

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
Кафедра Автоматизації проектування обчислювальної техніки
Рівень вищої освіти другий (магістерський)
Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Спеціалізовані комп'ютерні системи
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

“ _____ ” _____ 20__ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

здобувачеві Сліпенькому Володимирі Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Метод машинного навчання для торкання
рукою точок у просторі

затверджена наказом по університету від " 08 " 11 2024 р. № 1189 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії 21.01.2024

3. Вихідні дані до роботи

Операційна система розробки – Windows

Середовище розробки – Spyder

Розробка моделі торкання точок у просторі з використанням ІК та РСТ

Мова програмування - Python

4. Перелік питань, що потрібно опрацювати в роботі _____

1 Аналіз предметної області та постановка завдання

2 Аналіз сучасних концепцій побудови машинного руху

3 Аналіз створення рухових моделей кінцівки в освітній галузі

4. Розробка програмної реалізації проекту

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____ 18 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	Дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	01.09.2024 - 02.09.2024	
2	Аналіз поставленої задачі	03.09.2024 - 10.09.2024	
3	Дослідження існуючих методів вирішення поставленої задачі	11.09.2024 - 18.09.2024	
4	Дослідження методів заснованих на нейронних мережах	19.09.2024 - 10.10.2024	
5	Виявлення проблем і недоліків	11.11.2024 – 25.11.2024	
6	Дослідження можливих рішень виявлених проблем	26.11.2024 - 03.12.2024	
7	Оформлення пояснювальної записки	04.12.2024 - 18.12.2024	
8	Оформлення графічного матеріалу	19.12.2024 - 25.12.2024	
9	Перевірка виконаного проекту керівником	26.12.2024 -05.01.2025	
10	Захист роботи	24.01.2025	

Дата видачі завдання _____ 01 вересня 2024 р. _____

Здобувач _____
(підпис)

Керівник роботи _____
(підпис)

доц. Кулак Е.М.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 50 сторінки, 23 рисунків, 14 джерел за переліком посилань.

ПРИРОДНИЙ РУХ РУКИ, МАШИННЕ НАВЧАННЯ, ЗВОРОТНЯ КІНЕМАТИКА, ОПТИМІЗАЦІЯ, РЕАБІЛІТАЦІЯ, PCT, PUTHON

В кваліфікаційній роботі проаналізовано методи та технології створення моделей систем керування рухом руки, їх машинного навчання та використання. Основний фокус зроблений на застосуванні перцептивної теорії управління, що надає змогу збільшити ефективність і точність торкання цілі.

Об'єктом дослідження є концепції побудови руху верхньої кінцівки, предметом дослідження – моделі задання природньої (наближеної до людської) траєкторії руху роботизованої руки при торканні цілі у просторі.

Був проведений огляд досліджень у галузі роботизованого руху та фізичний аспект руху людської руки, способи покращення показників точності на певних ділянках траєкторії переміщення руки. Як результат було отримано загальні принципи створення моделі торкання точок у просторі за допомогою руки з врахуванням поставленої задачі та використання спеціальних концепцій та методів у цій галузі.

В роботі побудовано модель торкання рукою точок у просторі з використаннями технологіями машинного навчання.

Результати роботи можуть бути використані для подальшого застосування у розробці протезів та інших реабілітаційних пристроїв, призначених для допомоги відновлення руху.

ABSTRACT

The explanatory note contains 50 pages, 23 figures, 14 sources by the list of references.

NATURAL HAND MOVEMENT, MACHINE LEARNING, INVERSE KINEMATICS, OPTIMIZATION, REHABILITATION, PCT, PYTHON

The qualification work analyzes the methods and technologies for creating models of hand motion control systems, their machine learning, and use. The main focus is on the application of the perceptual control theory, which makes it possible to increase the efficiency and accuracy of touching the target.

The object of the study is the concepts of constructing the upper limb movement, and the subject of the study is the models for setting the natural (close to human) trajectory of the robotic arm when touching targets in space.

A review of research in the field of robotic motion and the physical aspect of human arm movement, ways to improve accuracy in certain parts of the arm trajectory was conducted. As a result, the general principles of creating a model of touching points in space with the help of a hand were obtained, taking into account the task and using special concepts and methods in this area.

In this work, a model of touching points in space with the hand was built using machine learning technologies.

The results of the work can be used for further application in the development of prostheses and other rehabilitation devices designed to help restore movement.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 ОГЛЯД КОНЦЕПЦІЙ ТА ТЕОРІЙ ПОБУДОВИ РУХУ	10
1.1 Концепція кінематики	10
1.1.1 Пряма кінематика	12
1.1.2 Обернена кінематика	13
1.2 Теорія перцептивного управління.....	15
1.3 Огляд досліджень руху людської руки.....	20
1.4 Мета та постановка завдання.....	22
2 РОЗРОБКА ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ РУХУ РУКИ ПІД ЧАС ТОРКАННЯ ДОВІЛЬНИХ ТОЧОК У ПРОСТОРИ.....	23
2.1 Опис використаних технологій	23
2.1.1 Мова програмування Python.....	23
2.1.2 Бібліотека OpenCV	24
2.1.3 Бібліотека NumPy	25
2.1.4 Бібліотека MediaPipe	26
2.1.5 Середовище розробки Anaconda/Spyder.....	26
2.2 Опис програмної реалізації моделей руху руки до точки	27
2.2.1 Створення еталонного зразку	28
2.2.2 Модель на основі інверсної кінематики.....	33
2.2.3 Модель на основі РСТ.....	36
2.3 Порівняльний аналіз розроблених моделей між собою.....	38
3 ОБЧИСЛЕННЯ ФІЗИЧНИХ ПОКАЗНИКІВ ДЛЯ РУХУ РУКИ	39
3.1 Опис особливостей руху руки	39
3.2 Опис алгоритмічної моделі руху руки.....	43
ВИСНОВКИ.....	48

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	49
ДОДАТОК А.....	51
ДОДАТОК Б	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

АЛП – арифметико-логічного пристрою

FK (Forward kinematic) – пряма кінематикаі

GPU (Graphics processing unit) – графічний процесор

IK (Inverse kinematic) – обернена (зворотна) кінематика

PCT (Perceptual control theory) – теорія перцептивного управління

ВСТУП

Актуальність дослідження концепцій побудови руху, особливо надання йому природності людського тіла є відображенням потреби світу у розвитку роботизованих систем, що будуть безпосередньо допомагати людині в різних сферах діяльності. Робототехніка та біомедична інженерія були основним напрямком для цього проекту. Важливим критерієм для роботи є потенціал розвитку та імплементації запропонованої системи у сфері безпосереднього лікування та реабілітації людей. Сьогоднішні методи допомоги надто залежні від людського фактору, що є і плюсом, і мінусом такої терапії. Такі практики можуть бути покращені завдяки додавання роботизованих елементів. Але постає проблема якісної заміни людської допомоги на механічну, тож дане питання і намагається закрити представлена робота. Завдяки наданню роботизованим рукам властивості природності руху, від цього утворюється перспектива плавності руху робота, як людини, завдяки чому можна якісно покращити вказані сфери та задати напрямком розвитку методів побудови руху.

Важливим доповненням є використання нейронних мереж у моделюванні системи, адже вони дають змогу підвищити швидкість і точність симуляції, забезпечуючи урахування всіх заданих параметрів та попередніх помилок. Відповідно, якщо побудовано модель програмно є валідною, то її фізична реалізація не буде проблемою при сучасних досягненнях конструкторів.

Отже, дослідження за даною темою є актуальним та необхідним для розвитку інженерії у напрямку медицини, особливо реабілітації в першу чергу. Додавання роботизованих елементів з новою методикою руху стане основою для покращення та розвитку медицини світу.

1 ОГЛЯД КОНЦЕПЦІЙ ТА ТЕОРІЙ ПОБУДОВИ РУХУ

В даному розділі розглянуто теорії, що існують та були розглянуті у науковій літературі, присвяченій людському руху руки та концепціями передачі цих властивостей на роботизовану систему. Відповідно, проведено огляд цих матеріалів та теоретично вказані їх недоліки відповідно до необхідних якостей у роботі. Підсумовуюче все перелічене, сформульовано мету та виконано постановку завдання роботи.

1.1 Концепція кінематики

Кінематика вивчає рух тіл без урахування сил або моментів, що викликають цей рух. Кінематика роботів – це аналітичне дослідження руху робота-маніпулятора. Формулювання відповідних кінематичних моделей для механізму робота є дуже важливим для аналізу поведінки промислових маніпуляторів. Існує в основному два різних простори, що використовуються в кінематичному моделюванні маніпуляторів, а саме: декартовий простір і простір кватерніонів. Перетворення між двома декартовими системами координат можна розкласти на обертання і трансляцію. Існує багато способів представлення обертання, включаючи наступні: Кути Ейлера, вектор Гіббса, параметри Кейлі-Клейна, спінові матриці Паулі, вісь і кут, ортонормальні матриці та кватерніони Гамільтона. З цих представлень найчастіше в робототехніці використовуються однорідні перетворення на основі дійсних матриць 4×4 (ортонормовані матриці). Денавіт і Хартенберг (1955) показали, що загальне перетворення між двома суглобами вимагає чотирьох параметрів. Ці параметри, відомі як параметри Денавіта-Гартенберга, стали стандартом для опису кінематики роботів. Хоча кватерніони є елегантним представленням обертання, вони не використовуються так часто, як однорідні перетворення в робототехнічній спільноті. Подвійний кватерніон може представити

обертання і трансляцію в компактній формі вектора перетворення одночасно. В той час як в однорідних перетвореннях орієнтація тіла представлена дев'ятьма елементами, подвійні кватерніони зменшують кількість елементів до чотирьох. Це дає значну перевагу з точки зору обчислювальної надійності та ефективності зберігання для роботи з кінематикою ланцюгів роботів [1].

Кінематику роботів можна розділити на пряму та зворотну кінематику. Задача прямої кінематики є простою і не вимагає складності у виведенні рівнянь. Отже, завжди існує рішення прямої кінематики маніпулятора. Зворотна кінематика є набагато складнішою задачею, ніж пряма кінематика. Розв'язання задачі оберненої кінематики вимагає великих обчислень і, як правило, займає дуже багато часу при управлінні маніпуляторами в реальному часі. Сингулярності та нелінійності, які ускладнюють розв'язання задачі. Отже, лише для дуже малого класу кінематично простих маніпуляторів (маніпулятори з зап'ястям Ейлера) існують повні аналітичні розв'язки (Kucuk & Bingul, 2004) [2]. Зв'язок між прямою та зворотною кінематикою проілюстровано на рисунку 1.1.

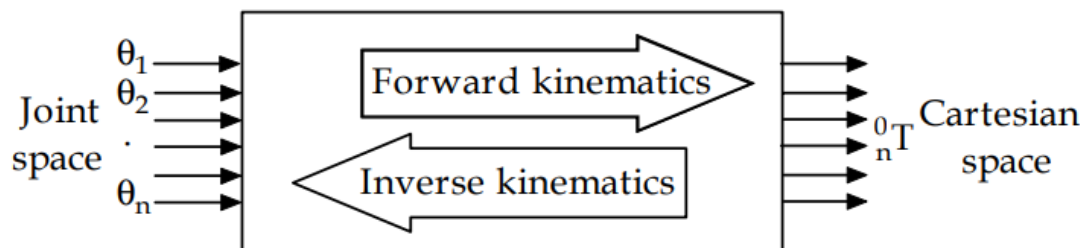


Рисунок 1.1 – Схема зв'язку прямої та зворотної кінематики

Кінематичні моделі у сфері роботопроекування побудовані у вигляді сегментів (беручи до уваги руку це плече, передпліччя, долоні тощо), що з'єднані шарнірами (відповідні суглоби руки). Взаємодію та поведінку цих суглобів описують різні напрямки кінематики, представлені нижче.

