показана на рис. 3.8.



Рисунок 3.8 – Вигляд імітатора штучного пальця та модуль керування у вигляді рукавиці з сенсором.

Алгоритм роботи програми для Arduino, яка керує імітатором штучного пальця з використанням сенсорів і сервоприводів складається з наступних основних етапів:

1) ініціалізація системи:

– оголошення пінів для підключення датчиків (flex, FSR, EMG);

- оголошення пінів для керування серводвигунами;
- ініціалізація бібліотек ('Servo.h', 'Wire.h' тощо);
- запуск серійного моніторингу для діагностики;
- 2) калібрування сенсорів:
 - зчитування початкових (нульових) значень з датчиків;
 - встановлення меж діапазону згину/сили;
- 3) основний цикл (loop):
 - зчитування даних з сенсорів:
 - отримати поточні значення з flex-сенсорів (кут згину);
 - отримати значення з датчиків сили (FSR);
 - (опціонально) зчитати EMG-сигнали;
 - обробка сигналів:
 - фільтрація шумів (середнє ковзне або експоненційне згладжування);
 - масштабування даних у діапазон, придатний для серводвигунів (0°-180°);
 - перевірка на граничні значення (захист від перевантажень);
 - формування керуючих дій:
 - обчислення кута відхилення пальця;
 - передача сигналів до відповідних сервоприводів;
 - реалізація алгоритму PID (або просто пропорційного керування) для плавності рухів;
 - зворотний зв'язок:
 - вивід даних на серійний монітор / дисплей;
 - передача даних на ПК або смартфон (через Bluetooth/Wi-Fi);
 - корекція рухів у разі перевищення сили/кута;
- 4) інтерфейс користувача (НМІ):
 - керування рухами через кнопки, потенціометри або джойстик;
 - можливість ручного вибору режиму роботи (автоматичний чи навчальний);

5) завершення роботи:

- зупинка двигунів у безпечному положенні;
- збереження даних калібрування (EEPROM, якщо потрібно).

Таким чином, Arduino-імітатор механізованого пальця може служити експериментальною моделлю для вивчення кінематики, динаміки та сенсорної взаємодії, фундаментом для розробки смарт-протезів, що інтегрують сенсори, інтелектуальне керування та механічно гнучкі структури, компонентом робототехнічних систем сервісного і реабілітаційного спрямування, здатним адаптуватися до користувача. Розроблену технологію біомеханічної системи на базі Arduino можна застосувати у медичній реабілітації для відновлення функцій кінцівок у кількох ключових напрямках, серед яких реабілітаційні екзоскелети та ортези, де Arduino може керувати серводвигунами чи лінійними актуаторами, які допомагають пацієнтові виконувати рухи (згинання, розгинання пальців, у подальшому – кисті, ліктя). Така система може працювати у двох режимах – пасивна реабілітація, коли пристрій сам рухає кінцівкою пацієнта за заданою програмою та активно-асистивна реабілітація, у якій система допомагає людині закінчити рух, який вона почала самостійно.

3.5 Обґрунтування вибору середовища програмування Arduino та розробка програмної частини установки

Розробка штучного пальця як частини біонічної кінцівки потребує використання компактної, надійної та енергоефективної системи керування, здатної взаємодіяти з датчиками та приводами в реальному часі. Одним із найбільш оптимальних рішень для цього завдання є використання платформи Arduino, що зумовлюється такими факторами:

– доступність та відкрита архітектура. Arduino є платформою з відкритим вихідним кодом, що дозволяє легко адаптувати апаратне та програмне забезпечення під специфічні вимоги розробки. Широка спільнота

користувачів забезпечує наявність бібліотек і прикладів коду для керування сервоприводами, сенсорами та інтерфейсами зворотного зв'язку;

– простота інтеграції з апаратними модулями. Arduino підтримує роботу з різними сервоприводами, датчиками сили та положення, тактильними сенсорами та інтерфейсами (Bluetooth, Wi-Fi, USB). Це дозволяє реалізувати управління рухом пальця з точністю до окремих фаланг, а також налаштувати систему для адаптивного реагування на навантаження;

– реалізація керування в реальному часі. Платформа забезпечує обробку сигналів і генерацію управляючих імпульсів у реальному часі. Це важливо для створення природної динаміки руху пальця та синхронізації з іншими елементами протезу (наприклад, кистю або передпліччям);

– гнучкість у програмуванні. Arduino IDE підтримує мову програмування C/C++, що дозволяє ефективно реалізувати алгоритми керування (наприклад, ПІД-регулятори або алгоритми оптимального керування). Є можливість підключення зовнішніх бібліотек для розширення функціоналу (робота з датчиками тиску, гіроскопами, біоелектричними сенсорами EMG тощо);

– масштабованість. Система, спроектована на Arduino, може бути легко розширена від одного пальця до цілої кисті чи навіть руки. Це забезпечує адаптивність рішення та створює основу для розробки комплексної біонічної кінцівки.

3.6 Розробка програмної частини установки

Програмна частина установки передбачає створення алгоритмів керування, що забезпечують обробку вхідних сигналів, зчитування даних із датчиків положення, сили або м'язової активності (EMG-сенсорів), виконання попередньої фільтрації сигналів для зменшення шумів.

Також забезпечується формування управляючих дій на базі використання алгоритмів ПІД-регулятора або оптимального LQR-контролю

після попереднього моделювання у MATLAB/Simulink, здійснюється розрахунок керуючих імпульсів для сервоприводів, що відповідають за згинання/розгинання фаланг пальця.

Програмний алгоритм здійснює реалізацію зворотного зв'язку через безперервне оновлення положення пальця відповідно до даних датчиків та адаптацію рухів до зовнішніх навантажень (наприклад, при захопленні предмета). Окремим компонентом є інтерфейс користувача, при цьому програмна частина може бути розширена модулем керування з планшета чи ПК, в тому числі тут передбачена можливість віддаленого керування через Bluetooth або Wi-Fi.

Використання Arduino як програмної платформи для керування смарт-пальцем є доцільним завдяки її простоті, гнучкості, підтримці роботи з сенсорами та актуаторами, можливостям масштабування та доступності для інтеграції з іншими системами. Програмна частина установки забезпечує реалізацію адаптивного керування рухом пальця з урахуванням зворотного зв'язку, що робить платформу ефективною основою для подальшої розробки біонічних кінцівок.

