

КВАЛІФІКАЦІЙНА РОБОТА

магістр

Освітній рівень

Метод оцінювання роботи маніпулятора Panda Arm на основі імітаційного

МОДЕЛЮВАННЯ

Назва теми

КвРАКІТР.2023162.01.23.ПЗ

Рівень вищої освіти магістр

Галузь знань 17 «Електроніка, автоматизація та електронні комунікації»

Шифр, назва

Спеціальність 174 «Автоматизація, комп'ютерно-інтегровані технології та
робототехніка»

Шифр, назва

Освітня програма «Автоматизація, комп'ютерно-інтегровані технології та
робототехніка»

Назва

Виконав:

студент ІІ курсу, група АКІТРм-23-1

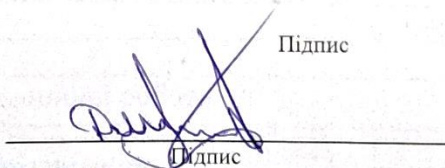


Андрій СЕМЕНИШЕН

Підпис

Ім'я, ПРІЗВИЩЕ

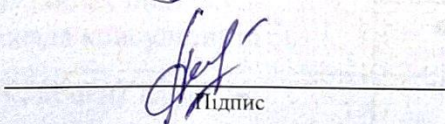
Керівник



Денис МАКАРИШКІН

Ім'я, ПРІЗВИЩЕ

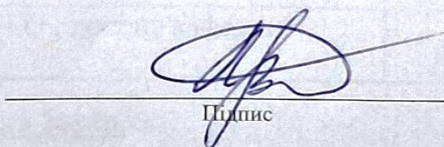
Нормоконтролер



Людмила КОРЕЦЬКА

Ім'я, ПРІЗВИЩЕ

До захисту допускаю:
зав. кафедри АКІТтаР



Валерій МАРТИНЮК

Ім'я, ПРІЗВИЩЕ

« 13 » грудня 2024 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

Рівень вищої освіти другий (магістерський)

Галузь знань 17 – Електроніка, автоматизація та електронні комунікації

Спеціальність 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Освітня програма Автоматизація, комп'ютерно-інтегровані технології та робототехніка

ЗАТВЕРДЖУЮ

Завідувач кафедри АКІТтаР

Валерій МАРТИНЮК

01 вересня 2024р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Семенишену Андрію Леонідовичу

Прізвище, ім'я, по батькові, науковий ступінь, учене звання

1 Тема роботи Метод оцінювання робота маніпулятора Panda Arm на основі імітаційного моделювання

Керівник роботи Макаришкін Денис Анатолійович, к.т.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, учене звання

Затверджено наказом ректора університету від 26.08.2024 р. №60





2 Строк подання студентом роботи на кафедру 02.12.2024р.

3 Вихідні дані до роботи технічні характеристики робота-маніпулятора Panda Arm, методика параболічного змішування, принцип розрахунку енергоспоживання за рекурсивним алгоритмом Ньютона-Ейлера, задачі функціонування робота-маніпулятора в межах промислового підприємства при зборі об'єктів з коробок та їх переміщення на конвеєр

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити) Проблематика моделювання роботи роботів-маніпуляторів типу «Рука», Розробка методу оцінювання функціонування моделі робота-маніпулятора, Математична модель проведення оцінки робота-маніпулятора, Дослідження результатів розробленого методу оцінювання

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень) презентаційні матеріали (слайди)

6 Консультанти розділів кваліфікаційної роботи


Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Антиплагіат	Федула М.В., доцент кафедри АКІТтаР		
Нормоконтроль	Корецька Л.О., доцент кафедри АКІТтаР		

7 Дата видачі завдання 01 вересня 2024р.

КАЛЕНДАРНИЙ ПЛАН

Назва розділу кваліфікаційної роботи	Строк виконання	Примітка
1. Вступ	15.09.2024 р.	Виконано
2. Проблематика моделювання роботи роботів-маніпуляторів типу «Рука»	30.09.2024 р.	Виконано
3. Розробка методу оцінювання функціонування моделі робота-маніпулятора	16.10.2024 р.	Виконано
4. Математична модель проведення оцінки робота-маніпулятора	1.11.2024 р.	Виконано
5. Дослідження результатів розробленого методу оцінювання	12.11.2024 р.	Виконано
6. Висновки	14.11.2024 р.	Виконано
7. Написання статті	25.11.2024р.	Виконано
8. Оформлення пояснювальної записки до КРМ	01.12.2024р.	Виконано
9. Оформлення презентаційних матеріалів	01.12.2024р.	виконано

Студент


Підпис

Андрій СЕМЕНИШЕН
Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи


Підпис

Денис МАКАРИШКІН
Ім'я, ПРІЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Метод оцінювання роботи маніпулятора Panda Arm на основі імітаційного моделювання».

Автор роботи: Семенишен Андрій Леонідович

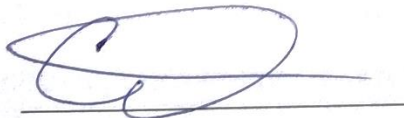
Керівник роботи: Макаришкін Денис Анатолійович, к.т.н., доцент

Пояснювальна записка: 94 с., 46 рис., 3 табл., 2 дод., 75джерел.

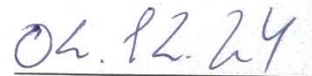
Графічна частина: 15 презентаційних слайдів.

РОБОТ-МАНІПУЛЯТОР, FRANKA ЕМІКА PANDA ARM, ПЕРЕТВОРЕННЯ ЛАГРАНЖА, РЕКУРСИВНИЙ АЛГОРИТМ НЬЮТОНА-ЕЙЛЕРА, МЕТОД ПАРАБОЛІЧНОГО ЗМІШУВАННЯ, МІНІМАЛЬНЕ ЕНЕРГОСПОЖИВАННЯ, КОНВЕЄРНА ЛІНІЯ, ЦИФРОВИЙ ДВІЙНИК, ВІРТУАЛЬНИЙ ПРОТОТИП, ІМІТАЦІЙНА МОДЕЛЬ

Мета роботи: розробка методу оцінювання функціональних можливостей роботу-маніпулятора Panda Arm на основі імітаційного моделювання. Розроблено математичну модель та модуль проведення оцінки функціональності роботу-маніпулятора Panda Arm на основі імітаційного моделювання. Наведено результати моделювання траєкторій руху Panda Arm при підйомі об'єктів з коробок та їх подальшого переміщення на конвеєр. Розглянуто оптимальні траєкторії виконання технологічного процесу при мінімальному часі виконання операції та мінімальному енергоспоживанні.



Підпис студента



Дата

ЗМІСТ

	с.
ВСТУП	4
1 ПРОБЛЕМАТИКА МОДЕЛЮВАННЯ РОБОТИ РОБОТІВ-МАНІПУЛЯТОРІВ ТИПУ «РУКА».....	10
1.1 Огляд та аналіз існуючих підходів щодо створення віртуальних прототипів	10
1.2 Особливості конструкції робота-маніпулятора Panda Arm.....	13
1.3 Постановка задач та запропоновані підходи до моделювання роботи роботів-маніпуляторів	16
1.4 Середовища розробки методу оцінки функціональності робота- маніпулятора Panda Arm.....	17
1.5 Висновки до першого розділу	23
2 РОЗРОБКА МЕТОДУ ОЦІНЮВАННЯ ФУНКЦІОНУВАННЯ МОДЕЛІ РОБОТА-МАНІПУЛЯТОРА.....	24
2.1 Архітектура фреймворку для проведення оцінки функціональності робота-маніпулятора.....	24
2.2 Визначення простору оцінки в рамках запропонованого методу	28
2.3 Принцип виконання та оцінки функціональності робота-маніпулятора..	31
2.4 Процедура візуалізації хмари точок із оптимальними параметрами роботи робота-маніпулятора.....	34
2.5 Висновки до другого розділу.....	37
3 МАТЕМАТИЧНА МОДЕЛЬ ПРОВЕДЕННЯ ОЦІНКИ РОБОТА- МАНІПУЛЯТОРА	39
3.1 Імітаційна модель Panda Arm	39
3.2 Оцінка траєкторії руху Panda Arm.....	41
3.2.1 Метод параболічного змішування	42

3.2.2 Оцінка споживання електричної енергії.....	46
3.2.3 Налаштування оцінки за математичною моделлю.....	50
3.3 Підйом об'єктів за допомогою Panda Arm.....	53
3.4 Висновки до третього розділу	58
4 ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ РОЗРОБЛЕНОГО МЕТОДУ ОЦІНЮВАННЯ РОБОТА МАНІПУЛЯТОРА.....	59
4.1 Встановлення залежностей між параметрами руху робота та енергоспоживанням	59
4.2 Оцінка підйому об'єктів із коробок.....	69
4.3 Аналіз параметрів функціонування робота-маніпулятора Panda Arm, отриманих в рамках методу	76
4.3.1 Оптимальні параметри робота-маніпулятора для мінімізації енергоспоживання.....	76
4.3.2 Оптимальні параметри робота-маніпулятора для підйому об'єктів..	80
4.4 Висновки до четвертого розділу	83
ВИСНОВКИ	84
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	87
ДОДАТКИ	95

ВСТУП

Мехатронні системи відносяться до технічних систем, які забезпечують виконання виробничих процесів, що засновані на комплексному використанні механічних, електричних та інформаційних технологій [54]. По своїй суті вони відносяться до сімейства всіх промислових процесів і продуктів, в яких керування, електроніка та механіка інтегровані в синергетичний спосіб і взаємодіють між собою. Така взаємодія забезпечує загальну продуктивність системи.

На ринку продуктів спостерігається тенденція до зростання рівня автоматизації технологічних процесів виробництва продукції наряду із запровадженням робототехнічного обладнання у виробничий процес. Тенденція на попит на роботизоване обладнання характеризується як швидкозмінний процес, якому властиві коливання цін, глобальна економічна конкуренція та майже щоденне оновлення та поява нових технологій.

По вищеперерахованим причинам відбувається зростання вимог до точності роботи мехатронних систем, а також ряду факторів, до яких відносять чіткий дизайн, якість, короткі терміни виготовлення продукції, тощо. З метою забезпечення цих показників на достатньому рівні фахівці з електроніки, механіки та управління запровадили тенденцію до інтеграції моделювання та симуляції функціонування таких систем в процес їх розробки [70].

Завдяки таким тенденціям протягом останніх десятиліть більшого поширення набула концепція віртуального введення робототехнічного обладнання в експлуатацію та віртуального прототипування такого обладнання. Ідея таких віртуальних підходів полягає у можливості тестування електронних та механічних конструктивних рішень наряду із керуючим програмним забезпеченням ще до початку або паралельно під час процесу фізичної розробки продукту. Завдяки такому підходу розробники робототехнічних систем можуть перевірити системи на елемент дотримання проектних специфікацій, а також провести всебічне дослідження варіантів проектування таких систем. До того ж, прототипування та віртуальне введення в експлуатацію дозволяють отримати

більш детальне уявлення про продуктивність системи та якість її виготовлення. В результаті такого віртуального прототипування та перевірки функціональності можна зменшити витрати на виробництво, а також час виходу продукції на ринок.

Слід зазначити, що проведення оцінки та проектування мехатронних систем в середовищі, заснованому на моделюванні, дозволяє виробництвам провести вдосконалення дизайну продукту, а також підвищити гнучкість виробництва та пришвидшити реакцію на зміну попиту на ринку продукції [24].

Для виконання віртуального тестування ефективності роботи електронних та механічних конструкцій необхідно розробити високоточні та складні віртуальні моделі мехатронної системи. Такі моделі часто називають цифровими двійниками. Цифровий двійник – це формальне цифрове представлення процесу чи системи, який фіксує поведінку та ознаки цього об'єкта, які підходять для зберігання, зв'язку, інтерпретації чи обробки в певному контексті [67, 73].

Враховуючи вищесказане, цифровий двійник можна поділити на три основні компоненти: фізичний продукт, віртуальну модель та дані, якими вони обмінюються. Однак цифровий двійник не обов'язково має фізичний аналог, а імітаційні моделі можуть використовувати на ранніх стадіях інженерного процесу, коли система, що розглядається, ще не існує фізично.

Використання цифрових двійників компаніями призводить до зниження витрати, подальшого збільшення об'єму продажів продукції та скорочення часу виходу на ринок. Враховуючи поширення використання таких цифрових двійників, **актуальною** стає задача розробки та запровадження відповідної методики оцінювання віртуальних прототипів роботів-маніпуляторів на основі імітаційного моделювання.

Метою роботи є розробка методу оцінювання функціональних можливостей роботу-маніпулятора Panda Arm на основі імітаційного моделювання.

Досягнення поставленої мети можна досягнути розв'язавши наступні **задачі**:

1. Проведення вибору критеріїв оцінювання роботи робота-маніпулятора.
2. Оцінка результатів, отриманих у віртуальній моделі.

3. Виокремлення оптимальних результатів траєкторії руху Panda Arm.

4. Виокремлення оптимальних результатів захвату об'єкту роботом-маніпулятором Panda Arm.

5. Виокремлення оптимальних параметрів функціонування робота-маніпулятора з точки зору мінімального споживання енергії.

Методи досліджень.

Використовувались основні положення загальної теорії систем, системного аналізу (ієрархічності, декомпозиції та ін.), теорії моделювання процесів. Внаслідок проведення моделювання процесу рухом роботом-маніпулятором Panda Arm із вантажем; використано теоретико-множинні підходи, алгебру систем, апарат модельно-орієнтованих підходів, методи концептуального та віртуального моделювання, принципи побудови баз знань та формування логічного висновку, евристичні оцінки.

Об'єкт дослідження – процес керування роботом-маніпулятором Panda Arm.

Предмет дослідження – метод оцінювання функціонування віртуальної моделі робота-маніпулятора Panda Arm.

Наукова новизна отриманих результатів: розроблено метод оцінки ефективності роботи віртуальної моделі робота-маніпулятора Panda Arm та виокремлено оптимальні параметри його функціонування.

Практична значимість отриманих результатів: запропоновані підходи можуть використовуватись на виробництвах та промисловостях, які використовують робот-маніпулятор Panda Arm в технологічному процесі виготовлення продукції.

Апробація результатів дослідження: опубліковано наукову статтю «Розрахунок параметрів цифрового двійника робота-маніпулятора Panda Arm» в журналі «Вимірювальна та обчислювальна техніка в технологічних процесах» №4 2024 року, який входить до фахових журналів категорії Б зі спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка».

Дипломна робота складається із наступних елементів: вступ, чотири розділи, висновки до кожного розділу, загальні висновки, список використаних

джерел, додатки. Загальний обсяг роботи складає 94 сторінки, у тому числі: 46 рисунків, 3 таблиці, список використаних джерел містить 75 найменувань.

В першому розділі було проведено огляд та аналіз підходів, які застосовуються при створенні віртуальних прототипів. Наведені характеристики віртуальних прототипів та переваги їх використання у порівнянні із фізичними прототипами.

В якості прикладу робота-маніпулятора, для якого буде проводитись оцінки функціональності його роботи, було обрано робот-маніпулятор типу «рука» Panda Arm. В розділі наведені технічні характеристики цього робота та його сфери застосування. Також у розділі було поставлено задачі та запропоновані підходи до проведення оцінки функціональності роботів-маніпуляторів.

В розділі також наведено короткий опис середовищ для розробки запропонованої оцінки функціональності робота-маніпулятора Panda Arm.

В другому розділі виконано огляд функціональних можливостей методу оцінки робота-маніпулятора на основі імітаційного моделювання як системи вцілому, так і як кожного її окремого компоненту. До цих компонентів відносяться архітектура фреймворку, що розробляється, в якому описуються незалежні компоненти, файли та дані, що оцінюються між цими компонентами.

Також в розділі ілюструється, яким чином можна визначити простір оцінки за допомогою редактора Unity. У розділі пояснюється концепція коду, що забезпечує функціональність роботи методу оцінки робота-маніпулятора. Крім того, у розділі підсумовується, яким чином проводиться оцінка симуляції робота-маніпулятора за допомогою коду, розробленого у .NET.

Наостанок, викладено ідею третього компоненту методу оцінки, а саме фреймворку, який зчитує дані з текстового файлу і в подальшому створює візуалізації результатів проведених у системі обчислень.

У третьому розділі наведено два конкретні приклади використання розробленого методу оцінки функціональності робота-маніпулятора, які демонструють можливість застосування методу.

Основна увага при цьому приділяється створенню імітаційної моделі роботи Panda Arm. Імітаційна модель представляє собою два додатки, в яких

проводиться моделювання роботи робота-маніпулятора Panda Arm. Крім того, в розділі показано параметри оцінки та об'єктивні значення, які використовувались для оцінки функціональності.

В розділі показано, що робот-маніпулятор Panda Arm можна перетворити на цифрового двійника в площині Unity, що здатен відобразити реального робота на високому за точністю рівні. Також в розділі наведено деталізацію використання розробленого додатку для оцінки траєкторії робота-маніпулятора та автоматизованого збору предметів, на прикладі збору сміття.

Також в розділі наведено математичну модель опису траєкторії руху робота із врахуванням його кінематичних та динамічних параметрів на основі методу параболічного змішування. Крім того, розглянута математична модель оцінки енергоефективності роботи Panda Arm, при якій рух по траєкторії буде виконуватись із максимально можливою швидкістю та мінімальною витратою енергії.

В четвертому розділі наведені результати моделювання траєкторій руху Panda Arm при підйомі об'єктів з коробок та їх подальшого переміщення на конвеєр. Ці траєкторії руху моделювались для двох можливих геометрій коробок. Створені траєкторії були проаналізовані з точки зору можливих зіткнень із іншими об'єктами. Такі хибні траєкторії були відсіяні та не приймали участь у подальшому аналізі.

Отримані траєкторії руху були проаналізовані з точки зору споживання енергії. В результаті було встановлено, що:

- зменшення довжини лінійної траєкторії руху робота призводить до збільшення енергоспоживання;
- зменшення вигину суглобів при руху по траєкторії призводить до зменшення енергоспоживання;
- найменше енергії потребує двигун першого суглоба;
- основну частку серед спожитої енергії має саме сумарна довжина траєкторії руху, пройденої всіма суглобами робота-маніпулятора.

Крім того, при аналізі процесу підйому об'єктів із коробок встановлено наступне:

– довжина лінійної траєкторії руху робота-маніпулятора не значним чином впливає на підйом об'єктів, тому її можна розглядати лише в розрізі енергоспоживання;

– визначені параметри, при яких досягається мінімальний час підйому об'єктів із коробок, а саме $x_p = -0,1$ м, $y_p = 0,1$ м, $z_p = 0$ м для коробок із складною геометрією та $x_p = 0$ м, $y_p = 0,05$ м, $z_p = -0,05$ м для коробок із простою геометрією.

У загальних висновках по кваліфікаційній роботі наведено отримані результати роботи на тему «Метод оцінювання робота-маніпулятора Panda Arm на основі імітаційного моделювання».

1 ПРОБЛЕМАТИКА МОДЕЛЮВАННЯ РОБОТИ РОБОТІВ-МАНІПУЛЯТОРІВ ТИПУ «РУКА»

1.1 Огляд та аналіз існуючих підходів щодо створення віртуальних прототипів

Створення віртуальних прототипів та проєктування роботів-маніпуляторів на основі імітаційного моделювання є одним із поширених методів в сучасній промисловості. Цей метод передбачає створення сценарію розробки продукту, який складається із ряду кроків, що розміщуються у V-подібній формі. По цій причині часто можна зустріти термін V-модель системної інженерії.

До цієї моделі входять наступні кроки:

1. Аналіз системних вимог.
2. Проєктування системи, яке включає в себе аналіз та моделювання моделей, які безперервно підтримують весь процес подальшої розробки.
3. Проєктування для конкретної предметної області, наприклад, для електротехніки, машинобудування, інформаційних технологій.
4. Системна інтеграція. На цьому етапі проводиться інтерактивна перевірка та валідація системних вимог наряду із забезпеченням властивостей системи.
5. Введення продукту в експлуатацію.

В процесі вищезазначених етапів розробки може виникати ряд проблем, які вимагають проведення перегляду проєктних специфікацій, властивостей самої системи та повторення етапів від першого до останнього. Діджиталізація та розвиток комп'ютерних технологій призводять до підвищення рівня інтеграції віртуальних прототипів в процес розробки продукції, що доповнює або повністю замінює фізичні прототипи. Однією із основних переваг віртуального прототипу у порівнянні із фізичним прототипом є значне збільшення рівня гнучкості проєктування та виробництва наряду із скороченням витрат на створення продукції та проведення оцінки прототипу [33]. За рахунок цього значно зменшується час виведення продукції, що прототипується, на ринок збуту. Крім того, створення віртуальних прототипів дозволяє зменшити кількість ітерацій, які

потребує процес розробки, а також має на меті виявити помилки в процесі розробки продукції.

Класифікація віртуальних прототипів проводиться на основі такого основного показника, як мета моделювання в контексті циклу розробки продукції. З такої точки зору виділяють 5 класів віртуальних прототипів:

- візуалізація – використовується для вивчення форми та зовнішнього вигляду продукції;

- механічна підгонка – використовується для оцінки стану збірки продукції як віртуального прототипу та відповідності його механічних компонентів визначеним вимогам та стандартам;

- тестування продуктивності та функціональності – по суті представляє собою ключову роль у створення віртуального прототипу, при якому проводиться перевірка властивостей продукту та можливість виконання ним функцій [35];

- оцінка виробництва – такі прототипу надають експертам з різних інженерних сфер можливість проводити дослідження поточного стану продукту на створювати подальший дизайн;

- аналіз людського фактору – дозволяють провести оцінку ергономічності та безпеки використання розробленого віртуального прототипу продукту, а також перевірити, наскільки реалізовано рівень ергономічних сумісностей в системі «людина-техніка-довкілля» [36, 37].

Наступним підходом є використання спеціалізованих програмних середовищ для моделювання мехатронних систем, в яких використовуються цифрові двійники. Вибір програмного забезпечення залежить від сфери застосування віртуального прототипу, типу системи, що моделюється, а також від цілей моделювання та його контексту (електронний, механічний, керування, тощо).

Імітаційне моделювання систем з дискретними подіями [5, 49, 69] – це своєрідна технологія, в якій моделювання відбувається шляхом обробки послідовності подій. При такому підході до моделювання модель системи може

бути представлена у вигляді скінченного автомату, списку поточних подій та подій, що відбудуться, а також специфікаціями часових параметрів логіки та подій.

В комп'ютерних мережах поширеною технологією є моделювання систем з дискретними подіями. Найяскравішими прикладами є такі програмні засоби, як ns-3 [18] та OMNet++ [19], які реалізують методику виконання моделювання системи із дискретними подіями.

В якості середовища розробки моделей на основі рівнянь та/або сигналів використовується MATLAB/Simulink [38, 51]. В цьому програмному засобі використовується мова програмування MATLAB, а також реалізується моделювання та імітація процесів у вигляді графічних блоків. В розрізі мехатронних систем цей програмний засіб використовує методику сигнально-орієнтованого моделювання для одновимірного моделювання віртуальних прототипів та моделей систем керування. При цьому слід відзначити, що в MATLAB є ряд незручностей при моделюванні тривимірної механіки та виявленні зіткнень при функціонуванні механізмів [47, 52].

Також використовуються ігрові підходи, наприклад Unreal Engine [9] та Unity [55]. Початково ці платформи використовувались лише для розробки комп'ютерних ігор, але в сучасних підходах спостерігається тенденція до розширення сфери використання цих платформ. Вони використовуються в якості середовища для розробки моделей та для методів тривимірної симуляції процесів в режимі реального часу в умовах фізично реалістичного середовища. Тобто, вони можуть використовуватись як середовища для моделювання тривимірної механіки.

Від компанії Dassault Systems для імітаційного моделювання можна використовувати такі програмні рішення, як CATIA [50, 74] та DELMIA [8]. В цих програмних середовищах у розробників є можливість виконувати багатодоменне моделювання, до того ж ці системи підтримують моделювання різноманітних віртуальних прототипів.

Комерційні пакети, такі як ADAMS [2], HyperWorks від Altair [3] та MATLAB Simscape Multibody [25], є чудовими інструментами для моделювання

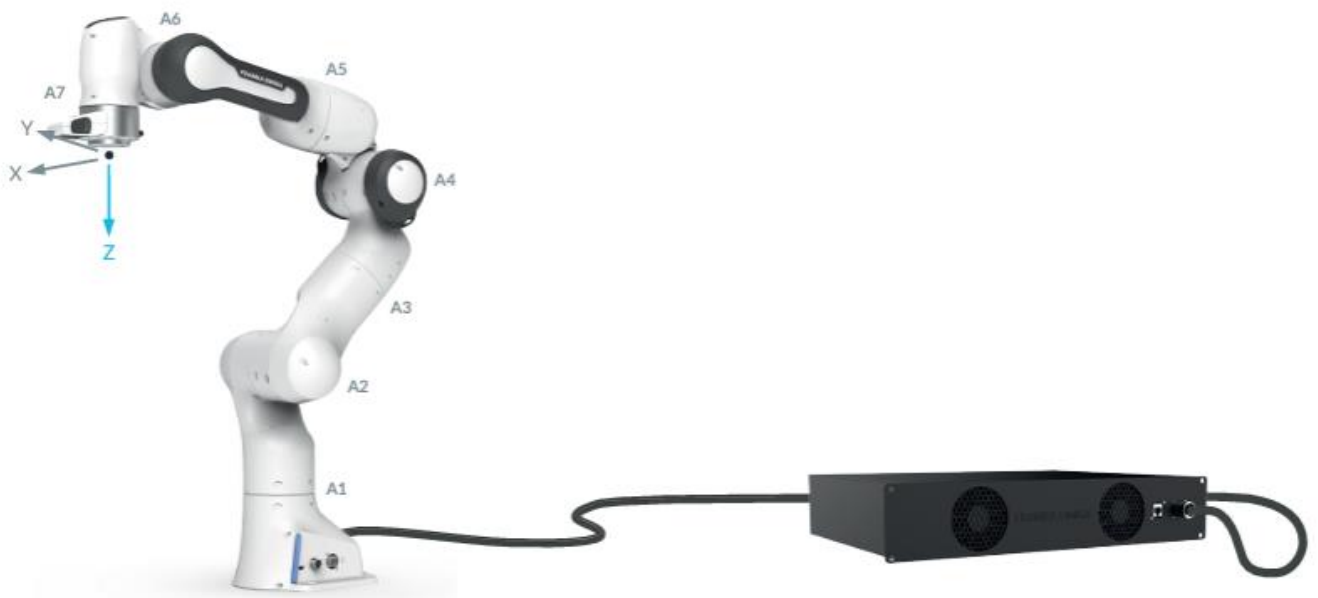
та імітації механіки багатотілесних систем, починаючи від визначення механічних розмірів і закінчуючи дослідженнями вібрацій.

Багатокритеріальне прийняття рішень (БПР), яке має ще іншу назву – багатокритеріальний аналіз рішень (БАЗР) використовується у випадках, коли на рішення впливає ряд суперечливих критеріїв [6]. БПР набуло широкого поширення в останні декілька років та вважається одним із найвідоміших підходів до прийняття рішень.

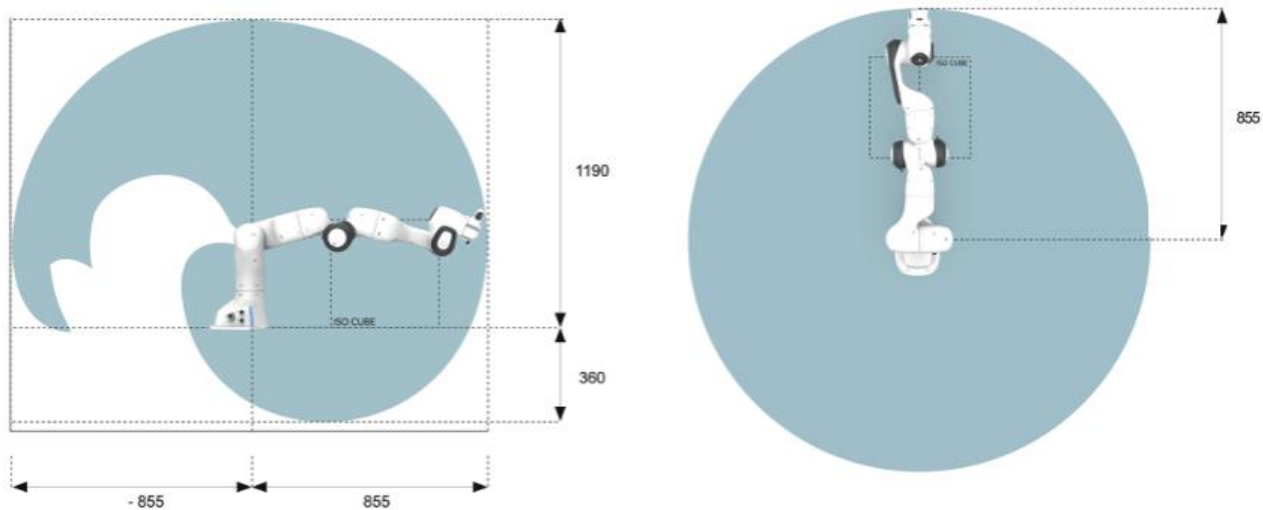
1.2 Особливості конструкції робота-маніпулятора Panda Arm

Franka Emika Panda (рис. 1.1) – це 7-осьовий робот-маніпулятор, який працює з навантаженням до 3 кг на відстані до 850 мм. Повторюваність робота складає 0,1 мм, при цьому вага становить 18 кг [11, 34]. Найчастішими сферами застосування Panda Arm є зварювання, зовнішній інструмент та дозована подача матеріалу.

Вартість робота-маніпулятора Franka Emika Panda Arm в Україні становить 868 370 гривень [48].



a)



б)

Рисунок 1.1 – Panda Arm:

а – зовнішній вигляд; б – простір роботи (вигляд зліва та зверху)

Panda Arm був розроблений німецькою компанією Franka Emika у 2016 році, яка власне спеціалізується у створенні роботизованого обладнання, що співпарцюють з людиною [12, 20]. Він відноситься до роботів-маніпуляторів типу «рука», які призначені для виконання різних завдань. В конструкцію робота входить датчик сили, який відповідає за автоматичну зупинку у випадку, якщо у робочу зону робота потрапляє щось неочікуване.

Технічні характеристики Panda Arm наведені у табл. 1.1.

Таблиця 1.1 – Технічні характеристики Panda Arm [7, 13]

Характеристика	Значення
<i>1</i>	<i>2</i>
Ступені свободи	7
Максимальна досяжність	855 мм
Навантаження	3 кг
Очікуваний термін служби	20 000 годин
Вага	17,8 кг
Робоча відносна вологість повітря	20-80 %

Продовження таблиці 1.1

1	2
Робочі температури навколишнього середовища	15-25 °С (ідеальні) 5-45 °С (розширені)
Границі розташування вузлів	A2: -101°/101° A1, A3, A5, A7: -166°/166° A6: -1°/215° A4: -176°/-4°
Енергоспоживання	80 Вт
Енергоживлення	100-240 В
Повторюваність позиціонування	<+/- 0.1 мм (відповідно до ISO 9283)
Обмеження швидкості по вузлам	A1, A2, A3, A4: 150°/с A5, A6, A7: 180°/с
Відхилення траєкторії	<+/- 1.25 мм
Регульована жорсткість обертання	0 – 300 Нм/рад
Регульована жорсткість при обертанні	0 – 3000 Н/м
Найгірший час реакції на зіткнення	<100 мс
Час виявлення зіткнення	<2 мс
Діапазон крутного моменту [Нм]	
M _x	-70 – 70
M _y	-16 – 12
M _z	-12 - 12
Діапазон зусиль [Н]	
F _x	-150 – 115
F _y	-275 – 275
F _z	-115 - 155
Мінімальне контрольоване зусилля (F _z)	0.05 Н
Відносна точність зусилля	0,8 Н

1.3 Постановка задач та запропоновані підходи до моделювання роботи роботів-маніпуляторів

Основна мета роботи полягає у розробці та впровадженні незалежної системи оцінювання, яка була б здатна проводити оцінку віртуальних прототипів мехатронних систем. Запланована система оцінювання повинна мати вигляд набору автономних інструментів, які б дозволили користувачу спростити процес проведення оцінки. Для спрощення сприйняття статистичної інформації, отриманої в результаті проведення оцінки, буде використовуватись так звана хмара точок, яка дозволить користувачеві отримати тривимірне сприйняття статистичних даних.

В якості платформи для створення фреймворку проведення оцінки була обрана Unity у поєднанні із функціональністю мови програмування C#. Крім того, в роботі планується здійснити розпаралелювання коду C# з метою отримання можливості виконання декількох симуляцій паралельно.

Перший сценарій буде відтворювати процес збору речей роботом, тобто його руху за різноманітними траєкторіями. На цьому етапі система оцінювання повинна визначати оптимальні геометричні та динамічні параметри руху робота-маніпулятора.

Другий сценарій призначений для визначення оптимального положення руки робота-маніпулятора в межах його використання на промисловому підприємстві при зборі матеріалу.

Вищезазначену мету кваліфікаційної роботи можна досягти, шляхом виконання ряду окремих задач, які тісно пов'язані між собою, а саме:

- створення повнофункціональної версії системи оцінювання, яка буде містити основні компоненти для проведення оцінки роботи мехатронної системи на прикладі Panda Arm;
- моделювання сценаріїв та представлення симуляції їх роботи в Unity;
- розробка додатків для функціонування системи проведення оцінки робота-маніпулятора;

– інтеграція операційної системи роботів-маніпуляторів (ROS) в архітектуру системи оцінювання робота-маніпулятора.

Основні компоненти функціонування системи оцінювання розроблялись із застосуванням підходу «зверху вниз». Цей підхід передбачає спочатку формулювання загальної концепції та ідеї оцінювання робота-маніпулятора в симуляції. В подальшому ця концепція поділяється на ряд окремих компонентів шляхом виокремлення незалежних функцій. Завдяки такому підходу можна окреслити функціональність кожної окремої частини мехатронної системи. Фактично порядок, в якому компоненти системи повинні реалізовуватись, буде відповідати раніше визначеній послідовності кроків, які необхідні для проведення оцінки імітаційної моделі мехатронної системи.

1.4 Середовища розробки методу оцінки функціональності робота-маніпулятора Panda Arm

В рамках роботи та виконання поставлених задач в кваліфікаційній роботі використовувались наступні середовища розробки:

- операційна система для розробки: Windows 10;
- робота з версіями Python: Anaconda 4.8.3;
- C# та .NET IDE: Microsoft Visual Studio Community 2019, 16.6.0;
- середовище розробки Python: Spyder 4.1.3.

Система оцінювання робота-маніпулятора базується на декількох середовищах розробки, інструментальних засобів та бібліотек коду. Основою фреймворку оцінки роботи стало три програмні середовища та одна бібліотека:

1. Unity;
2. Microsoft.NET;
3. Python;
4. ROS бібліотека.

Unity (рис. 1.2) - це програмне середовище, в якому можна створювати тривимірні та двовимірні симуляції, а також віртуальну та доповнену реальність.

Близько 50% всіх мобільних, консольних та комп'ютерних ігор працюють на Unity, тому кількість пристроїв, що використовують цю платформу, сягає приблизно 3 мільярдів.