1.1.1 Пряма кінематика

У даному напрямку маніпулятор складається з послідовних ланок, які кріпляться один до одного за допомогою поворотних або призматичних шарнірів від базової рами через кінцевий ефект. Розрахунок положення та орієнтації кінцевого ефектора в термінах змінних шарнірів називається прямою кінематикою. Для того, щоб мати пряму кінематику для механізму робота в систематизованому вигляді, необхідно використовувати відповідну кінематичну модель. Метод Денавіта-Гартенберга, який використовує чотири параметри, є найпоширенішим методом опису кінематики робота. Ці параметри a_{i-1} , α_{i-1} , d_i і θ_i – довжина ланки, поворот ланки, зсув ланки і кут з'єднання відповідно. До кожного з'єднання прикріплена координатна рамка для визначення параметрів ДН. Вісь Z_i координатної рамки спрямована вздовж напрямку обертання або ковзання з'єднання. На рисунку 1.2 показано призначення координатної рамки для загального маніпулятора.

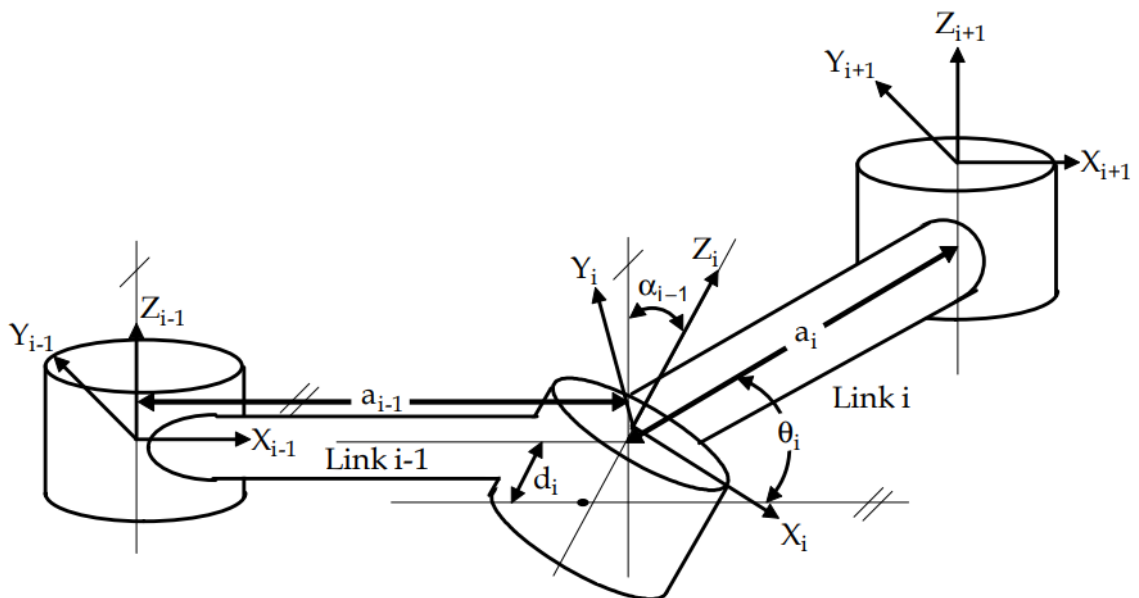


Рисунок 1.2 – Присвоєння координатних рамок для загального маніпулятора

Говорячи про аспекти цієї концепції, що є важливими для аналізу в роботі, варто зазначити, що модель, побудована за ФК парадигмою, має

ієрархічний ланцюжок «зверху вниз», тобто дочірні сегменти рухаються відносно батьківських. Спочатку положення та/або орієнтацію змінює батьківський сегмент. Ця зміна впливає на положення та/або орієнтацію всіх інших дочірніх сегментів. Далі змінюється положення наступного сегмента в ланцюжку, при цьому змінюється положення всіх наступних дочірніх до нього сегментів, а батьківські сегменти залишаються нерухомими. Відповідно до людського тіла спочатку буде виконано руху плечем, допоки його не зафіксовано в бажаному положенні, далі настане черга передпліччя при цьому плече має залишатися нерухомим. Такий шаблон буде повторюватися до останньої ланки моделі руки.

Відповідно до цього визначення, опису та принципу дії, концепція прямої кінематики не є доцільною для використання у темі роботи, адже принцип її дії протилежний людській природній поведінці руху, що не передбачає ієрархічний рух «згори-вниз».

1.1.2 Обернена кінематика

Задача оберненої кінематики серійних маніпуляторів вивчається вже багато десятиліть. Вона необхідна для керування маніпуляторами. Розв'язання оберненої кінематики вимагає великих обчислень і зазвичай займає дуже багато часу при управлінні маніпуляторами в реальному часі. Завдання, які повинен виконувати маніпулятор, знаходяться в декартовому просторі, в той час як приводи працюють у тривимірному просторі. Декартовий простір включає в себе матрицю орієнтації та вектор положення. Натомість суглобовий простір представлений суглобовими кутами. Перетворення положення та орієнтації кінцевого ефекту маніпулятора з декартового простору в шарнірний простір називається оберненою задачею кінематики [3]. На рисунку 1.3 зображено координати схематичної модель маніпулятора в оберненій кінематиці.

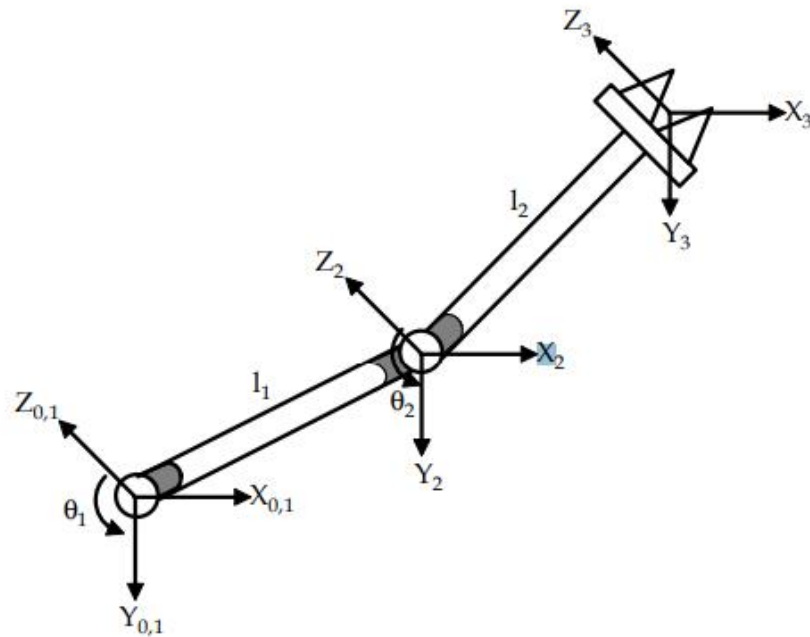


Рисунок 1.3 – Система координат для плоского маніпулятора

ІК оперує таким же ланцюгом сегментів, як і у прямій кінематиці, проте принцип дії є повністю протилежним. Ця концепція демонструє ієрархічний зв'язок «знизу вверху», тобто рух дочірніх сегментів впливає на положення батьківських.

В даній теорії рух та/або орієнтацію ініціює дочірній сегмент. Ця зміна впливає на положення та/або орієнтацію всіх інших батьківських сегментів. Задіяна ланка має назву ефектор. Якщо ефектор змінює своє положення/орієнтацію, то всі батьківські сегменти до нього так само змінюють своє положення/орієнтацію за законами інверсної кінематики. Всі сегменти після нього (дочірні відповідно поточного сегменту) будуть змінюватися за законом прямої кінематики.

Враховуючи, особливості руху людини, можна зазначити, що ця концепція набагато реалістичніше відображає рух руки в даному випадку, ніж пряма кінематика. Адже рух людини многокомпонентна дія, що не може бути описана простою ієрархією.

Обернена кінематика широко застосовується у роботехніці та моделюванні руху людини. Цей факт дозволяє розглядати цю теорію для розробки моделі торкання точок у просторі.

1.2 Теорія перцептивного управління

Теорія перцептивного управління – це теорія, що описує модель поведінки, заснована на властивостях контурів управління з негативним зворотним зв'язком. Контур керування підтримує контрольовану змінну на рівні або поблизу еталонного значення за допомогою впливу своїх виходів на цю змінну, опосередкованого фізичними властивостями навколишнього середовища.

В теорії інженерного управління еталонні значення встановлюються користувачем поза системою. Прикладом є термостат. У живому організмі референтні значення для контрольованих змінних сприйняття підтримуються ендогенно. Біологічний гомеостаз і рефлексії є простими, низькорівневими прикладами. Відкриття математичних принципів контролю дозволило змодельовати петлю негативного зворотного зв'язку, замкнену через навколишнє середовище (циркулярна причинність), що породило теорію перцептивного управління. Вона принципово відрізняється від деяких моделей у поведінковій та когнітивній психології, які моделюють стимули як причини поведінки (лінійна причинність). Дослідження РСТ (Perceptual control theory) опубліковані в експериментальній психології, нейронауках, етології, антропології, лінгвістиці, соціології, робототехніці, психології розвитку, організаційній психології та менеджменті, а також у ряді інших галузей. РСТ застосовувався для проектування та адміністрування освітніх систем і призвів до створення психотерапії, яка отримала назву метод рівнів.

Теорія перцептивного управління має глибоке коріння в біологічній кібернетиці, системній біології та теорії управління, а також у пов'язаній з нею концепції зворотного зв'язку. На відміну від деяких моделей у поведінковій та

когнітивній психології, вона виходить з концепції кругової причинності. Таким чином, вона поділяє свою теоретичну основу з концепцією управління рослинами, але відрізняється від неї тим, що наголошує на контролі внутрішньої репрезентації фізичного світу [4].

Сприйняття в РСТ будується і контролюється в ієрархії рівнів. Наприклад, візуальне сприйняття об'єкта базується на різниці в інтенсивності світла або різниці у відчуттях, таких як колір на його краях. Керування формою або розташуванням об'єкта вимагає зміни сприйняття відчуттів або інтенсивності (які контролюються системами нижчого рівня). Цей організаційний принцип діє на всіх рівнях, аж до найабстрактніших філософських і теоретичних конструкцій.

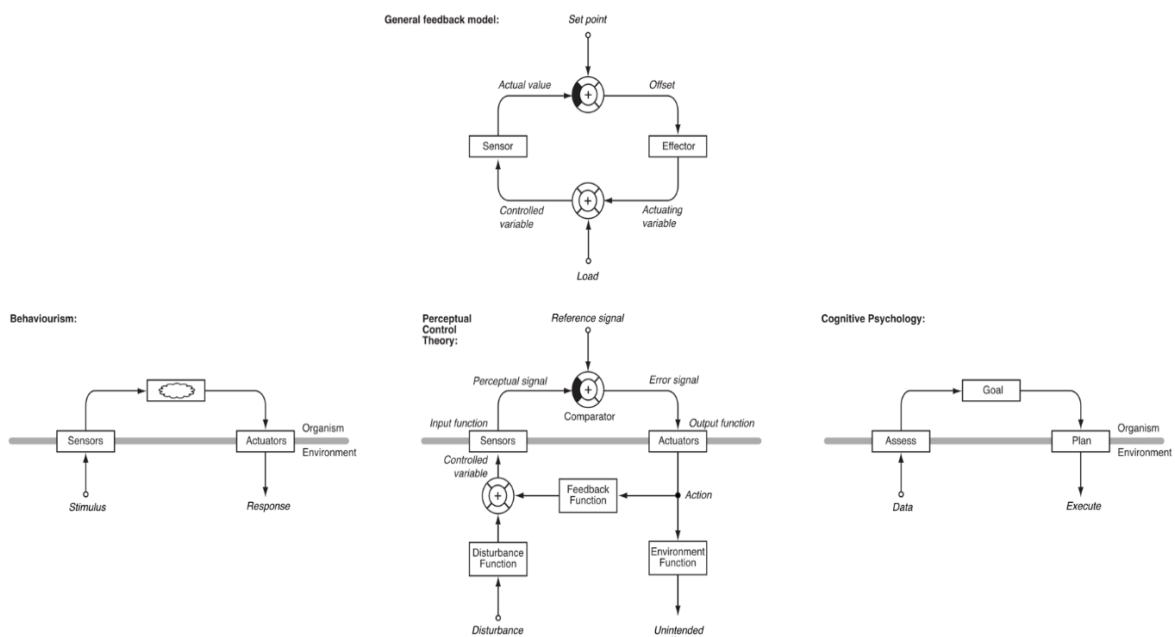


Рисунок 1.4 – Теорія перцептивного управління в порівнянні з поведінковою теорією та когнітивною психологією

У той час як інженерна система управління має еталонне значення або задане значення, яке регулюється зовнішнім органом, еталонне значення для біологічної системи управління не може бути встановлене таким чином.