3.7 Обґрунтування вибору середовища MATLAB і пакету Control System Toolbox

Розробка штучного пальця людини як елемента біонічної кінцівки вимагає поєднання точного математичного моделювання, синтезу систем керування та перевірки алгоритмів в умовах, наближених до реальних. Для цього середовище MATLAB з пакетом Control System Toolbox є одним із найбільш придатних інструментів, що зумовлюється такими чинниками:

– універсальність у побудові динамічних моделей. Control System Toolbox дозволяє описувати динаміку руху штучного пальця у вигляді системи диференціальних рівнянь, передавальних функцій або простору станів. Це спрощує моделювання біомеханічної структури пальця, де

враховуються параметри фаланг, пружні властивості шарнірів та динаміка приводів;

– інструменти оптимального керування. MATLAB має вбудовані функції для синтезу LQR-регуляторів, які часто застосовуються для систем із багатьма ступенями свободи. Це дозволяє проектувати керування, яке мінімізує відхилення від заданої траєкторії при зменшенні енергоспоживання приводів, що є критично важливим для протезів;

– аналіз стійкості та якості системи. Control System Toolbox надає широкий спектр методів для аналізу стійкості. Це дає змогу оцінити надійність роботи смарт-пальця в умовах різних навантажень та непередбачуваних зовнішніх впливів;

– можливості симуляції та візуалізації. MATLAB підтримує інтеграцію з Simulink, що дозволяє створювати візуальні блок-схеми роботи системи. Це полегшує імітацію рухів пальця, а також взаємодію сенсорів, актуаторів та системи керування у єдиному середовищі;

– гнучкість і масштабованість. Розроблені моделі можна адаптувати до більш складних систем – наприклад, від одного пальця до цілої кисті або руки. MATLAB дозволяє інтегруватися з реальними мікроконтролерами та апаратним забезпеченням через спеціальні пакети, зокрема, Arduino. MATLAB і Control System Toolbox широко використовуються у біомедичній інженерії, робототехніці та системах керування. Це забезпечує відтворюваність результатів, можливість порівняння з іншими дослідженнями та швидкий перехід від теоретичних моделей до практичної реалізації. Вибір середовища MATLAB з Control System Toolbox для проектування смарт-пальця є обґрунтованим, оскільки воно поєднує засоби математичного моделювання, оптимального керування, візуалізації та інтеграції з реальними прототипами. Це робить його ключовим інструментом у процесі створення адаптивних і надійних біонічних протезів.

3.8 Управління пристроями і реалізація програмно-функціонального зв'язку у середовищі MATLAB

MATLAB являє собою універсальне програмне забезпечення, яке можна використовувати для ряду різноманітних додатків. В даному проєкті реалізуємо управління сервоприводом за допомогою MATLAB та Arduino, на основі цих платформ (рис. 3.9).

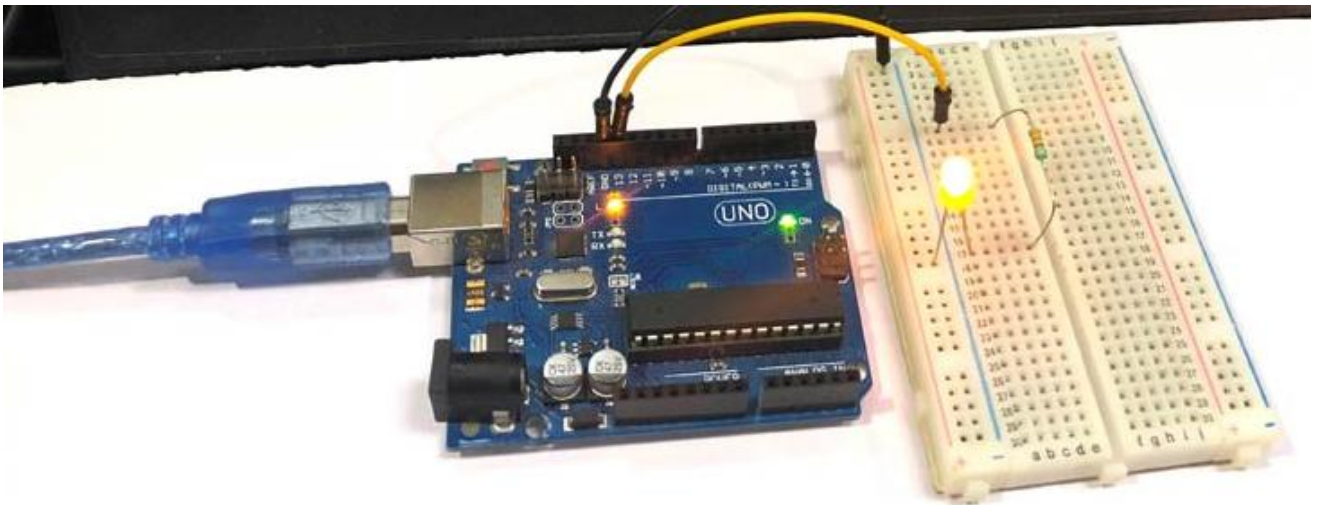


Рисунок 3.9 – Сценарій реабілітації з Arduino-реабілітаційним пальцем

Існує два основних способи використання послідовного зв'язку в MATLAB – за допомогою вікна команд для керування сервомотором за допомогою MATLAB, та за допомогою MATLAB GUI (Graphical User Interface) – графічний інтерфейс користувача. Код програми для плати Arduino в обох випадках буде однаковим.

Необхідні компоненти для цього це Плата Arduino Uno та математична система MATLAB. Схема проєкту для послідовного зв'язку між MATLAB та Arduino представлена на рисунку 3.10.

Зовнішній вигляд зібраної на макетній платі конструкції, що демонструє функціональний зв'язок показано на рисунку 3.10.

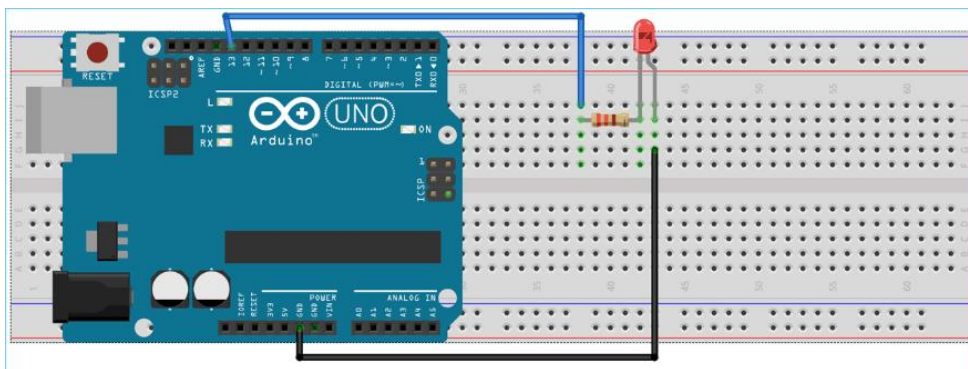


Рисунок 3.10 – Схема проекту для послідовного зв'язку між MATLAB та Arduino

В даному випадку реалізується один зі способів встановлення послідовного зв'язку між MATLAB та Arduino. У ньому передаються дані з MATLAB в Arduino, послідовно використовуючи вікно команд (command window), а потім плата Arduino зчитує послідовні дані, що надходять. Після цього ці послідовно передані дані можна використовувати для управління будь-яким пристроєм, підключеним до Arduino в залежності від послідовних даних, що надходять.