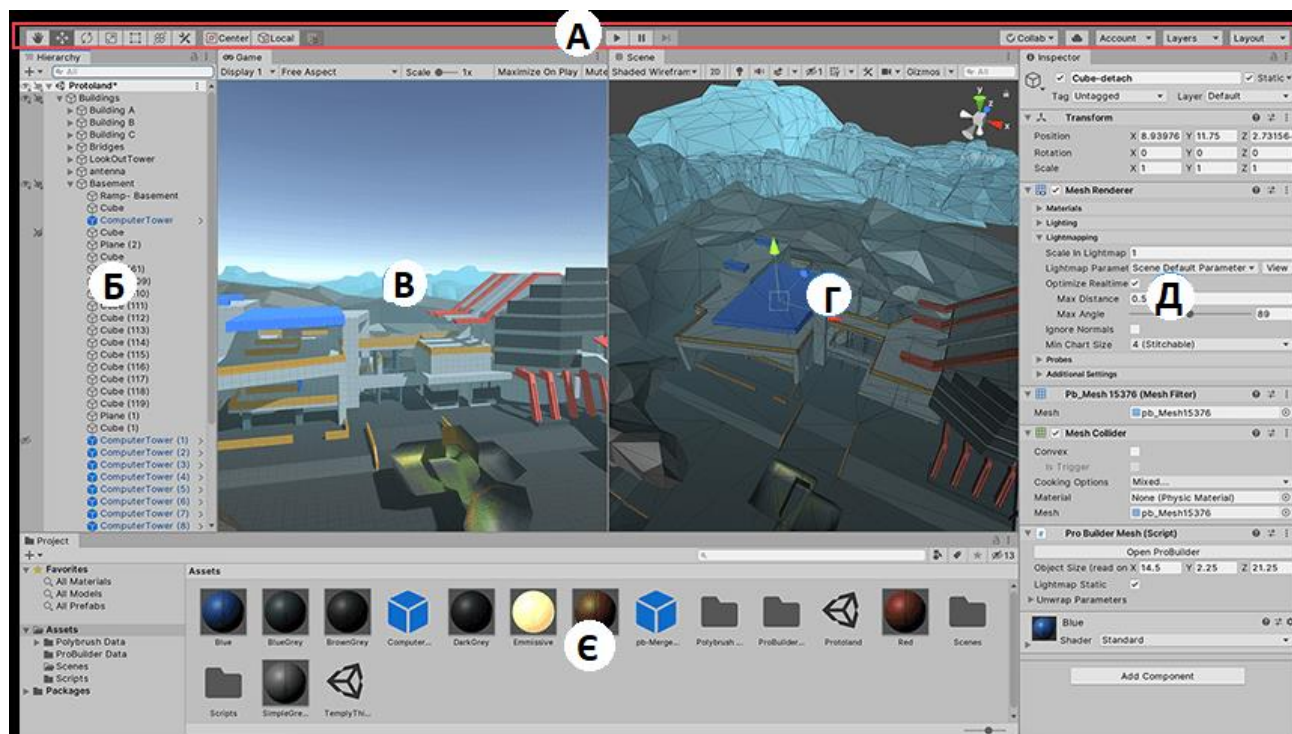


Рисунок 1.2 - Вікно редактора Unity

(а – панель інструментів; б – вікно ієрархії; в – вікно гри; г - вигляд сцени; д – вікно інспектора; е - вікно проекту)

Проект в Unity поділяється на рівні, які ще називають сценами. Ці рівні представляють собою окремі файли, які містять біти із набором налаштування, сценаріїв та об'єктів. Сцени при цьому можуть містити як об'єкти моделі, так і порожні об'єкти. Порожні об'єкти не мають моделі, але задають поведінку інших об'єктів. До об'єктів можна застосовувати скрипти, обертати, розташовувати, масштабувати, тощо. Різні моделі можуть об'єднуватись у своєрідні набори, що забезпечує швидкий доступ до них [31].

Однією із основних причин вибору платформи Unity стала його можливість підтримки фізики твердих тіл, тканин, фізику типу Ragdoll. В редакторі Unity наявна можливість успадкування об'єктів, при цьому дочірні об'єкти будуть повторювати зміни позиції, масштабу, повороту батьківського об'єкта.

Також платформа Unity підтримує міпмапінг, стиснення текстур і різні налаштування роздільності екрана для кожної платформи. Крім того платформа забезпечує паралакс-мапінг, бамп-мапінг, затінення навколишнього світла у екранному просторі, мапінг відображень, рендер у текстуру, динамічні тіні за картами тіней та повноекранні ефекти обробки зображення (глибина чіткості, зернистість, розмиття в русі, ореол навколо джерел світла, відблиски віртуальних лінз, тощо) [29].

Рендеринг зображень у Unity відбувається через віртуальну камеру огляду. Графічний рушій використовує OpenGL (на операційних системах Mac, Windows, Linux), DirectX (на ОС Windows), та власне API для Wii.

Unity підтримує ряд розширень файлів, які можна імпортувати та виконувати налаштування за допомогою інтерфейсу Unity. До цих файлів відносяться Maya, 3ds Max, modo, Blender, Softimage, Adobe Photoshop, ZBrush, Cheetah3D, Cinema 4D, Allegorithmic Substance [30].

Microsoft .NET – це програмна технологія, яка використовується як платформа створення звичайних програм наряду із веб-застосунками. На рис. 1.3 наведено стек-технології .NET Framework [1].

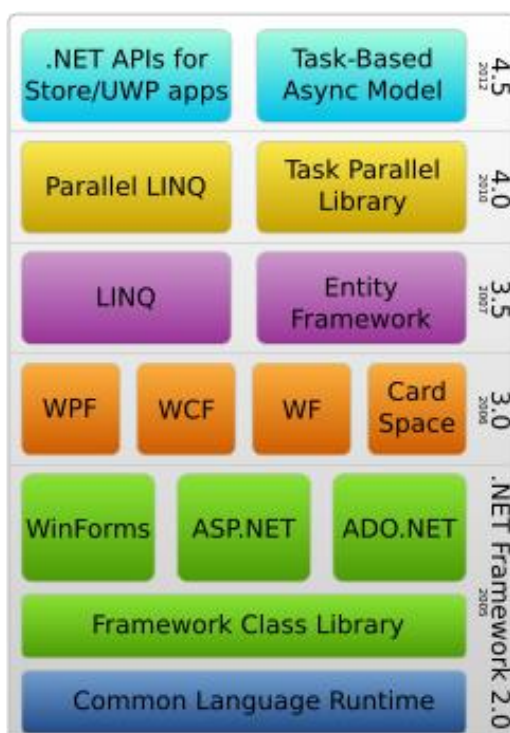


Рисунок 1.3 - Стек технологій .NET Framework

Ця платформа дозволяє використовувати різні мови програмування наряду із бібліотеками, які забезпечують необхідну функціональність. Microsoft .NET підтримує мови програмування F#, Visual Basic та C# наряду із трьома різними бібліотеками (платформами):

- 1) кросплатформенний засіб розробки .NET Core;
- 2) для створення вебсайтів .NET Framework;
- 3) для мобільних додатків Mono.

У всіх вищезазначених бібліотек є спільний набір стандартних бібліотек, які об'єднані у .NET Standard платформу.

Python – це багатоцільова мова програмування, за допомогою якої розробник може писати код, що легко читається. Мові характерний відносний лаконізм, завдяки чому можна створити програму, яка буде за кількістю коду набагато коротша за аналог такої ж програми, написаної на іншій мові програмування [75].

Вибір цієї мови програмування був продиктований рядом факторів, до яких відносяться:

- гнучкість відносно операційних систем, тобто можливість запуску програми, написаної на Python, на будь-якій операційній системі;
- наявність стандартної бібліотеки, які містить готові інструменти для роботи з ОС, базами даних, веб-сторінками, побудови графічного інтерфейсу, різними форматами даних, тощо;
- можливість створення програм у вигляді скрипта;
- Python безкоштовний [62].

Robot Operating System (ROS) – це операційна систем для роботів. Вона представляє собою екосистему для програмування робототехнічного обладнання, в якій наявний функціонал для розподіленої роботи. Першочергово ROS була розроблена у 2007 році [66]. ROS базується на архітектурі графів.

ROS забезпечує стандартні служби ОС, а саме:

- низькорівневий контроль приладів;
- апаратну абстракцію;

- керування пакетами;
- передачу повідомлень між процесами;
- реалізацію функцій [23].

Загалом розробка робота в ROS починається з CAD програми, наприклад, SolidWorks. На наступній фазі (рис. 1.4) необхідно створити URDF робота. URDF міститиме інформацію про межі робота, його вигляд та фізичні властивості, такі як інерція, вузли, вага, тощо. URDF чудово працює з примітивами, наприклад, циліндр, куля, паралелепіпед.

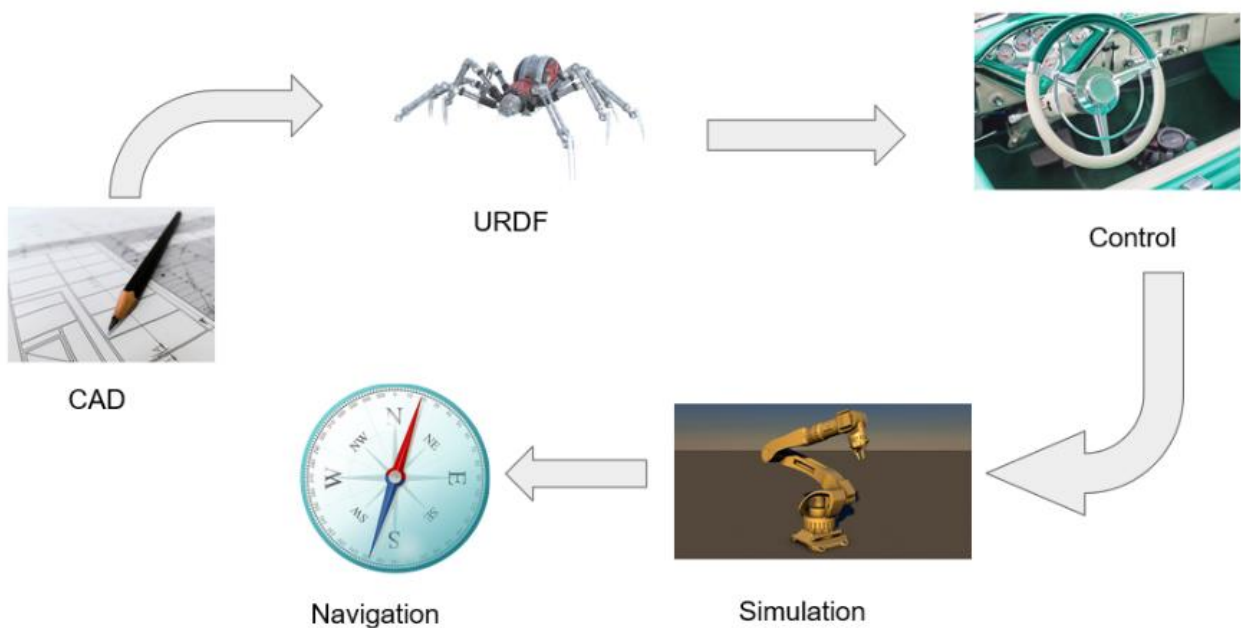


Рисунок 1.4 – Процес розробки мобільного робота у ROS

Наступний етап полягає у виборі регуляторів для робота. Наприклад, для роботизованого обладнання з двома моторами та парою активних коліс і третього пасивного колеса, можна використовувати диференціал. Слід відзначити, що симуляція роботи не здатна ідеально передати природу середовища, але наближається до передачі всього функціоналу та фізики його роботи. На рис. 1.5 наведено приклад симуляції у ROS, який був створений для робота Leobot у Львівській національній галереї мистецтв.

Найглибша інтеграція з ROS характерна симулятору Gazebo. Цей симулятор характеризується відкритим програмним кодом та безкоштовністю

його використання. Крім того, у цього симулятора є веб-розширення gzWeb, завдяки якому можна проводити симуляцію у веббраузері (рис. 1.6) [66].

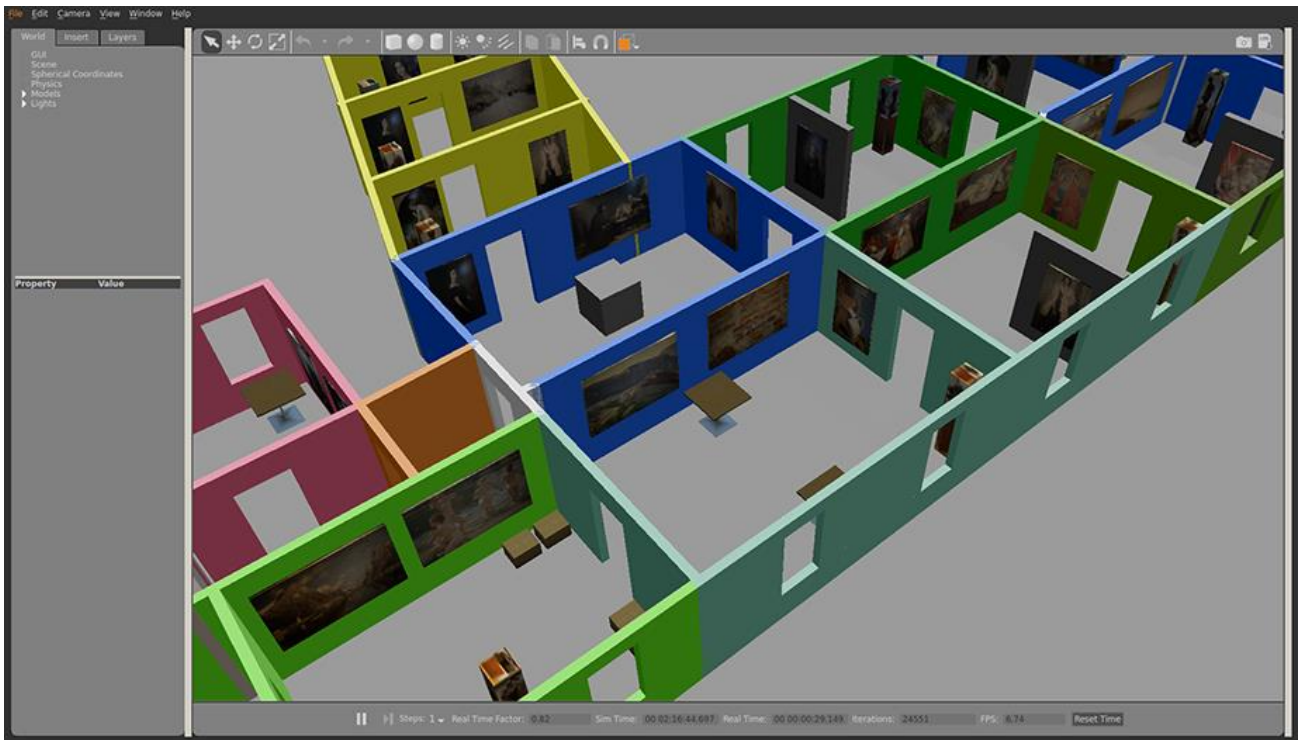


Рисунок 1.5 – Приклад симуляції робота у ROS

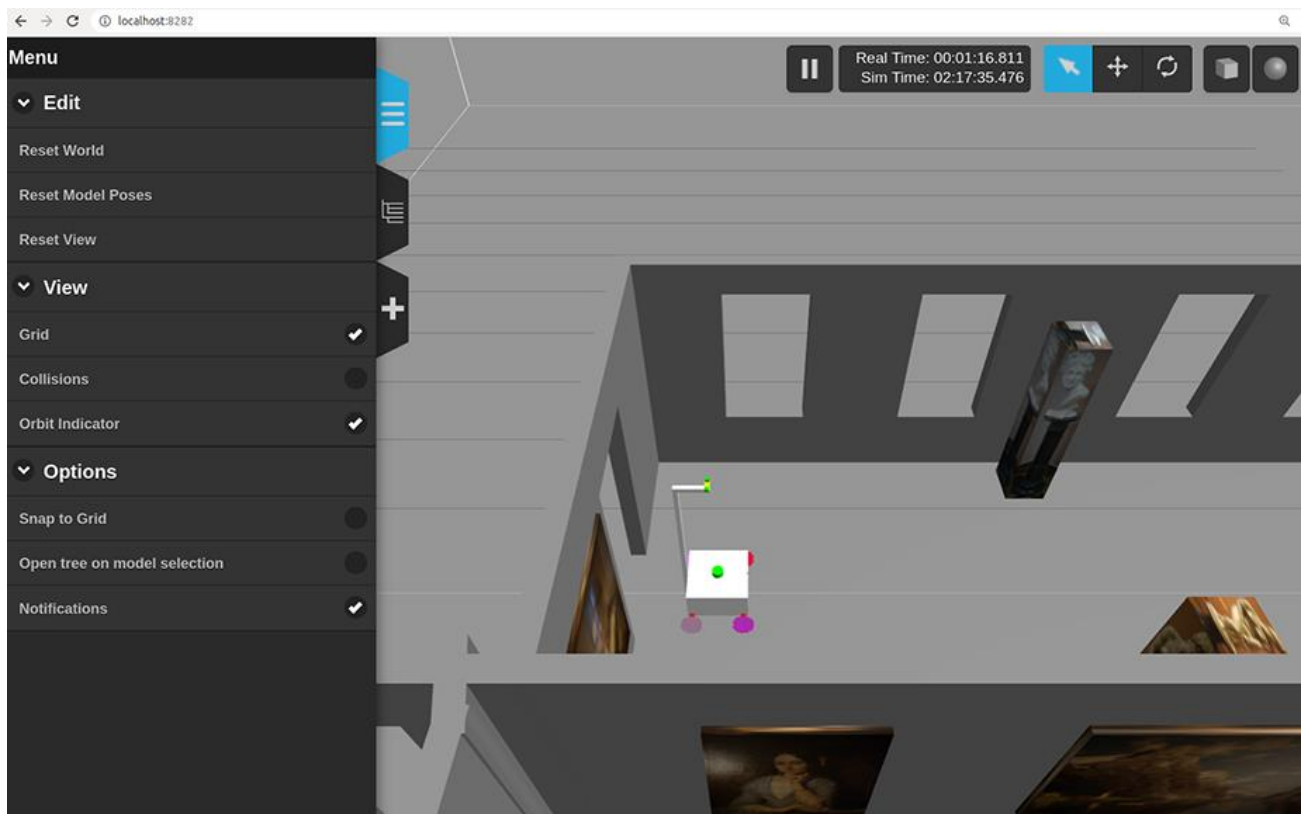


Рисунок 1.6 – Симуляція робота у gzWeb

Після цього можна створювати функціонал, а також додавати навігацію. Завдяки підбору ряду параметрів, навігаційна система ROS дозволяє проводити налаштування алгоритму пошуку робота на мапі, побудови мап, а також переміщення роботизованого обладнання з точки А у точку Б.

1.5 Висновки до першого розділу

В першому розділі було проведено огляд та аналіз підходів, які застосовуються при створенні віртуальних прототипів. Наведені характеристики віртуальних прототипів та переваги їх використання у порівнянні із фізичними прототипами.

В якості прикладу робота-маніпулятора, для якого буде проводитись оцінки функціональності його роботи, було обрано робот-маніпулятор типу «рука» Panda Arm. В розділі наведені технічні характеристики цього робота та його сфери застосування. Також у розділі було поставлено задачі та запропоновані підходи до проведення оцінки функціональності роботів-маніпуляторів.

В розділі також наведено короткий опис середовищ для розробки запропонованої оцінки функціональності робота-маніпулятора Panda Arm.

2 РОЗРОБКА МЕТОДУ ОЦІНЮВАННЯ ФУНКЦІОНУВАННЯ МОДЕЛІ РОБОТА-МАНІПУЛЯТОРА

2.1 Архітектура фреймворку для проведення оцінки функціональності робота-маніпулятора

Метод оцінки функціонування робота-маніпулятора Panda Arm на основі імітаційного моделювання, тобто на основі створення цифрового двійника, по своїй суті представляє систему, яка складається із ряду компонентів. При розробці цієї системи необхідно прийняти ряд рішень стосовно структури коду, архітектури, дизайну, зберігання даних.

На рис. 2.1 показано схему архітектури запропонованого методу оцінки робота-маніпулятора на основі імітаційного моделювання (МОІМ), який умовно поділений на три основні сегменти. В подальшому в кваліфікаційній роботі ці сегменти будуть називатись компонентами системи оцінки.

Роль кожного окремого компоненту в процесі проведення оцінки імітаційної моделі можна зрозуміти, в першу чергу, з його назви:

- визначення простору проведення оцінки;
- оцінка та виконання;
- візуалізація хмари точок.

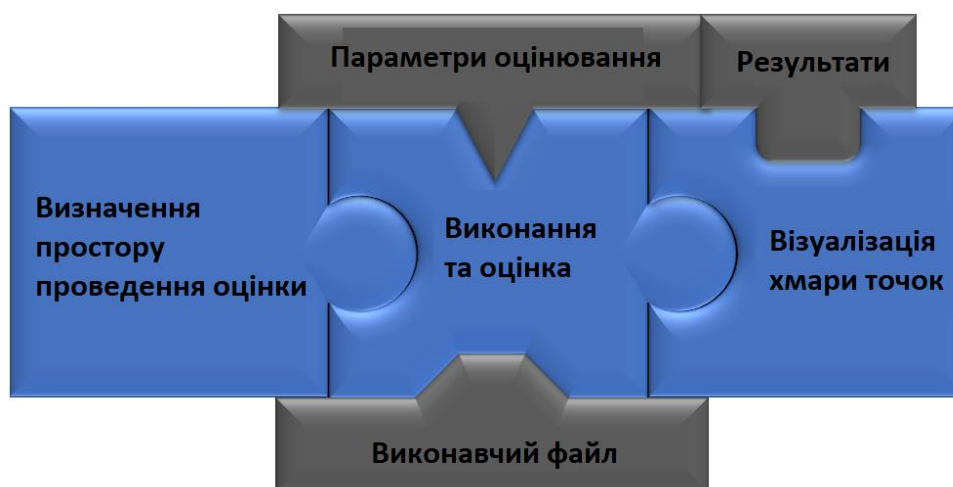


Рисунок 2.1 – Архітектура системи оцінки
(основні компоненти – синій колір, вхідні дані – сірий колір)

Як видно з архітектури, компоненти системи виконуються послідовним чином.

Окрім основних компонентів у систему оцінки МОІМ включено три сегменти, які відповідають за передачу інформації. На рис. 2.1 вони зображені сірими фрагментами. В подальшому в роботі ці компоненти отримали назву входів.

Оцінка мехатронної системи в симуляції потребує виконання декількох окремих кроків, які тісно пов'язані між собою. В загальному ці кроки відповідають методу віртуального прототипування.

Спочатку користувач повинен визначити простір, в якому проводиться оцінка роботи цифрового двійника. Такий підхід включає в себе специфікацію набору параметрів, які варіюються в імітаційній моделі мехатронної системи.

Зазвичай цей набір параметрів пов'язаний із властивостями компонентів робота-маніпулятора у вигляді мехатронної системи, що досліджується. Зміна таких параметрів буде мати вплив на зовнішній вигляд, функціональність, поведінку, безпеку системи або її ефективність. В програмній архітектурі, що розробляється, ці параметри отримують назву параметрів оцінювання.

Оскільки основною метою роботи користувача за запропонованим методом є визначення оптимальних параметрів налаштування, то він зацікавлений у моделюванні функціональності роботи робота-маніпулятора в межах певного діапазону значень. Тобто, користувач проводить вибір параметру оцінювання та зазначає діапазон значень, в якому обраний параметр повинен змінювати своє значення. Крім того, користувач зазначає розмір кроку зміни значення параметру оцінювання. Такий процес буде ітераційним і повторюватись до тих пір, поки система оцінки функціональності мехатронної системи не визначить всі бажані параметри оцінки наряду із діапазоном їх варіації.

Друга частина простору оцінювання представляє собою цільові значення. Такі значення можна розглядати як критерії прийняття рішень, за допомогою яких користувач бажає провести оцінку в процесі моделювання. Тобто, об'єктивні значення – це властивості мехатронної системи, значення яких цікавлять користувача при проведенні моделювання роботи системи.

Слід розглянути у вигляді прикладу дві зразкові мехатронні системи із наборами можливих параметрів оцінювання та цільових значень.

1. Автономний транспортний засіб.

Параметрами оцінки автономного ТЗ будуть контролер параметрів рульового керування та розподільча здатність сканування датчика, в той час як об'єктивними значеннями стануть здатність автономно слідувати по траєкторії руху та кількість виявлених перешкод у тестовому середовищі.

2. Робот-маніпулятор.

Параметрами оцінки для такої мехатронної системи стануть проміжна конфігурація суглобів на траєкторії, довжина ланок робота та тип двигуна, який використовується в його суглобах. В той час як цільовими значеннями стануть ступінь реалістичності рухів, що виконуються роботом з кожним типом двигуна, енергоефективність в процесі руху на траєкторіях та вартість виробництва ланок робота.

Загальний простір оцінювання в наведеному прикладі буде складатись із сітки, що охоплює зазначені параметри оцінювання, тобто в системі включені всі можливі комбінації параметрів оцінки наряду із визначеними користувачем цільовими функціями. Останні, в свою чергу, підлягають оцінюванню для кожного набору параметрів оцінювання.

При роботі системи оцінювання кожна комбінація оціночних параметрів передбачає моделювання мехатронної системи (етап виконання), а також розрахунок заздалегідь визначених цільових значень (проведення оцінки). Повне виконання такого оціночного циклу в подальшому називається кроком оцінювання.

На основі виконаної симуляції та отриманого набору цільових значень у користувача не має повної картини стосовно оцінки моделі функціональності робота-маніпулятора. Тому для спрощення сприйняття таких цифрових значень у фреймворку передбачений етап створення візуалізації хмар точок, яка згенерована шляхом багаторазового запуску оцінки для кожної із введених комбінацій параметрів роботи мехатронної системи. Оскільки системі також характерний інтуїтивно зрозумілий інтерфейс наряду із якісною візуалізацією

даних, у користувача є можливість оглянути простір проведення оцінки з різних точок зору. Крім того, користувач може виявити взаємозалежності та вплив певних параметрів оцінювання на об'єктивні значення роботи мехатронної системи, а в подальшому – вибрати оптимізовані набори параметрів та знайти найкращі компромісні рішення у випадку настання суперечливих ситуацій в роботі робота-маніпулятора.

Розбиття системи оцінки на окремі компоненти та їх взаємозв'язок показані на рис. 2.2.

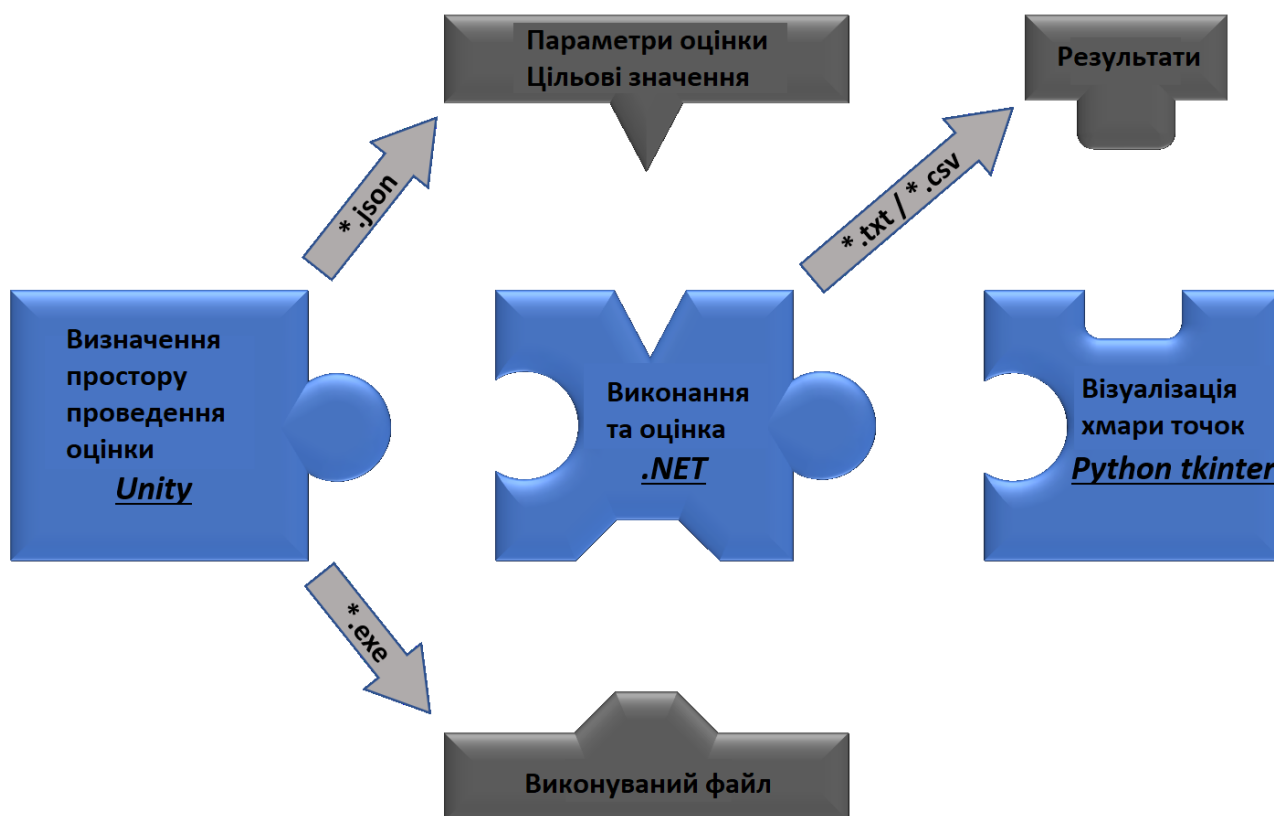


Рисунок 2.2 – Архітектура системи оцінки з розбивкою на компоненти

Три основні компоненти системи використовують три різні середовища розробки:

- Unity використовується для моделювання роботи мехатронної системи та визначення простору оцінки;
- Microsoft .NET використовується для виконання та оцінки кожного файлу;

– Python, а саме його бібліотека `tkinter`, використовується для візуалізації хмари точок.

Крім того, з рис. 2.2 видно, який компонент відповідає за передачу вхідних даних. Виконавчий файл симуляції створюється за допомогою Unity. До того ж, простір оцінки визначається в межах редактора Unity, які в подальшому експортуються за допомогою файлів із розширенням `.json`.

.NET фреймворк проводить оцінку двох файлів, експортованих з Unity, після чого періодично проводить виклик виконавчого файлу та надсилає комбінацію параметрів оцінки, а також отримує набір об'єктивних значень.

Після остаточної оцінки та моделювання всіх комбінацій параметрів системи, результати зберігаються у текстовому файлі. Цей файл в подальшому використовується для візуалізації хмари точок параметрів функціональності робота-маніпулятора.

2.2 Визначення простору оцінки в рамках запропонованого методу

Простір проведення оцінки функціональності робота-маніпулятора за допомогою розробленого методу можна визначити зсередини редактора Unity. Для цього було запропоновано концепцію, яка дозволяє користувачу проводити вибір параметрів оцінки та об'єктивні значення таких параметрів в той час як Unity перебуває в режимі редагування.

Ідея застосування сценаріїв редактора Unity відіграє головну роль в цьому контексті. На рис. 2.3 показано умовно схему структуру коду розробленого методу, який забезпечує функціональність вибору параметрів оцінки роботи робота-маніпулятора. Слід звернути увагу на те, що така сама стратегія використовується і для проведення оцінки об'єктивних значень.

У цій частині системи оцінки використовуються три концепції налаштування редактору Unity, а саме:

– використання класу `EditorWindow` у Unity з метою додавання додаткових вікон;

- використання CustomEditor в якості інструменту зміни вигляду компоненту роботи системи;
- використання в Unity об'єкта ScriptableObject в якості об'єкта для збереження даних.

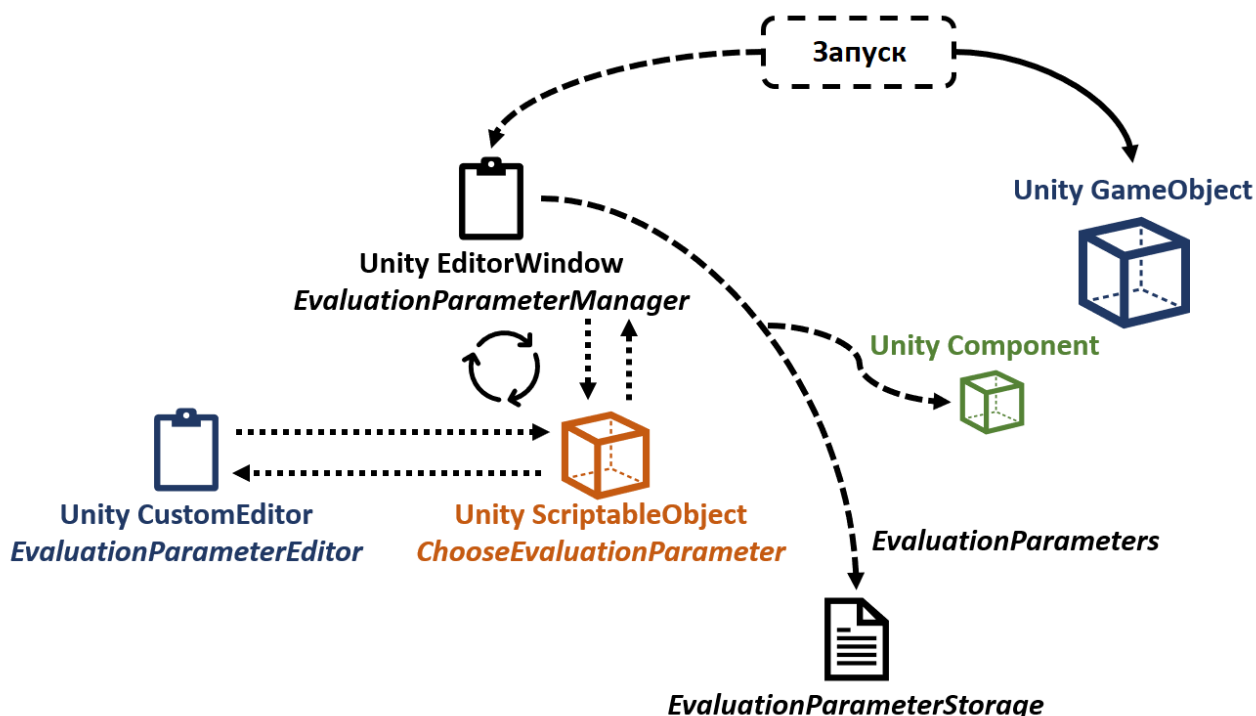


Рисунок 2.3 – Вибір параметрів оцінки в редакторі Unity

Запуск компоненту системи, написаного в Unity [40], відбувається шляхом вибору пункту меню у редакторі Unity. В результаті до ієрархії Unity додається порожній об'єкт GameObject, який має назву EvaluationFramework, а також з'являється кастомізоване вікно EditorWindow, яке має назву EvaluationParameterManager (рис. 2.3). В цьому вікні наявні кнопки для видалення та додавання параметрів оцінки, а також воно містить інформацію про вже обрані раніше параметри оцінки.

Отже, додавання користувачем параметру оцінки буде ініціювати наступну послідовність дій у компоненті системи, створеному на Unity:

1. Створюється об'єкт ScriptableObject, який буде зберігатись у папці ресурсів Unity.

2. Зміна вигляду скриптового об'єкта виконується за допомогою `EvaluationParameterEditor`.

3. Користувач може обрати `GameObject`, або ж один із його компонентів чи змінну. Вибір відбувається за допомогою елементів інтерфейсу користувача в `ScriptableObject`. Компонент повинен представляти собою скрипт C#, при чому він не може бути одним із вбудованих компонентів Unity.

4. Після того, як користувач обере змінну, йому буде запропоновано ввести налаштування для обраного параметру оцінки у відповідні поля. У фреймворку передбачена можливість вибору логічних, плаваючих та цілих змінних, а також передбачена можливість вибору плаваючих та цілих масивів. При цьому для змінних типу масив необхідно додатково вказувати розмір масиву. При виборі булевої змінної в якості параметру обчислення у користувача немає необхідності задавати додаткові вказівки.

5. За бажанням користувач може задати одиницю вимірю для обраного параметру оцінки. Використання цієї властивості на даному етапі може в подальшому спростити сприйняття візуалізації хмари точок.

Після завершення описаного процесу вибору у відповідному вікні редактора, що має назву `EvaluationParameterManager`, буде відображена інформація про параметр оцінки функціональності робота-маніпулятора. Ця інформація включатиме в себе назву обраного `GameObject`, змінну, компонент та визначені граничні значення і розмір кроку для зміни значення.

Тобто, при роботі системи оцінки для кожного параметра створюється окремий `ScriptableObject`. Слід відзначити, що видалення параметру оцінки призведе до видалення відповідного `ScriptableObject`.