Задане значення повинно виходити з якогось внутрішнього процесу. Якщо є спосіб впливати на поведінку, то будь-яке сприйняття може бути миттєво приведене до стану, визначеного вищими рівнями, а потім підтримуватися в цьому стані, незважаючи на непередбачувані збурення. В ієрархії систем управління вищі рівні коригують цілі нижчих рівнів як засіб досягнення власних цілей, встановлених ще вищими системами. Це має важливі наслідки для будь-якого запропонованого зовнішнього контролю автономної живої системи управління (організму). На найвищому рівні еталонні значення (цілі) встановлюються спадковістю або адаптивними процесами.

У 2008 році Пауерс створив симуляцію координації рухів руки [5]. Він припустив, що для того, щоб рухати рукою, задіяні чотирнадцять систем управління, які контролюють чотирнадцять суглобових кутів, і вони реорганізуються одночасно і незалежно. Було виявлено, що для оптимальної роботи вихідні функції повинні бути організовані таким чином, щоб вихід кожної системи контролю впливав лише на одну змінну навколишнього середовища, яку вона сприймає. У цій симуляції процес реорганізації працює належним чином, і саме так, як припускає Пауерс, він працює у людей, зменшуючи виходи, які спричиняють помилки, і збільшуючи ті, які зменшують помилки. Спочатку збурення мають великий вплив на кути з'єднань, але з часом кути з'єднань стають ближчими до еталонних сигналів завдяки реорганізації системи. Пауерс припускає, що для того, щоб досягти координації кутів суглобів для виконання бажаних рухів, замість того, щоб обчислювати, як повинні змінитися кути суглобів для виконання цього руху, мозок використовує системи негативного зворотного зв'язку, щоб генерувати необхідні кути суглобів. Один опорний сигнал, який змінюється в системі вищого порядку, може генерувати рух, який вимагає одночасної зміни кількох суглобових кутів [5].

Теорія перцептивного управління може надати пояснювальну модель нейронної організації, яка має відношення до поточних проблем. РСТ описує ієрархічний характер поведінки як такий, що визначається контролем

ієрархічно організованого сприйняття. Системи управління в організмі та у внутрішньому середовищі мільярдів взаємопов'язаних нейронів у мозку відповідають за утримання перцептивних сигналів у допустимих межах у непередбачувано мінливому середовищі, з якого ці сприйняття походять. РСТ не припускає, що існує внутрішня модель, в рамках якої мозок імітує поведінку перед тим, як віддати команди для її виконання. Натомість, однією з її характерних рис є принципова відсутність мозкової організації поведінки. Скоріше, поведінка є змінним засобом організму для зменшення розбіжностей між сприйняттям і еталонними значеннями, які базуються на різних зовнішніх і внутрішніх вхідних даних [6]. Поведінка повинна постійно адаптуватися і змінюватися, щоб організм міг підтримувати свої перцептивні цілі. Таким чином, ця теорія може пояснити абстрактне навчання через спонтанну реорганізацію ієрархії. РСТ припускає, що конфлікт виникає між різними еталонними значеннями для даного сприйняття, а не між різними реакціями, [7] і що навчання реалізується як зміна властивостей систем контролю методом проб і помилок, [8] а не як підкріплення якоїсь конкретної реакції. Таким чином, поведінка залишається адаптивною до навколишнього середовища, а не покладається на вивчені шаблони дій, які можуть не відповідати дійсності.

Теорія управління сприйняттям наразі пропонує ієрархію з 11 рівнів сприйняття, які контролюються системами в людському розумі та нейронній архітектурі. Це: інтенсивність, відчуття, конфігурація, перехід, подія, відношення, категорія, послідовність, програма, принцип і концепція системи. Різноманітні перцептивні сигнали на нижчому рівні (наприклад, візуальне сприйняття інтенсивності) об'єднуються у вхідній функції для побудови єдиного сприйняття на вищому рівні (наприклад, візуальне сприйняття відчуття кольору). Сприйняття, які створюються і контролюються на нижчих рівнях, передаються як вхідні дані на вищі рівні. Вищі рівні, у свою чергу, контролюють, коригуючи еталонні рівні (цілі) нижчих рівнів, фактично

вказуючи нижчим рівням, що сприймати [9] [10]. На рисунку 1.5 зображена схема організації рівнів сприйняття.

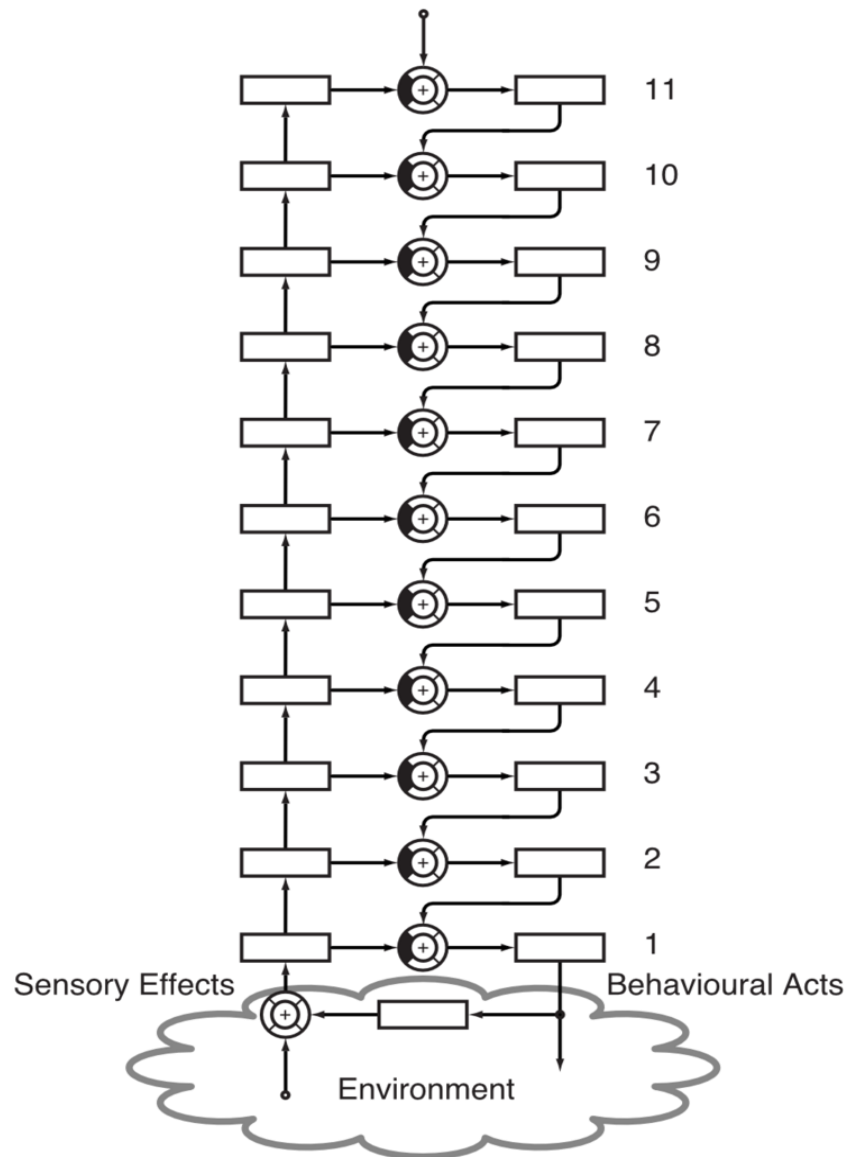


Рисунок 1.5 – Загальна блок-схема організації обробки інформації у РСТ

Хоча було розроблено безліч комп'ютерних демонстрацій принципів, запропоновані вищі рівні важко моделювати, оскільки занадто мало відомо про те, як працює мозок на цих рівнях. Ізольовані процеси управління більш високого рівня можна дослідити, але моделі великої ієрархії управління все ще є тільки концептуальними або в кращому разі рудиментарними.

Останні підходи використовують принципи теорії перцептивного управління для надання нових алгоритмічних основ для штучного інтелекту та машинного навчання [11].

1.3 Огляд досліджень руху людської руки

Важливу роль у розробці власної моделі відіграє огляд готових напрацювань, під час якого будуть здобуті необхідні для роботи знання.

Для симуляції руху людської руки варто приділити увагу і фізичним властивостям також: швидкість та траєкторія. Завдяки минулим дослідженням можна стверджувати, що швидкість руху буде нелінійною [12]. Нижче на рисунках 1.6-1.7 наведена інфографіка, що були створені у процесі роботи, з якою довелося ознайомитися.