Завантажимо в плату Arduino код програми, і реалізуємо програмування у вікні редагування (Editor Window) у MATLAB. Щоб відкрити вікно для нового скрипту, натискаємо 'New Script'.

Після цього вставимо у вікно редагування наступний код програм, який служить для здійснення послідовного зв'язку між MATLAB та Arduino (ліст. 3.1).

Лістинг 3.1 – здійснення послідовного зв'язку між MATLAB та Arduino

```
%MATLAB Code for Serial Communication between Arduino and MATLAB
x=serial('COM18','BAUD', 9600);
fopen(x);
go = true;
while go
a= input('Press 1 to turn ON LED & 0 to turn OFF:');
fprintf(x,a);
if (a == 2) go=false;
end
```

кінець лістингу 3.1

У поданому кодї програми команда `x=serial('COM18','BAUD', 9600)` використовується для встановлення послїдовного зв'язку у MATLAB.

Переконаємось, що в нїй записано саме номер COM порту комп'ютера, до якого підключена плата Arduino. Швидкїсть передачі необхідно встановити таку ж, як і в платї Arduino.

Щоб відкрити послїдовний код, використовується команда `foren(x)`. Наступна команда використовується для передачі даних від MATLAB до плати Arduino послїдовно, де x – це виклик послїдовного порту, а a – це значення, яке вводить користувач – `fprintf(x,a)`.

Функція `while` використовується для створення нескїнченного циклу, вихїд з нього буде здїйснюватися, якщо користувач введе цифру '2' (лїст. 3.2).

Лїстинг 3.2 – Використання функція `while` для створення нескїнченного циклу

```
while go
a= input('Press 1 to turn ON LED & 0 to turn OFF:');
fprintf(x,a);
if (a == 2)
go = false;
end
```

кїнець лїстингу 3.2

Пїсля запису цього скрипту, натискаємо кнопку 'RUN', щоб запустити його на виконання.

3.9 Програмна реалїзація управлїння пальцем на базї Arduino через середовище MATLAB

В даному параграфї показано інтеграцію Arduino з MATLAB/Simulink для керування штучним смарт-пальцем.

Програмна реалїзація управлїння пальцем на базї Arduino через середовище MATLAB базується на наступному: для реалїзації керування штучним пальцем у данїй роботї застосовується гїбридний пїдхїд, що поєднує

середовище MATLAB/Simulink для математичного моделювання та синтезу алгоритмів керування з платформою Arduino для їхньої фізичної реалізації. Така інтеграція дозволяє розробити оптимальні закони керування у MATLAB, протестувати їх у симуляційному режимі на основі динамічної моделі пальця, експортувати керуючі сигнали на Arduino для безпосереднього управління сервоприводами. Цей підхід забезпечує швидкий перехід від теоретичного моделювання до практичної реалізації.

Програмна реалізація включає кілька етапів. Перший це моделювання динаміки пальця у MATLAB/Simulink, що включає створення математичної моделі на основі системи рівнянь руху, визначення параметрів сервоприводів, обмежень кутів згину, симуляція різних сценаріїв руху (захоплення, згинання, утримання об'єкта). Наступний етап – це синтез алгоритму керування з використанням Control System Toolbox для побудови LQR-регулятора або ПІД-контролера, налаштуванням критеріїв оптимізації (мінімізація енергоспоживання, швидкодія, плавність рухів), тестування алгоритму в MATLAB.

Наступним етапом є передача сигналів на Arduino і використання MATLAB Support Package for Arduino Hardware, що дозволяє керувати Arduino з MATLAB, формування сигналів PWM для сервоприводів пальця, обробка зворотного зв'язку від датчиків (положення, сили).

Після цього реалізується програмна частина на Arduino, прошивка контролера базовим кодом для отримання та виконання команд від MATLAB, реалізація циклу читання сенсорних сигналів та формування управляючих дій, забезпечення коректної взаємодії з приводами в режимі реального часу.

Наступним кроком є тестування та верифікація. Для перевірки роботи системи проводиться Hardware-in-the-Loop (HIL) симуляція, коли математична модель у MATLAB працює паралельно з реальною установкою на Arduino. Це дає змогу: відстежувати відхилення між теоретичною моделлю та фізичною системою; налаштовувати параметри регулятора в реальному

часі; оптимізувати рухи пальця під індивідуальні задачі (наприклад, реабілітацію чи захоплення об'єктів).

Реалізація управління смарт-пальцем через інтеграцію MATLAB/Simulink та Arduino дозволяє поєднати сильні сторони обох середовищ. MATLAB забезпечує математичну точність і зручні інструменти моделювання, а Arduino надає можливість апаратної реалізації в реальному часі. Такий підхід створює основу для масштабування від одного пальця до всієї штучної кисті чи руки та забезпечує високу гнучкість і адаптивність системи (рис. 3.11).

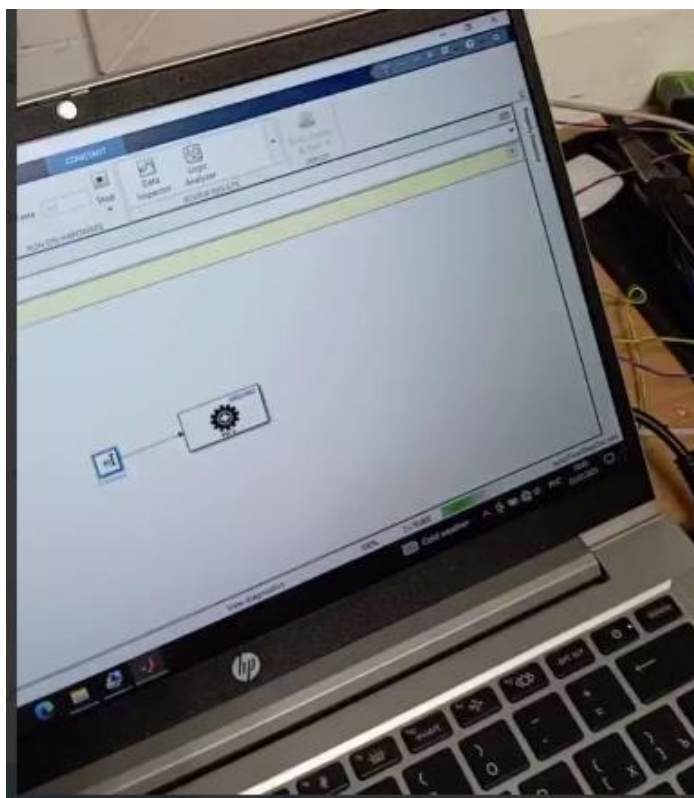


Рисунок 3.11 – Панель комп'ютерного керування

Схема програмної реалізації керування смарт-пальцем виконується через інтеграцію MATLAB з Arduino. Вона показує, як відбувається цикл: моделювання, синтез керування, передача на Arduino, керування сервоприводами, зчитування сенсорів, зворотний зв'язок до MATLAB.