Після того, як користувач виконає вибір всіх параметрів оцінки, він може зберегти інформацію за допомогою кнопки `EvaluationParameterManager`. Збереження даних буде виконано у вигляді масиву параметрів оцінки, при цьому він буде реалізований в окремому класі, написаному у мові програмування C# (назва класу - `Evaluation-ParameterStorage`). Також до початкового `GameObject` фремворк додасть унікальний скриптовий компонент `GameObject` `EvaluationFramework`. В цьому компоненті буде міститись по суті копія масиву

обраних параметрів оцінки. Визначення цільових значень проводиться аналогічним чином.

2.3 Принцип виконання та оцінки функціональності робота-маніпулятора

Розроблений в рамках роботи компонент виконання та оцінки функціонування робота-маніпулятора виконує велику кількість симуляцій, після чого зберігає отримані результати проведених симуляцій у вигляді хмари точок. Формат збереження хмари точок передбачений таким чином, щоб користувач в подальшому міг використовувати його за допомогою електронних таблиць для подальшого аналізу.

Кожен параметр обчислення в просторі по своїй суті представляє масив, оскільки він має діапазон певних значень, в межах якого він може змінюватись, а також розмір кроку в межах згаданого діапазону.

Метою роботи системи оцінки за запропонованим методом є моделювання всього можливого простору оцінок. Тобто всі параметри оцінки об'єднуються один з одним для подальшого формування багатовимірної хмари точок. Ця хмара містить всі можливі комбінації параметрів оцінки функціональності робота-маніпулятора.

Компоненту системи, який відповідає за виконання та оцінку, достатньо викликати симуляцію [61], що виконується, для кожного із обраних комбінацій параметрів оцінки, а також отримати в якості результатів заздалегідь визначені цільові значення.

На рис. 2.4 проілюстровано схему зчитування файлів із розширенням `.json`, що охоплює багатовимірну сітку над простором оцінювання та проведення симуляції для кожної комбінації параметрів оцінювання.

Точка входу від Unity до коду, написаного на `.NET`, представляє собою використання двох файлів із розширенням `.json`. Для спрощення зчитування інформації у фреймворку було реалізовано статичний клас `C#`, який називається `EvaluationData`. Він призначений для зчитування обидвох файлів на основі використання класу `Newtonsoft.Json.NET` [63].

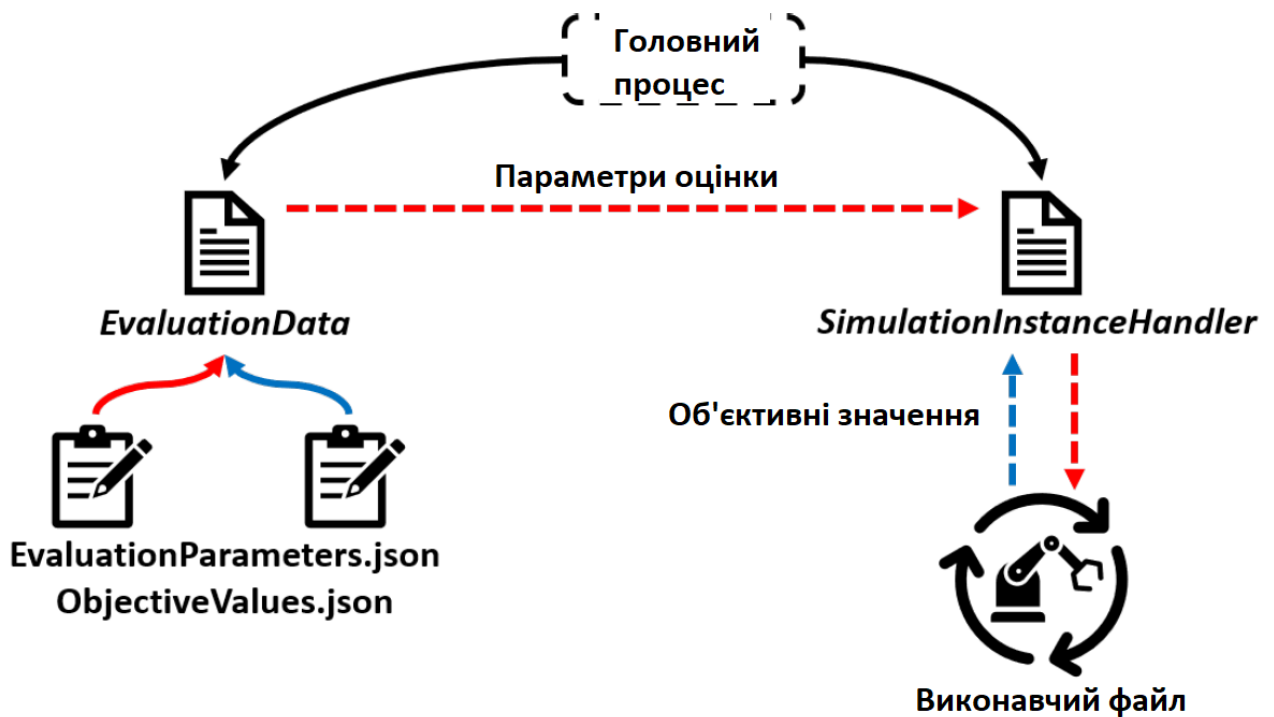


Рисунок 2.4 – Концепція виконання та оцінки результатів симуляції роботи
робота-маніпулятора

(червона стрілка – оцінка параметрів, синя стрілка – цільові значення)

Призначення класу полягає у зчитування файлів та подальшому охопленні раніше розглянутої багатовимірної сітки простору проведення оцінки функціональності робота-маніпулятора. Як це видно з рис. 2.4, шлях параметрів оцінки в рамках реалізованої системи показано червоними стрілками, в той час як синіми стрілками показано шлях цільових значень.

На платформі .NET також було реалізовано клас `SimulationInstanceHandler`, який отримує кожену комбінацію параметрів оцінювання. Після отримання таких комбінацій цей клас відповідає за виконання симуляції та подальше отримання цільових значень. Тобто, для кожної комбінації параметрів обчислення створюється окремий екземпляр класу `SimulationInstanceHandler` і отримує результати у вигляді об'єктивних значень.

На рис. 2.5 наведено приклад взаємодії одного `SimulationInstanceHandler` із одним виконавчим файлом. Тобто це приклад спрощеної структури міжпроцесорного зв'язку між компонентами системи оцінки, написаними на Unity та .NET.

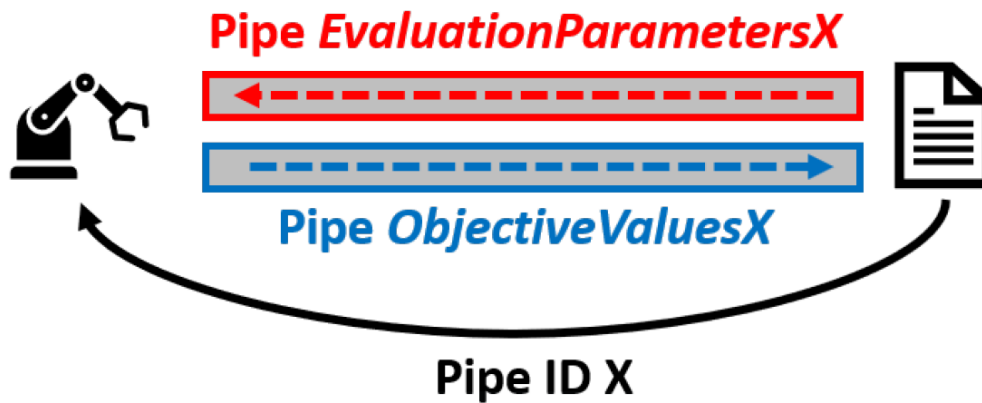


Рисунок 2.5 - Міжпроцесний зв'язок між виконавчим файлом Unity та .NET рішенням

При впровадженні компоненту виконання та оцінки, було передбачено розпаралелювання роботи системи. Оскільки в системі оцінки охоплюється великий масив можливих комбінацій, який може сягати десятків тисяч, то не доцільним і не раціональним стає синхронне моделювання одного виконавчого файлу для кожної окремої комбінації параметрів.

Для розпаралелювання процесу проведення оцінки було використано бібліотеку Microsoft .NET TPL, яка призначена для високорівненої реалізації паралельного коду [27]. На рис. 2.6 показано, як у компоненті виконання та оцінка TPL використовується, в результаті чого отримуємо ефективно розпаралелювання циклу. В такому циклі виконавчий файл Unity викликається кожною окремою комбінацією параметрів оцінки.

Як видно з рис. 2.6, кожна комбінація параметрів обчислення передбачає створення нового об'єкту класу `SimulationInstanceHandler`. Таке створення виконується до входу в основний цикл проведення оцінки параметрів. Крім того, максимальна ефективність виконання оцінки досягається в системі за умови запуску одного виконавчого файлу на одне ядро центрального процесора. Завдяки такому відходу забезпечується насичення ресурсів процесора, а коефіцієнт зайнятості процесора підтримується на рівні 100% протягом виконання всього процесу проведення оцінки функціональності робота-маніпулятора.

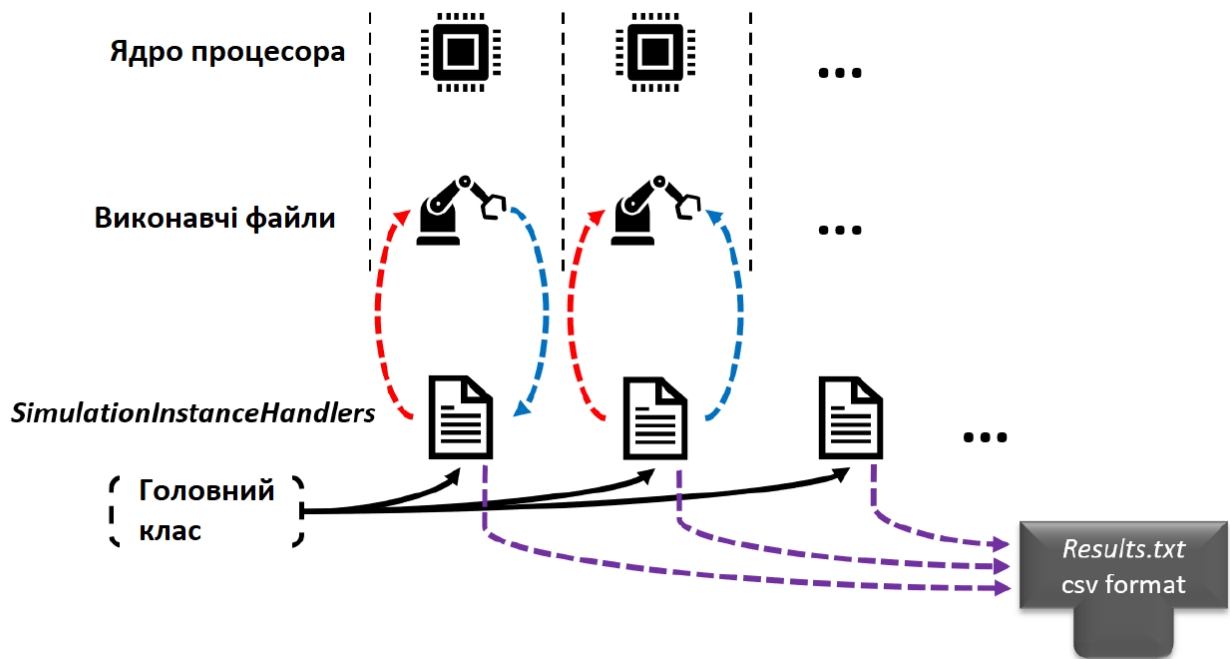


Рисунок 2.6 – Розпаралелювання взаємодії на ядрах процесора з виконавчими файлами

2.4 Процедура візуалізації хмари точок із оптимальними параметрами роботи робота-маніпулятора

Для покращення рівня проведення оцінювання імітаційної моделі робота-маніпулятора в контексті системи оцінки, що розробляється, необхідно розробити інтуїтивно зрозумілий та високоінформативний інструмент обробки масиву інформації. Для цього в методі передбачається візуалізація хмар точок даних, отриманих в процесі симуляції та в процесі обчислення об'єктивних значень параметрів оцінки. Для цього був розроблений третій компонент системи проведення оцінки, для якого використовувався Python tkinter [28].

По своїй суті це стандартний інтерфейс Python до інструментарію GUI Tcl/Tk. Tkinter підтримує низку версій Tcl/Tk, які створювались з або без підтримки потоків [28]. Він додає достатню кількість власної логіки у модулі та програми, при чому не є тонкою обгорткою [16].

На рис. 2.7 показано графічний інтерфейс, який було розроблено в рамках поставлених задач створення методу оцінки функціональності робота-

маніпулятора за допомогою імітаційного моделювання. Цей інтерфейс представляє собою хмару точок, вхідні дані для якої зберігаються в текстовому форматі, а за допомогою розробленого на Python tkinter компоненту, виконується візуалізація. В компоненті системи передбачена наявність деяких елементів інтерфейсу для користувача, за допомогою яких він зможе провести більш детальну оцінку отриманих результатів.

Для розробки графічного інтерфейсу використовувався Python tkinter, а обробка даних з текстового файлу у форматі .txt та їх подальша підготовка для візуалізації була виконана з використанням бібліотеки Pandas [21, 22, 59].

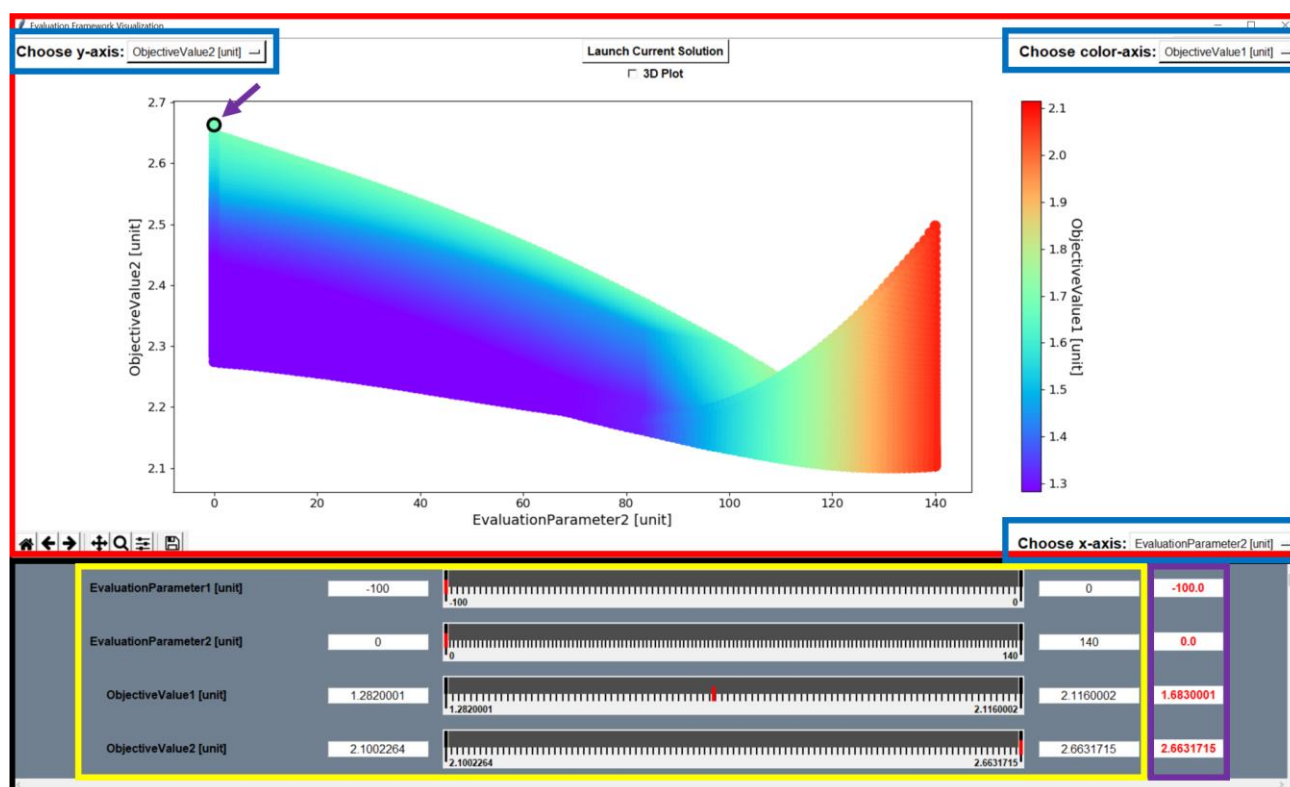


Рисунок 2.7 – Візуалізація хмари точок результатів проведеної оцінки

Зображений на рис. 2.7 графічний інтерфейс умовно поділений на дві основні частини, які представляють собою два різних фрейми tkinter. Верхній фрейм, виокремлений червоною рамкою, призначений для візуалізації двовимірного або тривимірного графіку простору оцінки робота-маніпулятора, а нижня частина інтерфейсу, яка виділена жовтим кольором, відображає цільові значення та параметри оцінки наряду із зазначенням діапазону їх значень.

Кожна точка, зображена на графіку, є однією точкою в просторі оцінки функціональності роботи робота-маніпулятора. Також наявні позначення осей абсцис та ординат для поточно обраних користувачем параметрів.

Кольорова смуга призначена для можливості додати третій вимір візуалізації. Завдяки градієнтному фарбуванню зображеної інформації в площині діаграми значень підвищується рівень сприйняття користувачем інформації.

В лівому нижньому кутку наявна панель інструментів для користувача, за допомогою якої він може:

- зберігати графік у вигляді зображення;
- перетягувати осі;
- масштабувати графік.

Кожен із розроблених віджетів має на меті збільшити рівень деталізації інформації для її подальшої оцінки в рамках розробленого методу для робота-маніпулятора.

Синім кольором на рис. 2.7 виокремлений віджет вибору вісі графіку. Натиснувши на будь-який із елементів, з'явиться випадаючий список, з якого користувач може провести вибір потрібного параметру, за яким проводиться оцінка.

Жовтим кольором виділений віджет, призначений для обмеження діапазону параметрів. Окрім наявної назви, повзунок відображає діапазон значень обраного параметру оцінки функціональності робота-маніпулятора в межах простору. У користувача є можливість перетягувати вправо та вліво обидва кінці повзунка, за допомогою чого відбувається обмеження діапазону параметрів, що відображаються в графічній області. У верхній частині відображаються лише ті точки, які попадають в діапазон між правим і лівим кінцевими значеннями цього повзунка. Решта точок за замовченням фарбуються в сірий колір.

Зона, виділена фіолетовим кольором (рис. 2.7), призначена для вибору конкретного значення. Наприклад, користувач може обрати певну точку на графіку шляхом звичайного натискання на неї, при цьому така точка автоматично

стає поточним обраним рішенням. Після такого вибору точка обводиться чорним кольором.

Крім того, передбачена можливість запуску поточного рішення за допомогою кнопки із відповідною назвою. Натискання цієї кнопки у верхній частині екрану призводить до виклику комбінації параметрів та виконання оцінки.

Розглянуті елементи інтерфейсу передбачені для збільшення рівня візуалізації та сприйняття і подальшої обробки виведеної інформації користувачем. За допомогою вибору осей діаграми, у користувача є можливість виявити залежності та зв'язки між об'єктивними значеннями та параметрами проведення оцінки функціональності робота-маніпулятора.

Слід відзначити, що декілька об'єктивних значень можуть відобразитись в області візуалізації одна навпроти одної. Запровадження кольорового відображення діаграми та можливості перемикачів з двовимірного на тривимірний вигляд та навпаки дозволяє оператору системи, який проводить оцінку функціональності роботи робота-маніпулятора, одночасно порівняти одразу декілька вимірів в області простору оцінки.

За допомогою розроблених додаткових віджетів користувач може оглянути, оцінити та порівняти лише частини простору оцінки, яка його цікавить. Обмеження простору оцінки можна проводити за об'єктивними значеннями. Таке обмеження особливо стане в нагоді в тому випадку, якщо в рамках симуляції оператора цікавить певний конкретний діапазон параметрів оцінки.

2.5 Висновки до другого розділу

В розділі виконано огляд функціональних можливостей методу оцінки робота-маніпулятора на основі імітаційного моделювання як системи в цілому, так і як кожного її окремого компоненту. До цих компонентів відносяться архітектура фреймворку, що розробляється, в якому описуються незалежні компоненти, файли та дані, що оцінюються між цими компонентами.

Також в розділі ілюструється, яким чином можна визначити простір оцінки за допомогою редактора Unity. У розділі пояснюється концепція коду, що забезпечує функціональність роботи методу оцінки робота-маніпулятора. Крім того, у розділі підсумовується, яким чином проводиться оцінка симуляції робота-маніпулятора за допомогою коду, розробленого у .NET.

Наостанок, викладено ідею третього компоненту методу оцінки, а саме фреймворку, який зчитує дані з текстового файлу і в подальшому створює візуалізацію результатів проведених у системі обчислень.

3 МАТЕМАТИЧНА МОДЕЛЬ ПРОВЕДЕННЯ ОЦІНКИ РОБОТА-МАНІПУЛЯТОРА

3.1 Імітаційна модель Panda Arm

В роботі розглядається оцінка функціональності на прикладі робота Panda Arm (рис. 3.1), технічні характеристики якого наведені у розділі 1.2. В своїй конструкції робот містить сім суглобів, що обертаються. Він здатен переносити навантаження у 3 кг, що по суті є достатньо великим відсотком від загальної ваги роботи (12 кг), а саме це становить 25%.

В конструкції робота наявно більше 100 датчиків, за допомогою яких підтримується висока точність вимірювання крутних моментів та сил. На всіх семи обертових осях наявні датчики крутного моменту.

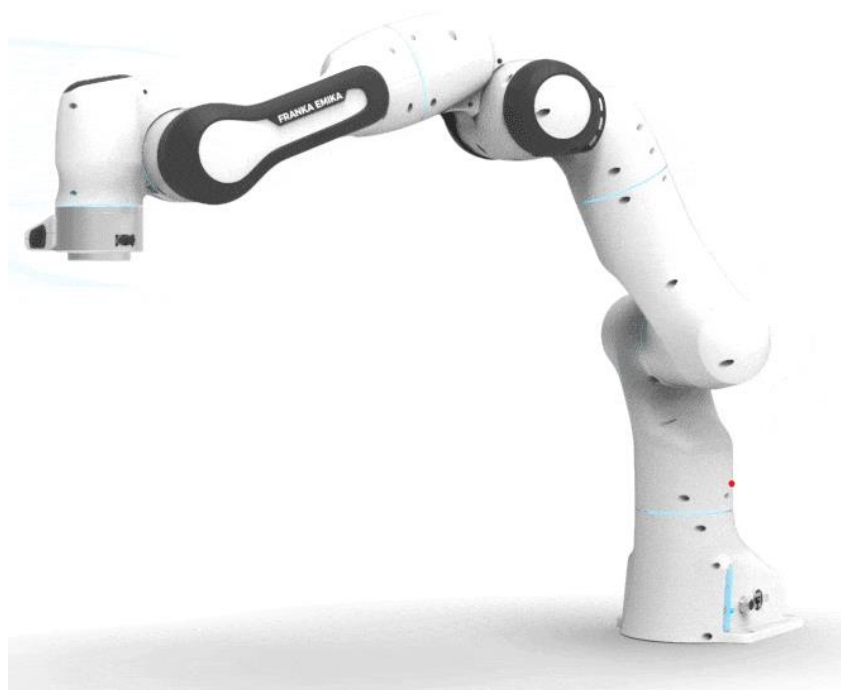


Рисунок 3.1 - Panda Arm від компанії Franka Emika

Для підтримки завантаження, керування роботом та візуалізації використовується пакет ROS із відкритим вхідним кодом. В цей пакет включений URDF-файл Panda Arm. Основною причиною вибору саме Panda Arm у якості

прикладу робота-маніпулятора є саме його відкритий код наряду із широкою сферою використання.

Для використання робота-маніпулятора в рамках методу, що розробляється, в першу чергу розглядаються його кінематичні параметри (наприклад, довжина ланки). Крім того, важливу інформацію надають його динамічні параметри, наприклад інерційні властивості та вага ланок. Якщо кінематичні параметри робота-маніпулятора наявні у відкритому доступі [7], то із динамічними параметрами в кваліфікаційній роботі ми будемо опиратись на дослідження положення центру мас кожної ланки при роботі та елементів тензора, які були отримані у роботі [14].

Після отримання динамічних параметрів робота, можна провести оцінку функціональності робота-маніпулятора в реалістичному середовищі Unity. Для імпортування робота-маніпулятора у сцену Unity використовувався UrdImporter з ROS. Результат такого імпортування наведено на рис. 3.2. Як видно з рис. 3.2, в лівій частині вікна відображено ієрархію робота-маніпулятора в сцені Unity.

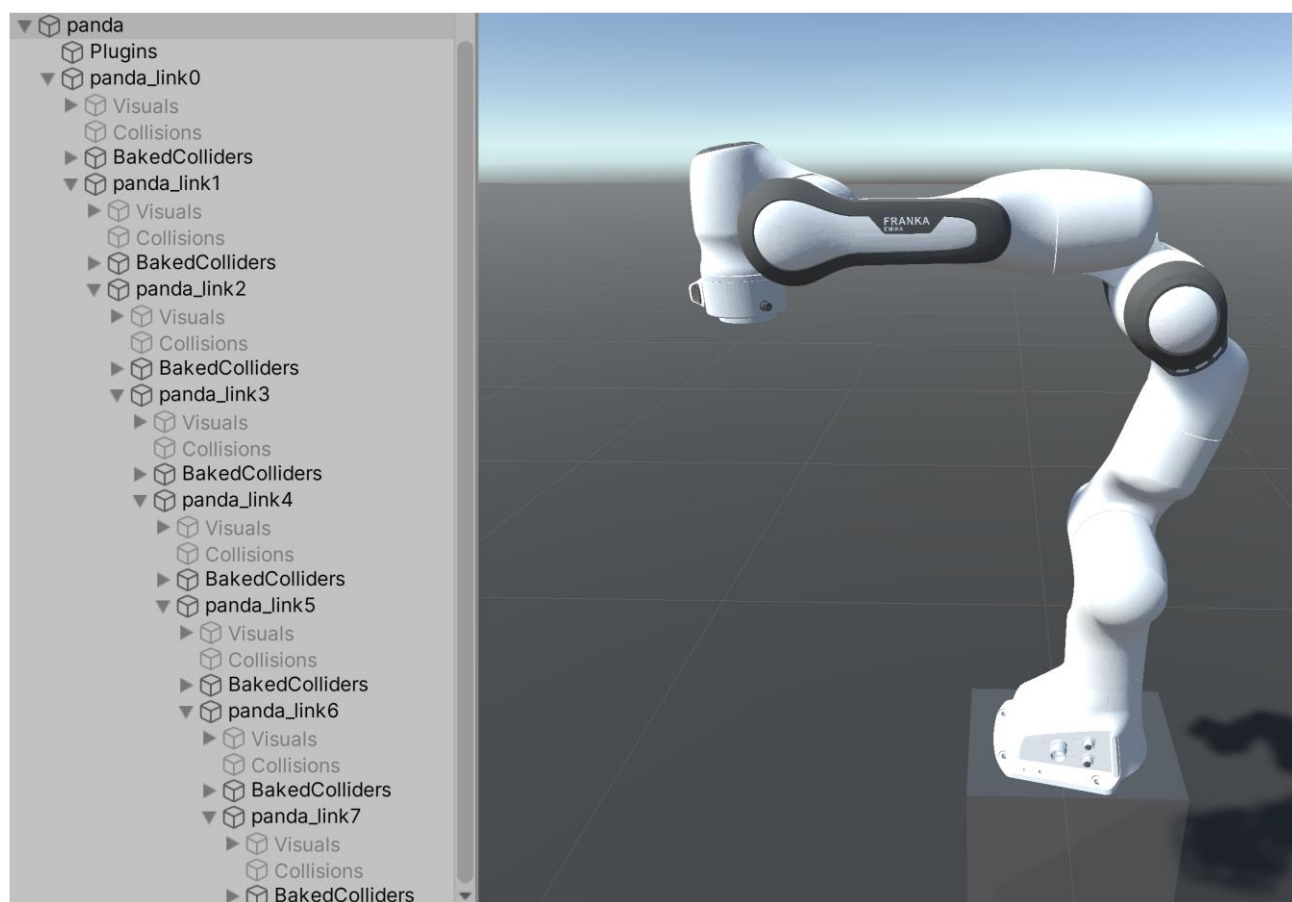


Рисунок 3.2 – Імітаційна модель Panda Arm в середовищі Unity

Утиліта URDF-Importer відповідає за повний процес імпортування імітаційної моделі робота-маніпулятора із URDF-файлу та його подальшого належного представлення в GameObjects [32]. Для представлення реального робота використовується метод наслідування. Всі сім ланок робота представляють собою один об'єкт GameObjects.

В імітаційній моделі Panda Arm використовуються наступні компоненти:

- HingeJoint (має кожен суглоб як жорстку ланку, параметром є вага ланки та тензор інерції);
- Rigidbody (через цей компонент відбувається з'єднання кожної дочірньої ланки GameObject із твердим тілом, при чому в якості властивостей задаються межі положення суглобів);
- MeshCollider (додає можливість симуляції зіштовхування ланок робота з іншими об'єктами);
- HingeJointMotor (двигун від Unity, за допомогою якого відбувається керування рухом кожною ланкою; за допомогою цього компоненту можна визначити максимальний крутний момент).

В розділі розглядається два приклади оцінки функціональності робота-маніпулятора Panda Arm, в кожному з яких використовується описана вище імітаційна модель робота.

3.2 Оцінка траєкторії руху Panda Arm

Використання роботів-маніпуляторів отримало широкого поширення. Вони можуть використовуватись для розпилення, фрезерування, складання, пакування, полірування, зварювання, тощо. У всіх схожих завданнях використання роботів-маніпуляторів для виконання технологічного процесу виготовлення продукції робот типу «рука» повинен виконувати рухи по різних траєкторіях, які можуть мати різну довжину, складність та інші параметри. При цьому рухи по траєкторії повинні характеризуватись високою точністю.

Для початку необхідно провести оцінку функціональності та руху по траєкторії робота-маніпулятора Panda Arm, якому властиві сім ступеней свободи.

Основними параметрами такої оцінки стануть точність дотримання траєкторії, енергоефективність, найменша відстань, тощо.

3.2.1 Метод параболічного змішування

Маршрутні точки, які поєднуються у траєкторію руху, параметризовані часом. Для гарантування, що рука робота-маніпулятора буде слідувати траєкторії, було використано метод параболічного змішування, розроблений Кунцем та Стілманом [26]. Метод полягає у перетворенні траєкторії на параметизовану в часі величину.

За цим методом, для уникнення помилок на траєкторії руху робота-маніпулятора, в його рух по траєкторії додаються параболічні рухи. На рис. 3.3 наведено ідею апроксимації прямої лінії, яка з'єднує дві кінцеві точки траєкторії руху робота-маніпулятора, при чому з'єднання виконується параболічними функціями навколо цих точок.

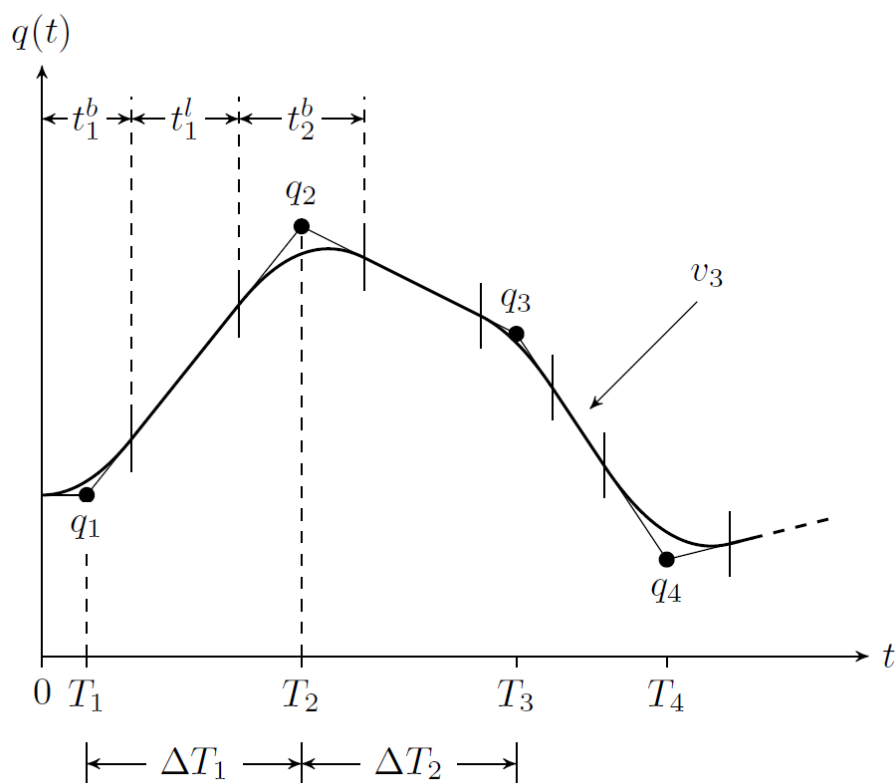


Рисунок 3.3 – Одновимірний приклад застосування методу параболічного змішування на траєкторії руху робота-маніпулятора

Згадані точки траєкторії руху робота позначені $q_i, i \in \{1 \dots n\}$, де n – це загальна кількість конфігурацій на шляху. Для Panda Arm, конфігурація q_i представлена вектором, який містить 7 елементів, по одній за кожен суглоб руху. З'єднання цих точок передбачено прямими відрізками.

Тобто, траєкторія руху Panda Arm буде представляти собою функцію q , яка на будь-який момент часу відображає загальну тривалість траєкторії t_f в конфігурації робота:

$$q: [0, t_f] \rightarrow C. \quad (3.1)$$

На обмеження траєкторії додатково впливає дві умови:

1. В кінцевих точках траєкторії руху робот-маніпулятор має нульову швидкість.
2. Протягом руху по всій траєкторії необхідно передбачати обмеження на прискорення та швидкість руху.

Ці умови можна записати як:

$$\forall t: |q'(t)| \leq v_{max} \wedge |\ddot{q}(t)| \leq a_{max}. \quad (3.2)$$

Слід відзначити, що немає повної гарантії в тому, що при проходженні траєкторії робота-маніпулятора він точно досягне кожну точку шляху. Тому траєкторія складається з двох основних фаз: лінійної (робот рухається по прямому відрізку) та параболічної.

При проходженні лінійної фази робот-маніпулятор рухається з нульовим прискоренням, тобто лінійно змінює своє положення в часі із постійною швидкістю. При вході в параболічну фазу, робот-маніпулятор переходить на постійне прискорення, тобто його конфігурацію змінюється відносно часу у квадратичному характері [71].

Таким чином, за час ΔT_i робот переміщується між точками q_i та $q_i + 1$. Тривалість фази в цій точці тоді t_i^b . Знаючи ці часові характеристики та постійну

швидкість при проходженні роботом-маніпулятором лінійної фази між зазначеними точками, обчислення швидкості зводиться до формули:

$$v_i = \frac{q_{i+1} - q_i}{\Delta T_i}. \quad (3.3)$$

Оскільки при русі робота-маніпулятора із такою швидкістю вздовж прямої траєкторії протягом всього часу роботи невідворотне призведе до наявності розривних швидкостей в кожній точці траєкторії, то частини двох сусідніх прямих відрізків замінюються параболою. Робот-маніпулятор в такому випадку рухається по параболі із постійним прискоренням, яке розраховується за формулою:

$$a_i = \frac{v_i - v_{i-1}}{t_i^b}. \quad (3.4)$$

В наступній формулі враховується важлива умова, яка гарантує, що дві параболічні фази змішування на сусідніх точках маршруту не будуть накладатись одна на одну:

$$t_i^b + t_{i+1}^b \leq 2\Delta T_i. \quad (3.5)$$

Після введення таких фаз змішування, робот-маніпулятор починає слідувати прямій траєкторії руху між точками q_i та q_{i+1} за час $T_i + \frac{t_i^b}{2}$. Як видно з рис. 3.3, траєкторія не починається за умови $t = T_0$, а також не закінчується за умови $t = T_n$.