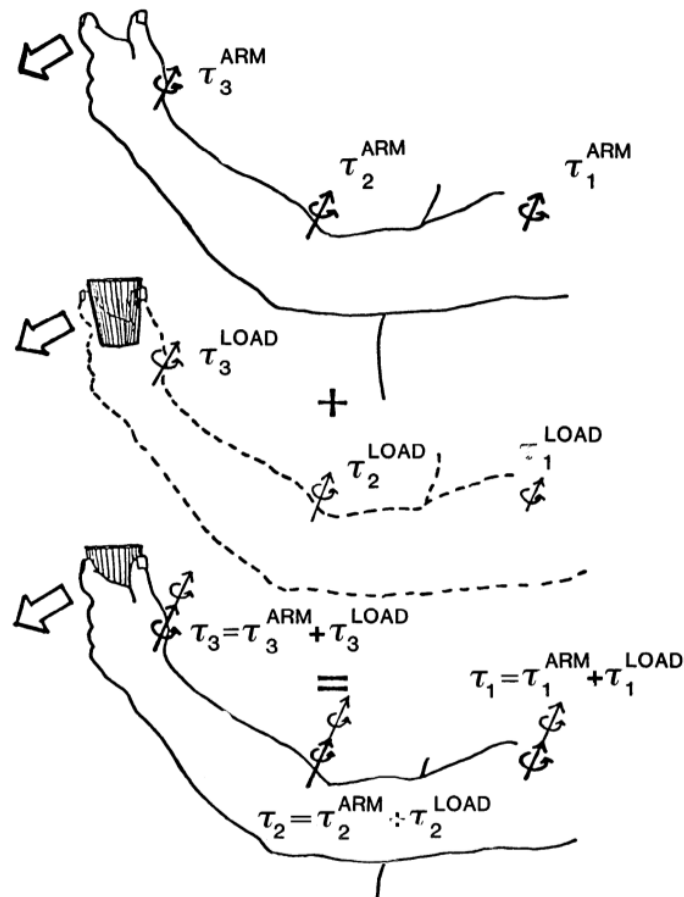


Рисунок 1.6 – Параметри, що вимірюються при зміні положення руки

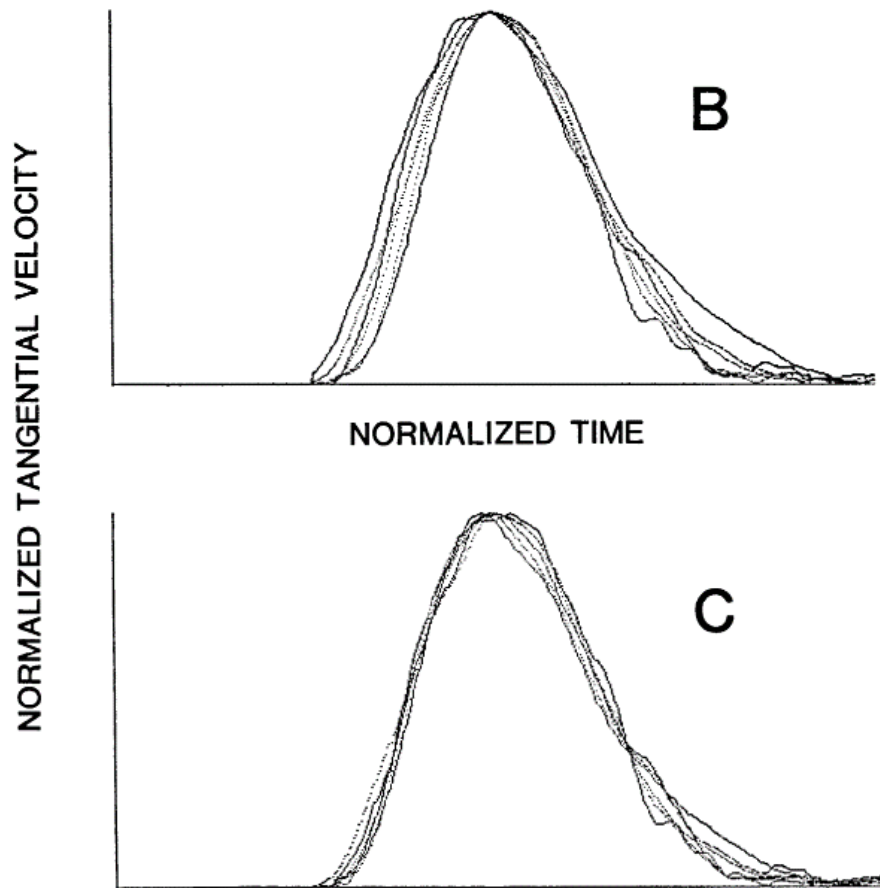


Рисунок 1.7 – Динаміка зміни траєкторії руху руки до цілі

На рисунку 1.7 був наведений графік нормалізованої тангенціальної швидкості руху руки до точки (через нормалізацію одиниці вимірювання на осях не мають змісту, тож не вказані). Графік відображає рух руки від старту до торкання цілі. На ньому можна чітко бачити, що швидкість частину руху буде майже лінійно зростати і досягнувши певного максимуму, почне зменшуватися, що демонструє реальну поведінку людини – сповільнення для корегування точності. Чим ближче рука до цілі, тим повільнішою буде ставати її швидкість.

Маючи ці доповнення про особливість руху людської руки, можна виконати розрахунки і для роботизованої моделі, що внесуть природність руху для неї та допоможуть здобути натуральність людського тіла.

Для логічного розділення кваліфікаційної роботи всі обчислення та супроводжуючі графічні матеріали будуть наведені у розділі 3.

1.4 Мета та постановка завдання

Метою кваліфікаційної роботи є розробка моделі роботизованої руки, що має торкатися довільних точок у просторі з використанням у процесі машинного навчання.

Об'єктом дослідження є концепції побудови руху людської руки, предметом – моделі та методи задання траєкторій руху руки для торкання точок у просторі.

Для досягнення поставленої мети визначено наступні завдання:

- огляд концепцій та дослідження принципів механіки та способів опису руху людського тіла;
- огляд та аналіз технологій, бібліотек та інструментів для розробки моделі;
- запис відео руху живої людини;
- створення програми для обробки відео руху живої людини;
- розробка моделей торкання рукою точок у просторі на основі інверсної кінематики та перцептивної теорії управління;
- порівняння розроблених моделей торкання рукою точок у просторі;
- створення алгоритму роботи руки під час руху.

2 РОЗРОБКА ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ РУХУ РУКИ ПІД ЧАС ТОРКАННЯ ДОВІЛЬНИХ ТОЧОК У ПРОСТОРИ

У даному розділі розроблено рухову модель кінцівки (руки), що дотикається до довільних точок у просторі. Акцент зроблено на побудові правильних траєкторій руху руки. Надано опис технологій, що були використані, бібліотек та інструментів для розробки, а саме мови програмування, бібліотек та технологій машинного навчання, здійснено опис розробленого коду з лістингами роботи програми.

2.1 Опис використаних технологій

При програмній розробці моделі руху руки при дотику до довільних просторових точок були задіяні наступні технології: мова програмування Python, і бібліотеки OpenCV, Numpy, Mediarpipe.

2.1.1 Мова програмування Python

Python є високорівневою інтерпретованою мовою програмування, яка здобула широку популярність завдяки своїй простоті та зручності використання. Його зрозумілий і лаконічний синтаксис сприяє легкому написанню та підтримці коду, що робить Python ідеальним інструментом для швидкої розробки програм і створення прототипів. Однією з визначальних характеристик Python є динамічна типізація, коли типи змінних встановлюються під час виконання програми. Це забезпечує високу гнучкість у процесі розробки.

Python широко застосовується для створення програмного забезпечення в різних галузях, включно з машинним навчанням та обробкою зображень та математичними обчисленнями, що є ключовими компонентами у побудові максимально наближеної моделі руху руки. Завдяки великій стандартній

бібліотеці та доступу до численних сторонніх модулів Python дозволяє ефективно реалізовувати складні алгоритми та працювати з великими масивами даних.

Для розробки кінематичної моделі рухи маніпулятора (руки) на Python можуть бути використані такі бібліотеки, як OpenCV для обробки зображень і NumPy для використання математичних операторів та функцій.

У свою чергу OpenCV надає широкий набір інструментів для роботи із зображеннями, включно з їхнім захопленням, обробкою та аналізом. Python підтримує інтеграцію з різноманітними платформами, що дає змогу створювати багатофункціональні системи, комбінуючи кілька сучасних технологій.

2.1.2 Бібліотека OpenCV

OpenCV, що розшифровується як Open Source Computer Vision Library, є потужним інструментом для комп'ютерного зору та обробки зображень, який надає широкий спектр алгоритмів і функцій для аналізу візуальної інформації. Бібліотека підтримує роботу як із двовимірними, так і з тривимірними зображеннями, що дозволяє виконувати складні завдання, такі як виявлення об'єктів, калібрування камери та стереоскопія. Однією з ключових переваг OpenCV є її висока ефективність, яка забезпечується оптимізованими алгоритмами на C/C++ із підтримкою SIMD-інструкцій та використанням апаратного прискорення, зокрема GPU. Завдяки цьому OpenCV є ідеальним рішенням для систем реального часу, які потребують мінімальної затримки при обробці зображень.

Бібліотека пропонує широкий набір інструментів для обробки та аналізу зображень, зокрема геометричні перетворення, сегментацію, виділення контурів, фільтрацію, згладжування, а також більш складні алгоритми, такі як гістограма напрямлених градієнтів і розпізнавання об'єктів за допомогою каскадів Хаара. Ці функції забезпечують основу для створення

інтелектуальних систем, здатних ідентифікувати та класифікувати об'єкти на зображеннях і відео.

Бібліотека легко інтегрується з іншими популярними інструментами машинного навчання. Це дозволяє розробникам використовувати глибокі нейронні мережі для підвищення точності та надійності алгоритмів.

Завдяки своїй гнучкості та широким можливостям OpenCV знаходить застосування в таких галузях, як робототехніка, доповнена реальність, автоматизований контроль якості та системи безпеки. Ця бібліотека є незамінним інструментом для розробників, які працюють у сфері моделювання нових прототипів та систем.

2.1.3 Бібліотека NumPy

NumPy є фундаментальною бібліотекою для наукових обчислень у Python, яка забезпечує потужний набір інструментів для роботи з багатовимірними масивами та матрицями, а також містить велику кількість математичних функцій для виконання складних обчислень. Завдяки ядру, оптимізованому на мові C, NumPy демонструє високу продуктивність, значно перевищуючи швидкість стандартних засобів Python, що робить її ключовим інструментом для чисельних розрахунків.

Однією з основних особливостей бібліотеки є підтримка багатовимірних масивів (`ndarray`), які дозволяють ефективно зберігати та обробляти великі обсяги даних. Масиви можуть мати будь-яку кількість вимірів, що забезпечує гнучкість у роботі з різними структурами даних – від одномірних векторів до багатовимірних тензорів, які широко застосовуються в машинному навчанні, обробці зображень та інших наукоємних галузях.

Завдяки своїй ефективності та функціональності, NumPy є незамінним інструментом для розробників, дослідників і науковців, які працюють із великими наборами даних або розв'язують складні математичні задачі.

2.1.4 Бібліотека MediaPipe

MediaPipe – це бібліотека для створення та реалізації проектів з використанням комп'ютерного зору. Вона надає високоефективні модулі для обробки відео і зображень у реальному часі. Дозволяє просто інтегрувати функціональні можливості, такі як розпізнавання облич, жестів рук, відстеження руху тіла, оцінка позиції обличчя та багато інших.

Бібліотека використовує сучасні алгоритми машинного навчання для точного аналізу та обробки відеоданих. Крім того, MediaPipe відрізняється своєю модульною архітектурою, що дозволяє комбінувати різні компоненти для створення складних оброблювальних конвеєрів. Завдяки цьому, вона є універсальним інструментом як для науковців, так і для розробників комерційних застосунків.

У роботі ця бібліотека буде використана саме для відстеження руху людини і його подальшої обробки.

2.1.5 Середовище розробки Anaconda/Spyder

Anaconda – це інтегроване середовище для роботи з Python і R, орієнтоване на наукові обчислення, аналіз даних і машинне навчання (рис 2.1). Воно надає зручний спосіб управління середовищами та пакетами, забезпечуючи стабільну інфраструктуру для розробки складних проектів. Однією з ключових особливостей Anaconda є використання менеджера пакетів Conda, який дозволяє встановлювати, оновлювати та ізолювати різні програмні компоненти. Anaconda також включає графічний інтерфейс Navigator, який забезпечує легкий доступ до інструментів, таких як Jupyter Notebook, RStudio та Spyder, і дозволяє запускати їх без необхідності налаштовувати середовище вручну.

Spyder, своєю чергою, є інтегрованим середовищем розробки, яке входить до складу Anaconda і спеціально розроблене для роботи з Python у наукових і аналітичних задачах. Воно поєднує в собі текстовий редактор із підтримкою Python, інтерактивну консоль для виконання коду та інструменти

візуалізації змінних і даних. Spyder забезпечує зручний підхід до дослідницького програмування, дозволяючи користувачам аналізувати дані в реальному часі, візуалізувати результати та швидко коригувати код.

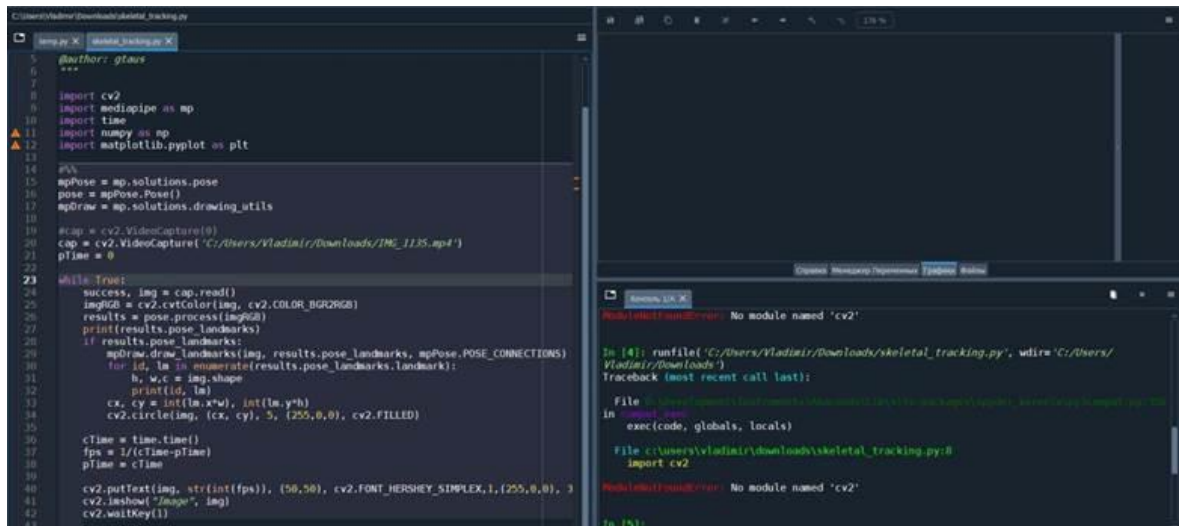


Рисунок 2.1 – Середовище розробки Anaconda/Spyder

Завдяки інтеграції з популярними бібліотеками для аналізу даних, такими як NumPy і Matplotlib, Spyder стає ідеальним вибором для аналітиків і дослідників, які працюють із великими обсягами даних або створюють математичні моделі.