Здійснимо налаштування Arduino для керування сервоприводом смарт-пальця і зчитування даних із сенсора:

– підключення Arduino до MATLAB здійснюється наступним чином:
% створення об'єкта Arduino, а = arduino('COM3', 'Uno'); COM-порт може відрізнятись (див. у диспетчері пристроїв). Якщо MATLAB успішно визначив плату, команда створить об'єкт 'а', який дозволить взаємодіяти з усіма пінами Arduino.

– керування сервоприводом пальця. Припустимо, що сервопривід підключений до цифрового піну D9 (ліст. 3.3).

Лістинг 3.3 – Керування сервоприводом пальця

```
% Ініціалізація сервопривода
s = servo(a, 'D9');
% Встановлення положення пальця
writePosition(s, 0); % мінімальний кут (повністю розігнутий палець)
pause(1);
writePosition(s, 0.5); % середнє положення
pause(1);
writePosition(s, 1); % максимальний кут (згинання)
```

кінець лістингу 3.3

Значення '0..1' відповідає діапазону кута сервопривода (залежно від моделі, це може бути приблизно 0-180°).

– зчитування даних із сенсора положення;

Як приклад сенсора використаємо ****потенціометр****, підключений до аналогового входу A0 (ліст. 3.4).

Лістинг 3.4 – Приклад сенсора

```
% Зчитування напруги з потенціометра
val = readVoltage(a, 'A0');
% Масштабування у кут (0..180 градусів)
angle = val * (180/5);
disp(['Поточний кут пальця = ', num2str(angle), ']);
writePosition(s, 1);
```

кінець лістингу 3.4

– реалізація зворотного зв'язку. Алгоритм має наступний вигляд. Необхідно задати бажаний кут, далі зчитати фактичний, після цього відкоригувати (лістинг 3.5).

Лістинг 3.5 – Алгоритм реалізації зворотного зв'язку

```
% Бажаний кут (наприклад, 90 градусів)
desired_angle = 90;
% Зчитуємо сенсор
val = readVoltage(a, 'A0');
current_angle = val * (180/5);
% Обчислюємо похибку
error = desired_angle - current_angle;
% Пропорційне регулювання (спрощений варіант контролю)
Kp = 0.005;
control_signal = current_angle/180; + Kp*error;
% Обмежуємо сигнал у діапазоні 0..1
control_signal = max(0, min(1, control_signal));
% Відправляємо сигнал на сервопривід
writePosition(s, control_signal);
☒ автоматизація процесу (цикл керування).
for i = 1:50
    desired_angle = 90; % бажане положення
        val = readVoltage(a, 'A0');           % зчитування сенсора
        current_angle = val * (180/5);       % перетворення у градуси
        error = desired_angle - current_angle;
        Kp = 0.005;                          % коефіцієнт регулятора
        control_signal = current_angle/180 + Kp*error;
        control_signal = max(0, min(1, control_signal));
        writePosition(s, control_signal);    % керування приводом
        pause(0.1); % затримка між ітераціями
end
```

кінець лістингу 3.5

За допомогою MATLAB Support Package for Arduino Hardware реалізується повний цикл: ініціалізація Arduino, керування сервоприводом пальця, зчитування даних із сенсора, формування простого регулятора на базі MATLAB. Цей підхід дозволяє легко інтегрувати математичне моделювання, контроль у реальному часі та експериментальні дослідження у єдиній платформі.

Блок-схема алгоритму роботи ардуїно для пальця зображена на рисунку 3.12.

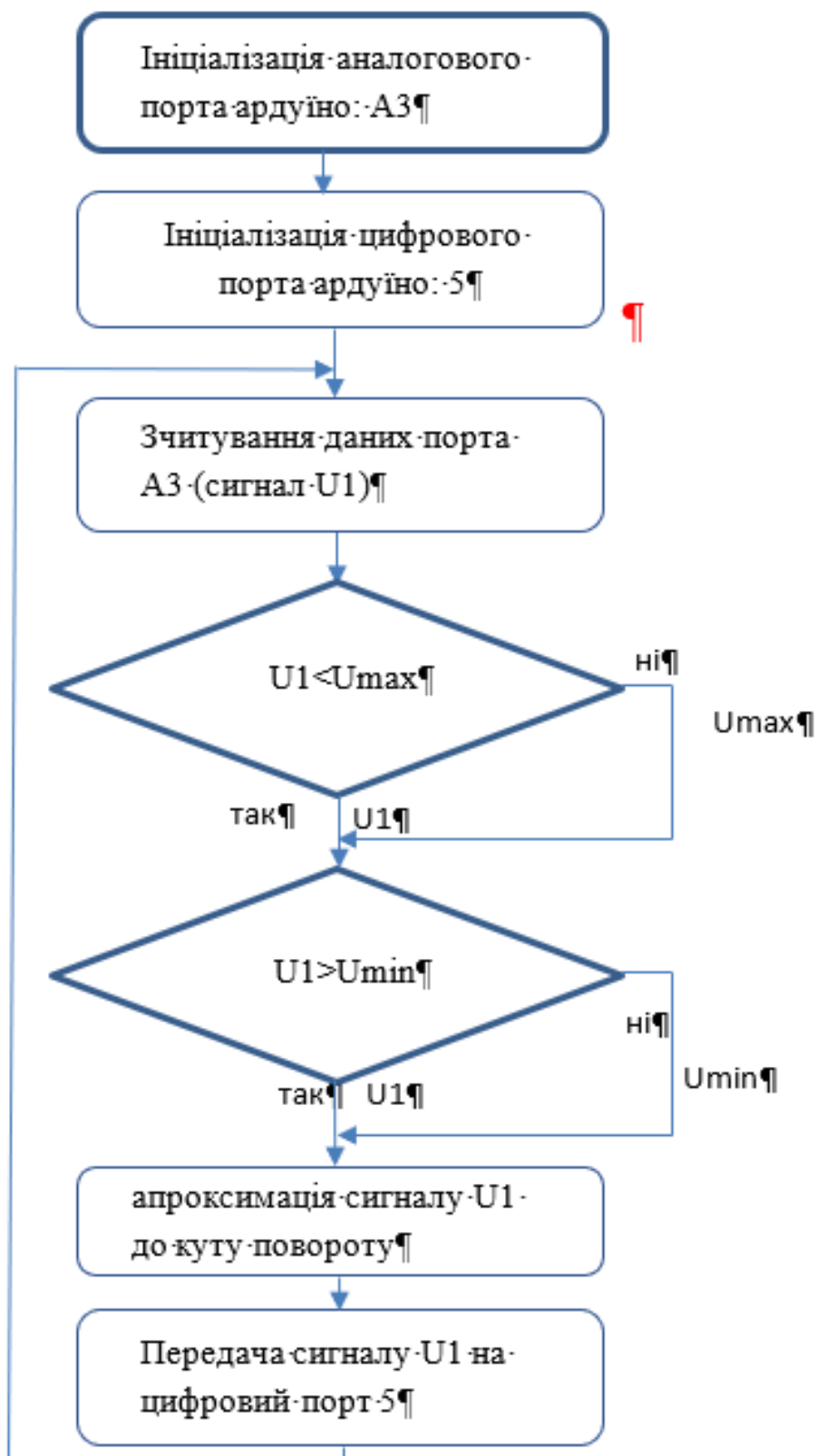


Рисунок 3.12 – Блок-схема алгоритму роботи ардуїно для пальця

Блок-схема алгоритму роботи ардуїно для руки (5 пальців) зображена на рисунку 3.13.