Розглядати першу і останню точку можна за однаковими підходами, вважаючи, що першій точці завжди передують відрізок прямої з нульовою швидкістю. Також можна вважати, що за останньою точкою слідує лінійний відрізок. Звідси виходить, що загальна тривалість траєкторії руху робота-маніпулятора з точки 0 до точки n розраховується за формулою:

$$t_f = \frac{t_0^b}{2} + \sum_{i=1}^n \Delta T_i + \frac{t_n^b}{2}, \quad (3.6)$$

де t_f – час проходження роботом траєкторії шляху, хв;

t_0^b і t_n^b – час проходження роботом траєкторії першого та останнього лінійного відрізків відповідно, хв;

ΔT_i – час, за який робот переміщується між двома точками, що не є кінцевими в траєкторії руху, хв.

Тобто, остаточна траєкторія руху робота-маніпулятора визначається за формулою:

$$q(t) = \begin{cases} q_i + v_{i-1}(t - T_i) + \frac{1}{2}a_i \left(t - T_i + \frac{t_i^b}{2} \right)^2, \\ \text{якщо } T_i - \frac{t_i^b}{2} \leq t \leq T_i + \frac{t_i^b}{2}, \quad i \in \{1 \dots n\}, \\ q_i + v_i(t - T_i), \\ \text{якщо } T_i + \frac{t_i^b}{2} \leq t \leq T_{i+1} + \frac{t_{i+1}^b}{2}, \quad i \in \{1 \dots n - 1\} \end{cases} \quad (3.7)$$

За цією формулою було отримано значення прискорення та швидкості руху робота-маніпулятора Panda Arm в імітаційній моделі, що використовувалась.

За цим методом розрахована траєкторія може використовуватись лише в тому випадку, якщо нам відомі таймінги фаз і точок. Однак, найчастіше, розробнику не відомі ці змінні часу при роботі з імітаційною моделлю робототехнічного обладнання.

Для визначення цих значень Кунц та Стілман пропонують отримувати ці значення за запропонованим ним ітераційним підходом. За цим підходом в першу чергу обираються такі часові змінні, щоб досягалися максимальні швидкості руху суглобів робота-маніпулятора. Для такого розрахунку використовується формула:

$$\Delta T_i = \max \frac{|q_{i+1}[j] - q_i[j]|}{v_{max}}, \quad (3.8)$$

де j – номер вузла.

Крім того, за допомогою максимізації швидкостей рухів суглобів робота-маніпулятора означає розрахунок часу досягнення точок траєкторії таким чином, щоб хоча б один із суглобів робота-маніпулятора рухався із максимальною швидкістю.

На другому етапі проводиться вибір тривалості фази змішування t_i^b таким чином, щоб хоча б один із суглобів Panda Arm за таких умов рухався із максимальним прискоренням:

$$t_i^b = \max \frac{|v_{i+1}[j] - v_i[j]|}{a_{max}}. \quad (3.9)$$

3.2.2 Оцінка споживання електричної енергії

Для проведення змістовної оцінки енергоспоживання робота-маніпулятора при його проходженні по різних траєкторіях як вихідні дані будуть виступати електрична потужність, яка подається на кожен двигун. Тому необхідно розглянути принцип розрахунку енергії, що споживається Panda Arm в його імітаційній моделі.

Основні дані для проведення розрахунку були отримані з технічних деталей та специфікації робота-маніпулятора [7, 11, 20]. Слід враховувати, що певні із наведених значень можуть мати незначні відхилення в імітаційній моделі у порівнянні із реальними значеннями при роботі робота-маніпулятора.

В загальному рівняння динаміки руху робота по траєкторії може бути отримано із формулювання Лагранжа [44] або Ньютона-Ейлера [42]:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q), \quad (3.10)$$

де τ – крутний момент суглоба робота-маніпулятора;

q – кутове положення робота-маніпулятора;

\dot{q} – швидкість руху робота-маніпулятора;

\ddot{q} – прискорення руху робота-маніпулятора;

$M(q)$ – матриця мас у просторі;

$C(q, \dot{q})$ – відцентрові та коріолісові умові [65], які діють на робота;

$g(q)$ – гравітаційна сила.

У випадку застосування формули до Panda Arm буде отримано наступне:

$$\tau, q, \dot{q}, \ddot{q} \in R^{7 \times 1}. \quad (3.11)$$

На основі наявних даних про крутні моменти в елементах руху робота-маніпулятора можна провести розрахунок енергоспоживання двигуна в суглобах робота Panda Arm. При наявних значеннях швидкості та прискорення, а також відомих координат положення робота, в процесі симуляції можна в будь-який момент часу визначити крутні моменти в суглобах роботу типу «рука». Такий підхід називається зворотною динамікою або рекурсивним алгоритмом Ньютона-Ейлера [4, 10].

Цей рекурсивний алгоритм Ньютона-Ейлера (RNEA) забезпечує ефективний засіб для обчислення оберненої задачі динаміки. Найбільш широкого застосування він набув для розрахунку системи з багатьма тілами.

Змінні імпульсу та кінематичні змінні в прямій рекурсії робота обчислюються для всіх суглобових з'єднань робота, починаючи з основи та крокуючи в сторону кінцевого елемента.

Для проведення обчислень за методом RNEA в модулі на Unity було реалізовано вивід динамічних параметрів кожної ланки. В кожен момент часу, тобто після виконання роботом кожного кроку, за допомогою методу RNEA проводиться розрахунок крутних моментів, які в подальшому використовуються для розрахунку енергоспоживання.

Електрична потужність двигуна $P_{el}^j(t)$, яка необхідна для роботи двигуна на суглобі j , визначається за формулою:

$$P_{el}^j(t) = P_{motor}^j(t) + P_{loss}^j(t), \quad (3.12)$$

де $P_{motor}^j(t)$ – потужність двигуна, яка перетворюється в механічну енергію з метою виконання роботом рухів;

$P_{loss}^j(t)$ – потужність, яка втрачена через струм двигуна $I_{motor}^j(t)$, що протікає через клему із опором R_T . Розраховується за формулою:

$$P_{loss}^j(t) = I_{motor}^j(t)^2 \cdot R_T. \quad (3.13)$$

Опір в основному лежить в межах 1 Ома, для цієї моделі значення опору становить 0,22 Ома. В цій моделі передбачається нехтування витратами енергії через інвертор та блок живлення.

Механічна потужність розраховується як добуток крутного моменту на стороні з'єднання та кутової швидкості з'єднання за наступною формулою:

$$P_{mech}^j(t) = \omega^j(t) \cdot \tau^j(t), \quad (3.14)$$

де $\tau^j(t)$ – крутний момент j -ого з'єднання (суглоба робота) у момент часу t ;
 $\omega^j(t)$ – кутова швидкість руху суглоба.

Для визначення потужності двигуна, яка необхідна для генерування бажаної механічної потужності (3.14), необхідно при розрахунку враховувати передаточний момент та швидкість двигуна, беручи до уваги коефіцієнт корисної дії як двигуна, так і редуктора. Суглоб двигуна Panda Arm обертається із заданою швидкістю $\omega^j(t)$, яка пов'язана із частотою обертання двигуна через відношення $R = 100$ наступним чином:

$$n_{motor}^j(t) = R \cdot \frac{60}{2\pi} \cdot \omega^j(t). \quad (3.15)$$

Коефіцієнт корисної дії (ККД) редуктора n_g напряму залежить від швидкості обертання двигуна. Спостерігається тенденція до залежності чим вище обертів на хвилину, тим нижчий ККД редуктора, а саме:

$$\begin{aligned}
 n_g (n_{motor} \leq 500) &= 0,75; \\
 n_g (n_{motor} = 1000) &= 0,69; \\
 n_g (n_{motor} = 2000) &= 0,63; \\
 n_g (n_{motor} = 3000) &= 0,58.
 \end{aligned}
 \tag{3.16}$$

Між зазначеними у 3.16 кутовими точками передбачається, що ККД зменшується за лінійною залежністю. На рис. 3.4 показано зазначену залежність між швидкістю двигуна та ККД редуктора.

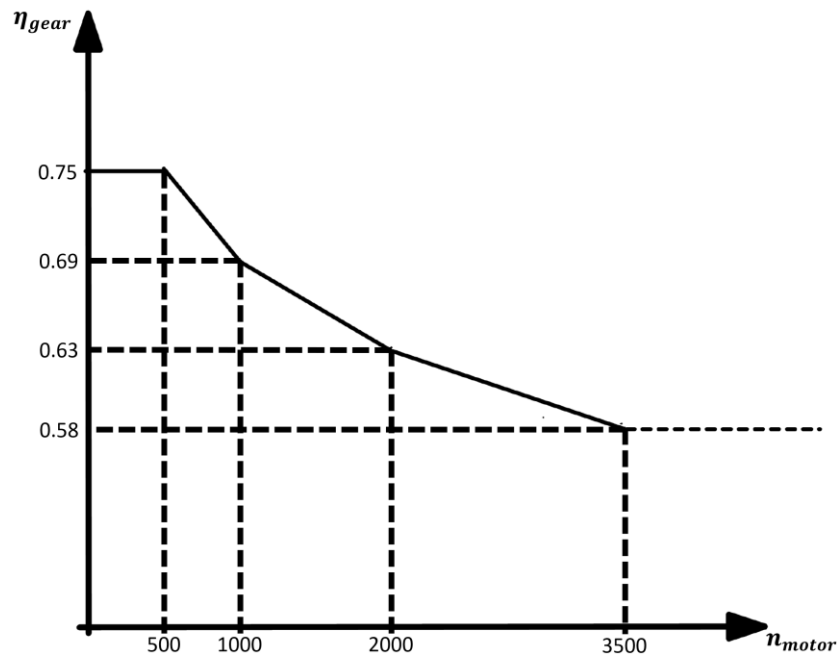


Рисунок 3.4 – Лінійна залежність між ККД редуктора та швидкістю обертання двигуна

В подальшому для проведення розрахунку крутного моменту двигуна, необхідно виділити два відмінні випадки – якщо двигун знаходиться в стані рекуперації та якщо двигун знаходиться у стані прискорення. Для цього використовуються наступні формули, відповідно:

$$\tau_{motor}^j(t) = \frac{\tau^j(t)}{R} \cdot n_{motor} \cdot n_g.
 \tag{3.17}$$

$$\tau_{motor}^j(t) = \frac{\tau^j(t)}{R} \cdot \frac{1}{n_{motor} \cdot n_g}.
 \tag{3.18}$$

ККД двигуна в такому випадку є константою і становить $n_{motor} = 0,92$. Таким чином, розрахунок потужності двигуна зводиться до формули:

$$P_{motor}^j(t) = 2\pi \cdot n_{motor}^j(t) \cdot \tau_{motor}^j(t). \quad (3.19)$$

Для визначення втрат потужності необхідно мати значення струму $I_{motor}^j(t)$, який буде пропорційний крутному моменту:

$$\tau_{motor}^j(t) \sim I_{motor}^j(t). \quad (3.20)$$

Використовуючи таке співвідношення, струм двигуна та крутний момент можна між собою пов'язати константами пропорційності, яка отримає назву константи крутного моменту k_M . Для робота-маніпулятора Panda Arm ця константа становить 0,134. Звідси випливає:

$$\tau_{motor}^j(t) = I_{motor}^j(t) \cdot k_M \rightarrow I_{motor}^j(t) = \frac{\tau_{motor}^j(t)}{k_M}, \quad (3.21)$$

де k_M – константа крутного моменту;

$I_{motor}^j(t)$ – струм двигуна;

$\tau_{motor}^j(t)$ – крутний момент двигуна.

На основі 3.21 та 3.12 можна визначити загальну електричну потужність, яка передається на двигун:

$$P_{el}^j(t) = P_{motor}^j(t) + P_{loss}^j(t) = P_{motor}^j(t) + R_T \cdot I_{motor}^{j^2}(t). \quad (3.22)$$

3.2.3 Налаштування оцінки за математичною моделлю

Основна ідея роботи полягає в проведенні оцінки ефективності руху імітаційної моделі робота по траєкторіях. Генерування траєкторій руху робота в

Unity відбувається за концепцією параболічного змішування, описаної у розділі 3.2.1. Декілька конфігурацій шляху в просторі робота-маніпулятора Panda Arm визначались та перетворювались у траєкторію за формулою 3.7.

В загальному при моделюванні проводиться оцінка ряду траєкторій руху, які складаються із конфігурацій q_1, q_2, q_3 , які позначають відповідно початкову, проміжну та кінцеву конфігурацію та положення робота-маніпулятора на траєкторії руху. Для всіх траєкторій, які проходять оцінку, параметри q_1 та q_3 залишаються незмінними, оскільки відповідають початковому та кінцевому положенню (рис. 3.5). Змінюється при цьому лише проміжний параметр q_2 .

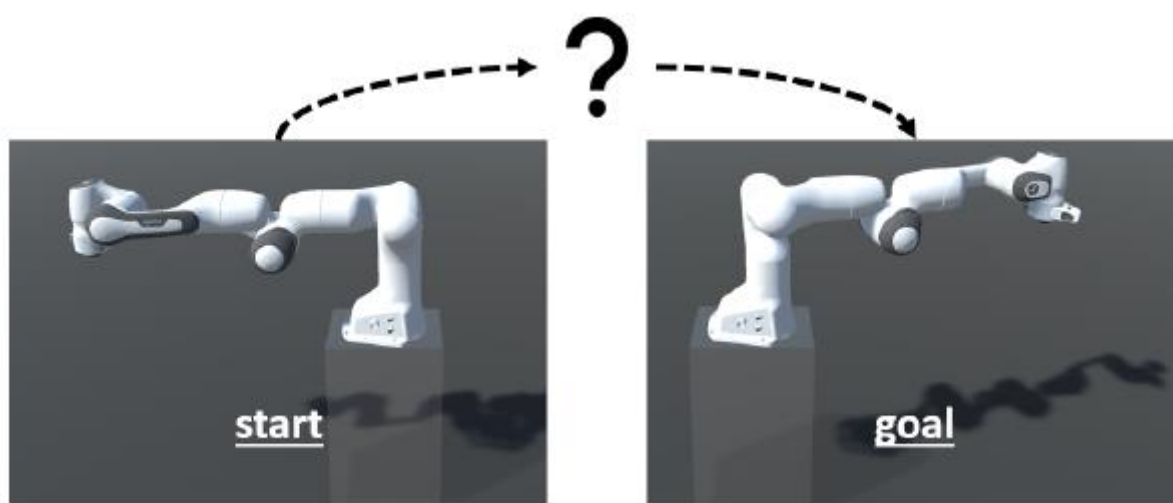


Рисунок 3.5 – Початкове, проміжне та кінцеве положення траєкторії руху Panda Arm

Обидва положення (кінцеве та початкове) відповідають положенню, коли рука робота-маніпулятора повністю витягнута у горизонтальному напрямку. Єдина відмінність між цими параметрами полягає у куті другого суглоба, тобто у 180 градусів. Таким чином, вибір початкової та кінцевої конфігурації, можна звести до наступного:

$$q_n = (q_n[1] \ 90^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 90^\circ \ 0^\circ), n \in \{1, 3\}, \quad (3.23)$$

де $q_1[1] = 0^\circ$ і $q_3[1] = -180^\circ$.

Шостий суглоб робота-маніпулятора в межах імітаційної моделі був додатково повернутий на 90° з метою вирівнювання кінцевого положення робота по вертикалі та імітації перенесення роботом якогось об'єкту.

Слід відзначити, що в Unity початкова конфігурація робота q_1 у залежності 3.23 автоматично встановлюється як змінна, в якій всі кути дорівнюють 0° . Тоді 3.23 для подальшої оцінки набуде вигляду:

$$q_n = (q_n[1] \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ), n \in \{1, 3\}, \quad (3.24)$$

де $q_1[1] = 0^\circ$ і $q_3[1] = -180^\circ$.

Такий рух робота можна прирівняти із рухами людини, яка бере предмет з горизонтальної поверхні, при чому предмет розташовується по ліву сторону від об'єкта (робота-маніпулятора).

Для спрощення процесу проведення оцінки за умови збереження різноманітності траєкторій руху робота-маніпулятора Panda Arm було введено проміжну конфігурацію q_2 :

$$q_2 = (-90^\circ \ q_2[2] \ 0^\circ \ q_2[4] \ 0^\circ \ 0^\circ \ 0^\circ). \quad (3.25)$$

Граничними випадками положень робота-маніпулятора тоді будуть:

- повністю витягнуте положення руки робота - $q_2[4] = 0^\circ, q_2[2] = -100^\circ$;
- плече в рівному положенні, а лікоть зігнутий під кутом 140° - $q_2[4] = 140^\circ, q_2[2] = 0^\circ$;
- рука горизонтально розташована та рівна, відбувається лише обертання навколо другого суглоба - $q_2[4] = -0^\circ, q_2[2] = 0^\circ$;
- обидва суглоби знаходяться в русі, при чому другий відповідає за підняття руки догори, а четвертий за згинання ліктя - $q_2[4] = -100^\circ, q_2[2] = 140^\circ$.

Для виконання реальної оцінки траєкторії руху робота-маніпулятора необхідно використати об'єктивні значення, тобто властивості мехатронної системи Panda Arm.

3.3 Підйом об'єктів за допомогою Panda Arm

В якості прикладу в процесі моделювання роботи робота-маніпулятора Panda Arm було обрано принцип автоматизованого збору та підйому коробок. Такий вибір базується на все ширшому застосуванні роботів Panda Arm саме для роботи із коробками на роботизованих підприємствах.

На рис. 3.6 наведено приклад використання Panda Arm на промисловому підприємстві. Як видно з рис., завдання робота полягає у виявленні в коробках предметів з метою неможливості пропуску на лінію пустої коробки.



Рисунок 3.6 – Приклад використання Panda Arm при роботі із коробками

Розглянутий приклад передбачає напівавтоматизоване виробництво, в якому поєднується ручна та роботизована праця. Тому при розрахунку траєкторії руху робота слід враховувати безпеку працівника на зображенні 3.6. Другий оператор використовує модуль телеуправління, за допомогою якого можна відновити правильні положення робота в процесі роботи. Для цього оператор використовує джойстик.

Початковою точкою для оцінки імітаційної моделі функціонування Panda Arm буде збір вмісту коробок. Така робота передбачає рух по складним траєкторіям. Генерація траєкторій руху по серії точок проводиться у декартовому просторі. Позиції руху при зборі коробок представлені наступним чином:

1. Положення над коробкою, яка містить об'єкт, що необхідно забрати (p_1);
2. Положення руки всередині коробки при зборі об'єкта (p_2);
3. Положення над коробкою, об'єкт знаходиться в захваті ($p_3 = p_1$);
4. Положення над конвеєром, куди необхідно розмістити об'єкт із руки (p_4);
5. Положення, близьке до конвеєра, для опущення об'єкта на конвеєр (p_5);
6. Положення над конвеєром після того, як об'єкт було вивільнено ($p_6 = p_4$);
7. Переміщення над коробкою для збору наступного об'єкту ($p_7 = p_3 = p_1$).

Вищезазначені точки задаються у відповідній бібліотеці планування траєкторії руху суглобів робота-маніпулятора через ROS, в результаті чого можна отримати клас `GetCartesianPathResponse`. Таке декартове планування траєкторії руху представляє собою список позицій суглоба, прискорень та швидкостей його руху наряду із таймінгами. Крім того, у цьому класі міститься число із зазначенням відсотку того, які траєкторії можуть виконуватись без зіткнень з перешкодами.

Для створення симуляції в Unity було використано модель промислового заводу, яка представляє собою CAD-файл [17]. Для генерації траєкторій руху робота та підйому об'єктів з конвеєрної лінії необхідно провести моделювання тих об'єктів в рамках підприємства, які можуть представляти собою перешкоду на шляху траєкторії.

Для створення віртуального прототипу процесу збору предметів із коробки в Unity, необхідно додати згенеровану траєкторію руху до відповідного виконавчого файлу. Зміна положення робота-маніпулятора в межах промислового підприємства передбачає перепланування траєкторії руху при зборі контейнерів. Це пояснюється тим, що всі перешкоди на траєкторії руху та сім декартових координат руху суглобів робота залежать від його просторового положення. Отже, для кожного такого положення необхідно проводити планування нової траєкторії руху.

Для проєктування процесу інтеграції ROS в середовище, що розробляється, було виконано наступне:

- встановлено з'єднання між ROS та кожного виконавчого файлу Unity. Ці файли викликаються під час проведення оцінки функціонування віртуального прототипу. Таким чином, для кожної позиції робота в тривимірному просторі встановлюється з'єднання між операційною системою ROS та модулем, розробленим на Unity. Після цього кожне із з'єднань використовується для подальшого планування траєкторії руху робота, а також ця траєкторія передається назад до виконавчого файлу Unity;

- створення підключення .NET модулю для виконання та проведення оцінки системи ROS один раз перед входом у виконання основного циклу проведення оцінки функціональності імітаційної моделі. Такий підхід дозволяє встановити з'єднання із ROS, після чого проводиться послідовне виконання та планування всіх траєкторій руху робота, які відповідають його положенням, зазначеним користувачем. Після цього виконується передача відповідної траєкторії кожному із виконавчих файлів Unity.

Першому варіанту виконання характерна повільна робота. Індивідуальне підключення кожного із виконавчих файлів Unity до ROS буде означати, що для кожного віртуального прототипу роботи Panda Arm необхідно встановити з'єднання та відправити поточне положення робота. Лише після цього відбувається планування траєкторії руху та її повернення у виконавчий файл. Тому, в рамках виконання кваліфікаційної роботи, було обрано другий підхід. Йому характерне паралельне виконання процесів та запуску одного виконавчого файлу на кожному із ядер процесора.

Підйом об'єктів з конвеєра за допомогою робота-маніпулятора передбачає проведення оцінки різних позицій Panda Arm між конвеєром та коробкою (рис. 3.7). При установці робота-маніпулятора занадто далеко або занадто близько до коробки може призвести до того, що Panda Arm не зможе виконати повний цикл своєї роботи. Слід відзначити, що різні положення робота можуть збільшувати ймовірність утворення зіткнень на траєкторії його руху. Як додатковий елемент

можна провести оцінку між різними геометриями коробки та власне здатністю Panda Arm дістатися до об'єктів, які знаходяться всередині коробки.

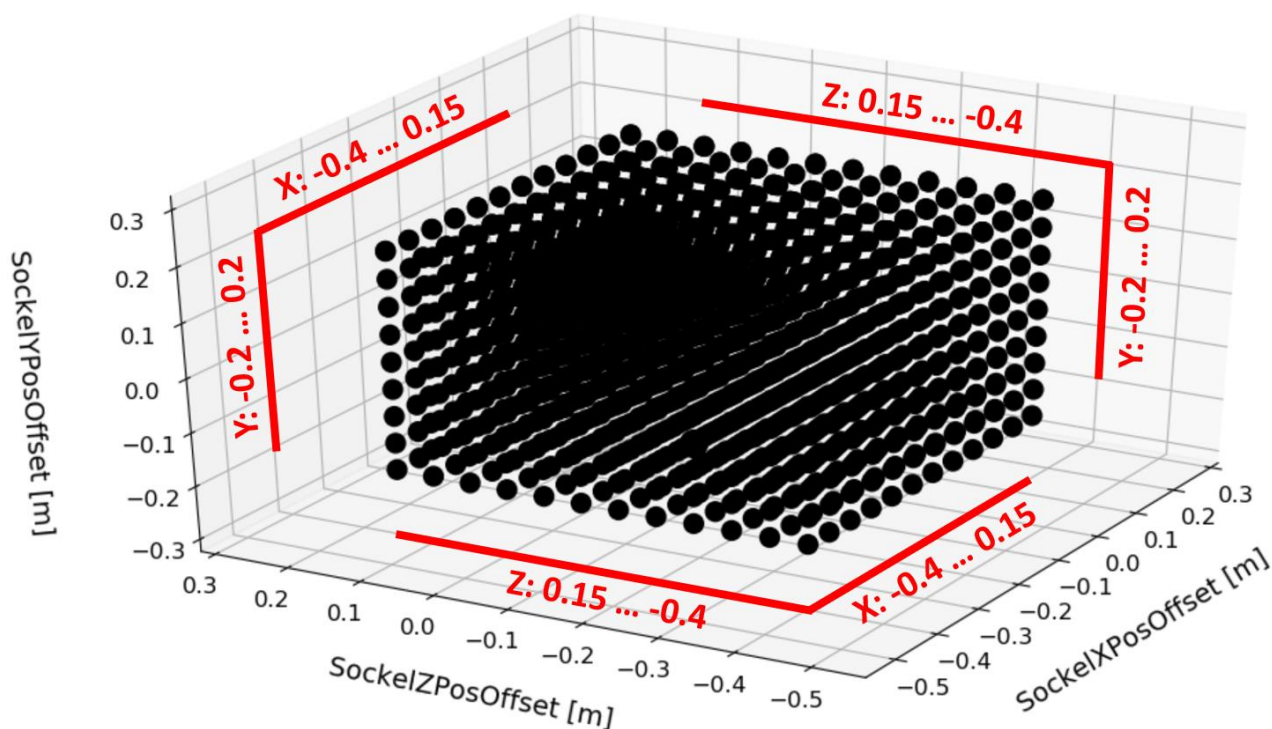


Рисунок 3.7 – Тривимірні координати робота-маніпулятора на промисловому підприємстві в процесі підйому вмісту коробок з конвеєра

При проведенні оцінки функціонування цифрового двійника враховуються параметри зміщення по координатам x , y та z , які відповідають змінним x_p , y_p , z_p відносно нульової позиції. На рис. 3.7 зображено параметри проведення оцінки траєкторії руху та всіх комбінацій цих параметрів. В результаті параметри утворюють куб в просторі проведення оцінки. Розміри по осям відповідають напрямкам та відповідають положенням руху робота-маніпулятора відповідно до рис. 3.6 наступним чином:

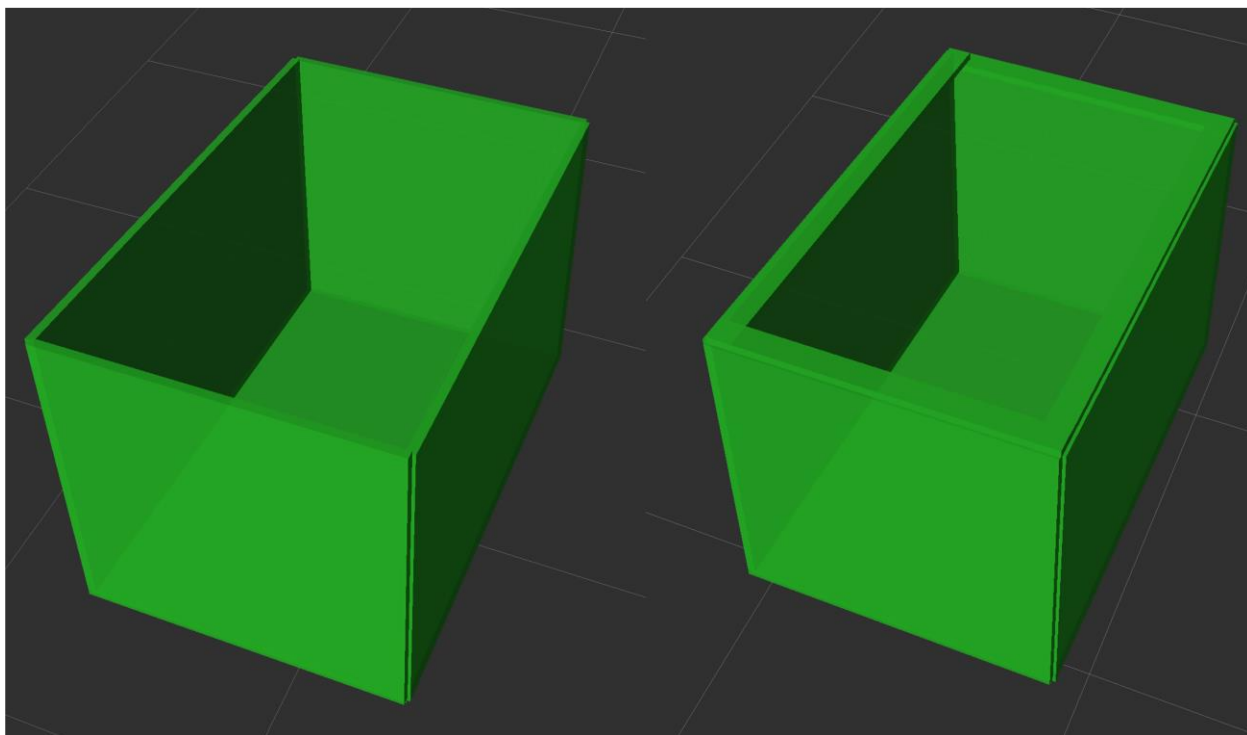
- додатна вісь x відповідає напрямку на конвеєр;
- додатна вісь y відповідає напрямку, перпендикулярному конвеєру;
- додатна вісь z відповідає напрямку в сторону коробки.

Максимальні та мінімальні значення, які були обрані для кожного переміщення позиції робота показані через формулу:

$$x_p \in [-0,4 \text{ м}, 0,15 \text{ м}]; y_p \in [-0,2 \text{ м}, 0,2 \text{ м}]; z_p \in [-0,4 \text{ м}, 0,15 \text{ м}]. \quad (3.26)$$

Діапазон значень по осям були обрані із врахуванням правильного положення робота-маніпулятора в межах простору підприємства без можливого зіткнення з перешкодами. Діапазон зміщення робота були обмежені положенням, при яких робот-маніпулятор не зміг би досягнути конвеєра або коробки. При проведенні оцінки параметрів крок був встановлений на значенні $s = 0,05 \text{ м}$.

Для проведення оцінки до простору оцінки було додано додатковий параметр. Цей параметр описується булевою змінною та відповідає за власне особливу геометрію коробки на конвеєрі. Наприклад, на рис. 3.8 зображено дві коробки із різною геометрією. Особливою геометрією коробки вважається не звичайний куб, а наявність в неї звисаючих частин. Тобто отвір коробки з положення зверху стає меншим за розмірами, що може збільшити складність в роботі Panda Arm при захваті об'єкта із коробки без зіткнення із останньою. В подальшому такі коробки умовно називатимуться простою (рис. 3.8, а) та складною (рис. 3.8, б).



а)

б)

Рисунок 3.8 – Проста та складна геометрія коробки

3.4 Висновки до третього розділу

У розділі наведено два конкретні приклади використання розробленого методу оцінки функціональності робота-маніпулятора, які демонструють можливість застосування методу.

Основна увага при цьому приділяється створенню імітаційної моделі роботи Panda Arm. Імітаційна модель представляє собою два додатки, в яких проводиться моделювання роботи робота-маніпулятора Panda Arm. Крім того, в розділі показано параметри оцінки та об'єктивні значення, які використовувались для оцінки функціональності.

В розділі показано, що робот-маніпулятор Panda Arm можна перетворити на цифрового двійника в площині Unity, що здатен відобразити реального робота на високому за точністю рівні. Також в розділі наведено деталізацію використання розробленого додатку для оцінки траєкторії робота-маніпулятора та автоматизованого збору предметів, на прикладі збору сміття.

Також в розділі наведено математичну модель опису траєкторії руху робота із врахуванням його кінематичних та динамічних параметрів на основі методу параболічного змішування. Крім того, розглянута математична модель оцінки енергоефективності роботи Panda Arm, при якій рух по траєкторії буде виконуватись із максимально можливою швидкістю та мінімальною витратою енергії.

4 ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ РОЗРОБЛЕНОГО МЕТОДУ ОЦІНЮВАННЯ РОБОТА МАНІПУЛЯТОРА

4.1 Встановлення залежностей між параметрами руху робота та енергоспоживанням

Основною метою проведення оцінки є аналіз енергоефективності робота-маніпулятора Panda Arm за умови різних траєкторій руху. Параметрами оцінки будуть четвертий та другий суглоби, позначені q_4 [4] та q_2 [2] відповідно. Загальне енергоспоживання роботом розраховується як сума енергоспоживання на кожному із зазначених суглобів, які, в свою чергу, розраховувались за методикою, наведеною в розділі 3.

На рис. 4.1 показано загальне енергоспоживання Panda Arm для всіх комбінацій параметрів траєкторії руху, тобто для всіх траєкторій, які були створені в Unity.

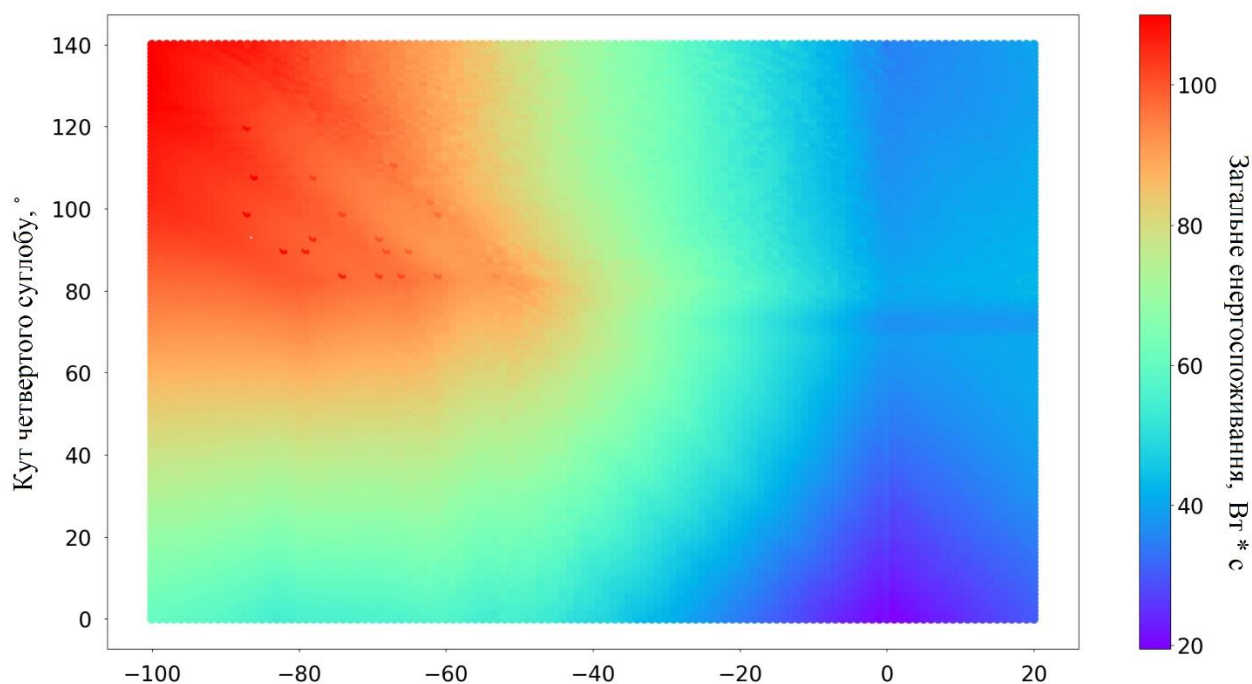


Рисунок 4.1 – Загальне енергоспоживання Panda Arm для всіх траєкторій руху

На рис. 4.1 вісь x відповідає параметру q_2 [2], діапазон якого становить від -100 до 20 (крок становить 1). Вісь y відповідає параметру q_4 [4], діапазон становить від 0 до 140. Енергоспоживання, яке необхідне для виконання роботом руху по траєкторії, вимірюється у $\text{Вт} \cdot \text{с}$ і пофарбоване у кольори, де червоний відповідає максимальним значенням, а синій – мінімальним.

Як видно з рис. 4.1, для траєкторій з низьким енергоспоживанням можна виокремити скупчення значень оціночних параметрів q_2 [2] і q_4 [4]. В загальному спостерігається тенденція до зростання енергоспоживання із збільшенням швидкостей другого та четвертого суглобів робота-маніпулятора.

При виконанні руху по траєкторії Panda Arm здійснює рухи в основному на першому, другому та четвертому суглобах. Енергоспоживання на цих суглобах зображено на рис. 4.2, 4.3 та 4.4 відповідно.