Саме такі інструменти, які є ефективними, зручними та не надлишковими, були обрані для реалізації дослідницької моделі торкання рукою точок у просторі.

2.2 Опис програмної реалізації моделей руху руки до точки

Для розробки моделі будуть залучені дві концепції, що мають найбільші показники точності передачі руху: зворотна кінематика та теорія перцептивного управління. По кожній з них буде створена своя модель торкання рукою точки. Порівняння буде відбуватися з використанням еталонного зразку. Всі моделі та зразок виконані на двовимірній площині –

така міра дозволяє наочно бачити результат без ускладнення моделі, як первинної у цій розробці.

2.2.1 Створення еталонного зразку

Для коректного виконання завдання та подальшої перевірки спроектованих моделей, було зроблено зразок руху руки людини. Ця процедура була здійснена шляхом зйомки процесу руху на відео: доброволець виконував довільні рухи рукою з метою доторкнутися до уявних точок у просторі. Відео було записано вдруге, адже в першій версії було замало варіантів руху, тож у другій кількість «точок» була збільшена, відповідно кількість нових траєкторій теж зросла. Такий матеріал дозволяє бачити основне для проектування моделей: як формується траєкторія людського руху та яких кутів згинання/розгинання набувають фрагменти руки. Кадр з відео наведено на рисунку 2.2.



Рисунок 2.2 – Приклад руху руки живої людини

Надалі отримане відео було оброблено для формування на руці основних точок суглобів та ліній плеча, передпліччя за схемою, що зображена на рисунку 2.3.

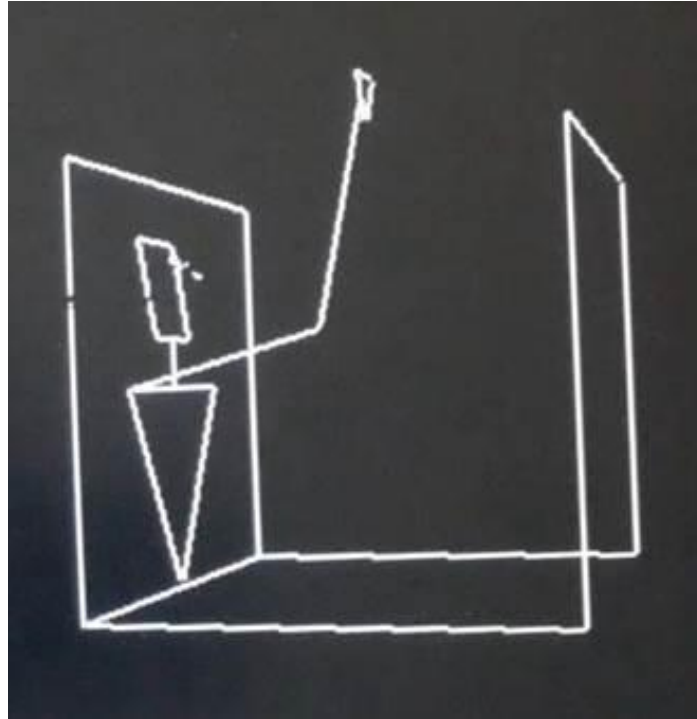


Рисунок 2.3 – Схематична модель руху кінцівки

Таким чином відео було завантажено до програми у Spyder, яка в результаті відсканувала людину і нанесла відповідні точки та відрізки. Лістинг коду та результат роботи програми наведені нижче.

Лістинг 2.1 – Код для відскановування та розмічування руки

```
import cv2
import mediapipe as mp
import time
import numpy as np
import matplotlib.pyplot as plt
mpPose = mp.solutions.pose
pose = mpPose.Pose()
mpDraw = mp.solutions.drawing_utils
cap =
cv2.VideoCapture('C:/Users/Vladimir/Downloads/IMG_1135.mp4')
```

```

pTime = 0
while True:
    success, img = cap.read()
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    results = pose.process(imgRGB)
    print(results.pose_landmarks)
    if results.pose_landmarks:
        mpDraw.draw_landmarks(img, results.pose_landmarks,
mpPose.POSE_CONNECTIONS)
        for id, lm in enumerate(results.pose_landmarks.landmark):
            h, w, c = img.shape
            print(id, lm)
            cx, cy = int(lm.x*w), int(lm.y*h)
            cv2.circle(img, (cx, cy), 5, (255,0,0), cv2.FILLED)
        cTime = time.time()
        fps = 1/(cTime-pTime)
        pTime = cTime
        cv2.putText(img, str(int(fps)), (50,50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255,0,0), 3)
        cv2.imshow("Image", img)
        cv2.waitKey(1)

```

Програма побудована з використанням бібліотеки MediaPipe, що дозволяє відстежувати рухи та оброблювати їх. В основній секції циклу while відбувається зчитування завантаженого відео, далі завдяки бібліотеці mediapipe та її вбудованим функціям виконується фіксація пози, зчитування основних фрагментів тіла (відрізки ставляться по частині кінцівки, а точки по суглобу) та нанесення цих даних по верх початкового відео (за допомогою бібліотеки opencv) з урахуванням масштабу довжини та ширини. Така обробка виконується кожен фрейм відео, що дозволяє отримати достатньо точну картину руху тіла за відео, яке створюється після закінчення роботи програми. Фрагмент роботи програми винесений на рисунок 2.4.



Рисунок 2.4 – Відсканована модель з живої людини

Для проведення майбутніх перевірок отриманих моделей розроблюваних систем з використанням зворотної кінематики та перцептивного управління з еталонними даними, буде використовувалося вище згадане відео руху людини, яке попередньо було оброблено програмою-трекером. З цього відео були вилучені окремі кадри, що відображають рух від однієї точки до іншої, для подальшого порівняння траєкторій, які визначила програма, з фактичними рухами, що здійснює людина під час природного пересування кінцівки.

Групи кадрів були накладені один на одного, щоб створити суцільне зображення тіла, з особливим акцентом на рух руки. Це дало змогу чітко простежити траєкторію руху руки протягом всього циклу руху, забезпечуючи точну візуалізацію та більш глибоке розуміння динаміки руху людини (детальніше рис. 2.5).



Рисунок 2.5 – Готове зображення після накладання

Далі була виконана розмітка основних ліній кінцівки, яку можна побачити на рисунку 2.6.

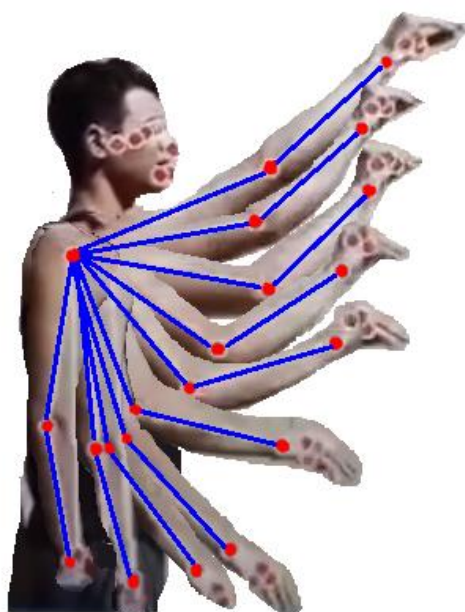


Рисунок 2.6 – Макет траєкторій руху людини для порівняння

2.2.2 Модель на основі інверсної кінематики

Інверсна кінематика сьогодні є основною концепцією для графічного моделювання рухів живих істот (в області створення відео, комп'ютерних ігор). Тож в першу чергу необхідно розглянути її як концепцію для створення моделі по завданню і дізнатися наскільки точно з її використанням можна виконати поставлені умови.

Для реалізації моделі був створений програмний код, що схематично описує траєкторію руху руки людини до точки завдяки системі зв'язків відрізків, що відображають сегменти руки, та кутів між ними. Даний код відображений у лістингу 2.2.

Лістинг 2.2 – Модель торкання точок на основі інверсної кінематики

```
L1 = 2.0 # Length of the first link
L2 = 1.5 # Length of the second link
# Initial position of the end effector
initial_x = 1
initial_y = -1
# Define multiple target points for the end effector
targets = [(1.5, 0.5), (3, 1), (0, 3), (2, 2.8), (2, 2.5), (2, -1), (-1, -2), (-1, 2.5), (1, 3), (3, -1), (3.4, 0.5), (0, -3.5)]
# Inverse Kinematics calculations
def inverse_kinematics(x, y):
    r = np.sqrt(x**2 + y**2)
    if r > (L1 + L2) or r < abs(L1 - L2):
        print("Target point is out of reach.")
        return None
    cos_theta2 = (x**2 + y**2 - L1**2 - L2**2) / (2 * L1 * L2)
    sin_theta2 = np.sqrt(1 - cos_theta2**2)
    theta2 = np.arctan2(sin_theta2, cos_theta2)
    k1 = L1 + L2 * cos_theta2
    k2 = L2 * sin_theta2
    theta1 = np.arctan2(y, x) - np.arctan2(k2, k1)
    return np.degrees(theta1), np.degrees(theta2)
# Lists to store the trajectory of the end effector
trajectory_x = []#[initial_x]
trajectory_y = []#[initial_y]
plt.figure()
# Process each target point
for target in targets:
    target_x, target_y = target
    angles = inverse_kinematics(target_x, target_y)
    if angles:
```

```

theta1, theta2 = angles
# Calculate the new position of the end effector
x1 = L1 * np.cos(np.radians(theta1))
y1 = L1 * np.sin(np.radians(theta1))
x2 = x1 + L2 * np.cos(np.radians(theta1 + theta2))
y2 = y1 + L2 * np.sin(np.radians(theta1 + theta2))
# Append the new position to the trajectory lists
trajectory_x.append(x2)
trajectory_y.append(y2)
plt.plot([0, x1, x2], [0, y1, y2], marker='o', color='b')
plt.plot(trajectory_x, trajectory_y, 'or')
plt.scatter([t[0] for t in targets], [t[1] for t in targets],
            color='blue', label='Targets')
plt.xlim(-L1 - L2 - 1, L1 + L2 + 1)
plt.ylim(-L1 - L2 - 1, L1 + L2 + 1)
plt.title('End Effector Trajectory')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.grid(True)
# plt.legend(['End Point', 'Targets'])
plt.show()

```

Основна робота у цій програмі виконується у функції `inverse_kinematics(x, y)`, що приймає у якості аргументів координати точки, до якої буде торкатися «рука». Всередині функції завдяки групі математичних формул зворотної кінематики, обчислюються кути згинання фрагментів кінцівки та положення (координати) ліктьового суглоба на площині. У програмі вважається, що плечовий суглоб розміщений за координатами (0, 0) та є нерухомим. Також програма містить перевірку на здатність дістати «рукою» до певної точки. Якщо результат буде негативним програма повідомить про це повідомленням у консоль, а відрізки до такої точки не будуть будуватися. Результати роботи наведені на рисунку 2.7.