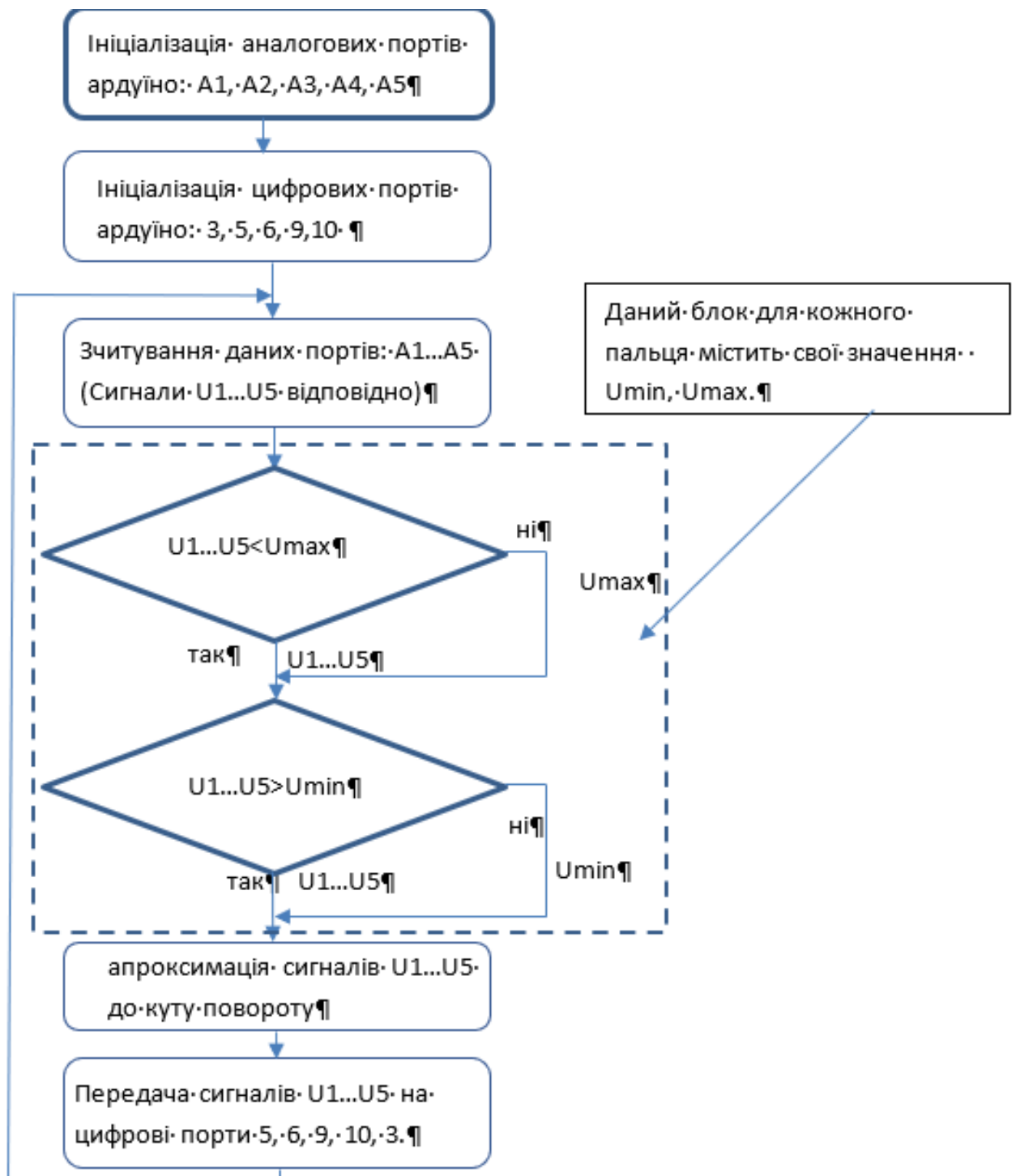


Рисунок 3.13 – Блок-схема алгоритму роботи ардуїно для руки

Після монтажу сервомотора, визначено діапазон кутів повороту пальця. Дані результати використані при програмуванні загального алгоритму.

При роботі виявлено, що датчики згину можуть працювати у різних діапазонах. Тому було застосовано така схема:

- датчик було протестовано на діапазон показів, в яких він працює;
- при написанню загального скетча враховано діапазон показів.

Опис алгоритму роботи пальця:

- з датчика згину на аналоговий порт ардуїно попадає сигнал;
- даємо короткочасну затримку близько 5-15 мс, щоб ардуїно обробив сигнал, що поступив, так як одночасно має поступати 5 сигналів(з кожного пальця);
- даний сигнал обробляється функцією, яка підрізає мінімальні і максмальні значення, що поступають від пальця;
- спеціальною функцією робимо апроксимацію сигналу із датчика до куту повороту серво мотора;
- передаємо отримані дані на сервомотор.

3.10 Дослідження способів практичного застосування установки у реабілітації

3.10.1 Перспективи біонічного протезування

Розглянемо приклад сценарію практичного використання Arduino-системи людського смарт-пальця для реабілітації пацієнта з порушенням функції кисті після інсульту.

Спочатку дослідимо перспективи застосування демонстраційної установки – імітатора механізованого пальця робота-гуманоїда на базі платформи Arduino. Розроблена система побудована за принципами біоміметичного моделювання, забезпечуючи відтворення ключових характеристик кінематики й динаміки природного руху людської кінцівки, яка може мати застосування у відновлення дрібної моторики для людей після інсульту чи травм, яким особливо важливо

тренувати згинання пальців і хватку. Імітатор роботизованого людського пальця на базі Arduino має потенційне використання як прототип тренажера хватальних рухів, у якій датчики допомагають контролювати силу стиснення, що знижує ризик травм і поступово збільшує навантаження. Це є модельний прототип штучної кінцівки із функціональним смарт-керуванням, який поєднує сенсори зворотного зв'язку, мікроконтролер на базі Arduino та інтелектуальний алгоритми керування, що дозволить вирішити значну кількість проблем у сфері медичної реабілітації.