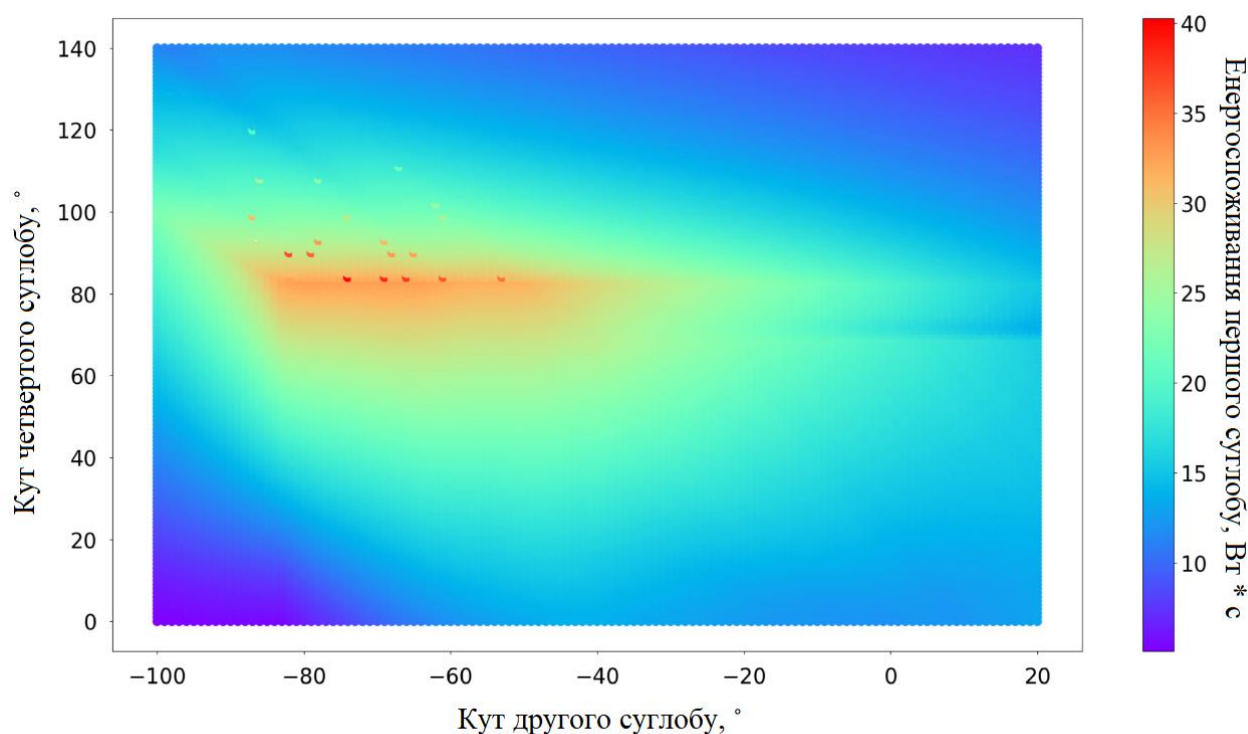


Рисунок 4.2 – Енергоспоживання першого суглобу Panda Arm

Як видно з рис. 4.2, найвищі значення енергоспоживання для першого суглобу досягається при виконанні руху по траєкторії, яка лежить в межах від -85° до -40° для параметру q_2 [2] і приблизно 85° для параметру q_4 [4].

Максимальне споживання енергії при цьому становить $E_{el} = 40,27$ Вт · с.
 Найменше споживання електричної енергії для першого суглобу становить $E_{el}[1] = 5,08$ Вт · с досягається при значеннях $q_2[4] = 0^\circ, q_2[2] = -100^\circ$.

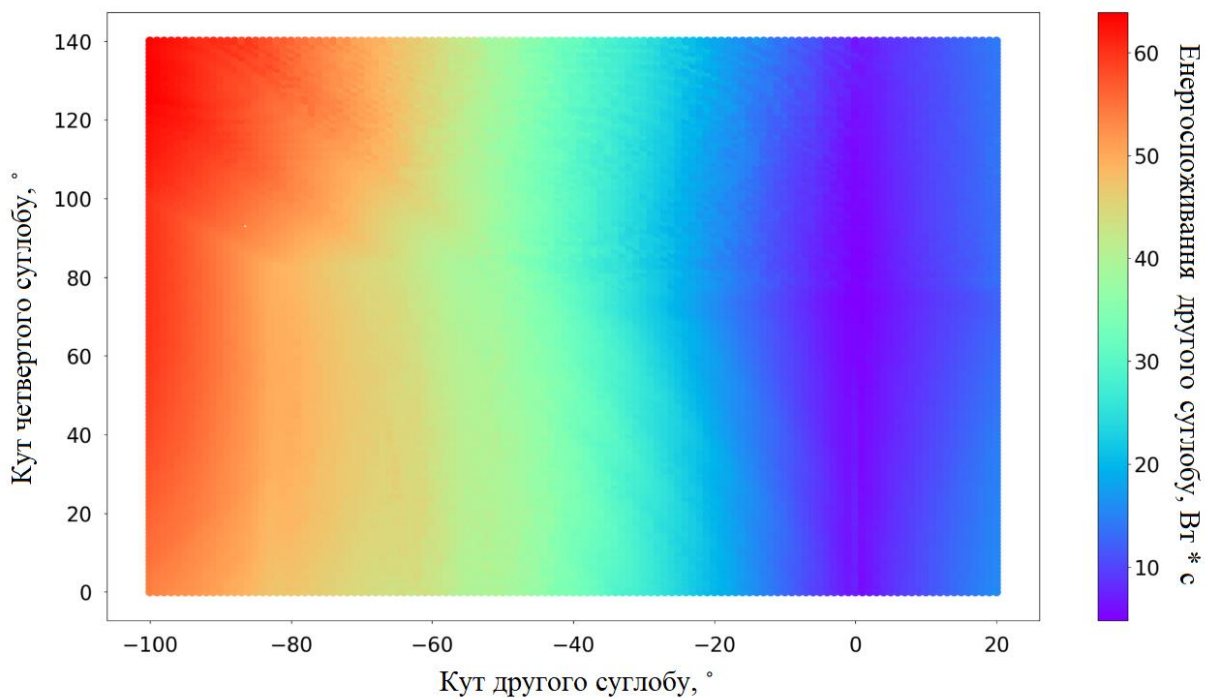


Рисунок 4.3 – Енергоспоживання другого суглобу Panda Arm

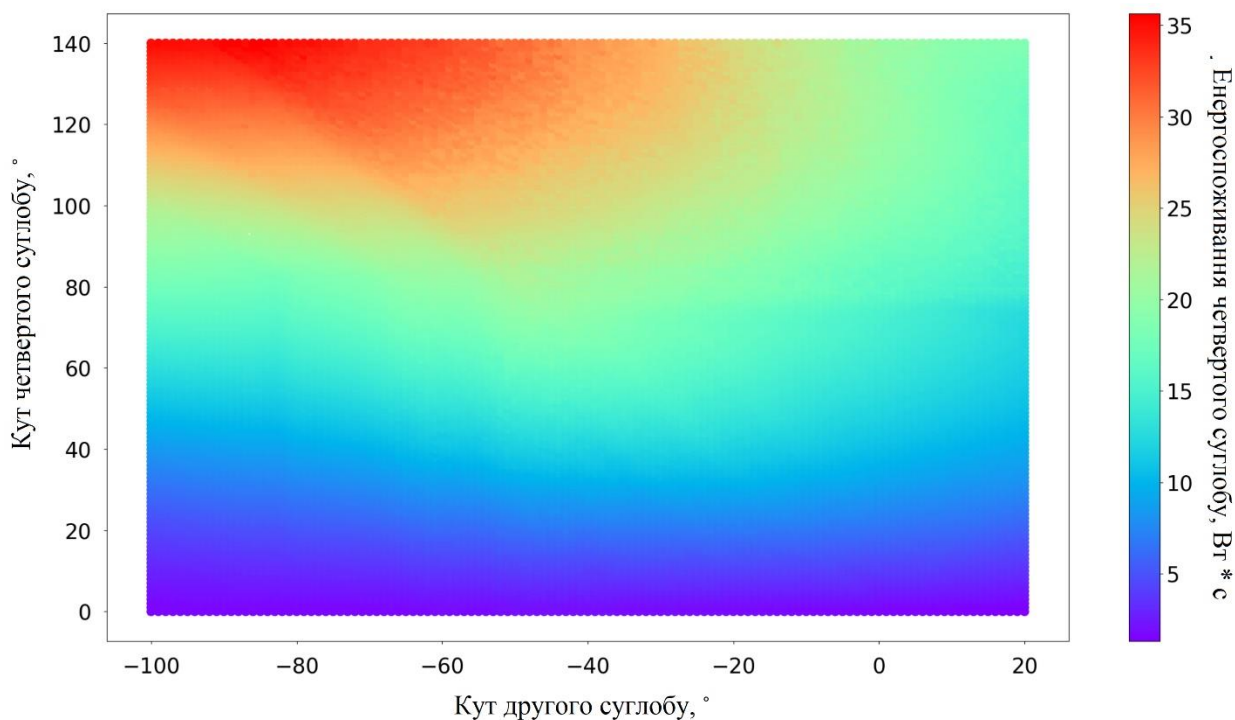


Рисунок 4.4 – Енергоспоживання четвертого суглобу Panda Arm

Аналогічно для другого суглобу мінімальне енергоспоживання становить $E_{el}[2] = 4,78 \text{ Вт} \cdot \text{с}$ досягається при значеннях $q_2[4] = 76^\circ, q_2[2] = 1^\circ$, а максимальне становить $E_{el}[2] = 63,88 \text{ Вт} \cdot \text{с}$ і досягається при значеннях $q_2[4] = 126^\circ, q_2[2] = -100^\circ$.

Для четвертого суглобу ці значення становлять $E_{el}[2] = 1,31 \text{ Вт} \cdot \text{с}$ (мінімальне) при $q_2[4] = 0^\circ, q_2[2] = 20^\circ$ і $E_{el}[2] = 35,64 \text{ Вт} \cdot \text{с}$ (максимальне) при $q_2[4] = 140^\circ, q_2[2] = -86^\circ$.

В результаті спостерігається залежність, що чим менший кут четвертого суглоба $q_2[4]$, тим менше енергії буде споживати робот-маніпулятор при руху по траєкторії.

Для кращого сприйняття інформації стосовно того, як відбувається споживання енергії роботом-маніпулятором при виконанні різних траєкторій руху було побудовано графіки, показані на рис. 4.5 та 4.6. На цих графіках показана залежність між загальним енергоспоживанням та відстанню, яку пройшов робот Panda Arm.

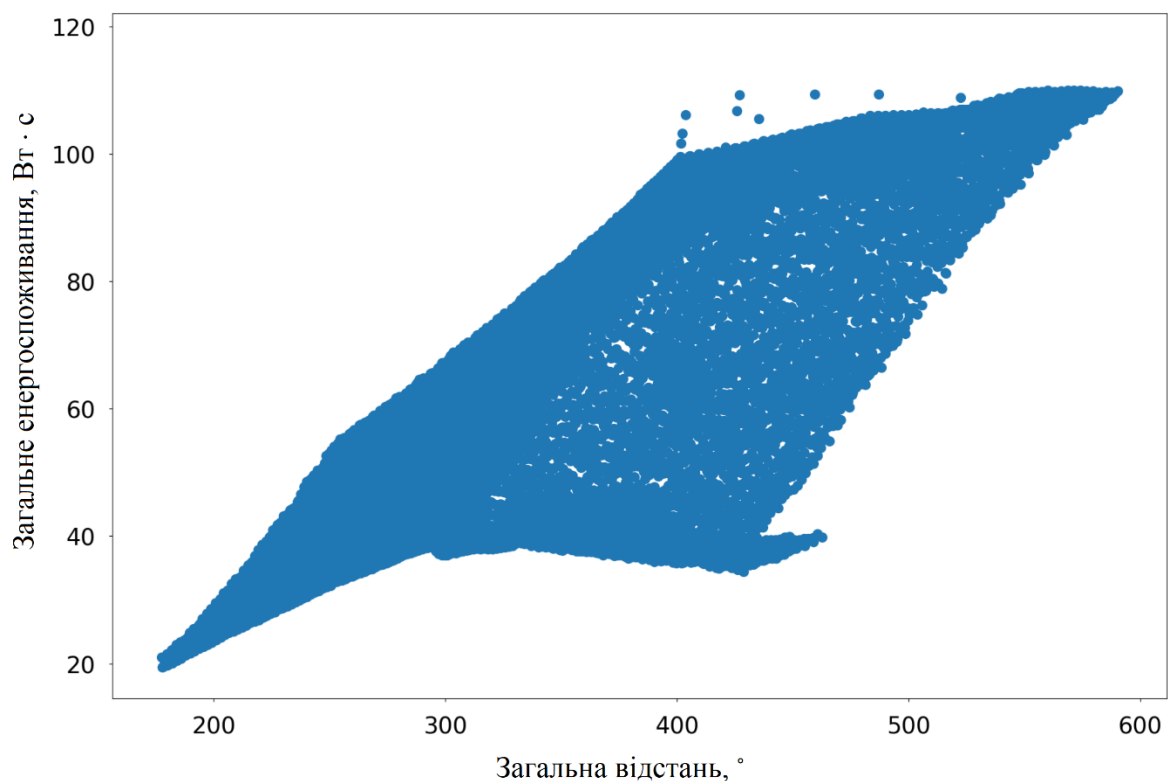


Рисунок 4.5 – Споживання електроенергії при проходженні загальної відстані

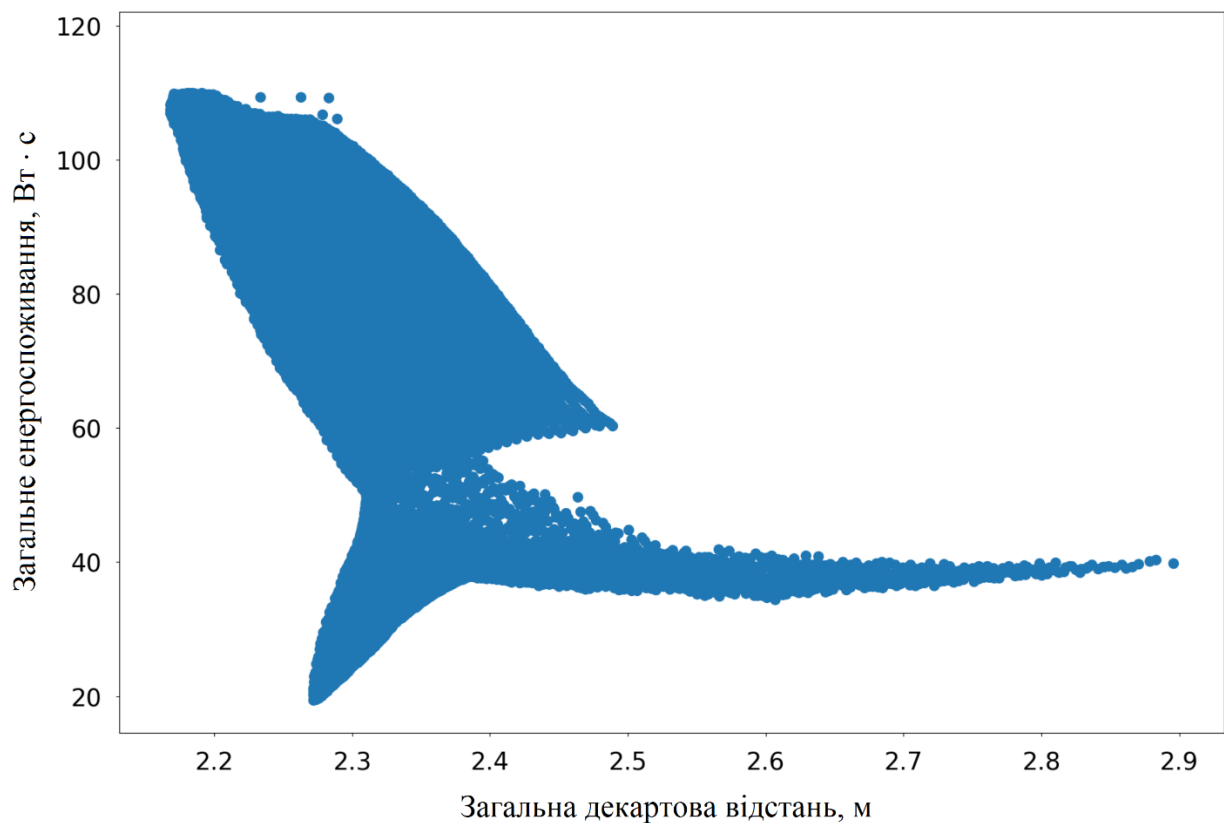


Рисунок 4.6 - Споживання електроенергії при проходженні відстані в декартовій системі

Для всіх траєкторій руху, які моделювались в процесі виконання поставлених в кваліфікаційній роботі задач, наявна залежність між енергоспоживанням робота-маніпулятора та загальною відстанню, яку виконують суглоби при русі по траєкторії (s_j). Ця залежність показана на рис. 4.5. Загальна відстань руху суглобу отримується простим підсумовуванням значень зміни кута суглоба для всіх проміжків часу.

Як видно з рис. 4.5, на всіх траєкторіях, які були змодельовані, найменша відстань становить $s_j = 177,5^\circ$ при значеннях $q_2[4] = 0^\circ, q_2[2] = 0^\circ$, а найбільша відстань, яку проходить робот становить $s_j = 590,2^\circ$ при значеннях $q_2[4] = -100^\circ, q_2[2] = 140^\circ$. Слід відзначити, що траєкторія, якій характерна найбільша відстань (найбільша довжина) відповідає тій траєкторії руху, при якій досягається максимальне значення енергоспоживання ($E_{el} = 109,95$ Вт · с). Аналогічним чином, траєкторія із найменшою загальною відстанню, тобто найменшою

довжиною, відповідає траєкторії із найменшим споживанням енергії ($E_{el} = 19,46 \text{ Вт} \cdot \text{с}$).

Максимальне споживання енергії досягається при виконанні роботоманіпулятором рухів, які відповідають куту 400° або більше. При цьому на певному значенні досягається стрімке зростання енергоспоживання. Наприклад, при проходженні суглобами загального кута $428,8^\circ$ споживання енергії становить $E_{el} = 34,42 \text{ Вт} \cdot \text{с}$, а при значенні 429° споживання енергії стрімко зростає до $100,21 \text{ Вт} \cdot \text{с}$.

Аналогічним чином розглядалось споживання електричної енергії в розрізі проходження загальної відстані, яку можна виразити в метрах, а не в куті згину суглобів. Ці залежності показані на рис. 4.6. При значеннях $q_2[4] = 140^\circ$, $q_2[2] = -100^\circ$ робот Panda Arm проходить найменшу відстань в метрах, яка становить $s_{dec} = 2,17 \text{ м}$. На відміну від суглобової відстані, яка виражається в загальному куті згину, мінімальному значенню декартової відстані відповідає найвище споживання енергії, яке становить $E_{el} = 110,02 \text{ Вт} \cdot \text{с}$. При цьому спостерігається зворотна залежність, оскільки мінімальному значенню енергоспоживання відповідає максимальне значення відстані, яку проходить робот-маніпулятор, а саме $E_{el} = 39,86 \text{ Вт} \cdot \text{с}$ при $s_{dec} = 2,90 \text{ м}$. Ці значення досягаються при $q_2[4] = 140^\circ$, $q_2[2] = 20^\circ$.

При більш детальному розгляді рис. 4.6 можна помітити скупчення траєкторій руху, яким характерний високий рівень споживання електроенергії при відносно малих відстанях траєкторій руху. Після аналізу змодельованих даних можна зробити висновок, що найбільш оптимальним значенням відстані в розрізі мінімального енергоспоживання стане траєкторія руху, яка має загальну довжину $2,27 \text{ м}$.

Крім того, необхідно оцінити час проходження робота-маніпулятора по тій чи іншій траєкторії руху. Це моделювання відбувалось відповідно до математичної моделі, наведеної в розділі 3, а результати зображені на рис. 4.7. Аналогічним чином на рис. 4.8 показано загальну відстань, яку проходить робот-маніпулятор, при виконанні тих чи інших траєкторій.

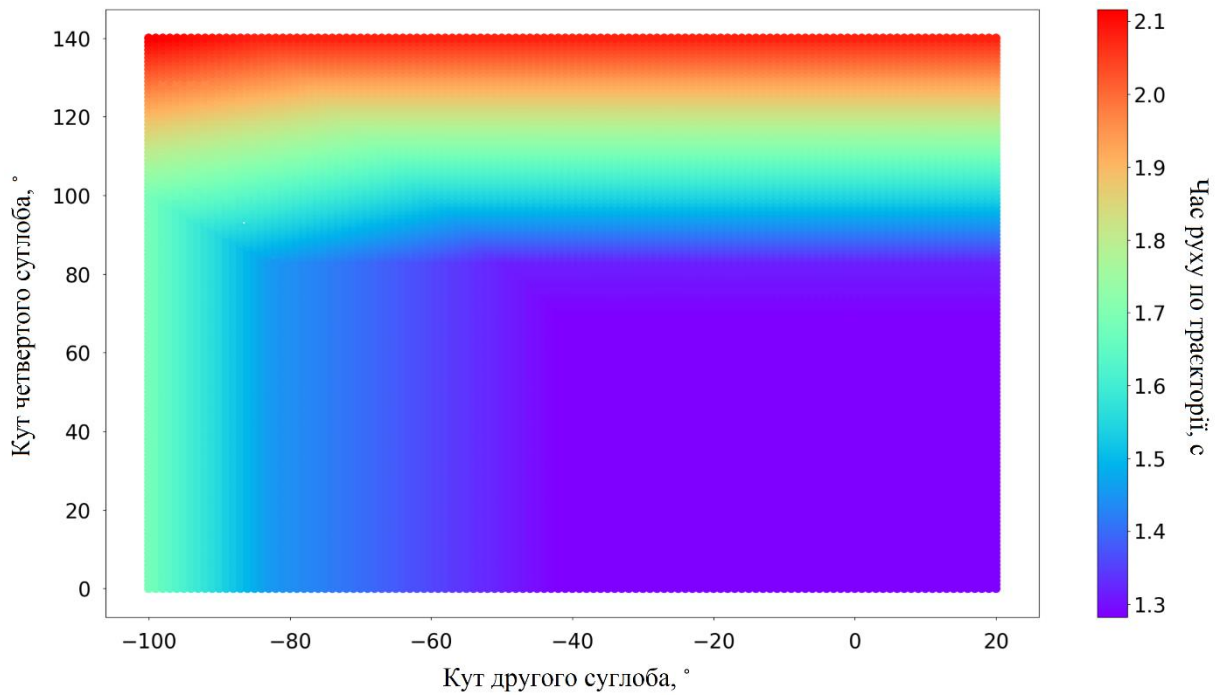


Рисунок 4.7 – Час руху по змодельованим траєкторіям

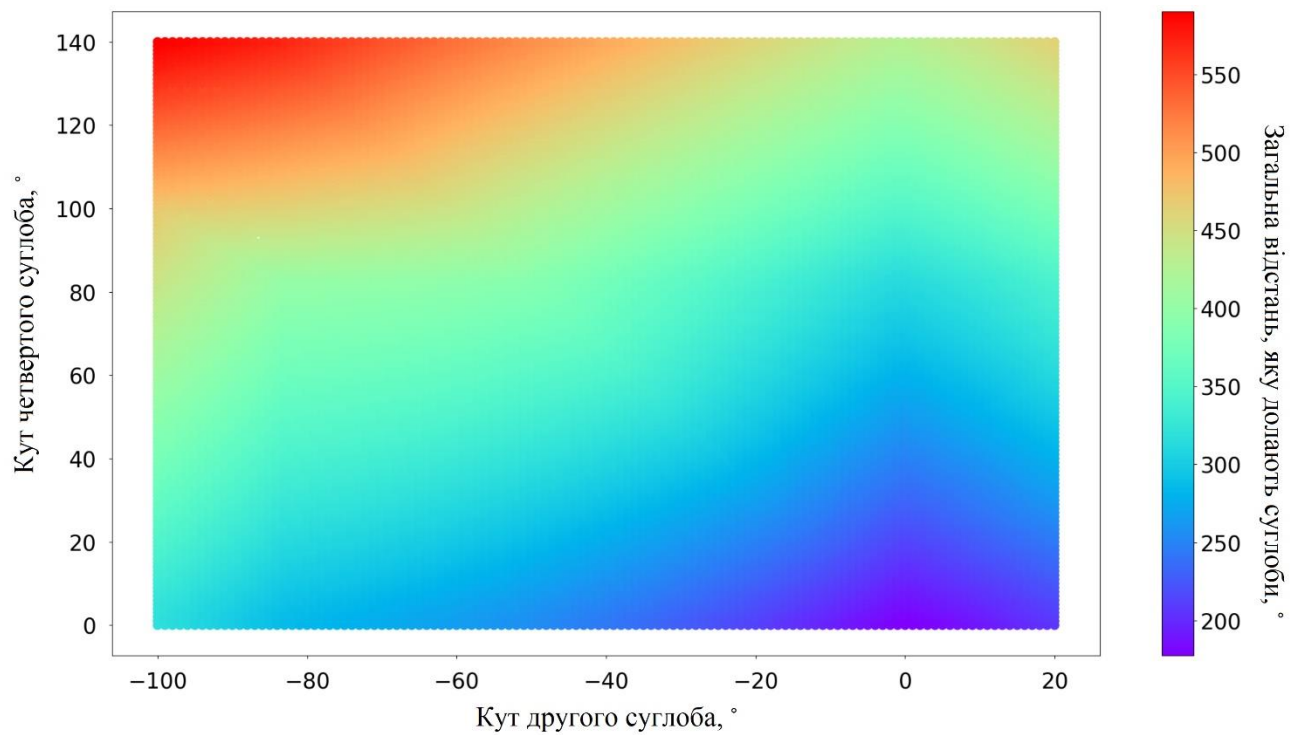


Рисунок 4.8 – Загальна відстань, яку долають суглоби при проходженні всіх траєкторій

Додатковими параметрами для оцінки енергоспоживання можуть бути не тільки виконання роботом-маніпулятором рухів в розрізі зміни загального кута суглоба, але й довжина траєкторії, по якій виконуються рухи. Також параметром

для оцінки буде швидкість проходження по траєкторії. Змодельовані довжини всіх траєкторій наведені на рис. 4.9, а швидкості – на рис. 4.10-4.12.

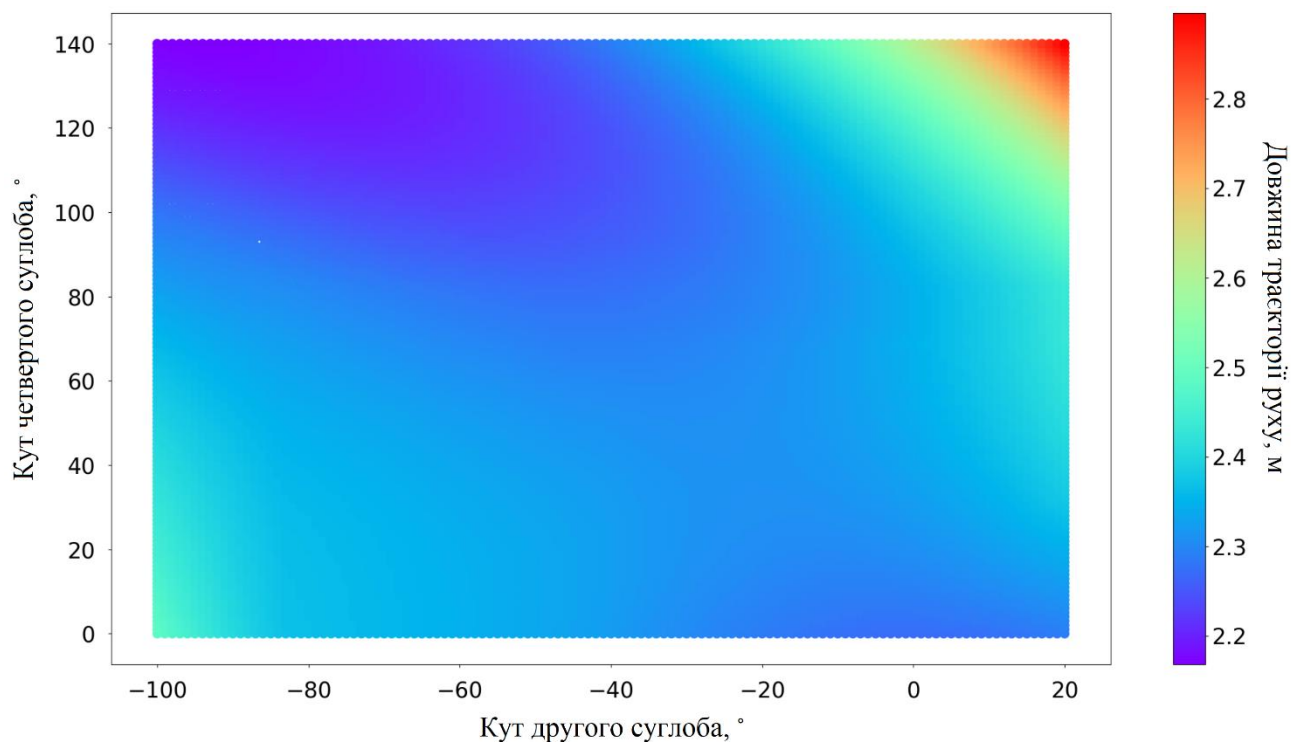


Рисунок 4.9 – Зміна довжини траєкторії руху робота-маніпулятора

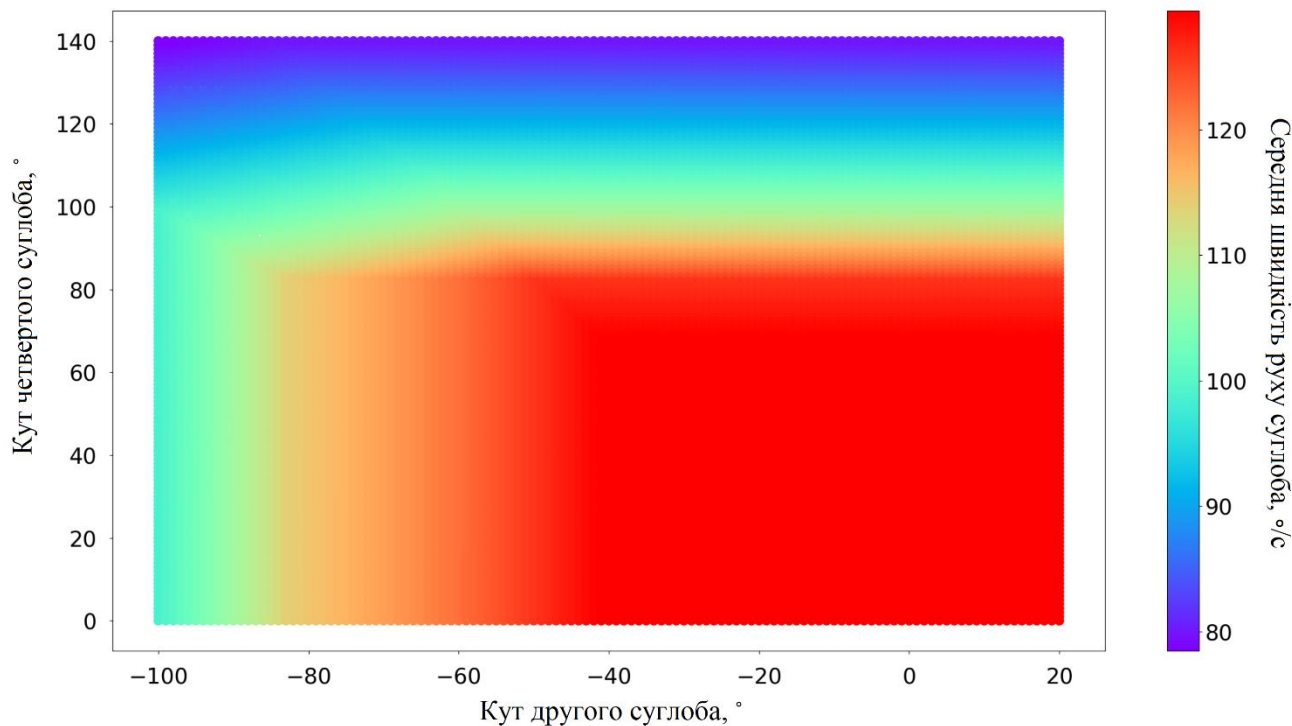


Рисунок 4.10 – Середня швидкість руху першого суглоба по всім траєкторіям руху

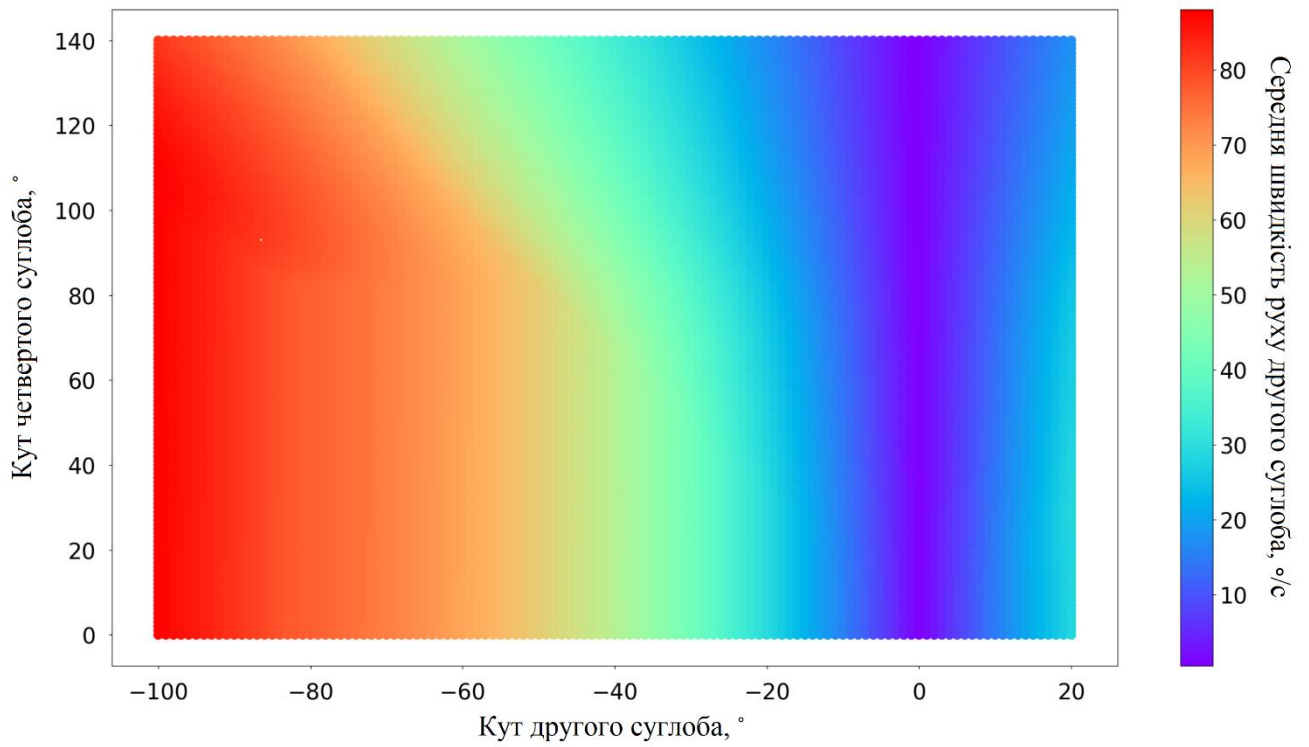


Рисунок 4.11 - Середня швидкість руху другого суглоба

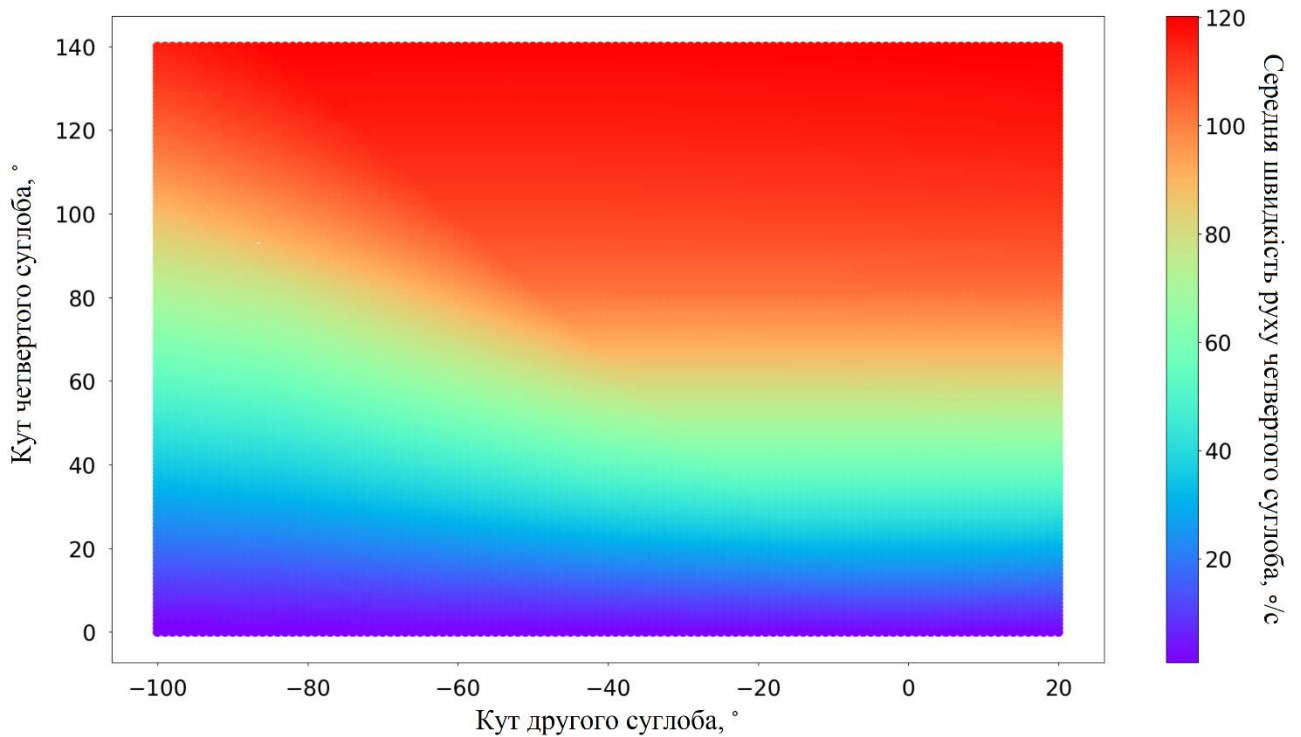


Рисунок 4.11 - Середня швидкість руху четвертого суглоба

На рис. 4.12 показане загальне споживання електроенергії при різних значеннях довжини та куту згину суглобів на різних змодельованих траєкторіях руху. Нажаль, чітка тенденція не спостерігається, але завдяки проведеному

моделювання стало можливим знайти скупчення значень, яким характерний високий рівень енергоспоживання. Цим значенням також характерна мала довжина траєкторії руху робота-маніпулятора, високі значення кутів згину суглобів робота та тривалий час руху по траєкторії.