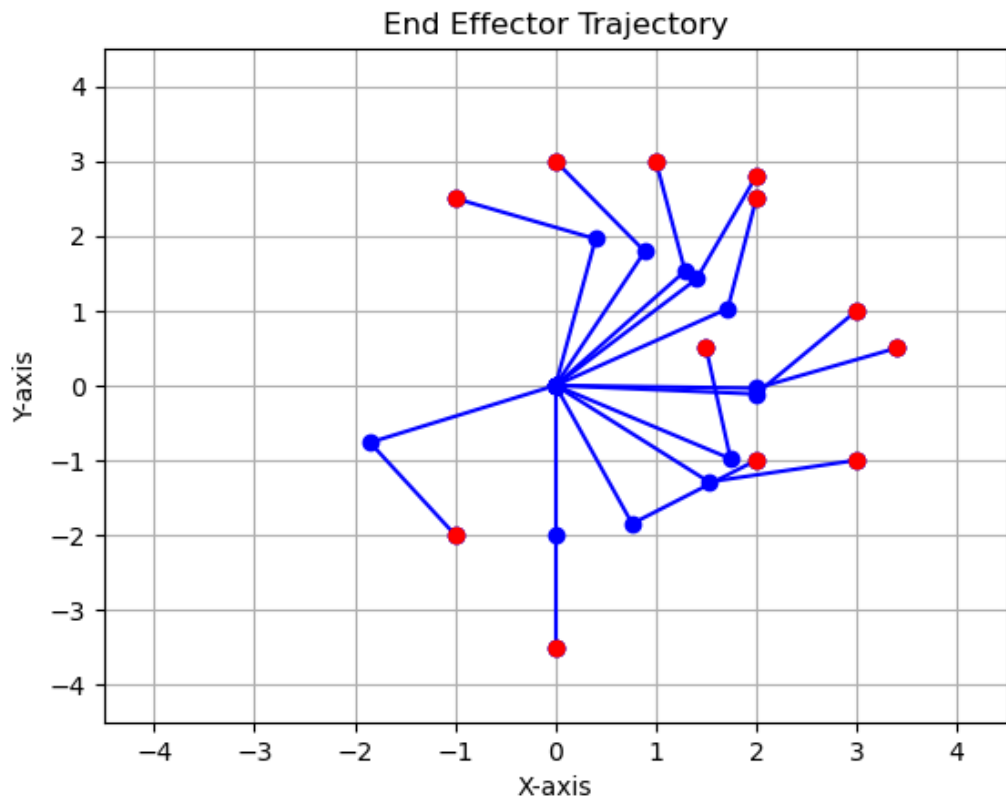


Рисунок 2.7 – Траєкторії руки у моделі зворотної кінематики

Далі важливо виконати порівняння отриманої моделі з еталоном шляхом зіставлення окремих точок та загальної траєкторії. Такий метод дозволяє виявити можливі розбіжності між прогнозованими та фактичними рухами, що сприяє вдосконаленню моделі та її налаштуванню для більш точного відтворення людської кінематики.

Для цієї задачі задаємо у програмі координати точок, приблизно наведені на рисунку 2.6, що містить розмічений еталон. Виконується це шляхом заповнення масиву `targets` у програмному коді (лістинг 2.2). Для порівняння було виставлено 10 точок.

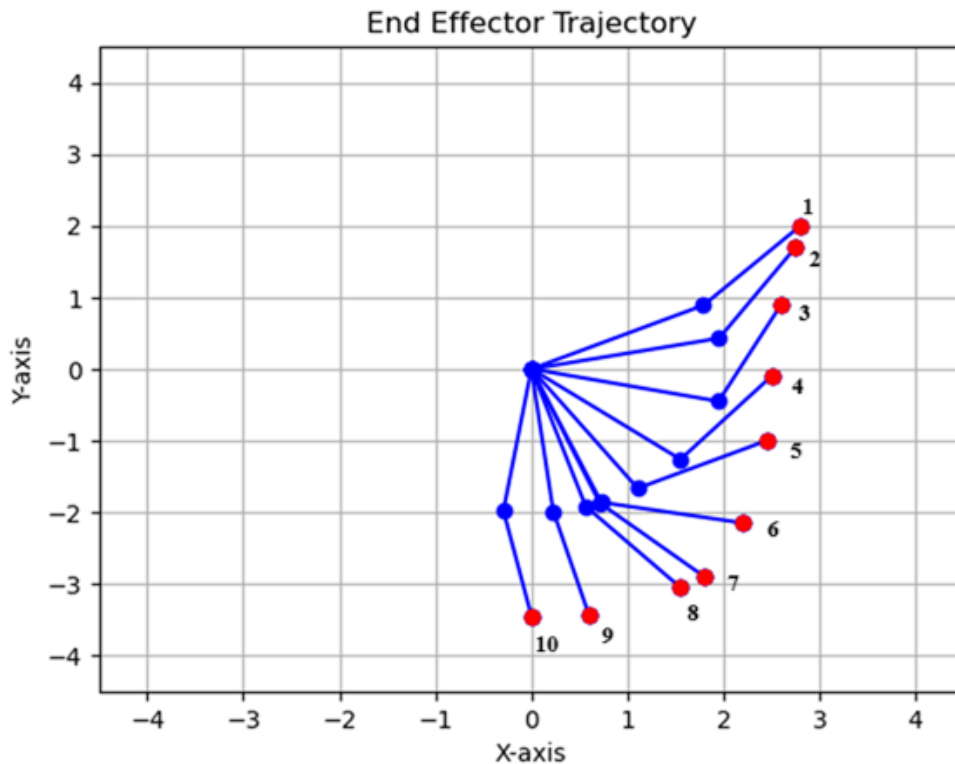


Рисунок 2.8 – Траекторій руху руки зворотної кінематики для порівняння

Варто наголосити, що розглядається рух від положення у точці «1» до точки «10». Точки, розташовані між ними надані, щоб продемонструвати динаміку траекторії руху. Отже, зіставляючи рисунки 2.6 та 2.8 можна помітити, що загальна схема руху достатня, щоб вважати її коректною, проте слід зауважити, що точки «4-5» відрізняються від еталону, що демонструє недоліки роботи алгоритму інверсної кінематики.

2.2.3 Модель на основі РСТ

Модель, що використовує перцептивну теорію управління базується на сприйнятті навколишнього середовища та обробкою внутрішньою системою. Така система стає самокерованою, так само як і людський організм.

Основною задачею моделі є рух руки до точки, тож для поточного процесу на кінцеву точку руки встановлюється сенсор для фіксації відстані до цілі (у людину цю функцію сприймають органи відчуття як очі та вуха). Коли

рука наближається до точки сенсор надає зворотній зв'язок моделі, яка в свою чергу підлаштовує руку під цю точку. Обробка подібних сигналів від сенсора виконується у нескінченному циклі з певними часовими інтервалами (подібно до нервових імпульсів). Оновлюючи інформацію від сенсора, алгоритм РСТ може коригувати траєкторію руки, відповідно такий рух буде вважатися більше подібним до людського, що і є метою роботи. Результат роботи алгоритму можна спостерігати на рисунку 2.9.

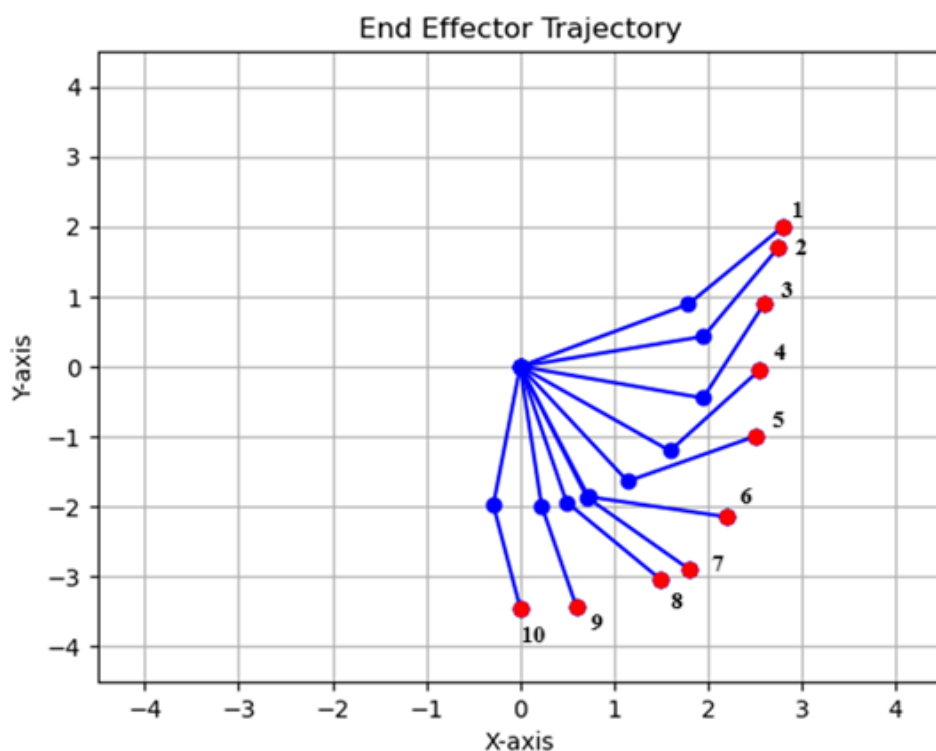


Рисунок 2.9 – Траєкторій руху руки РСТ для порівняння

Таким чином отриманий результат моделі РСТ буде порівнюватися з еталоном (рисунку 2.6), можна зазначити, що відповідність є доволі високою і всі точки можна вважати такими, що зійшлися з оригіналом, що говорить виконання поставленого завдання з урахуванням нюансів розробки проекту та першої спроби імплементації теорії перцептивного управління у модель.

2.3 Порівняльний аналіз розроблених моделей між собою

Порівняння двох моделей необхідне для зауваження недоліків та сильних сторін кожної. Для порівняння будуть використані рисунки результатів (2.8 та 2.9) та часові показники роботи програм.

Графічні результати обох моделей демонструють достатню схожість один до одного, проте у точках «4-5» та «7-8» можна помітити головну відмінність: інверсна кінематика приділяє більше уваги повороту найближчому до кінцевої точки відрізка (передпліччя), у той час як РСТ виконує оберт одразу обох відрізків. Ця особливість робить рух другої моделі більш природнім та наближеним до людини, адже виконується умови плавності руху та розподілу сили, які властиві людському тілу.

Зміщуючи акцент на швидкодію, можна чітко говорити, що зворотна кінематика працює швидше за РСТ, що обумовлено більшою кількістю перевірок самого алгоритму перцептивного управління, та постійним обміном інформації із сенсором, але варто наголосити, що різниця у часі становить 2-3 мілісекунди (варіативність показника обумовлена різними шаблонами точок). Той факт, що швидкодія не несла основну інформативність для даної роботи не означає, що цей показник не має бути врахований. Тож для подальших покращень це варто прийняти до уваги.

За висновком можна стверджувати, що модель перцептивного управління є кращою за модель інверсної кінематики у завдання торканні рукою точок у просторі. РСТ також має точніші показники до еталону, тож модель вважається успішною за побудовою траєкторій. Вона може бути використана надалі у синтезі більш комплексної роботи описаної у третьому розділі даної роботи.

3 ОБЧИСЛЕННЯ ФІЗИЧНИХ ПОКАЗНИКІВ ДЛЯ РУХУ РУКИ

У даному розділі надано математичний та фізичний опис розробленої моделі. Використовуючи матеріали світових дослідження та власні розрахунки, наведені відповідні формули для надання моделі властивостей природнього руху людської руки.

3.1 Опис особливостей руху руки

Згідно з дослідженням швидкість руху пальця руки не є лінійною, а має закон розподілу, наближений до нормального [12]. Траєкторія руху пальця руки, як багато процесів у природі, що йдуть шляхом найменшого опору, буде йти приблизно по прямій, якщо не стоїть завдання обійти перешкоди, які стоять на шляху руки. Якщо розглядати одну й ту саму точку в просторі як мету, то відстань до мети є константним значенням $L = \text{const}$. Досягти цієї мети, значить подолати цю відстань за якийсь проміжок часу. Для натурального руху це може бути 1-2 секунди, а може бути 30 мілісекунд, можна рухати руку також набагато довше. Для кожного зі значень часу t_1, t_2, \dots, t_n , необхідного для досягнення мети (торкання), зміна швидкості v руху пальця руки буде різною (рис. 3.1).