Такі системи можуть адаптуватися до індивідуальних особливостей пацієнта, забезпечувати навчання рухів і навіть відновлювати втрачені функції за допомогою інтерактивних протоколів реабілітації. Основні складові даної розробки включають функціональну схему автоматизації, що описує потоки сигналів і етапи керування, інтеграцію сенсорних і виконавчих компонентів, включно з датчиками сили чи гнучкості, електричну схему під'єднання сенсорів і приводів до контролера. Розроблено також програмне забезпечення для інтелектуального модуля керування, яке реалізує алгоритми реакції на сенсорні дані та показано приклад, що підтверджує ефективність і доцільність використання даної розробки для цілей медичної реабілітації.

Таким чином, сценарій реабілітації з Arduino-імітаційним пальцем має наступний вигляд:

– підготовчий етап. Мета: визначити рівень моторики та силу м'язів пацієнта. Дії: лікар або реабілітолог підключає на руку пацієнта flex-сенсори та датчики сили (FSR); Arduino зчитує початкові показники згину пальців та силу стиснення; встановлюються індивідуальні межі руху для безпечної тренування. Результатом є створення профілю пацієнта та калібрування пристрою;

– пасивна реабілітація. Мета: розробка та розслаблення м'язів, відновлення рухового стереотипу. Дії: Arduino керує сервоприводами пальця, виконуючи згинання та розгинання; пацієнт пасивно спостерігає рухи, повторюючи їх за можливістю. Зворотний зв'язок виявляється у тому, що дані про кут згину та силу стиснення відображаються на дисплеї або планшеті;

– активно-асистивна реабілітація. Мета: навчити м'язи самостійно виконувати рухи, підтримка системою при слабких сигналах. Дії: пацієнт пробує самостійно зігнути або розігнути палець; Arduino аналізує EMG-сигнали або зчитує легкий рух з flex-сенсорів; якщо пацієнт не може завершити рух, система підключає сервопривод для допомоги. Результатом є поступове підвищення сили та контролю м'язів;

– біологічний зворотний зв'язок (Biofeedback). Мета: стимулювати нейропластичність та мотивацію. Дії: Arduino передає дані на ПК або смартфон; пацієнт бачить графіки кута згину та сили стиснення в реальному часі; програма може інтегруватися у просту гру, де кожне успішне згинання пальця – це прогрес у грі. Результатом є покращення мотивації та точності рухів;

– інтерактивна практика. Мета: відновлення дрібної моторики через ігрову та інтерактивну діяльність. Дії: пацієнт бере участь у вправі «захоплення та переміщення об'єктів» за допомогою смарт-пальця; Arduino контролює положення пальця та силу хватки. Результатом є розвиток координації, сили та швидкості реакції;

– моніторинг і корекція. Мета: оцінка прогресу та корекція програми. Дії: лікар аналізує історію рухів, сили та кількість повторів; змінює режим роботи: від пасивного до активно-асистивного, у подальшому – до активного тренування. Результатом є індивідуальна програма реабілітації для максимального відновлення функцій кисті.

Розроблений сценарій (рис. 3.14) має такі переваги як доступність і низька вартість завдяки застосуванню смарт-технологій на базі Arduino, можливість поступової адаптації рівня допомоги пацієнту, застосування біосигналів для відновлення природних рухів людини, інтерактивність та мотивація пацієнта через biofeedback та ігрові вправи. Блок-схема цього сценарію має вигляд показаний на рисунку 3.14.

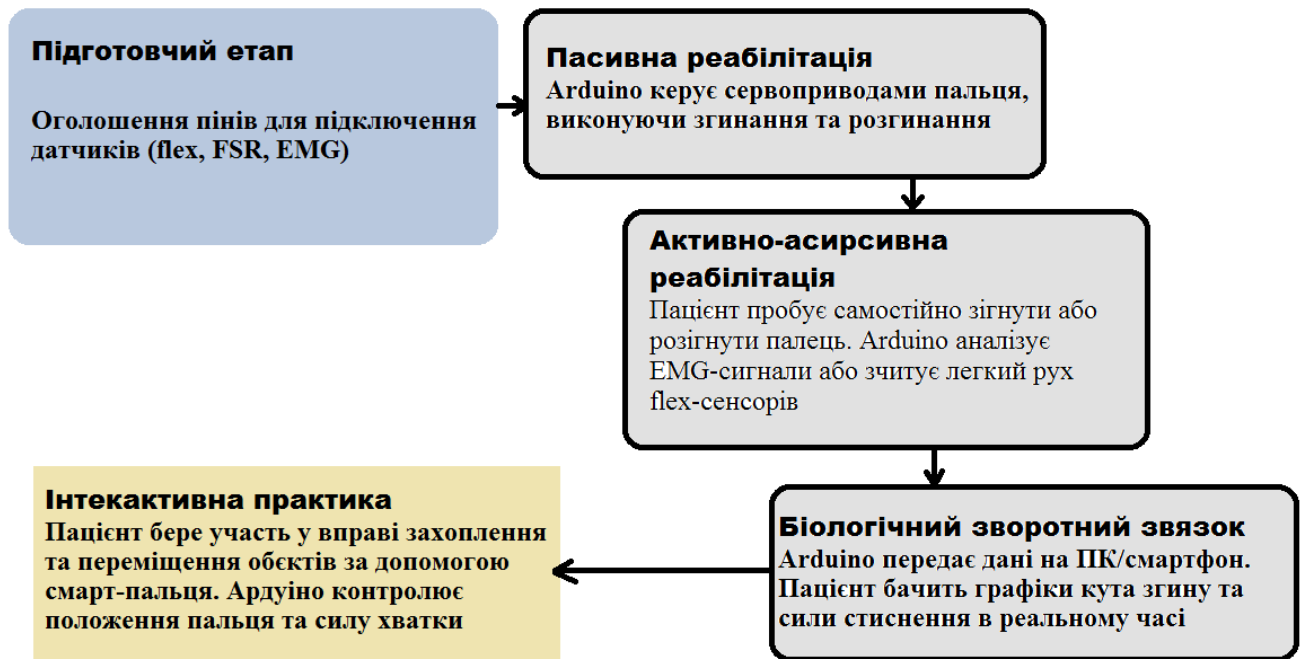


Рисунок 3.14 – Сценарій реабілітації з Arduino-реабілітаційним пальцем

Таким чином, розробка демонстраційного модуля штучного пальця на базі Arduino не лише надає зручну платформу для навчання і досліджень, але й закладає технологічну базу для майбутніх інновацій у галузі біомеханіки, смарт-протезування і реабілітаційної робототехніки.