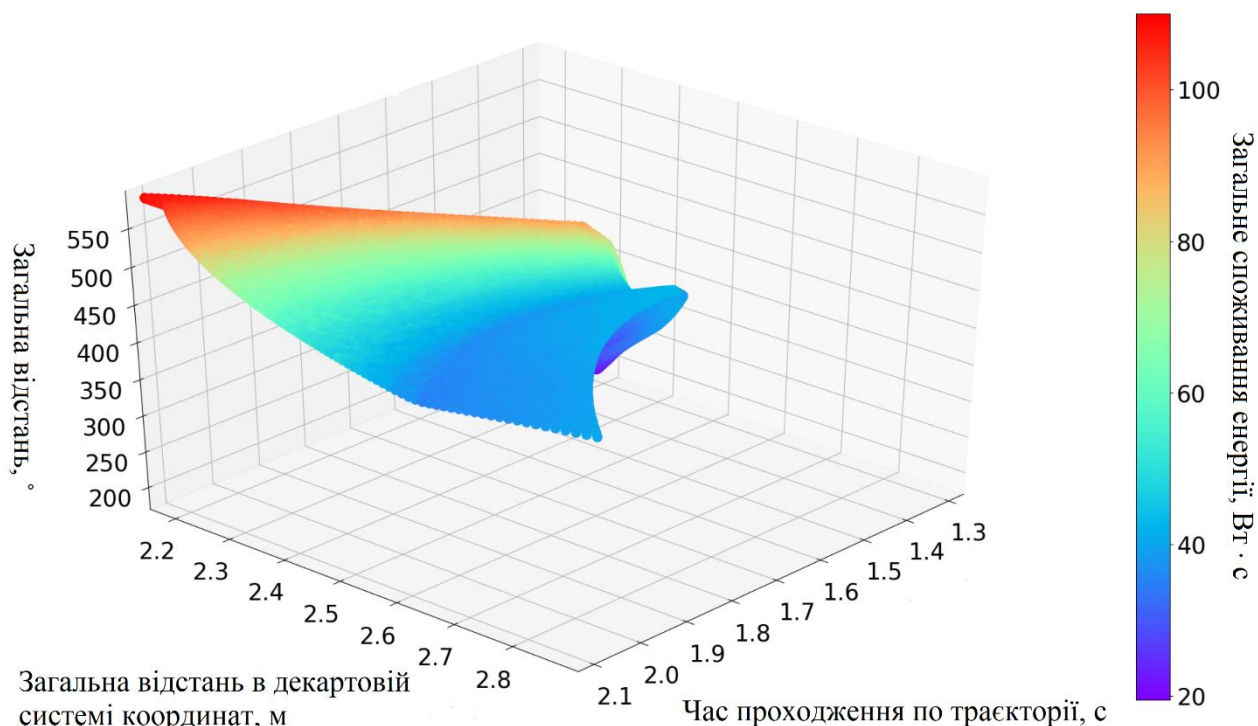


Рисунок 4.12 – Загальне споживання енергії при проходженні всіх траєкторій

Отже, мінімальні та максимальні значення енергії, що споживається роботом-маніпулятором при проходженні по всіх траєкторіях руху, що були змодельовані, можна звести до табличного вигляду. Ці значення наведені в табл. 4.1.

Таблиця 4.1 – Отримані параметри оцінки траєкторії руху робота

Назва	Позначення	Значення		Одиниці вимірювання
		Мін.	Макс.	
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
Кут четвертого суглоба	q_2 [4]	0	140	°
Кут другого суглоба	q_2 [2]	-100	20	°

Продовження таблиці 4.1.

1	2	3	4	5
Час виконання траєкторії	t	1,28	2,12	с
Середня швидкість суглоба 1	$q_{\text{сер}} [1]$	78,5	129,5	°/с
Середня швидкість суглоба 4	$q_{\text{сер}} [4]$	0,8	120,3	°/с
Середня швидкість суглоба 2	$q_{\text{сер}} [2]$	0,5	88	°/с
Загальна декартова відстань	s_{dec}	2,17	2,9	м
Загальна відстань суглобів	s_j	177,5	590,2	°
Енергоспоживання суглоба 1	$E_{el} [1]$	5,08	40,27	Вт · с
Енергоспоживання суглоба 4	$E_{el} [4]$	1,31	35,64	Вт · с
Енергоспоживання суглоба 2	$E_{el} [2]$	4,78	63,88	Вт · с
Загальне енергоспоживання	E_{el}	19,46	110,02	Вт · с

4.2 Оцінка підйому об'єктів із коробок

Проведення оцінки процесу підйому об'єктів із коробок, які рухаються по конвеєру, за допомогою Panda Arm буде використовуватись з точки зору проведення оцінки позиціонування робота-маніпулятора. Для цього необхідно провести визначення позицій робота-маніпулятора, при яких він здатен виконувати технологічний процес, передбачений на промисловому підприємстві.

В подальшому необхідно провести аналіз позиціонування роботу з точки зору відстані, яку він проходить для виконання технологічного процесу, а також часу виконання цього процесу. Для такого порівняння була побудована тривимірна сітка, на якій показані основні параметри проведення оцінки (див. рис. 3.7).

Для успішного виконання процесу підйому об'єктів із коробок з конвеєрної лінії необхідно, щоб робот пройшов всі маршрутні точки p_i в кількості семи штук, при чому без зіткнень, тобто $i \in \{1 \dots 7\}$.

На рис. 4.13 та 4.14 показано здатність робота-маніпулятора повністю виконати процес підйому об'єктів із коробки на конвеєрній лінії. Цей процес був

змодельований для всіх позицій робота, які відповідають змінним зміщення по координатам x , y , z . Червоним кольором позначені позиції, при яких процес збору об'єктів із коробок був проведений успішно, в той час як всі можливі процеси збору, включаючи помилкові та ті, які мали на своїх траєкторіях руху потенційні зіткнення, показані фіолетовим кольором.

В рамках створення імітаційної моделі слід враховувати, що розглядаються дві різні геометрії коробок – проста та складна. На рис. 4.13 показані зміщення при підйомі об'єктів із коробок із простою геометрією, а на рис. 4.14 – із складною геометрією.

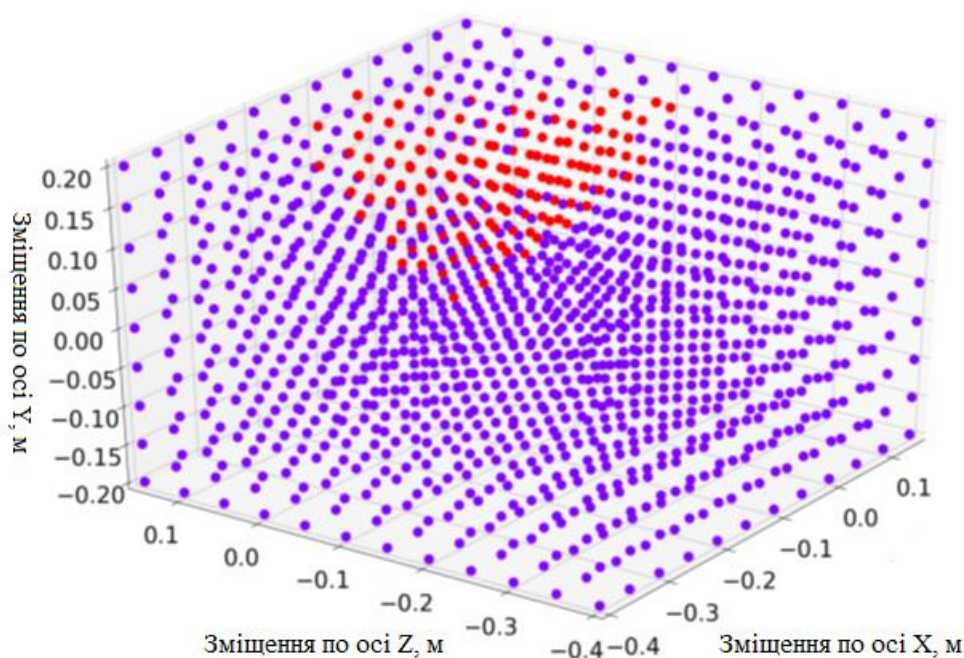


Рисунок 4.13 – Тривимірне відображення всіх можливих траєкторій при підйомі об'єктів з коробок з простою геометрією (фіолетові – проблемне виконання, червоні – успішне виконання)

В загальному в процесі проведення оцінки було змодельовано 1296 позицій робота-маніпулятора для обох типів коробок – із простою та складною геометрією. Як видно з рис. 4.13 у порівнянні із рис. 4.14, при простій геометрії коробки вірних траєкторій руху більше, ніж при складній геометрії коробок. Після проведених розрахунків встановлено, що серед змодельованих траєкторій

руху при підйомі об'єктів із коробок з простою геометрією вірними виявились 139 позицій робота, що становить 10,73% від всіх можливих варіантів. Для коробок із складною геометрією цей показник становить лише 46 можливих позицій, тобто всього 3,55%.

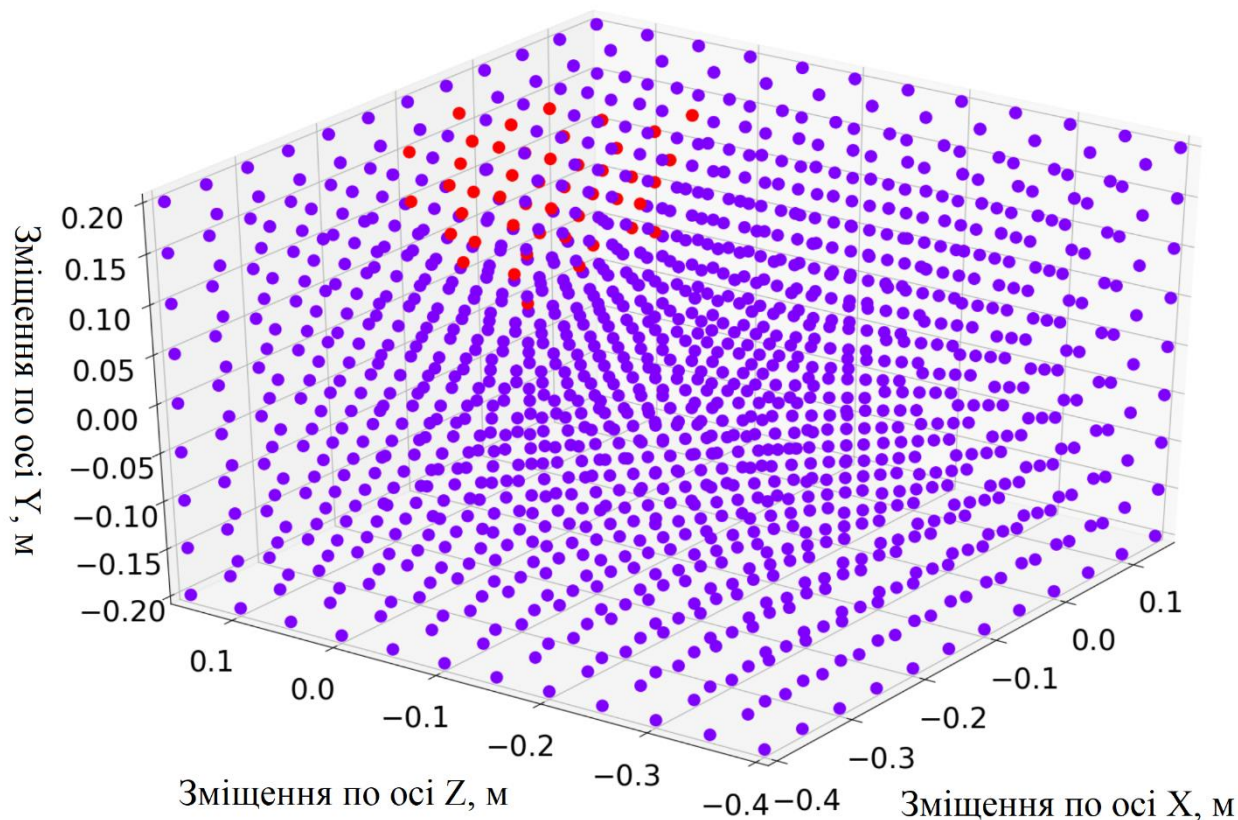


Рисунок 4.14 - Тривимірне відображення всіх можливих траєкторій при підйомі об'єктів з коробок із складною геометрією (фіолетові – проблемне виконання, червоні – успішне виконання)

При моделюванні часу виконання траєкторії при підйомі об'єктів з коробки із простою геометрією було отримано достатньо широкий діапазон значень, який показаний на рис. 4.15. Мінімальний час виконання процесу становив 11,94 с при положенні Panda Arm $x_p = 0$ м, $y_p = 0,05$ м, $z_p = -0,05$ м. Максимальний час виконання при цьому становив 19,39 с при параметрах $x_p = 0$ м, $y_p = -0,05$ м, $z_p = 0,1$ м.

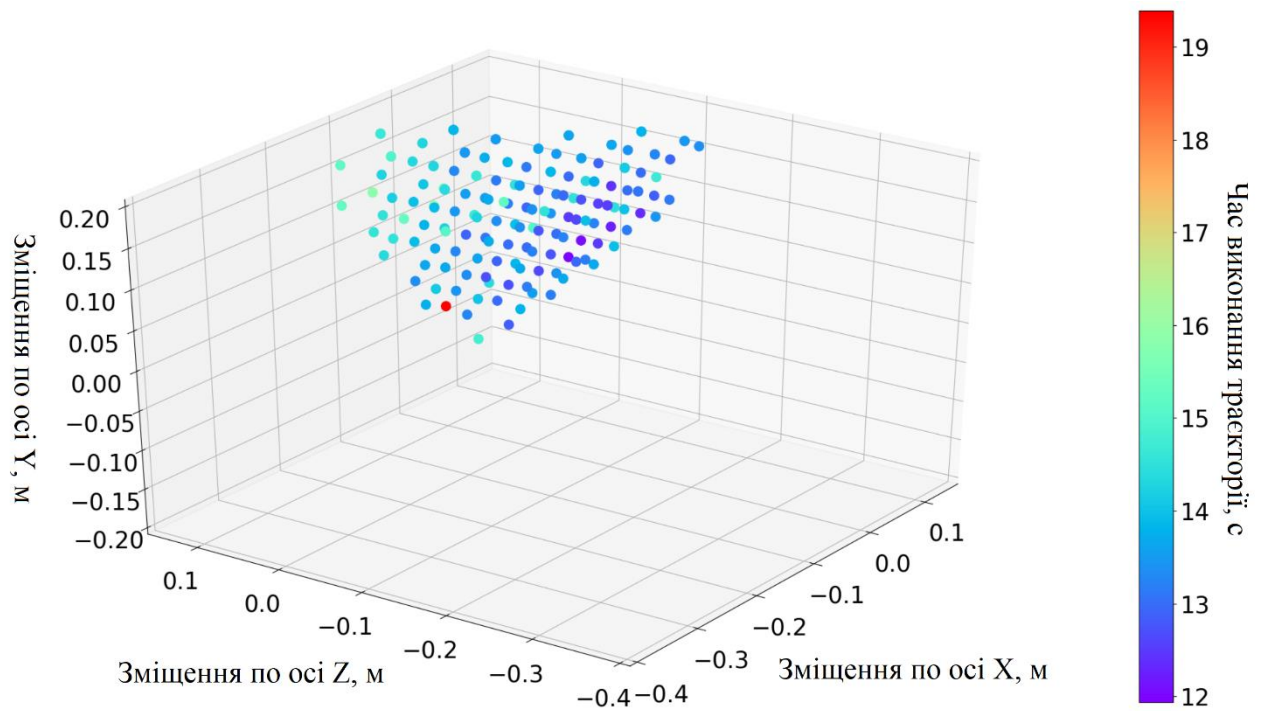


Рисунок 4.15 – Час виконання траєкторії при підйомі об’єктів з коробки із простою геометрією

Якщо провести порівняння із підйомом об’єктів з коробки із складною геометрією (рис. 4.16), то мінімальний час виконання процесу становив 12,9 с при положенні Panda Arm $x_p = -0,1$ м, $y_p = 0,1$ м, $z_p = 0$ м . Максимальний час виконання при цьому становив 17,46 с при параметрах $x_p = 0$ м, $y_p = 0,1$ м, $z_p = 0$ м.

В процесі моделювання функціонування Panda Arm під час підйому об’єктів з коробок виконувалось моделювання ще й загального часу виконання підйому та переміщення об’єктів наряду із моделюванням та розрахунком траєкторій руху, загальна довжина яких розраховувалась в метрах та в загальній зміні куту суглобу.

В процесі оцінки було змодельовано час виконання траєкторії руху при підйомі об’єктів із коробки зі складною геометрією. Результати такого моделювання для траєкторій, на яких відсутні зіткнення, показані на рис. 4.16 для часу виконання траєкторії, на рис. 4.17 – результати моделювання загального вигину суглобу робота-маніпулятора, а на рис. 4.18 – результати моделювання загальної довжини траєкторії руху, вираженої в метрах.

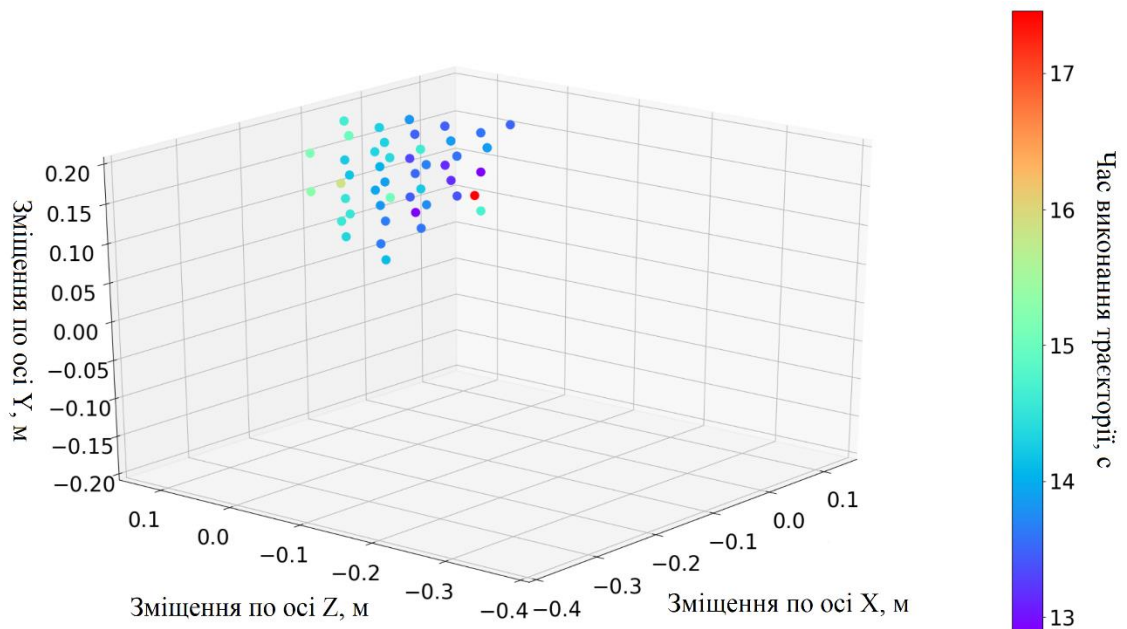


Рисунок 4.16 – Час виконання підйому об’єктів із коробки зі складною геометрією

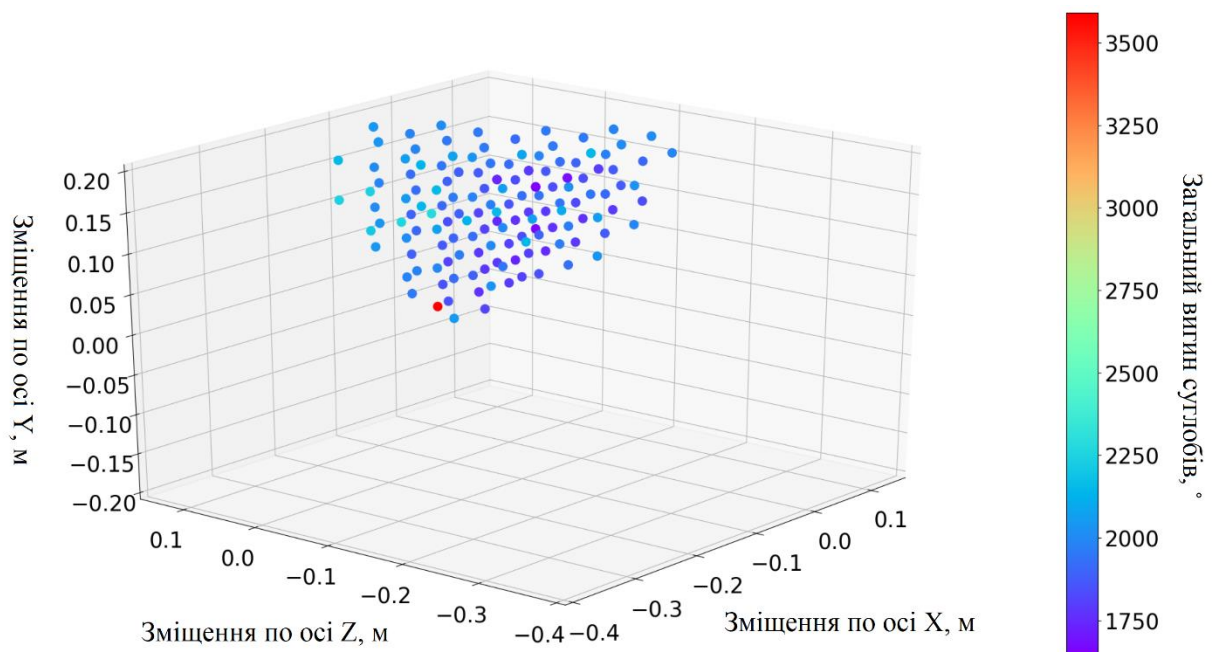


Рисунок 4.17 – Загальний вигин суглобів при підйомі об’єктів із коробки зі складною геометрією

Для спрощення сприйняття результати моделювання для коробок із простою та складною геометрією були зведені в один графік, який містить всі три параметри оцінки. Всі допустимі позиції робота-маніпулятора та час виконання процесу за умови простої геометрії коробки наведені на рис. 4.18, а на рис. 4.19 –

за умови складної геометрії коробки. Слід відзначити, що не коректні траєкторії руху відкинуті та не відображені на графіках.

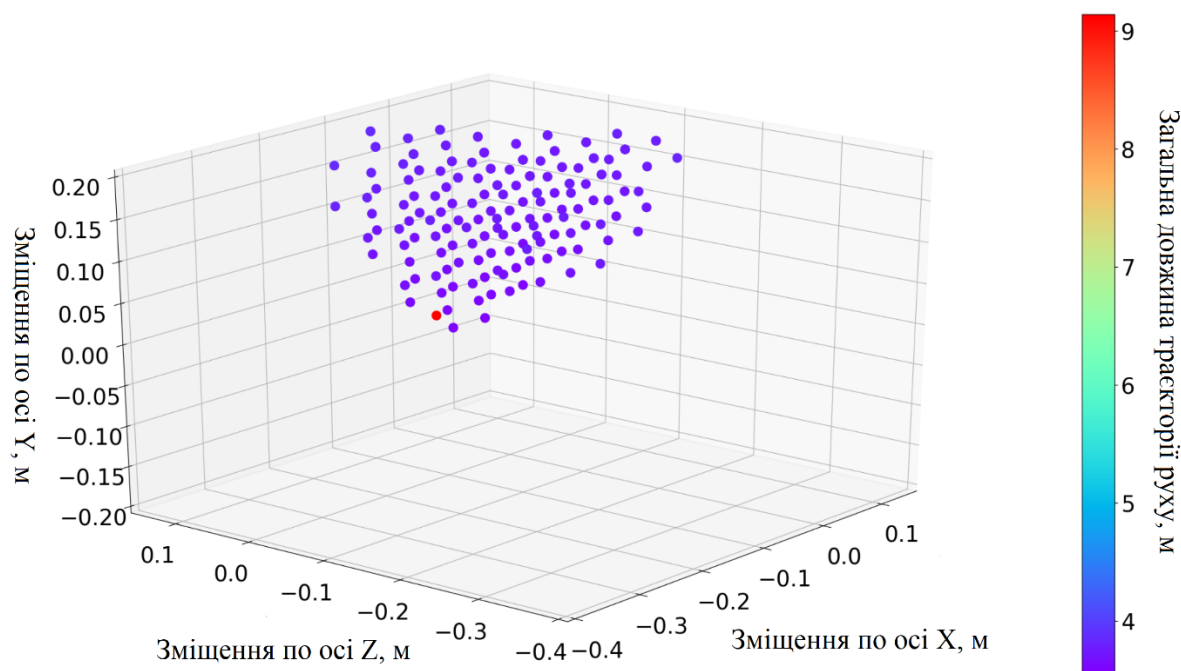


Рисунок 4.18 – Загальна довжина траєкторії руху при підйомі об’єктів із коробки зі складною геометрією

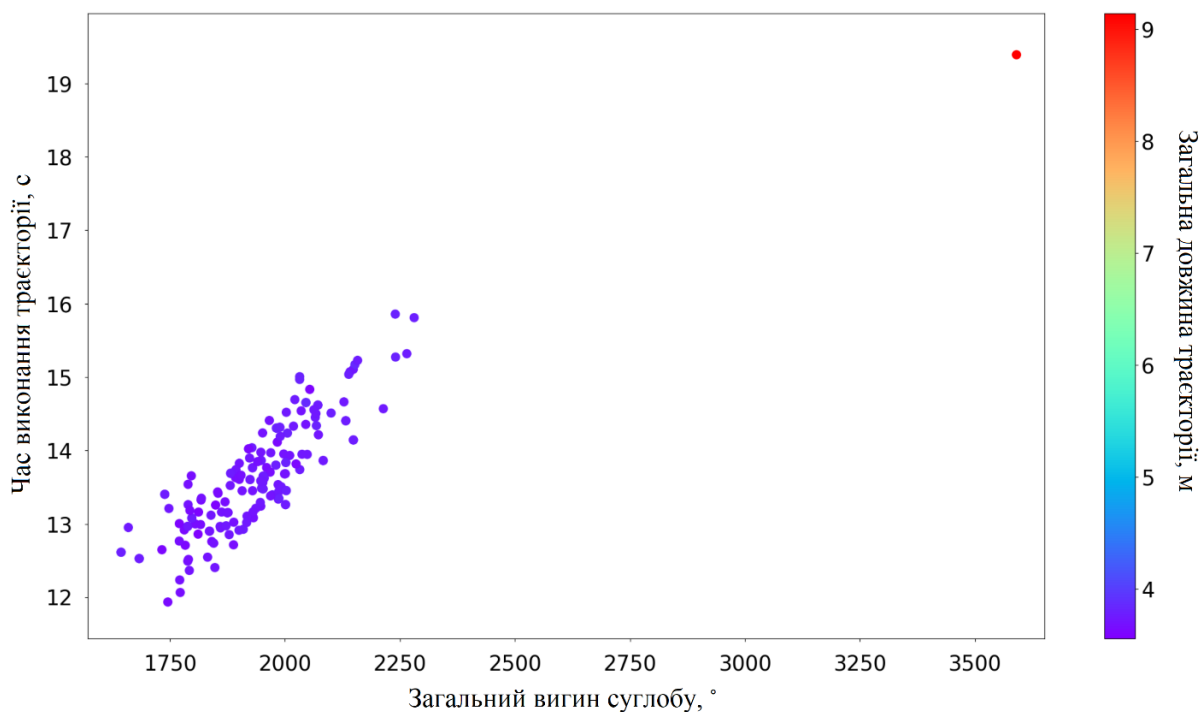


Рисунок 4.19 – Взаємозв’язок між загальною довжиною траєкторії, кутовим згином суглобів та часом виконання підйому об’єктів з простої коробки

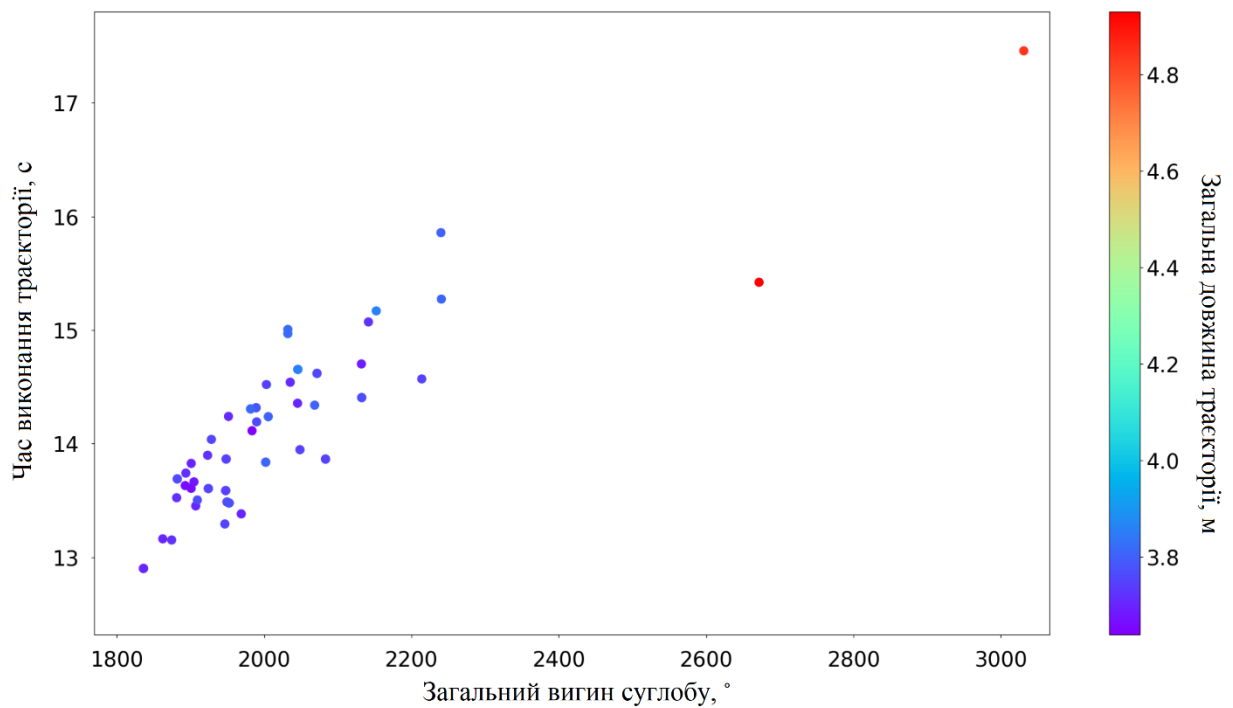


Рисунок 4.20 – Взаємозв’язок між загальною довжиною траєкторії, кутовим згином суглобів та часом виконання підйому об’єктів з складної коробки

Отже, отримані в результаті моделювання процесу підйому об’єктів з коробок із простою та складною геометрією параметри наведені в табл. 4.2.

Таблиця 4.2 – Оптимальні параметри

Назва	Позначення	Значення		Одиниці вимірювання
		Мін.	Макс.	
<i>l</i>	<i>z</i>	<i>z_p</i>	<i>z_p</i>	<i>z</i>
Зміщення по координаті Z	z_p	-0,4	0,15	м
Зміщення по координаті Y	y_p	-0,2	0,2	м
Зміщення по координаті X	x_p	-0,4	0,15	м
Проста геометрія коробки				
Пройдена відстань (загальна декартова відстань)	s_{dec}	3,56	9,14	м

Продовження таблиці 4.2

1	2	3	4	5
Загальна відстань між суглобами	s_j	1643,9	3590,3	°
Час виконання траєкторії	t	11,94	19,39	с
Складна геометрія коробки				
Пройдена відстань (загальна декартова відстань)	s_{dec}	3,64	4,84	м
Загальна відстань між суглобами	s_j	1836,3	3031,0	°
Час виконання траєкторії	t	12,9	17,46	с

4.3 Аналіз параметрів функціонування робота-маніпулятора Panda Arm, отриманих в рамках методу

4.3.1 Оптимальні параметри робота-маніпулятора для мінімізації енергоспоживання

Цей розділ присвячений аналізу оптимальних параметрів функціонування робота-маніпулятора, які були отримані за допомогою імітаційного моделювання та описані в розділах 4.1 та 4.2. Також необхідно встановити взаємозв'язок між цільовими значеннями та параметрами проведення оцінки.

В процесі функціонування робота-маніпулятора Panda Arm наявні складні взаємозв'язки між суглобами та ланками робота, оскільки крутні моменти в суглобах робота підпорядковуються рівнянням руху за методом параболічного змішування. При аналізі рисунку 4.1 можна сказати, що оптимальна траєкторія руху робота з точки зору мінімального енергоспоживання відповідає параметрам $q_2[2] = 0^\circ, q_2[4] = 0^\circ$. Такі траєкторії руху можуть бути отримані при параметрах для суглобів Panda Arm, які наведені у 4.1.