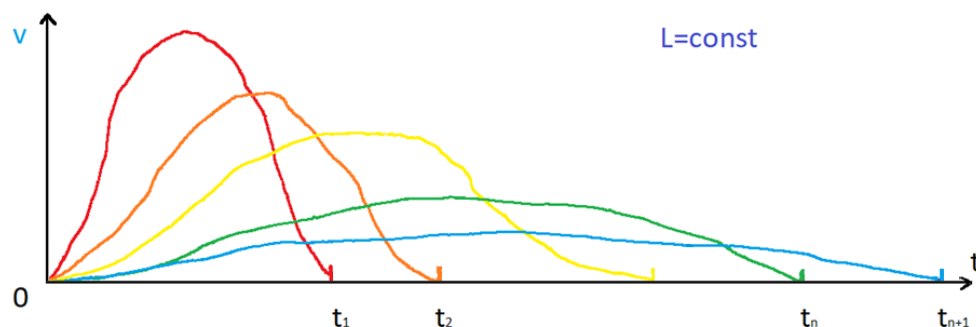


Рисунок 3.1 – Залежність функції швидкості руки від часу

Що менший час досягнення мети, то більша швидкість руху відповідно. Максимальне значення швидкості руху пальця буде приблизно в момент часу $t_n/2$, потім вона зменшуватиметься і в момент часу t_n вона дорівнюватиме 0.

Коли людина торкається якоїсь точки в просторі, вона оцінює очима відстань, куди рухатися, і в міру наближення до цілі сповільнюється, і зі сповільненням відбувається підгін (коригування) напрямку руху для точнішого влучання з першого разу. Це адаптивний процес. При цьому мозок не заміряє фактичної, тобто абсолютної відстані. З боку очей – ця відстань виглядає інакше, меншою, тобто спотворено. Людина використовує своє внутрішнє відчуття, відпрацьоване в процесі навчання управління рукою в просторі.

У даній роботі пропонується не розраховувати кути в суглобах, що безперервно змінюються, не використовувати обернену кінематику або предиктивних (прогнозуючих) обчислень, а використовувати елементи теорії перцептивного управління. Теорія перцептивного управління – теорія, що описує поведінку живого організму моделлю, яка ґрунтується на властивостях контурів управління з негативним зворотним зв'язком [9, 12, 13]. Контур управління підтримує розглянуту змінну на рівні або близько контрольного значення за допомогою впливу її виходів на цю змінну, опосередкованого фізичними властивостями навколишнього середовища. В інженерній теорії управління контрольні значення встановлюються користувачем поза системою. Прикладом є круїзконтроль. У живому організмі контрольні значення для контрольованих перцептивних змінних підтримуються внутрішньо. Рефлекси – це прості приклади низького рівня. Принципи теорії перцептивного управління використовуються сьогодні для надання нових алгоритмічних засад для штучного інтелекту та машинного навчання [5].

Для робота, який має імітувати натуральний рух руки, одним із можливих рішень буде використовувати, наприклад, систему для визначення відносної відстані з позиції його машинного зорового апарату до об'єкта, якого необхідно торкнутися. У цій роботі розглядатиметься варіант, за якого вимірюватиметься реальна (абсолютна) відстань до цільового об'єкта.

Наприклад, вимір може бути виконано за допомогою датчика на кінці пальця робота. Це простіший випадок. Такий робот може бути використаний у реабілітаційному центрі для тренування пацієнта з метою навчання. Робот рухає руку людини багаторазово, навчаючи її цього руху заново, утворюючи тим самим нові нейронні зв'язки в мозку. У цьому випадку важлива траєкторія краю руки людини (пальця). Якщо робот забезпечує рух пальця людини, суглоби і частини руки утворюють необхідні кути і траєкторії. Першу половину шляху $L/2$ робот-рука може пройти грубо за першим виміром без корекції. У точці $L/2$ передбачається зробити новий замір відстані, щоб скорегувати напрямок руху, щоб компенсувати помилку. В ідеальній моделі це було б $L/2$ (рис. 3.2), але в реальній буде відхилення від цього значення, тому необхідно зробити новий вимір L' до цілі (рис. 3.3), де $L' \neq L/2$, і рухатися на дистанцію $L'/2$ (не $(L/2)/2$) та робити наступний вимір, перераховуючи координати наступної точки контролю, і так до мінімальної відстані, наприклад, розміру фаланги або пучка пальця.



Рисунок 3.2 – Відстані проміжних замірів в ідеальній моделі

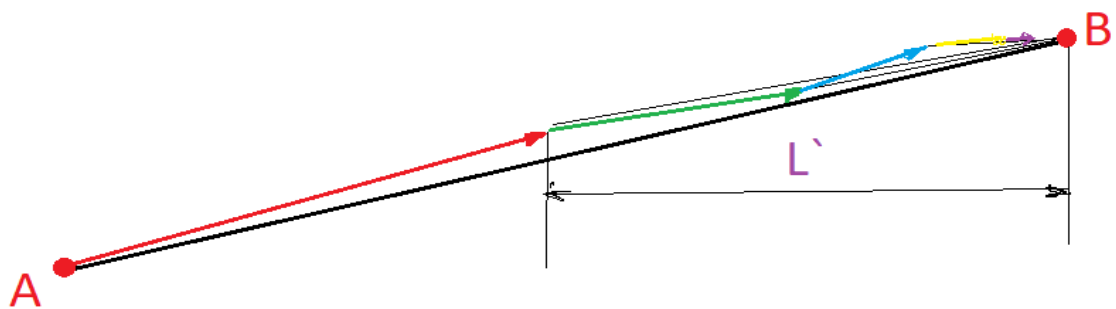


Рисунок 3.3 – Рух із точки А в точку В із корекцією напрямку руху в реальній моделі

Якщо поєднати рисунки 3.1 і 3.2 так, що L і t_n були б в одній точці, отримаємо рис. 3.4, з якого видно, що в момент часу $t_n/2$ палець руки буде приблизно на половині відстані до цілі, при цьому швидкість буде максимальною.

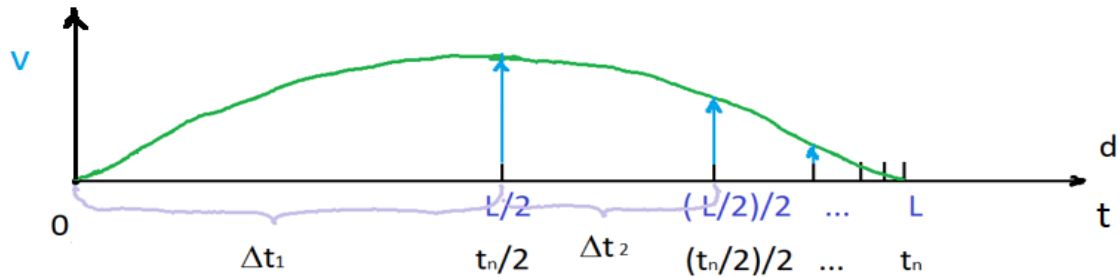


Рисунок 3.4 – Розподіл швидкості руху пальця руки

$$\Delta t_1 = \frac{t_n}{2} \quad (3.1)$$

де Δt_1 – час проходження половини шляху,

t_n – час руху до мети, заданий ззовні.

$$v_1 = \frac{L/2}{\Delta t_1} \quad (3.2)$$

де v_1 – середня швидкість проходження половини шляху або максимальна швидкість у момент часу,

L – відстань до цілі.

$$a_1 = \frac{v_1 - v_0}{\Delta t_1} \quad (3.3)$$

де a_1 – прискорення,

v_0 – швидкість на початок шляху ($t = 0$).

На проміжку Δt_1 прискорення додатне. На наступних проміжках прискорення від'ємне. У загальному вигляді рівняння для матимуть вигляд:

$$\Delta t_i = \frac{\Delta t_{i-1}}{2} \quad (3.4)$$

де Δt_i – час проходження i -го проміжку шляху,

Δt_{i-1} – час проходження попереднього $i-1$ проміжку шляху,

$$v_i = \frac{L'_i/2}{\Delta t_i} \quad (3.5)$$

де v_i – середня швидкість проходження половини шляху або максимальна швидкість у дискретний момент часу i ,

L'_i – заміряний проміжок шляху.

$$a_i = \frac{v_i - v_{i-1}}{\Delta t_i} \quad (3.6)$$

де v_{i-1} – швидкість у попередній момент часу виміру $i-1$,

a_i – прискорення,

v_0 – швидкість на початку шляху ($t = 0$).

3.2 Опис алгоритмічної моделі руху руки

На рис. 3.5 зображено схему керування рухом руки. Вона має ієрархічну структуру. На верхньому рівні ієрархії знаходиться таймер, який ітеративно обчислює Δt_i і забезпечує очікування наступного дискретного моменту часу для виконання виміру відстані та коригування напрямку (df – розмір фаланги пальця).

На наступному рівні ієрархії знаходиться пристрій, що виконує вимірювання відстані до цілі, ініційоване сигналом «виміряти L' ». Цей пристрій передає значення L' таймеру у зворотному зв'язку і на вхід пристрою, що перебуває на нижчому рівні ієрархії, це обчислювач швидкості та прискорення. Цей обчислювач передає значення швидкості та прискорення на вхід пристрою керування двигуном руки. Для переміщення пальця руки в тривимірному просторі необхідно розраховувати координати x , y , z , але це наразі не розглядається.

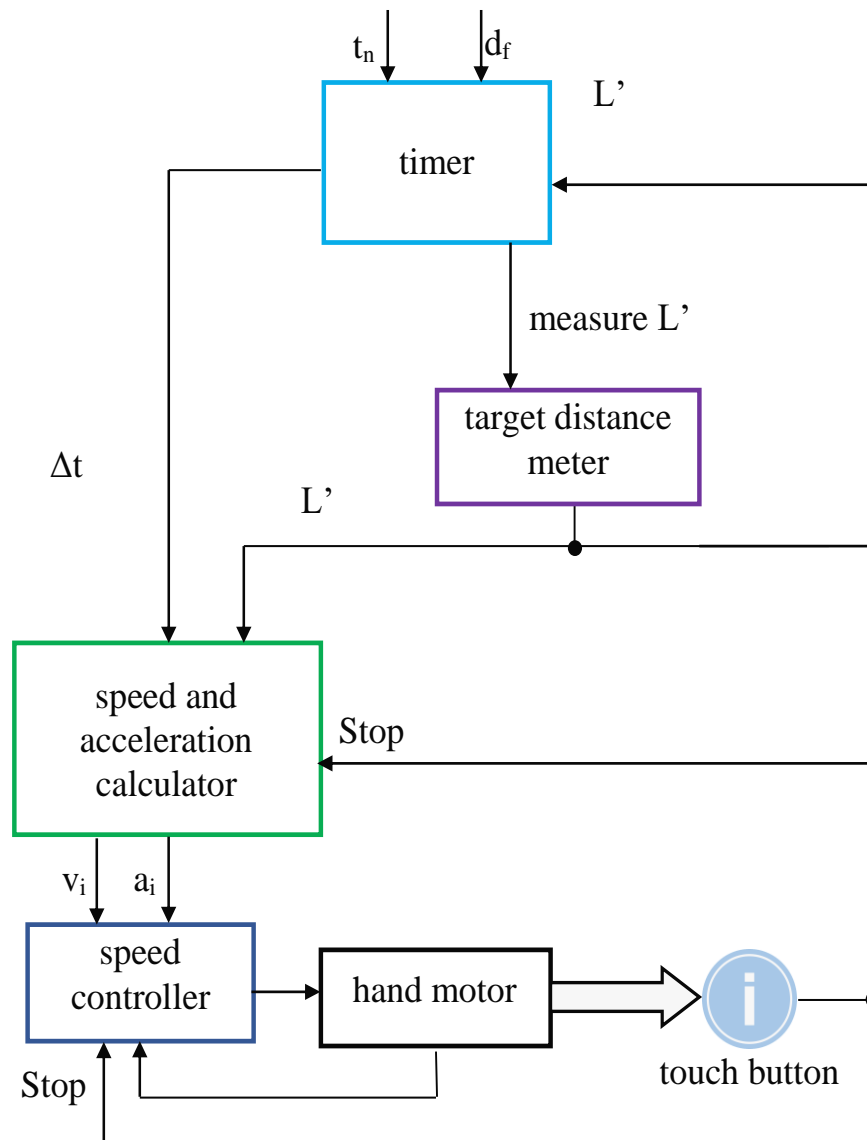


Рисунок 3.5 – Схема управління рухом руки

На рис. 3.6 зображено алгоритм роботи таймера.