3.10.2 Імітаційне експериментальне дослідження

У дослідженні (рис. 3.15) задіяно трьох учасників із легкими та помірними проявами тунельного синдрому, що сформувався внаслідок тривалої роботи з комп'ютерною мишею. Метою експерименту було оцінити, як короткий курс щоденних вправ зі смарт-пальцем впливає на динаміку больових відчуттів упродовж одного тижня. Смарт-палець виконував роль керованого реабілітаційного інструмента, який забезпечував стандартизовані рухи та контрольоване силове навантаження згідно з індивідуальними параметрами користувача.

Кожен учасник протягом семи днів виконував однаковий комплекс вправ: повільне розгинання, ритмічні опори на сенсорну подушку та пасивно-активну мобілізацію середнього пальця. Після завершення щоденного сеансу учасники самостійно оцінювали рівень болю за візуально-аналоговою шкалою

(0-10 балів). На основі цих щоденних суб'єктивних оцінок було побудовано індивідуальні криві зміни больових відчуттів.

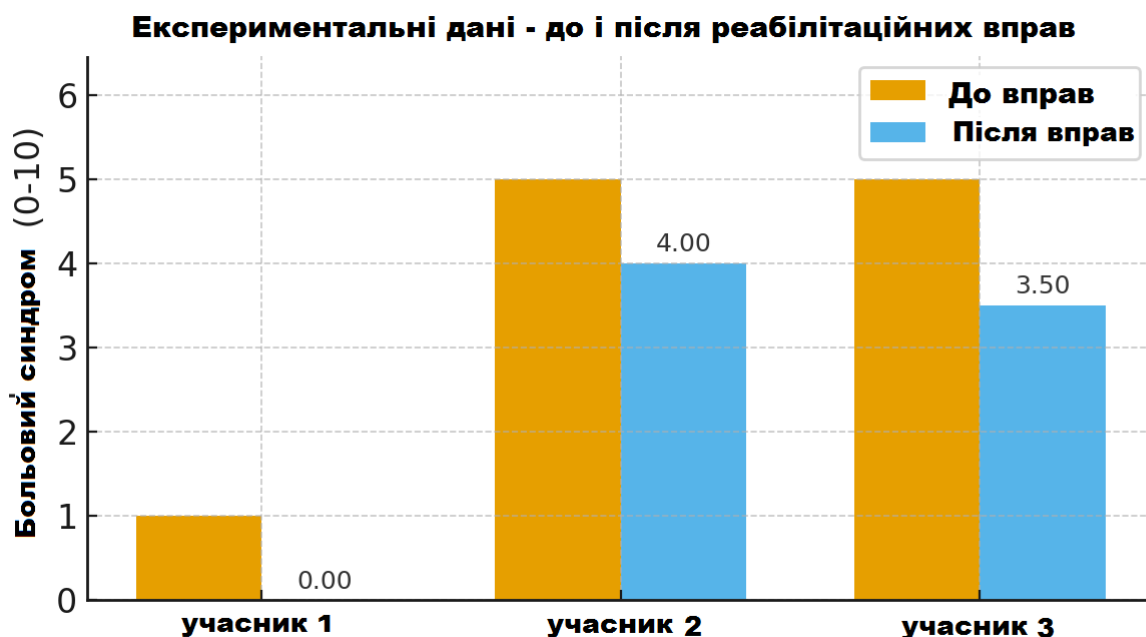


Рисунок 3.15 – Модельні експерименти

Отримані дані демонструють різну динаміку адаптації у трьох учасників. У першого учасника спостерігалось помірне й поступове зниження больових відчуттів (близько 10 % від початкового рівня). У двох інших учасників виявлено більш виражений терапевтичний ефект: зменшення болю становило орієнтовно 20-30 % протягом тижня. Усі три криві характеризувалися типовим трендом спадання із незначними коливаннями, що узгоджується з очікуваними процесами нейром'язової адаптації під час відновлення функції сухожильного каналу. Таким чином, попередні результати свідчать, що навіть короткий курс стандартизованих тренувальних вправ зі смарт-пальцем може сприяти зниженню больового синдрому та помірному покращенню функціонального стану кисті. Хоча ефекти варіюють між учасниками, сама методика показує потенційність для подальшого вдосконалення та масштабування в рамках реабілітаційних програм. Зауважимо, що проведені експерименти це – імітація потенційних можливостей цього смарт-пальця, оскільки це є установка, протитип, а не

повноцінний пристрій. Тому наведені вище експерименти є імітаційними та покликані продемонструвати потенційні можливості смарт-пальця, оскільки на цьому етапі йдеться лише про прототип установчого рівня.

Отже, представлені результати експериментів мають імітаційний характер і відображають потенційні можливості запропонованої системи, і при подальшому вдосконаленні можуть спрогнозувати ефективність повністю сформованого медичного пристрою. На момент проведення дослідження смарт-палець є на стадії прототипу, який включав базовий модуль сенсорики та виконавчої механіки, однак не містить повного спектра функціональних елементів, які характерні для серійних реабілітаційних систем – зокрема, відсутні сертифіковані медичні датчики, адаптивні режими керування та засоби повноцінного біомеханічного калібрування.

З огляду на це, експерименти проведено як модельні дослідження, спрямовані на виявлення можливих тенденцій, які могли б виникати під час реабілітації із використанням майбутньої розвинутої версії пристрою. Зміни рівня больових відчуттів оцінювалися за суб'єктивною шкалою, що дозволяє дослідити загальну динаміку, але не забезпечує точних клінічних вимірювань. Крім того, комплекс вправ був стандартизований лише на базовому рівні, оскільки прототип ще не підтримує інтелектуальні режими адаптації навантаження, які плануються відповідно до кінцевої концепції.

Таким чином, отримані криві зниження болю відображають моделювання очікуваного ефекту, який може бути притаманний більш технологічно розвинутому пристрою. Метою цих імітаційних досліджень було не доведення клінічної ефективності, а підтвердження доцільності обраної концепції та демонстрація того, що навіть базовий прототип здатен забезпечити мінімальний терапевтичний вплив. Ці результати створюють підґрунтя для подальших інженерних вдосконалень, розробки точних алгоритмів керування та проведення повноцінних клінічних випробувань у майбутньому.

ВИСНОВКИ

За результатами виконання кваліфікаційної роботи можна зробити відповідні висновки: у цій роботі розроблено і досліджено демонстраційну установку – імітатор механізованого пальця робота-гуманоїда на базі платформи Arduino.

Система побудована за принципами біоміметичного моделювання, забезпечуючи відтворення ключових характеристик кінематики й динаміки природного руху людської кінцівки, яка може мати застосування у відновлення дрібної моторики для людей після інсульту чи травм, яким особливо важливо тренувати згинання пальців і хватку. Імітатор роботизованого людського пальця на базі Arduino має потенційне використання як прототип тренажера хватальних рухів, у якій датчики допомагають контролювати силу стиснення, що знижує ризик травм і поступово збільшує навантаження. Це є модельний прототип штучної кінцівки із функціональним смарт-керуванням, який поєднує сенсори зворотного зв'язку, мікроконтролер на базі Arduino та інтелектуальний алгоритми керування, що дозволить вирішити значну кількість проблем у сфері медичної реабілітації. Такі системи можуть адаптуватися до індивідуальних особливостей пацієнта, забезпечувати навчання рухів і навіть відновлювати втрачені функції за допомогою інтерактивних протоколів реабілітації.