Візуально ці п'ять конфігурацій можна представити у вигляді, який зображений на рис. 4.21. Слід відзначити, що найбільше енергоспоживання

відповідає траєкторії із параметрами $q_2[2] = -100^\circ, q_2[4] = 140^\circ$. Для досягнення максимального енергоспоживання Panda Arm повинно працювати із параметрами, які зазначені у 4.2, а конфігурації наведені на рис. 4.22.

$$\begin{aligned}
 q_1 &= (0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ), \\
 q_{1-2} &= (-45^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ), \\
 q_2 &= (-90^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ), \\
 q_{2-3} &= (-135^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ), \\
 q_3 &= (-180^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ)
 \end{aligned}
 \tag{4.1}$$

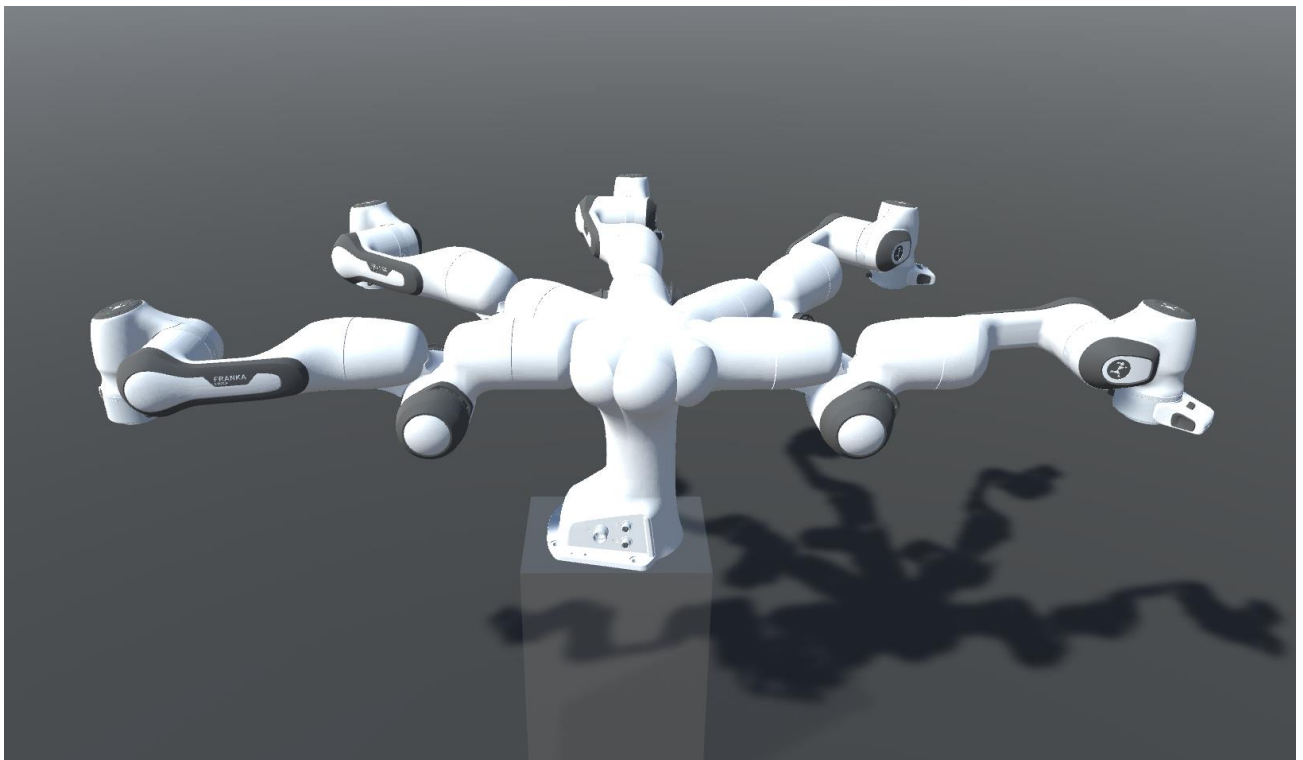


Рисунок 4.21 – Конфігурації суглобів робота, які відповідають оптимальним траєкторіям руху (рух за годинниковою стрілкою)

$$\begin{aligned}
 q_1 &= (0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ), \\
 q_{1-2} &= (-45^\circ \ -50^\circ \ 0^\circ \ 70^\circ \ 0^\circ \ 0^\circ \ 0^\circ), \\
 q_2 &= (-90^\circ \ -100^\circ \ 0^\circ \ 140^\circ \ 0^\circ \ 0^\circ \ 0^\circ), \\
 q_{2-3} &= (-135^\circ \ -50^\circ \ 0^\circ \ 70^\circ \ 0^\circ \ 0^\circ \ 0^\circ), \\
 q_3 &= (-180^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ)
 \end{aligned}
 \tag{4.2}$$

Якщо проводити аналогію із рухом людини, яка піднімає об'єкт зі стола, то спостерігається цікава ситуація. Людина, скоріш за все, зігне лікоть і опустить руку з плеча, тобто утворить умовну V-подібну форму між верхньою і нижньою частинами руки. Таким чином, вони наблизить об'єкт до свого тулуба. У просторі руху робота-маніпулятора така траєкторія руху відповідає підйому з другого суглоба, коли параметр $q_2[2] \ll 0^\circ$, та згину ліктя з четвертого суглоба робота донизу, коли параметр $q_2[4] \gg 0^\circ$.

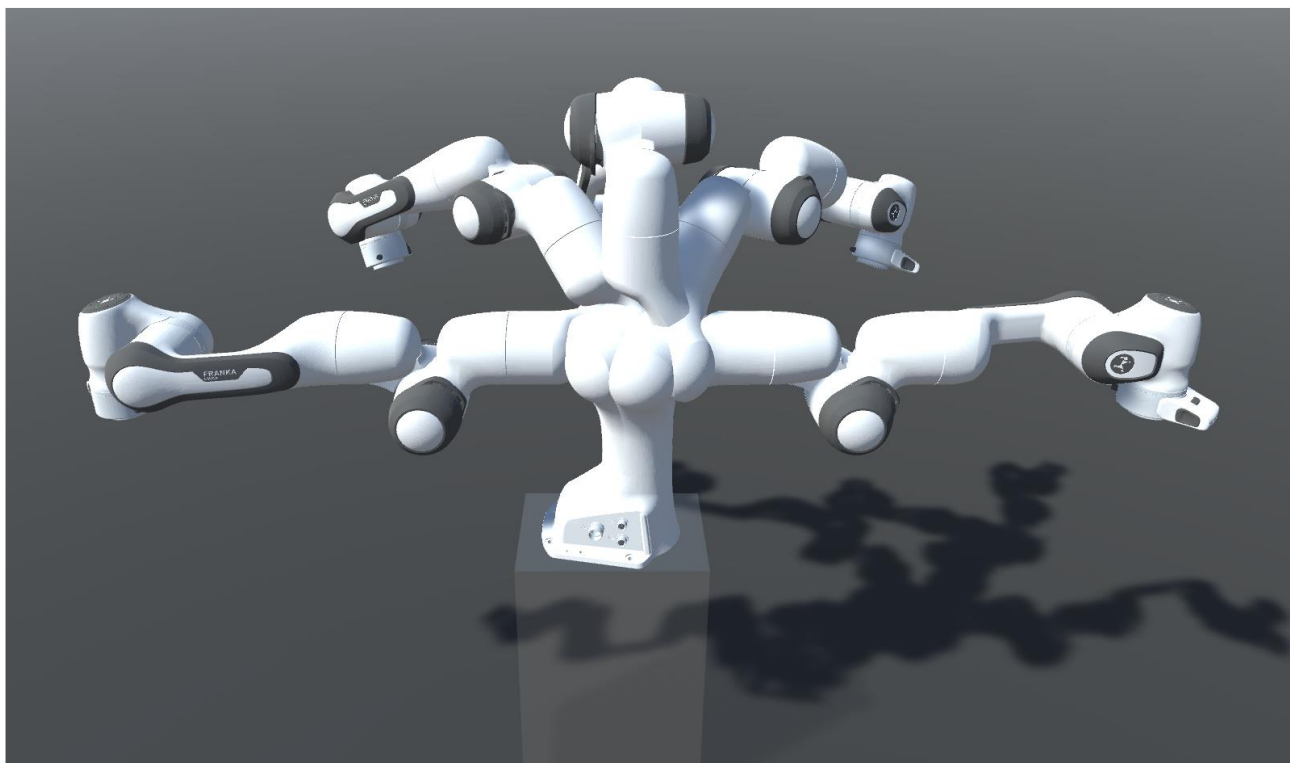


Рисунок 4.22 - Конфігурації суглобів робота, які відповідають найбільшому енергоспоживанню при русі (рух за годинниковою стрілкою)

Перераховані вище траєкторії зображені на рис. 4.2 і відповідають максимальному енергоспоживанню робота-маніпулятора. Така відмінність у русі людини та робота-маніпулятора розглядалась у роботі [15], в якій автори стверджують, що не зрозуміло, чому саме людина виконує підйом об'єктів по такій траєкторії. Тобто можна зробити висновок, що при створенні траєкторій руху роботів-маніпуляторів не завжди повторювання руху людей будуть найбільш оптимальними рішеннями, в тому числі з точки зору енергоефективності.

За результатами проведених досліджень між збільшенням сумарного руху суглобів Panda Arm та енергоефективністю наявний прямий зв'язок – чим більший згин, тим більші витрати енергії. Тоді найбільші значення споживання енергії накопичуються на траєкторіях, в яких параметри $q_2[2] < -60^\circ$ та $q_2[4] > 80^\circ$. В той же час, зменшення вигину суглобів робота призводить до зменшення енергоспоживання. Результати цих досліджень показані на рис. 4.5 та 4.6.

Більша відстань в суглобовому просторі руху робота-маніпулятора по траєкторії відповідає меншій лінійній відстані. Після проведення оцінки не вдалося встановити прямої залежності між довжиною траєкторії та енергоспоживанням, але була встановлена тенденція, що збільшення довжини траєкторії призводить до зменшення енергоспоживання.

В основному найбільше значення крутного моменту на другому суглобі відповідає супротиву силі тяжіння, тобто положенню, при якому робот-маніпулятор тримає руку прямо, що відповідає параметрам $q_2[2] = 0^\circ$ та $q_2[4] = 0^\circ$. Зменшення впливу на другий суглоб спостерігається при піднятті руки робота-маніпулятора та його згину на другому суглобі, тобто досягнення значень $q_2[4] > 0^\circ$.

Як видно з рис. 4.3, друге з'єднання потребує найбільшого споживання енергії серед усіх семи суглобів конструкції Panda Arm. Це можна пояснити тим, що більша частка маси робота підключена саме до цього двигуна.

В загальному отримані результати досліджень в розрізі енергоспоживання можна розбити на наступні складові:

- зменшення довжини лінійної траєкторії руху робота призводить до збільшення енергоспоживання;
- зменшення вигину суглобів при руху по траєкторії призводить до зменшення енергоспоживання;
- найменше енергії потребує двигун першого суглоба;
- основну частку серед спожитої енергії має саме сумарна довжина траєкторії руху, пройденої всіма суглобами робота-маніпулятора.

4.3.2 Оптимальні параметри робота-маніпулятора для підйому об'єктів

Отримані на рис. 4.13 та 4.14 залежності траєкторій руху робота-маніпулятора при підйомі об'єктів із коробок можна пояснити тим, що при підйомі із коробок зі складною геометрією отвір становиться вузький, що призводить до зменшення кількості можливих траєкторій приблизно на 100 штук у порівнянні із результатами для коробок із простою геометрією. Крім того, одна із позицій відповідає положенню, в якому робот-маніпулятор не здатен підняти об'єкт із коробки, оскільки $y_p = 0$. При такому положенні четвертий та п'ятий суглоб робота стикаються із геометрією коробки.

В результаті проведеного аналізу було встановлено, що максимальне зміщення позицій по осям x та z становить по 0,15 м для кожної. Такі зміщення положення робота-маніпулятора відповідають позиціям, при яким він знаходиться близько до конвеєра, на який необхідно покласти об'єкт. При збільшенні цієї відстані робот-маніпулятор не буде здатен схопити об'єкт або досягти площини конвеєра.

З рис. 4.14 також видно, що при занадто близькому розташуванні Panda Arm до конвеєра він не буде здатен повністю виконати процес збору об'єктів з коробок із складною геометрією.

При детальному аналізі всіх можливих позицій робота-маніпулятора було встановлено, що оптимальним за часом значення розташування робота за умови підйому об'єктів із коробок з простою геометрією відповідає параметрам $x_p = 0$ м, $y_p = 0,05$ м, $z_p = -0,05$ м. Хоча це значення є оптимальним, наявний ряд близьких значень, в яких час виконання підйому об'єктів відрізняється лише на кілька десятих секунди.

У випадку підйому об'єктів із коробок зі складною геометрією, оптимальними параметрами є $x_p = -0,1$ м, $y_p = 0,1$ м, $z_p = 0$ м. Ці ж самі параметри відповідають параметрам, при яким робот-маніпулятор проходить найменшу відстань, тобто траєкторія руху є найменшою за довжиною. В загальному спостерігається тенденція до збільшення часу виконання операції при

збільшенні зміщення робота по осі z. При цьому такий зв'язок для осей x та y не спостерігається.

Причина такого зв'язку була встановлена при запуску цифрового двійника Panda Arm із зазначеними параметрами. В ході симуляції видно, що чим ближче робот-маніпулятор знаходиться до коробки, тим складніше йому опуститись в площину коробки та схопити об'єкт, що знаходиться в ній. Слід пам'ятати, що при цьому Panda Arm не повинно зіштовхуватись із самою коробкою. При занадто малій відстані між роботом та коробкою робот зіштовхнеться із задньою стінкою коробки. Тому при плануванні траєкторії підйому об'єктів враховувались менші швидкості та прискорення в критичних положеннях робота-маніпулятора по відношенню до коробки. В зв'язку з цим збільшується час виконання підйому об'єктів.

Прикладом такої траєкторії є максимальне значення на рис. 4.15, при якому час виконання підйому об'єктів становить 19,39 с. Ця траєкторія має найбільшу лінійну траєкторію руху та найбільший вигин суглобів робота-маніпулятора.

На рис. 4.23 зображено положення Panda Arm в імітаційній моделі, створені в Unity, при якій досягається найменший (рис. 4.23, а) та найбільший (рис. 4.23, б) час виконання процесу. З рис. 4.23 б видно, що робот-маніпулятор знаходиться занадто близько до коробки.

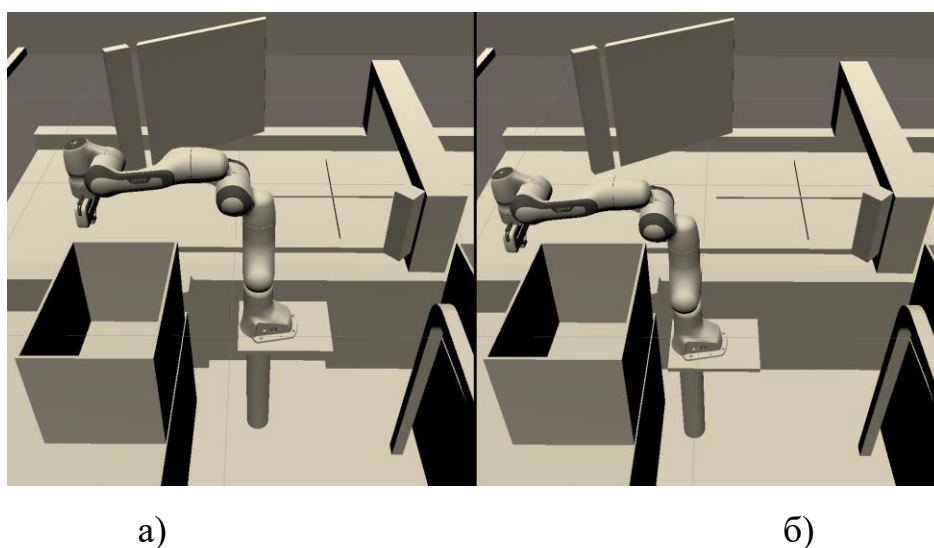


Рисунок 4.23 – Розташування робота при найменшому (а) та найбільшому (б) часі виконання підйому об'єктів з коробки

На рис. 4.24 та 4.25 показані три різні траєкторії руху робота-маніпулятора при підйомі об'єктів. На рис. 4.24 зображена траєкторія, яка відповідає найменшому часу виконання за умови складної геометрії коробки. Ця траєкторія відповідає параметрам $x_p = -0,1$ м, $y_p = 0,1$ м, $z_p = 0$ м. В свою чергу, на рис. 4.25 показана траєкторія, при якому досягається найбільший час виконання 15,81 с та найбільшому вигину суглобів 2281° та відповідає параметрам $x_p = 0,05$ м, $y_p = 0,05$ м, $z_p = 0,15$ м.

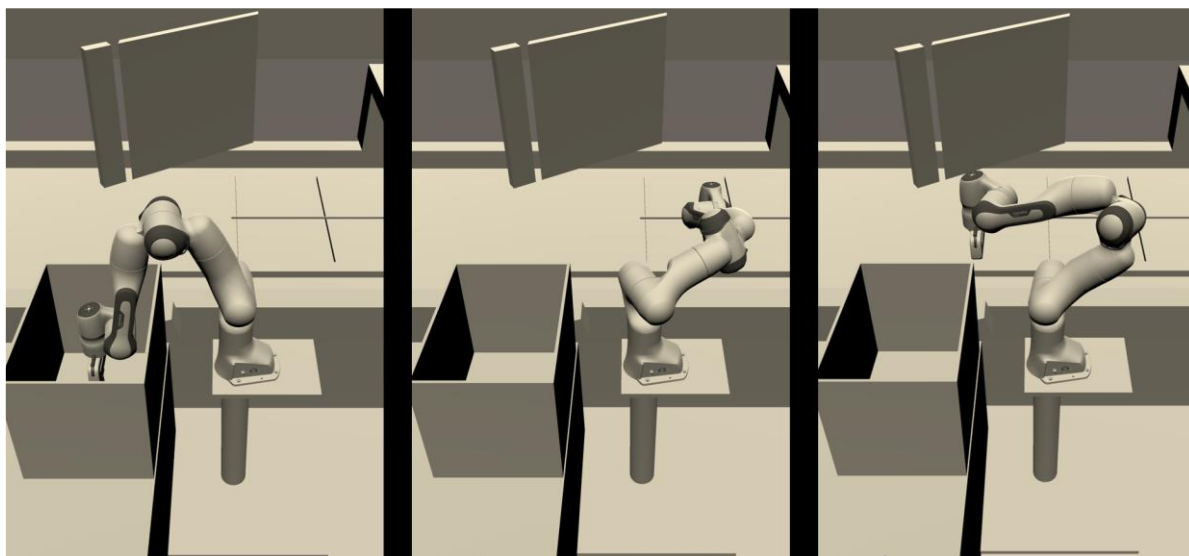


Рисунок 4.24 – Траєкторія, при якій досягається найменша лінійна відстань та час виконання підйому об'єктів

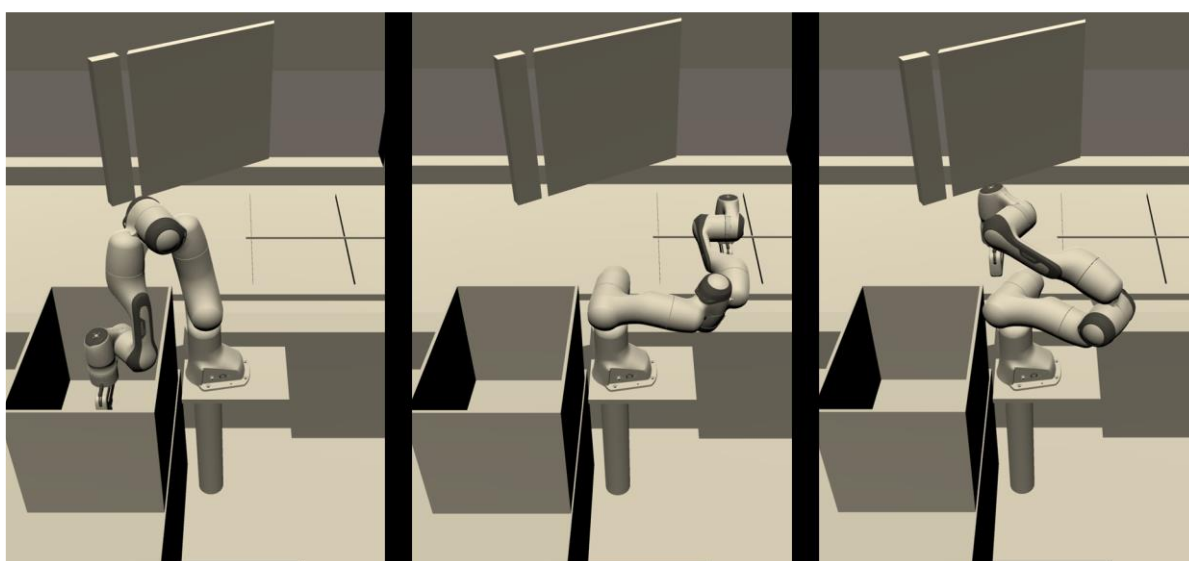


Рисунок 4.25 - Траєкторія, при якій досягається найбільша лінійна відстань та час виконання підйому об'єктів

4.4 Висновки до четвертого розділу

В розділі наведені результати моделювання траєкторій руху Panda Arm при підйомі об'єктів з коробок та їх подальшого переміщення на конвеєр. Ці траєкторії руху моделювались для двох можливих геометрій коробок. Створені траєкторії були проаналізовані з точки зору можливих зіткнень із іншими об'єктами. Такі хибні траєкторії були відсіяні та не приймали участь у подальшому аналізі.

Отримані траєкторії руху були проаналізовані з точки зору споживання енергії. В результаті було встановлено, що:

- зменшення довжини лінійної траєкторії руху робота призводить до збільшення енергоспоживання;
- зменшення вигину суглобів при руху по траєкторії призводить до зменшення енергоспоживання;
- найменше енергії потребує двигун першого суглоба;
- основну частку серед спожитої енергії має саме сумарна довжина траєкторії руху, пройденої всіма суглобами робота-маніпулятора.

Крім того, при аналізі процесу підйому об'єктів із коробок встановлено наступне:

- довжина лінійної траєкторії руху робота-маніпулятора не значним чином впливає на підйом об'єктів, тому її можна розглядати лише в розрізі енергоспоживання;
- визначені параметри, при яких досягається мінімальний час підйому об'єктів із коробок, а саме $x_p = -0,1$ м, $y_p = 0,1$ м, $z_p = 0$ м для коробок із складною геометрією та $x_p = 0$ м, $y_p = 0,05$ м, $z_p = -0,05$ м для коробок із простою геометрією.

ВИСНОВКИ

В першому розділі було проведено огляд та аналіз підходів, які застосовуються при створенні віртуальних прототипів. Наведені характеристики віртуальних прототипів та переваги їх використання у порівнянні із фізичними прототипами.

В якості прикладу робота-маніпулятора, для якого буде проводитись оцінки функціональності його роботи, було обрано робот-маніпулятор типу «рука» Panda Arm. В розділі наведені технічні характеристики цього робота та його сфери застосування. Також у розділі було поставлено задачі та запропоновані підходи до проведення оцінки функціональності роботів-маніпуляторів.

В розділі також наведено короткий опис середовищ для розробки запропонованої оцінки функціональності робота-маніпулятора Panda Arm.

В другому розділі виконано огляд функціональних можливостей методу оцінки робота-маніпулятора на основі імітаційного моделювання як системи вцілому, так і як кожного її окремого компоненту. До цих компонентів відносяться архітектура фреймворку, що розробляється, в якому описуються незалежні компоненти, файли та дані, що оцінюються між цими компонентами.

Також в розділі ілюструється, яким чином можна визначити простір оцінки за допомогою редактора Unity. У розділі пояснюється концепція коду, що забезпечує функціональність роботи методу оцінки робота-маніпулятора. Крім того, у розділі підсумовується, яким чином проводиться оцінка симуляції робота-маніпулятора за допомогою коду, розробленого у .NET.

Наостанок, викладено ідею третього компоненту методу оцінки, а саме фреймворку, який зчитує дані з текстового файлу і в подальшому створює візуалізації результатів проведених у системі обчислень.

У третьому розділі наведено два конкретні приклади використання розробленого методу оцінки функціональності робота-маніпулятора, які демонструють можливість застосування методу.

Основна увага при цьому приділяється створенню імітаційної моделі роботи Panda Arm. Імітаційна модель представляє собою два додатки, в яких

проводиться моделювання роботи робота-маніпулятора Panda Arm. Крім того, в розділі показано параметри оцінки та об'єктивні значення, які використовувались для оцінки функціональності.

В розділі показано, що робот-маніпулятор Panda Arm можна перетворити на цифрового двійника в площині Unity, що здатен відобразити реального робота на високому за точністю рівні. Також в розділі наведено деталізацію використання розробленого додатку для оцінки траєкторії робота-маніпулятора та автоматизованого збору предметів, на прикладі збору сміття.

Також в розділі наведено математичну модель опису траєкторії руху робота із врахуванням його кінематичних та динамічних параметрів на основі методу параболічного змішування. Крім того, розглянута математична модель оцінки енергоефективності роботи Panda Arm, при якій рух по траєкторії буде виконуватись із максимально можливою швидкістю та мінімальною витратою енергії.

В четвертому розділі наведені результати моделювання траєкторій руху Panda Arm при підйомі об'єктів з коробок та їх подальшого переміщення на конвеєр. Ці траєкторії руху моделювались для двох можливих геометрій коробок. Створені траєкторії були проаналізовані з точки зору можливих зіткнень із іншими об'єктами. Такі хибні траєкторії були відсіяні та не приймали участь у подальшому аналізі.

Отримані траєкторії руху були проаналізовані з точки зору споживання енергії. В результаті було встановлено, що:

- зменшення довжини лінійної траєкторії руху робота призводить до збільшення енергоспоживання;
- зменшення вигину суглобів при руху по траєкторії призводить до зменшення енергоспоживання;
- найменше енергії потребує двигун першого суглоба;
- основну частку серед спожитої енергії має саме сумарна довжина траєкторії руху, пройденої всіма суглобами робота-маніпулятора.

Крім того, при аналізі процесу підйому об'єктів із коробок встановлено наступне:

– довжина лінійної траєкторії руху робота-маніпулятора не значним чином впливає на підйом об'єктів, тому її можна розглядати лише в розрізі енергоспоживання;

– визначені параметри, при яких досягається мінімальний час підйому об'єктів із коробок, а саме $x_p = -0,1$ м, $y_p = 0,1$ м, $z_p = 0$ м для коробок із складною геометрією та $x_p = 0$ м, $y_p = 0,05$ м, $z_p = -0,05$ м для коробок із простою геометрією.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. .NET Framework. [Електронний ресурс]. - Режим доступу: https://uk.wikipedia.org/wiki/.NET_Framework (дата звернення 26.09.2024)
2. ADAMS. [Електронний ресурс]. - Режим доступу: https://hexagon.com/products/product-groups/computer-aided-engineering-software/adams?utm_easyredir=www.mssoftware.com (дата звернення 17.09.2024)
3. Altair HyperWorks. [Електронний ресурс]. - Режим доступу: <https://altair.com/altair-hyperworks> (дата звернення 17.09.2024)
4. Ang. M.H. Encyclopedia of Robotics / M.H. Ang, O. Khatib, B. Siciliano. – Living reference work. – 2020. – 623 p.
5. Banks J. Discrete-event system simulation / J. Banks, J. S. Carson II, L. Barry. – 2005. – 423 с.
6. Cirovic G. Multiple-Criteria Decision Making / G. Cirovic, D. Pamucar. - Basel : MDPI, 2022. – 310 p.
7. Data Sheet Robot Arm & Control. [Електронний ресурс]. - Режим доступу: [/https://pkj-robotics.dk/wp-content/uploads/2020/09/Franka-Emika_Brochure_EN_April20_PKJ.pdf](https://pkj-robotics.dk/wp-content/uploads/2020/09/Franka-Emika_Brochure_EN_April20_PKJ.pdf) (дата звернення 20.09.2024)
8. DELMIA. [Електронний ресурс]. - Режим доступу: <https://www.3ds.com/ru/products/delmia> (дата звернення 16.09.2024)
9. Doran J.P. Unreal Engine Game Development Cookbook / J.P. Doran. – 2015. – 326 с.
10. Featherstone R. Inverse dynamics — the recursive Newton-Euler method / R. Featherstone // Robot Dynamics Algorithms. - Boston, MA: Springer US. – 1987. – pp. 65-77
11. Franka Emika Panda Arm. [Електронний ресурс]. - Режим доступу: <https://robodk.com/robot/ru/Franka/Emika-Panda> (дата звернення 20.09.2024)
12. Franka Emika Panda. [Електронний ресурс]. - Режим доступу: <https://cyberbotics.com/doc/guide/panda?version=R2022b> (дата звернення 15.09.2024)

13. Franka Emika Robot's Instruction Handbook. – Germany, Munich. – 2021. – 184 p.
14. Gaz C. Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization / C. Gaz, M. Cagnetti, A. Oliva, P. Robuffo Giordano, A. De Luca // IEEE Robotics and Automation Letters. - vol. 4, no. 4. – 2019. – с. 4147-4154
15. Marshall R. Performance objectives in human movement: A review and application to the stance phase of normal walking / R. Marshall, G. Wood, L. Jennings // Human Movement Science. – 1989. – vol. 8, no. 6. – pp. 571-594
16. Moore D. Python GUI Programming with Tkinter: Develop responsive and powerful GUI applications with Tkinter / D. Moore. – Packt: Paperback, 2019. – 452 с.
17. Nonaka Y. Fast Path Finding System with GPGPU Computing for Replacement Tasks in Plant Maintenance / Y. Nonaka, A. Enomoto, N. Fukii, K. Shibuta // Procedia CIRP. – 22 (1). – 2014. – p. 23-32
18. NS-3. [Електронний ресурс]. - Режим доступу: <https://www.nsnam.org/> (дата звернення 16.09.2024)
19. OMNeT++. [Електронний ресурс]. - Режим доступу: <https://omnetpp.org/> (дата звернення 16.09.2024)
20. Panda Arm. [Електронний ресурс]. - Режим доступу: <https://ori.ox.ac.uk/robots/panda-arm/> (дата звернення 20.09.2024)
21. Pandas Tutorial. [Електронний ресурс]. - Режим доступу: <https://www.w3schools.com/python/pandas/default.asp> (дата звернення 05.10.2024)
22. Python Підручник Pandas: DataFrame, діапазон дат, використання Pandas. [Електронний ресурс]. - Режим доступу: <https://www.guru99.com/uk/python-pandas-tutorial.html> (дата звернення 07.10.2024)
23. Quigley M. Programming Robots with ROS: A Practical Introduction to the Robot Operating System / M. Quigley, B. Gerkey, W.D. Smart – O'Reilly, 2015. – 448 с.
24. R. Wagner. Challenges and potentials of digital twins and industry 4.0 in product design and production for high performance products / Wagner R., Schleich B.,

Haefner B., Kuhnle A., Wartzack S., Lanza G. // Procedia CIRP. – 2019. – vol. 84, p. 88-93

25. Simscape Multibody. [Электронный ресурс]. - Режим доступа: <https://ch.mathworks.com/products/simscape-multibody.html> (дата звернення 17.09.2024)

26. T. Kunz. Turning paths into trajectories using parabolic blends / T. Kunz, M. Stilman. - Georgia Institute of Technology. – 2011. -213 p.

27. Task Parallel Library (TPL). [Электронный ресурс]. - Режим доступа: <https://learn.microsoft.com/uk-ua/dotnet/standard/parallel-programming/task-parallel-library-tpl> (дата звернення 02.10.2024)

28. Tkinter – Python interface to Tcl/Tk. [Электронный ресурс]. - Режим доступа: <https://docs.python.org/uk/3/library/tkinter.html> (дата звернення 05.10.2024)

29. Unity at 10: For better – or worse – game development has never been easier. [Электронный ресурс]. - Режим доступа: <https://arstechnica.com/gaming/2016/09/unity-at-10-for-better-or-worse-game-development-has-never-been-easier/> (дата звернення 24.09.2024)

30. Unity User Manual (2019.3). [Электронный ресурс]. - Режим доступа: <https://docs.unity3d.com/2019.3/Documentation/Manual/UnityManual.html> (дата звернення 24.09.2024)

31. Unity. [Электронный ресурс]. - Режим доступа: [https://uk.wikipedia.org/wiki/Unity_\(%D1%96%D0%B3%D1%80%D0%BE%D0%B2%D0%B8%D0%B9_%D1%80%D1%83%D1%88%D1%96%D0%B9\)](https://uk.wikipedia.org/wiki/Unity_(%D1%96%D0%B3%D1%80%D0%BE%D0%B2%D0%B8%D0%B9_%D1%80%D1%83%D1%88%D1%96%D0%B9)) (дата звернення 22.09.2024)

32. Unity-Technologies/URDF-Importer. [Электронный ресурс]. - Режим доступа: <https://github.com/Unity-Technologies/URDF-Importer> (дата звернення 10.10.2024)

33. Vanderelst D. Rational imitation for robots: the cost difference model / D. Vanderelst, A. Winfield // Adaptive Behavior. – 2017. – 25(2). – p. 1-22

34. What is Panda Arm? [Электронный ресурс]. - Режим доступа: <https://riders.ai/en-blog/what-is-panda-arm> (дата звернення 15.09.2024)

35. Zorriassatine F. A survey of virtual prototyping techniques for mechanical product development / F. Zorriassatine, C. Wykes, R. Parkin, N. Gindy // Proceedings of the institution of mechanical engineers, Part B: Journal of engineering manufacture. – 2003. - vol. 217, no. 4. – p. 513-530
36. Брусенцов В.Г. Основи Ергономіки: навчальний посібник / В.Г. Брусенцов, О.В. Брусенцов, І.І. Бугайченко, С.О. Кисельова. – Харків: УкрДАЗТ, 2011. – 141 с.
37. Гервас О.Г. Ергономіка. Навчально-методичний посібник / О.Г. Гервас. – Умань: видавничо-поліграфічний центр «Візаві». – 2011. – 130 с.
38. Гоблик Н.М. MATLAB в інженерних розрахунках. Комп'ютерний практикум: навчальний посібник / Н. М. Гоблик, В.В. Гоблик. - Львів: Видавництво Львівської політехніки, 2020. – 190 с.
39. Гуржій А.М. Основи автоматики та робототехніки: навчальний посібник / А.М. Гуржій, А.Т. Нельга, В.М. Співак, О.С. Ітякін: Дніпро: «Гарант СВ», 2021. – 243 с.
40. Девіс Е. Розробка 3D-ігор у Unity / Е. Девіс, Т. Батіст, Р. Крейг, Р. Станкел. – 2019. – 298 с.
41. Ельперін І.В. Автоматизація виробничих процесів: підручник / І.В. Ельперін, О.М. Пупена, В.М. Сідлецький, С.М. Швед. – К.: Вид. Ліра-К, 2017. – 378 с.
42. Задачин В.М. Чисельні методи : навчальний посібник / В. М. Задачин, І. Г. Конюшенко. – Х. : Вид. ХНЕУ ім. С. Кузнеця, 2014. – 180 с.
43. Засорнов О.С. Програмування мікроконтролерних та робототехнічних систем. Навчальний посібник / О.С. Засорнов, І.О Засорнова. – Кондор, 2023. – 280 с.
44. Карбованець М.І. Теоретична механіка. Частина І. Метод Лагранжа: навчальний посібник / М.І. Карбованець, В.Ю. Лазур, Є.А. Нодь. – Ужгород: Видавництво УжНУ «Говерла», 2019. – 82 с.
45. Ковальов Ю.А. Проектування промислових роботів та маніпуляторів: навчальний посібник / Ю.А. Ковальов, С.О. Кошель, О.П. Манойленко. – Центр навчальної літератури.. – 2019. – 256 с.

46. Когутяк І.М. Сучасні комп'ютерні технології в автоматизації. Навчальний посібник / І.М. Когутяк – Івано-Франківськ: Факел, 2009. – 279 с.

47. Козбур І.Р. Моделювання систем керування в пакеті MATLAB SIMULINK, методичні вказівки до виконання лабораторної роботи по курсу «Комп'ютерні методи дослідження систем автоматичного управління», для студентів 4 курсу спеціальності 6.050201 «Системна інженерія» / укл.: І.Р. Козбур, Г.В. Козбур, Р.І. Михайлишин. – Тернопіль : ТНТУ, 2019. - 23 с.