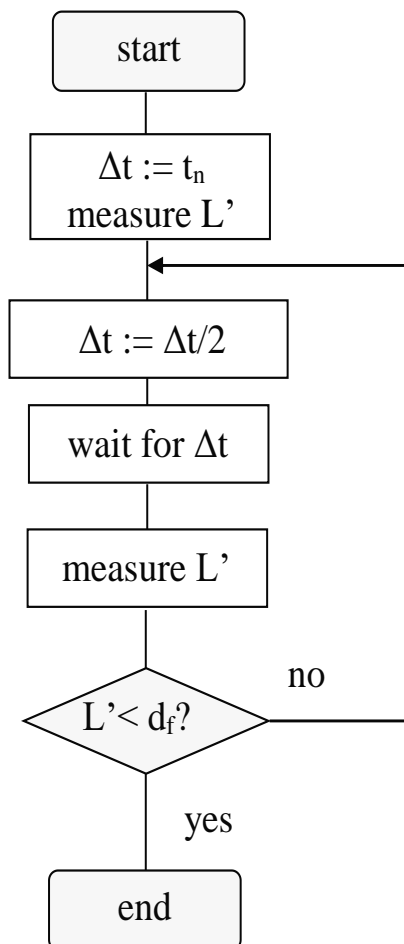


Рисунок 3.6 – Блок-схема алгоритму роботи таймера

Поеднуючи концепції попередніх досліджень та розрахунків, можна сформулювати загальну алгоритмічну схему. На рис. 3.7 зображено алгоритм роботи обчислювача швидкості та прискорення, який може бути реалізовано також програмно або за допомогою керуючого ітераційного автомата.

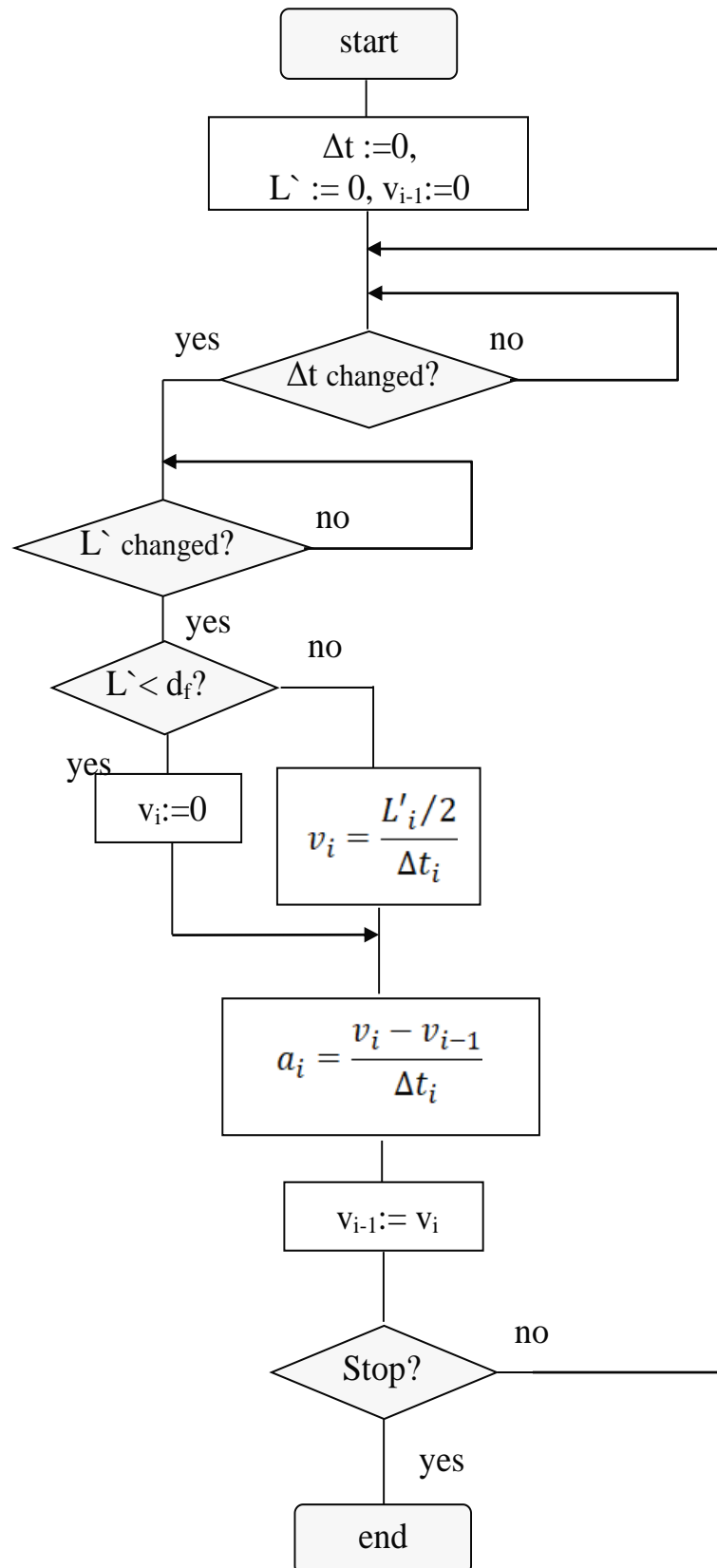


Рисунок 3.7 – Алгоритм роботи обчислювача швидкості та прискорення

Алгоритми керування можуть бути реалізовані як програмно, так і апаратно, залежно від вимог до продуктивності, гнучкості та енергоспоживання. Програмна реалізація, зазвичай, виконується на процесорах загального призначення або мікроконтролерах, де програмне забезпечення створюється за допомогою мов програмування, таких як C або Python, з використанням відповідних бібліотек і драйверів. Такий підхід забезпечує високу гнучкість і можливість оновлення алгоритмів через перепрограмування, що особливо важливо в умовах швидкого розвитку технологій або змін у вимогах до системи.

Апаратна реалізація, у свою чергу, передбачає впровадження алгоритмів безпосередньо у вигляді електронних схем. Це може бути досягнуто через використання мікроконтролерних (embedded) систем, де алгоритм виконується на апаратній платформі, інтегрованій у кінцевий пристрій. Такий підхід часто застосовується в системах, які потребують низького енергоспоживання або виконання алгоритмів у реальному часі.

Альтернативно, алгоритми можуть бути реалізовані у вигляді спеціалізованих керуючих ітераційних автоматів з елементами автомата тимчасового керування, що реалізуються на програмованих логічних інтегральних схемах [13]. У цьому випадку створюється спеціалізована апаратна архітектура, яка дозволяє досягти високої швидкодії та точності роботи. Для впровадження такої системи необхідне використання арифметико-логічного пристрою (АЛП), що є ключовим компонентом, який забезпечує виконання математичних і логічних операцій. Як АЛП може бути використано готовий IP core (інтелектуальний модуль), що зменшує час розробки та забезпечує оптимізацію апаратних ресурсів.

Така архітектура підходить для застосувань, де потрібна максимальна продуктивність, наприклад, у системах автоматичного керування, робототехніці, або телекомунікаціях. У кожному з описаних варіантів підходів вибір реалізації залежить від вимог до системи, обмежень щодо вартості, енергоспоживання, швидкодії та інших специфічних умов експлуатації.

ВИСНОВКИ

Кваліфікаційна робота присвячена створенню моделі торкання рукою точок у просторі з використанням елементів машинного навчання, яка продемонструвала покращені показники руху, а саме стала більш подібною до рухів живої людини.

При виконанні завдання проведено аналіз предметної області теорій роботизованого руху, а також розглянуто основні методи, що використовуються для розробки таких моделей, а саме: метод, заснований на зворотній кінематиці, метод перцептивного управління.

Було створено відеоматеріал із рухом людини, розроблено програму обробки відео та отримано схематичну модель, на основі якої проводився порівняльний аналіз. Розроблено моделі руху на основі інверсної кінематики та перцептивної теорії управління. Виконано порівняння моделей між собою та з еталонним зразком, в результаті чого було виявлено, що РСТ відтворює рухи більш точно ніж ІК. У цьому полягає наукова новизна даної роботи. Згадані результати були опробовані у публікації тез доповіді конференції [14] та стали основою для подальшої роботи над моделлю руху.

Для результируючої моделі (РСТ) розроблена алгоритмічна схема руху руки шляхом опису відповідних показників як швидкість та прискорення, записані рівняння, що розраховують ці показники. По завершенню модель була представлена блок-схемою алгоритму, який описує рух руки та окремих його частин, що беруть участь у роботі моделі.

В результаті проведеного дослідження можна стверджувати, що розширення можливостей існуючих напрацювань руху, подібного до руху людини є перспективним напрямком, особливоу сферах, які потребують саме точного але природнього руху роботизованих рук, як галузь протезування та центри реабілітації людей, що втратили нейроні зв'язки для управління власним тілом.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Denavit, J., Hartenberg, R. S. A kinematic notation for lower-pair mechanisms based on matrices // Journal of Applied Mechanics. Vol., 1. June 1955 pp. 215-221.
2. Kucuk, S. The Inverse Kinematics Solutions of Industrial Robot Manipulators, / S. Kucuk, Z. Bingul, // IEEE Conferance on Mechatronics, Turkey, June 2004, Istanbul, pp. 274-279.
3. D'Souza. Learning inverse kinematics / D'Souza, Aaron, Sethu Vijayakumar, and Stefan Schaal // Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180). Vol. 1. IEEE, 2001, pp. 298-303.
4. Cybernetics: Or Control and Communication in the Animal and the Machine Paris: Hermann & Cie. 1948. 2nd revised ed. 1961, MIT Press, Cambridge, MA.
5. Powers, William T. Living Control Systems III: The fact of control. New Canaan, CT: Benchmark Publications, 2008.
6. Cools, A. R. Brain and Behavior: Hierarchy of Feedback Systems and Control of Input. / In Bateson, P. P. G.; Klopfer, Peter H. (eds.) // Perspectives in Ethology. Boston, MA: Springer US., 1985, pp. 109–168.
7. Mansell, Warren. Control of perception should be operationalized as a fundamental property of the nervous system // Topics in Cognitive Science., 2011, pp. 257–261.
8. Marken, Richard S.; William T., Powers, Levels of intention in behavior in Hershberger, Wayne (ed.) // Volitional Action, Advances in psychology, vol. 62, Amsterdam: Elsevier B.V., 1989, pp. 409–430.
9. Powers, William T. Behavior: The Control of Perception. Wildwood House, 1973.

10. Cziko, Gary. *Without Miracles*. MIT Press, 1995.
11. Monaco, Joseph D.; Hwang, Grace M. *Neurodynamical Computing at the Information Boundaries of Intelligent Systems // Cognitive Computation*. 16 (5): 2022, pp. 1–13.
12. Atkeson Christopher G. *Kinematic Features of Unrestrained Vertical Arm Movements / Christopher G. Atkeson and John M. Hollerbach, Artificial Intelligence Laboratory and Department of Psychology, Massachusetts Institute of Technology. Cambridge, Massachusetts // The Journal of Neuroscience Vol. 5, No. 9, September 1985, pp. 2318-2330.*
13. Pedroni V. A. *Finite state machines in hardware : theory and design (with VHDL and SystemVerilog) Cambridge, MA: MIT Press., 2013. p. 338 p.*
14. Сліпенький В. В. *Модель перцептивного контролю для відтворення природного руху руки людини при торканні точки в просторі / Сліпенький В. В // Тези доповіді VII Міжнародна студентська конференція «Діджиталізація науки як виклик сьогодення» 25 жовтня 2024, Полтава, Україна, 2024. – С. 539-540. [Електронний ресурс] – Режим доступу: <https://archive.liga.science/index.php/conference-proceedings/issue/view/inter-25.10.2024-15.11.2024> p.*