Основні складові розробки включають функціональну схему автоматизації, що описує потоки сигналів і етапи керування, інтеграцію сенсорних і виконавчих компонентів, включно з датчиками сили чи гнучкості, електричну схему під'єднання сенсорів і приводів до контролера. Розроблено також програмне забезпечення для інтелектуального модуля керування, яке реалізує алгоритми реакції на сенсорні дані та показано приклад, що підтверджує ефективність і доцільність використання даної розробки для цілей медичної реабілітації.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мороз Б. С., Багнюк Н. В. Модель роботизованого штучного пальця руки людини з перспективами біонічного протезування. *Наука, освіта і суспільство в умовах змін: виклики та інновації* : зб. тез доп. міжнар. наук.-практ. конф., м. Кременчук, 10 жовт. 2025 р. Кременчук, 2025. С. 232-235.
2. Вплив біомеханіки на оптимізацію дизайну та функціональності протезів : Новий Колегіум / Ю.Ю.Худецький та ін. 3-тє вид. Київ, 2024. 115 с.
3. Кафедра фізіології та функціональної біомеханіки: вплив біомеханіки на оптимізацію дизайну та функціональності протезів / І. Ю. Шликовський та ін. Київ, 2024. 253 с.
4. EksoNR. URL: <https://eksobionics.com/eksonr/> (дата звернення: 06.07.2025).
5. 3D reconstruction of human bodies from single-view and multi-view images. URL: <https://www.sciencedirect.com/science/article/pii/S0169260723002857> (дата звернення: 06.07.2025).
6. X-one-se. URL: <https://www.youjoyhealth.com/uk/-x-one-se-> (дата звернення 12.07.2025).
7. МРТ - магнітно-резонансна томографія. URL: <https://ito.gov.ua/poslugi/viddil-diagnostiki/mrt.html> (дата звернення 06.10.2025).
8. A Survey on Design and Control of Lower Extremity Exoskeletons. URL: https://www.dfki.de/fileadmin/user_upload/import/12297_20220324_applsci_12_02395.pdf (дата звернення: 17.07.2025).
9. Medical Imaging-Based Kinematic Modeling for Biomimetic Finger Joints and Hand Exoskeleton Validation. URL: <https://www.mdpi.com/2313-7673/10/10/652> (дата звернення: 22.07.2025).
10. Personalized biomechanical modeling of the human limb using motion capture and medical imaging. URL: <https://doi.org/10.1016/j.compbiomed.2023.106568> (дата звернення: 06.10.2025).

11. Geomagic Freeform. URL: <https://www.artec3d.com/3d-software/geomagic-freeform> (дата звернення 25.07.2025).
12. Deep learning approach to control of prosthetic hands with electromyography signals. URL: <https://arxiv.org/abs/1909.09910> (дата звернення 03.08.2025).
13. Concept of a shadow hand. URL: <https://shadowrobot.com/shadow-hand-glove/> (дата звернення: 08.08.2025).
14. Роботизований біонічний протез. URL: <https://braveinventors.com/invention/robotyzovanyj-bionichnyj-protezh/> (дата звернення 08.08.2025).
15. EMG-based neural network control of a robotic prosthetic forearm. URL: <https://doi.org/10.3389/fnbot.2023.112233> (дата звернення: 12.08.2025).
16. Variable-stiffness prosthesis improves biomechanics of walking across speeds compared to a passive device. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11255255/> (дата звернення 12.08.2025).
17. Dynamic Analysis of a High-Performance Prosthetic Leg: Experimental Characterisation and Numerical Modelling. URL: <https://www.mdpi.com/2076-3417/13/20/11566> (дата звернення 17.08.2025).
18. A reduced-order closed-loop hybrid dynamic model for design and development of lower limb prostheses. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10936358/> (дата звернення 17.08.2025).
19. Using embedded prosthesis sensors for clinical gait analyses in people with lower limb amputation: a feasibility study. URL: <https://pubmed.ncbi.nlm.nih.gov/37230008/> (дата звернення 17.08.2025).
20. Jyothish, K. J., & Mishra, S. A Survey on Robotic Prosthetics: Neuroprosthetics, Soft Actuators, and Control Strategies. Publication History. 2024. 44 p.
21. Будова кисті. URL: <https://orto-med.com.ua/articles/articulat/> (дата звернення 17.08.2025).

22. Central Compartment Group. URL: <https://learnmuscles.com/glossary/central-compartment-group/> (дата звернення 28.08.2025).

23. A 3D-Printed Soft Fingertip Sensor for Providing Information about Normal and Shear Components of Interaction Forces. URL: <https://www.mdpi.com/1424-8220/21/13/4271> (дата звернення 28.08.2025).

24. Identification and validation of the dynamic model of a tendon-driven anthropomorphic finger. URL: <https://arxiv.org/abs/2408.13044> (дата звернення 28.08.2025).

25. Електронейроміографія. URL: <https://awatage.com/ua/service/elektronejromiografiya/> (дата звернення 28.08.2025).

26. MATLAB Graphics and Visualization. URL: <https://www.mathworks.com/products/matlab.html> (дата звернення: 10.10.2025).

27. App Designer User's Guide. URL: <https://www.mathworks.com/help/matlab/app-designer.html> (дата звернення: 28.08.2025).

28. Simulink Model-Based Design. URL: <https://www.mathworks.com/products/simulink.html> (дата звернення: 10.09.2025).

29. Parallel Computing Toolbox Documentation. URL: <https://www.mathworks.com/help/distcomp/> (дата звернення: 10.09.2025).

30. MATLAB Web App Server. URL: <https://www.mathworks.com/products/web-app-server.html> (дата звернення: 10.09.2025).

31. Mechanisms and component design of prosthetic knees: a biomechanical review. *Frontiers in Bioengineering and Biotechnology* / Liang, W. et. al. *Frontiers in Bioengineering and Biotechnology*, 2022. 232p.

32. Arduino Board. URL: <https://iot-guider.com/iot/what-is-arduino-introduction/> (дата звернення 10.09.2025).

33. Arduino UNO. URL: <https://itmaster.biz.ua/directory/kits-nabory/arduino-uno.html> (дата звернення 13.10.2025).

34. Arduino UNO Pinout. URL: <https://de.scribd.com/document/129681857/Arduino-Uno-Pinout> (дата звернення 13.10.2025).

35. NanitLib. URL: <https://nanitrobot.github.io/NanitLib/Ukrainian/> (дата звернення 13.10.2025).