48. Колаборативний робот Franka Emika Panda. [Електронний ресурс]. - Режим доступу: https://hydrolider.com.ua/ua/p595553818-kollaborativnyj-robot-franka.html?srsId=AfmBOoqAsVdUBwSS87m1QK2_bnrmsZ5hkDx4uH3-OzjC4cDM14ucUdDe (дата звернення 10.09.2024)

49. Кравець І.О. Імітаційне моделювання. Навчальний посібник / І.О. Кравець. – ЧДУ ім. Петра Могили, 2010. – 107 с.

50. Крищук М.Г. Робота в програмному продукті САТІА. Загальні відомості. Частина 1: методичні вказівки до виконання комп'ютерного практикуму / М.Г. Крищук, А.В. Трубін, Н.Ф. Тертишна, В.О. Єщенко – К.: НТУУ «КПІ імені Ігоря Сікорського», 2017. – 78 с.

51. Лазарев Ю.Ф. Довідник з MATLAB. Електронний навчальний посібник з курсового і дипломного проектування / Ю.Ф. Лазарев – К.: НТУУ "КПІ", 2013. – 132 с.

52. Лазарев Ю.Ф. Моделювання динамічних систем у Matlab. Електронний навчальний посібник / Ю.Ф. Лазарев. – Київ: НТУУ "КПІ", 2011. – 421 с.

53. Лісовець С.М. Робототехніка. Практикум. Частина 1: навчальний посібник для здобувачів ступеня бакалавра за освітньо-професійною програмою «Комп'ютерноінтегровані системи та технології в приладобудуванні» спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка (151 Автоматизація та комп'ютерноінтегровані технології) / КПІ ім. Ігоря Сікорського ; автори: С. М. Лісовець, С. В. Барилко. – Електронні текстові дані. – Київ : КПІ ім. Ігоря Сікорського, 2023. – 78 с.

54. Ловейкін В.С. Мехатроніка. Навчальний посібник / В.С. Ловейкін, Ю.О. Ромасевич, Ю.В. Човнюк. – К., 2012. – 357 с.

55. Ляшенко О.М. Розробка комп'ютерних ігор за допомогою Unity 3D: електронний навчальний посібник для підготовки студентів спеціальності 121 «Інженерія програмного забезпечення» / Укладач: О.М. Ляшенко. – Херсон: видавництво ФОП Вишемирський В.С., 2018. – 220 с.

56. Морзе Н.В. Основи робототехніки: навчальний посібник / Н.В. Морзе, Л.О. ВарченкоТроценко, М.А. Гладун. – Кам'янець-Подільський : ПП Буйницький О.А., 2016. – 184 с.

57. Невлюдов І. Ш. Комп'ютерно-інтегровані технології виробництва технічних засобів автоматизації. Частина 2: Підручник Кривий Ріг: видавець Чернявський Д. О., 2022. – 424 с.

58. Невлюдов І.Ш. Пневматичні пристрої та засоби автоматизації мехатронних систем: Навчальний посібник / І.Ш. Невлюдов, Л.О. Кривопляс-Володіна, С.П. Новоселов, О.В. Сичова. – Харків: ФОП Панов А.М., 2020 . – 256 с.

59. Пасхавер Б. Pandas в дії / Б. Пасхавер. - Packt: Paperback, 2018. – 512 с.

60. Пелевін Л.Є. Синтез робототехнічних систем в машинобудуванні: Підручник / Л.Є. Пелевін, К.І. Почка, О.М. Гаркавенко, Д.О. Міщук, І.В. Русан. – К.: ТОВ «НВП «Інтерсервіс», 2016. – 258 с.

61. Програмування для робототехніки: мови та інструменти. [Електронний ресурс]. - Режим доступу: <https://itproger.com/ua/news/programmirovanie-dlya-robototehniki-yaziki-i-instrumenti> (дата звернення 03.10.2024)

62. Путівник мовою програмування Python. [Електронний ресурс]. - Режим доступу: https://pythonguide.rozh2sch.org.ua/#_python_%D0%B2_%D1%80%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%BC%D1%83_%D1%81%D0%B2%D1%96%D1%82%D1%96 (дата звернення 27.09.2024)

63. Ріхтер Дж. CLR via C#. Програмування на платформі Microsoft.NET Framework 4.5 мовою C#. / Дж. Ріхтер. – 2020. – 896 с.

64. Сенько В.І. Електроніка і мікропроцесорна техніка / В.І. Сенько, В.П. Лисенко, О.М. Юрченко, В.Є. Лукін, А.А. Руденський — К. «Агроосвіта», 2015. — 676 с.

65. Сила Коріоліса. [Електронний ресурс]. - Режим доступу: https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BB%D0%B0_%D0%9A%D0%BE%D1%80%D1%96%D0%BE%D0%BB%D1%96%D1%81%D0%B0 (дата звернення 20.10.2024)

66. Створення роботів за допомогою ROS (Robot Operating System). [Електронний ресурс]. - Режим доступу: <https://dou.ua/lenta/articles/robots-in-containers/> (дата звернення 27.09.2024)

67. Стефанюк О. Аналіз концепції цифрових двійників / О. Стефанюк, О.В. Маціук // VI Міжнародна студентська науково - технічна конференція "ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ. АКТУАЛЬНІ ПИТАННЯ". – 2023. - с. 179-180

68. Томашевський В.М. Моделювання систем. Підручник / В.М. Томашевський.- К .: Видавнича група ВНУ, 2015. – 352 с.

69. Уривський Л.О. Імітаційне моделювання систем і процесів у телекомунікаціях. Навчальний посібник / Л.О. Уривський, А.В. Мошинська, С.О. Осипчук. – Київ, КПІ ім. Ігоря Сікорського. – 2022. – 202 с.

70. Хорольський В.П. Мехатроніка (мехатроніка та інтелектуальна автоматика): навчальний посібник / В.П. Хорольський, Ю.М. Коренець. – Кривий Ріг: ДонНУЕТ, 2023. – 342 с.

71. Хусаїнов Д.Я. Введення в моделювання динамічних систем: Навчальний посібник / Д.Я. Хусаїнов, І.І. Харченко, А.В. Шатирко. – Київський національний університет імені Тараса Шевченка. – 2010. – 132 с.

72. Цвіркун Л.І. Робототехніка та мехатроніка: навч. посіб. / Л.І. Цвіркун, Г. Грулер ; під заг. ред. Л.І. Цвіркуна ; М-во освіти і науки України, Нац. гірн. ун-т. – 3-тє вид., переробл. і доповн. – Дніпро: НГУ, 2017. – 224 с.

73. Цифрові двійники для промислового застосування. [Електронний ресурс]. – Режим доступу: <https://atop.kpi.ua/wp->

content/uploads/2021/12/iic_digital_twins_industrial_apps_white_paper_2020-02-18-ukr.pdf (дата звернення 15.09.2024)

74. Швед О. Введення в систему автоматизованого проектування САТІА V5 / О. Швед. – 71 с.

75. Яковенко А.В. Основи програмування Python. Частина 1: підручник для студ. спеціальності 122 "Комп'ютерні науки", спеціалізації "Інформаційні технології в біології та медицині" / А. В. Яковенко ; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 195 с.

Додаток А

Стаття у журналі «Вимірювальна та обчислювальна техніка в технологічних процесах»

УДК 681.5:629.3

DOI: <https://doi.org/10.31891/2219-9365-2024-80-25>

СЕМЕНИШЕН А.Л.

Студент АКІТРМ-23-1

СОКОЛАН Ю.С.

<https://orcid.org/0000-0002-0273-5719>

sokolan.julia@gmail.com

МАЙДАН П.С.

<https://orcid.org/0000-0003-3319-8730>

maidanps@gmail.com

МАКАРИШКІН Д.А.

<https://orcid.org/0000-0003-3447-811X>

makaryshkinde@khmnu.edu.ua

Хмельницький національний університет

РОЗРАХУНОК ПАРАМЕТРІВ ЦИФРОВОГО ДВІЙНИКА РОБОТА-МАНІПУЛЯТОРА PANDA ARM

В останні десятиліття спостерігається зростання ступеня автоматизації технологічних процесів шляхом впровадження роботизованого обладнання. Відповідно, зростають вимоги до точності функціонування такого обладнання. Одним із шляхів пришвидшення та більш точного забезпечення цих показників є інтеграція моделювання та подальшої симуляції роботи такого обладнання як невід'ємний елемент розробки роботизованого обладнання. Віртуальне тестування ефективності функціонування роботизованого обладнання виконується на основі розробки високоточних віртуальних моделей, складних за конфігурацією, які отримали назву цифрових двійників.

У роботі розглядається створення цифрового двійника робота-маніпулятора Panda Arm на етапі розрахунку траєкторії його руху із забезпеченням максимальної ефективності та мінімального енергоспоживання. Для розрахунку траєкторії руху використовувався метод параболічного змішування, який передбачає розподіл траєкторії на лінійні ділянки руху із нульовим прискоренням та фіксованою швидкістю та параболічні фази руху, яким характерні максимальне прискорення.

Також в роботі проведено аналіз математичної моделі оцінки енергоспоживання на основі перетворення Лагранжа та рекурсивного алгоритму Ньютона-Ейлера. Виокремлення зазначених математичних залежностей дозволить в подальшому при роботі із цифровим двійником виконувати функції робота-маніпулятора при мінімальному енергоспоживанні. При розрахунку енергоспоживання Panda Arm окрім основних характеристик також враховувались такі критичні умови роботи, як максимальна швидкість та прискорення, а також враховувався коефіцієнт корисно дії та втрати струму через опір двигуна.

Ключові слова: робот-маніпулятор, Panda Arm, віртуальний прототип, імітаційна модель, цифровий двійник, метод параболічного змішування, рекурсивний алгоритм Ньютона-Ейлера, перетворення Лагранжа.

SEMENYSHEN Andriy, SOKOLAN Iuliia,

MAIDAN Pavlo, MAKARYSHKIN Denys,

Khmelnytskyi National University

CALCULATION OF THE PARAMETERS OF THE DIGITAL TWIN OF THE PANDA ARM ROBOT MANIPULATOR

In recent decades, there has been an increase in the degree of automation of technological processes through the introduction of robotic equipment. Accordingly, the requirements for the accuracy of such equipment are increasing. One of the ways to accelerate and more accurately ensure these indicators is to integrate modeling and subsequent simulation of the operation of such equipment as an

integral element of robotic equipment development. Virtual testing of the efficiency of robotic equipment is carried out on the basis of the development of high-precision virtual models, complex in configuration, which are called digital twins.

This paper considers the creation of a digital twin of the Panda Arm robot manipulator at the stage of calculating its motion trajectory to ensure maximum efficiency and minimum power consumption. To calculate the motion trajectory, the parabolic mixing method was used, which involves dividing the trajectory into linear motion sections with zero acceleration and fixed speed and parabolic motion phases characterized by maximum acceleration.

The paper also analyzes the mathematical model for estimating energy consumption based on the Lagrange transform and the recursive Newton-Euler algorithm. The isolation of these mathematical dependencies will allow further work with a digital twin to perform the functions of a robot manipulator with minimal power consumption. When calculating the power consumption of Panda Arm, in addition to the main characteristics, such critical operating conditions as maximum speed and acceleration were also taken into account, as well as the efficiency and current losses due to motor resistance.

Keywords: robotic manipulator, Panda Arm, virtual prototype, simulation model, digital twin, parabolic mixing method, recursive Newton-Euler algorithm, Lagrange transform.

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОКІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

В останні десятиліття спостерігається зростання ступеня автоматизації технологічних процесів шляхом впровадження роботизованого обладнання. Відповідно, зростають вимоги до точності функціонування такого роботизованого обладнання, а також до таких його властивостей, як якість, дизайн, терміни виготовлення продукції, тощо.

Одним із шляхів пришвидшення та більш точного забезпечення цих показників є інтеграція моделювання та подальшої симуляції роботи такого обладнання як невід'ємний елемент розробки роботизованого обладнання [1]. Інструментом для тестування механічних конструктивних та електронних рішень роботизованого обладнання стала концепція віртуального введення роботів в експлуатацію шляхом створення їх віртуальних прототипів. Це дозволяє розробникам провести перевірку системи на дотримання в ній проектних специфікацій, а також розширює можливості у дослідженнях варіантів проектування роботів.

Віртуальне тестування ефективності функціонування роботизованого обладнання виконується на основі розробки високоточних віртуальних моделей, складних за конфігурацією. Такі моделі отримали назву цифрових двійників. Вони представляють собою формальне цифрове представлення певної системи чи процесу, який здатен зафіксувати поведінку та функціональні властивості роботизованого обладнання [2, 3]. Умовно цифровий двійник можна розділити на три складові: віртуальну модель, фізичний продукт та дані, якими вони обмінюються.

Запровадження тенденції до використання цифрових двійників у технологічному процесі може призвести до зменшення витрат, скорочення загального часу випуску продукції, яка передбачається технологічним процесом, зменшення кількості можливих помилок в процесі експлуатації, більш детальну перевірку граничних значень роботи системи, тощо.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

При створенні цифрових двійників та проектування роботів-маніпуляторів існує ряд підходів, серед яких один із найбільш поширених – це створення імітаційної моделі. Такий метод передбачає створення певного сценарію розробки продукту, в який входить декілька послідовних кроків. Виконання цих кроків передбачається у V-подібній формі, тому в літературі достатньо часто можна зустріти поняття створення V-моделі системної інженерії. До цих кроків відносяться:

1. Проведення аналізу системних вимог.
2. Проведення безперервного аналізу та моделювання системи, що проектується.
3. Спеціалізоване проектування для заданої предметної області, наприклад, машинобудування.
4. Проведення інтерактивної перевірки системи, яка повинна при цьому забезпечувати всі властивості, виокремлені на першому етапі.
5. Власне введення системи в експлуатацію.

В загальному існує 5 класів віртуальних прототипів, до яких відносяться візуалізація, механічна підгонка, оцінка виробництва, аналіз людського фактору та тестування продуктивності і функціональності. Остання клас є ключовим при розробці будь-якого віртуального прототипу [4].

В роботі розглядається модель цифрового двійника робота-маніпулятора Panda Arm. Franka Emika Panda (рис. 1) -це робот-маніпулятор, який має 7 осей та працює на відстані до 850 мм із навантаженням до 3 кг. Технічні характеристики Panda Arm наведені у табл. 1. [5, 6].

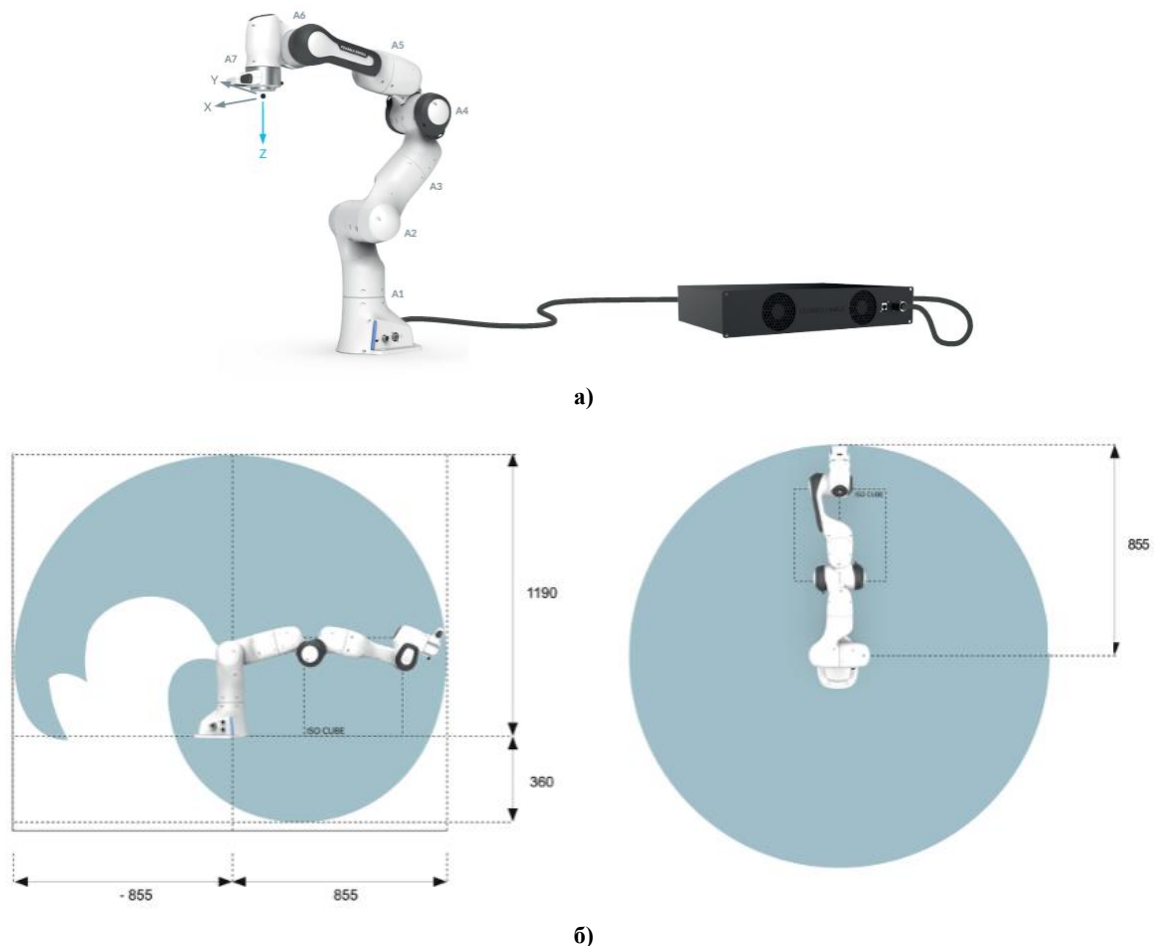


Рис. 1. Робот-маніпулятор Panda Arm: а – зовнішній вигляд; б – робочий простір

Конструкція робота передбачає наявність семи вузлів, що обертаються. Крім того, в ньому наявно більше 100 датчиків, призначених для контролю та високої точності при вимірюванні сил та крутних моментів. Керування роботом передбачається через Robot Operating System (ROS) [7, 8].

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Метою роботи є: розробка моделі цифрового двійника робота-маніпулятора Panda Arm на основі імітаційного моделювання.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Для створення моделі цифрового двійника Panda Arm в першу чергу необхідно розглянути його кінематичні параметри, до яких відносяться довжина ланки, навантаження, тощо. Динамічні параметри, такі як вага ланки або інерційні властивості, також несуть важливу інформацію. В рамках проведення дослідження кінематичні параметри обирались із технічних характеристик, зазначених у технічному паспорті робота-маніпулятора [6]. Динамічні параметри функціонування Panda Arm розглядались у роботі [9].

В якості середовища для оцінки функціональності Panda Arm було обрано Unity. Unity представляє собою програмне середовище із можливістю створення дво- та тривимірних симуляцій, а також він містить функціонал для доповненої та віртуальної реальності [10]. Вибір цієї платформи продиктований тим, що в ній підтримується фізика твердих тіл, тканин, стиснення структур, тощо. Імпортування Panda Arm в середовище було виконано за допомогою UrdfImporter з ROS (рис. 2).

Незалежно від сфери застосування робота-маніпулятора, основною його характеристикою в першу чергу виступає траєкторія руху. Траєкторії, в залежності від поставлених задач та вимог технологічного процесу, можуть мати різну складність, довжину, кривизну та інші параметри. Незалежно від перерахованих параметрів, точність повинна підтримуватись на високому рівні.

Для створення віртуальної моделі *траєкторії руху* робота-маніпулятора Panda Arm використовувався метод Кунца та Стілмана, який отримав назву методу параболічного змішування [11]. По своїй суті метод передбачає перетворення траєкторії руху робота на величину, параметризовану в часі. На рис. 3 показана ідея апроксимації прямої лінії траєкторії руху, яка з'єднує дві кінцеві точки, а з'єднання виконується параболічними функціями.

Технічні характеристики робота-маніпулятора Panda Arm [6]

Характеристика	Значення
Вага	17,8 кг
Максимальна досяжність	855 мм
Ступені свободи	7
Очікуваний термін служби	20 000 годин
Навантаження	3 кг
Робоча відносна вологість повітря	20-80 %
Енергоспоживання	80 Вт
Робочі температури навколишнього середовища	Ідеальні: 15-25 °С; розширені: 5-45 °С
Енергоживлення	100-240 В
Відхилення траєкторії	<+/- 1.25 мм
Час виявлення зіткнення	<2 мс
Найгірший час реакції на зіткнення	<100 мс
Обмеження швидкості по вузлам	A1-A4: 150°/с; A5-A7: 180°/с
Повторюваність позиціонування (відповідно до ISO 9283)	<+/- 0.1 мм
Мінімальне контрольоване зусилля	0.05 Н

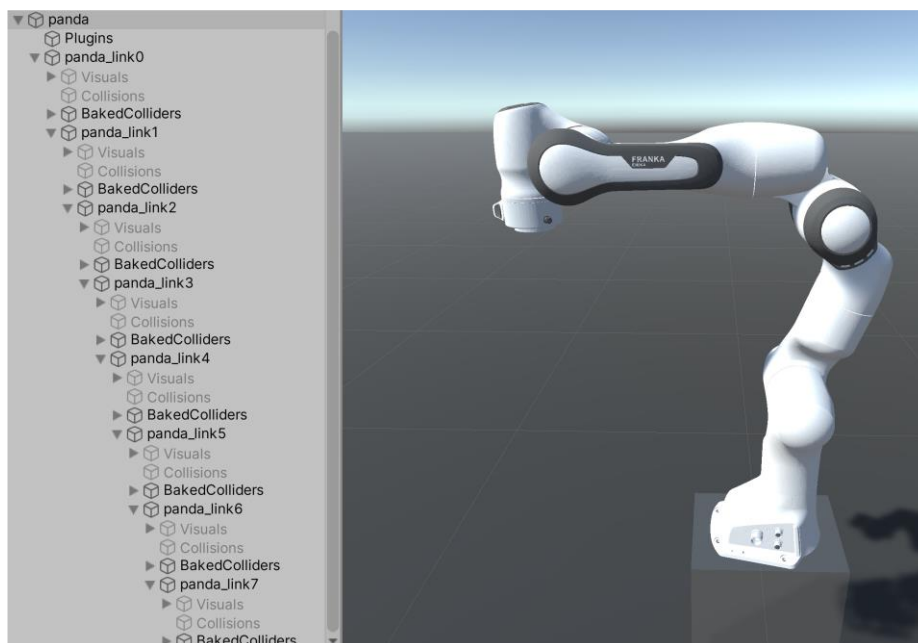


Рис. 2. Робот-маніпулятор Panda Arm в програмному середовищі Unity

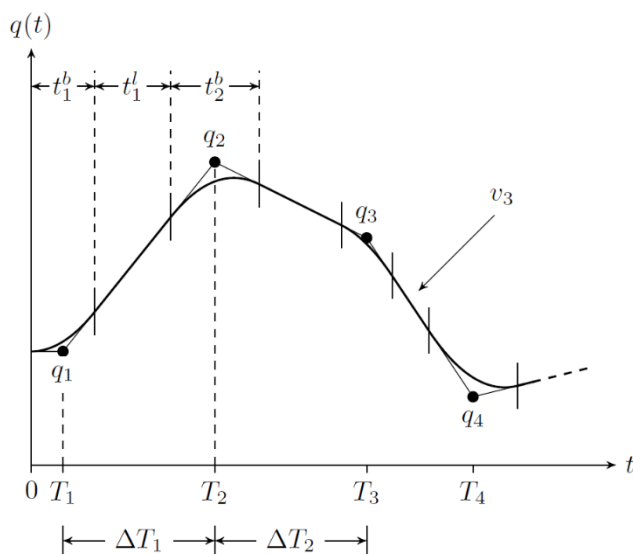


Рис. 3. Траєкторія руху за методом параболічного змішування [11]

Точки початку та кінця траєкторії робота-маніпулятора позначаються $q_i, i \in \{1 \dots n\}$, де n – це загальна кількість конфігурацій на шляху. У випадку робота-маніпулятора Panda Arm, конфігурація представлена вектором, що складається із 7 елементів, які відповідають суглобам руху. З'єднання виконується прямими відрізками. Таким чином, траєкторія представляє собою функцію q , яка на певний момент часу відображає тривалість траєкторії t_f , тобто:

$$q: [0, t_f] \rightarrow C. \quad (1)$$

Для траєкторії руху робота-маніпулятора застосовується два основних обмеження: обмеження на швидкість руху та прискорення протягом переміщення робота-маніпулятора по всій траєкторії та нульова швидкість в кінцевих точках траєкторії. Ці умови записуються у наступному вигляді:

$$\forall t: |q(\dot{t})| \leq v_{max} \wedge |\ddot{q}(t)| \leq a_{max}. \quad (2)$$

Траєкторія руху робота-маніпулятора складається з двох фаз, які є основними: з параболічної та лінійної, коли робот виконує рухи по прямому відрізку. На лінійній фазі траєкторії прискорення нульове, а швидкість постійна. Вхід в параболічну фазу характеризується переходом в постійне прискорення, тобто спостерігається залежність від часу в квадратичному характері. Тобто, за час ΔT_i Panda Arm виконує переміщення між точками q_i та q_{i+1} . Якщо відомі ці характеристики та швидкість руху робота-маніпулятора при його проходженні в лінійній фазі між точками є постійною величиною, то розрахунок швидкості можна представити у вигляді:

$$v_i = \frac{q_{i+1} - q_i}{\Delta T_i}. \quad (3)$$

При русі Panda Arm з такою швидкістю по лінійній траєкторії руху протягом всього періоду часу призведе до наявності швидкостей, що відрізняються одна від одної, на кожному переході з одного лінійного відрізка на інший. В такому випадку, частини двох сусідніх траєкторій перетворюються на рух по параболі, якому характерне постійне прискорення. Розрахунок такого прискорення виконується за наступною формулою:

$$a_i = \frac{v_i - v_{i-1}}{t_i^b}. \quad (4)$$

Для врахування того, що дві параболічні фази траєкторії руху не накладатимуться одна на одну, використовується залежність:

$$t_{i+1}^b + t_i^b \leq 2\Delta T_i. \quad (5)$$

Ввівши такі параболічні фази змішування, робот-маніпулятор починає виконувати рухи по лінійній траєкторії за час $T_i + \frac{t_i^b}{2}$ між точками q_i та q_{i+1} . На рис. 3 видно, що початок траєкторії руху не відповідає умові $t = T_0$, при чому кінець траєкторії також не відповідає умові $t = T_n$. Розрахунок загальної тривалості руху робота-маніпулятора по траєкторії від нульової точки до точки n виконується за формулою:

$$t_f = \frac{t_0^b}{2} + \sum_{i=1}^n \Delta T_i + \frac{t_n^b}{2}, \quad (6)$$

де ΔT_i – час переміщення робота між двома точками, які не відповідають кінцевим точкам траєкторії, хв;

t_0^b і t_n^b – час проходження траєкторії на першому та останньому лінійному відрізку відповідно, хв;

t_f – загальний час руху робота по траєкторії, хв.

В результаті розрахунок остаточної траєкторії руху робота-маніпулятора Panda Arm за методом параболічного змішування можна визначити за формулою:

$$q(t) = \begin{cases} q_i + \frac{1}{2}a_i \left(t - T_i + \frac{t_i^b}{2} \right)^2 + v_{i-1}(t - T_i), \\ \text{якщо } T_i - \frac{t_i^b}{2} \leq t \leq T_i + \frac{t_i^b}{2}, \quad i \in \{1 \dots n\}, \\ q_i + v_i(t - T_i), \\ \text{якщо } \frac{t_i^b}{2} + T_i \leq t \leq \frac{t_{i+1}^b}{2} + T_{i+1}, \quad i \in \{1 \dots n - 1\} \end{cases} \quad (7)$$

На основі формули 7 можна провести розрахунок траєкторії в тому випадку, якщо в цифровому двійнику робота-маніпулятора відомі часові проміжки точок та фаз. У випадку, якщо ці проміжки невідомі, за методикою Стілмана та Кунца [11] пропонується визначення цих значень за ітераційним підходом. Він передбачає вибір часових змінних таким чином, щоб на траєкторії робот-маніпулятор досягав максимальної швидкості руху. Такий розрахунок виконується наступним чином:

$$\Delta T_i = \max \frac{|q_{i+1}[j] - q_i[j]|}{v_{max}}, \quad (8)$$

де j – це номер вузла.

При такому виборі максимальних швидкостей руху суглобів Panda Arm враховується, що хоча б один із них буде рухатись із максимальною швидкістю. Також за наступною формулою виконується максимізація можливого прискорення робота-маніпулятора на траєкторії руху:

$$t_i^b = \max \frac{|v_{i+1}[j] - v_i[j]|}{a_{max}}. \quad (9)$$

Після розрахунку за умови максимального прискорення та швидкості можна провести остаточний розрахунок траєкторії руху та застосовувати її в імітаційній моделі робота-маніпулятора Panda Arm.

Оцінка енергоспоживання Panda Arm для різних траєкторій руху проводиться базуючись на електричній потужності, що подається на кожен двигун. Загальне перетворення динаміки руху робота по розрахованій та змодельованій траєкторії можна отримати за допомогою перетворень Ньютона-Ейлера [12] та Лагранжа [13]:

$$\tau = M(q)\ddot{q} + g(q) + C(q, \dot{q})\dot{q}, \quad (10)$$

де q – кутове положення Panda Arm;

\dot{q} – швидкість руху Panda Arm;

\ddot{q} – прискорення руху Panda Arm;

τ – крутний момент суглоба;

$g(q)$ – гравітаційна сила;

$M(q)$ – матриця мас у просторі;

$C(q, \dot{q})$ – діючі на робота відцентрові та коріолісові умови.

Оскільки значення прискорення та швидкості руху по ділянках траєкторії нам відомі наряду із координатами положення робота, то при моделюванні цифрового двійника можна провести визначення крутних моментів суглобів Panda Arm в будь-який момент часу його роботи. Цей підхід отримав назву рекурсивного алгоритму Ньютона-Ейлера (RNEA) [14, 15].

Необхідна для роботи двигуна Panda Arm електрична потужність $P_{el}^j(t)$ на суглобі j обчислюється наступним чином:

$$P_{el}^j(t) = P_{loss}^j(t) + P_{motor}^j(t), \quad (12)$$

де $P_{motor}^j(t)$ – це перетворювана в механічну енергію потужність двигуна;

$P_{loss}^j(t)$ – потужність, що втрачається через протікання струму двигуна $I_{motor}^j(t)$ через клему із опором R_T . Ця величина розраховується за наступною формулою:

$$P_{loss}^j(t) = R_T \cdot I_{motor}^j(t)^2. \quad (13)$$

Механічна потужність, в свою чергу, отримується із наступної залежності:

$$P_{mech}^j(t) = \tau^j(t) \cdot \omega^j(t), \quad (14)$$

де $\omega^j(t)$ – це кутова швидкість руху суглоба Panda Arm;

$\tau^j(t)$ – це крутний момент j -ого суглоба Panda Arm у момент часу t .

При розрахунку потужності двигуна, необхідного для генерування заданої механічної потужності за формулою 14, також необхідно враховувати швидкість двигуна та передаточний момент із врахуванням коефіцієнта корисної дії (ККД) редуктора та двигуна. Суглоб Panda Arm обертається із відомою швидкістю $\omega^j(t)$, яка, в свою чергу, пов'язана із частотою обертання двигуна наступним співвідношенням:

$$n_{motor}^j(t) = R \cdot \omega^j(t) \cdot \frac{60}{2\pi}. \quad (15)$$

ККД редуктора n_{gear} залежить від швидкості обертання двигуна, тобто зберігається залежність – чим більше обертів на хвилину, тим меншим буде ККД редуктора. Ця залежність показана на рис. 4 і становить:

$$\begin{aligned} n_{gear}(n_{motor} = 3000) &= 0,58; \quad n_{gear}(n_{motor} = 2000) = 0,63; \\ n_{gear}(n_{motor} = 1000) &= 0,69; \quad n_{gear}(n_{motor} \leq 500) = 0,75. \end{aligned} \quad (16)$$

Слід враховувати, що можуть існувати два випадки, що відрізняються між собою. Перший відповідає стану, коли двигун перебуває у стані рекуперації, а другий – коли двигун перебуває у стані прискорення. Відповідно, використовуються дві наступні формули:

$$\tau_{motor}^j(t) = n_{gear} \cdot \frac{\tau^j(t)}{R} \cdot n_{motor} \quad (17)$$

$$\tau_{motor}^j(t) = \frac{1}{n_{motor} \cdot n_{gear}} \cdot \frac{\tau^j(t)}{R} \quad (18)$$

Тобто, розрахунок потужності двигуна можна представити у вигляді:

$$P_{motor}^j(t) = n_{motor}^j(t) \cdot 2\pi \cdot \tau_{motor}^j(t). \quad (19)$$

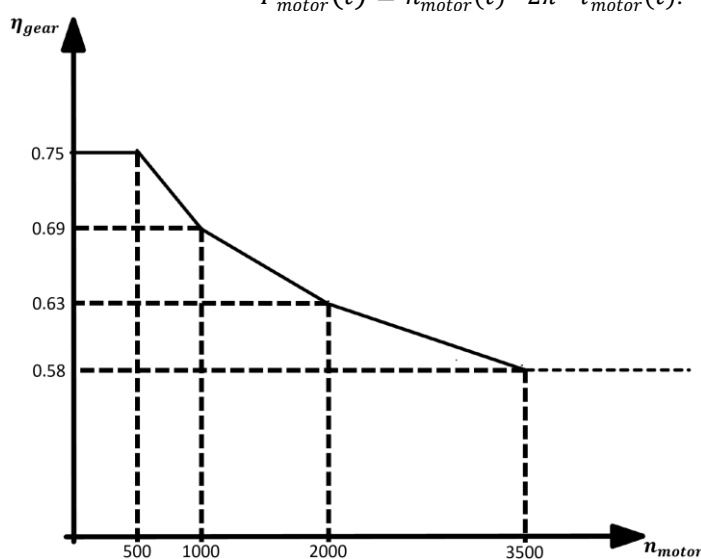


Рис. 4. Залежність між швидкістю обертання двигуна та ККД редуктора

При визначенні можливих втрат потужності необхідно в якості вхідних даних мати значення струму $I_{motor}^j(t)$, яке пропорційне крутному моменту. Крутний момент та струм двигуна можна пов'язати константами пропорційності, які за методом називають константами крутного моменту k_M [15]. У випадку Panda Arm ця константа рівна 0,134, звідки випливає:

$$\tau_{motor}^j(t) = k_M \cdot I_{motor}^j(t) \rightarrow I_{motor}^j(t) = \frac{\tau_{motor}^j(t)}{k_M}, \quad (20)$$

де $\tau_{motor}^j(t)$ – крутний момент двигуна Panda Arm;

$I_{motor}^j(t)$ – струм двигуна;

k_M – константа крутного моменту.

На основі формул 12 та 20 стає можливим визначення загальної електричної потужності, що передаватиметься на двигун Panda Arm:

$$P_{el}^j(t) = P_{loss}^j(t) + P_{motor}^j(t) = R_T \cdot I_{motor}^j{}^2(t) + P_{motor}^j(t). \quad (21)$$

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

В роботі розглядається процес створення цифрового двійника робота-маніпулятора Panda Arm на основі його кінематичних та динамічних параметрів. Моделювання роботи робота-маніпулятора передбачає розрахунок траєкторії руху за заданими умовами.

Для розрахунку траєкторії руху було обрано метод параболічного змішування, при якому траєкторія

представляє собою поєднання лінійної та параболічної фази руху. В розрахунку траєкторії було враховано можливість руху Panda Arm із максимальною швидкістю та прискоренням на певних ділянках траєкторії з метою збільшення ефективності та швидкості виконання технологічного процесу роботом-маніпулятором Panda Arm.

Крім того, в роботі проведена оцінка ефективності енергоспоживання робота-маніпулятора Panda Arm за розрахованою траєкторією руху. Для такої оцінки використовувались перетворення Ньютона-Ейлера та Лагранжа.

Проведені дослідження та виокремлення математичних залежностей функціонування робота Panda Arm дозволить в подальшому провести більш детальне моделювання його роботи із передбаченням більшості можливих проблем, які можуть виникнути, таких як, наприклад, зіткнення конструктивних частин робота із іншими предметами.

Література

1. Хорольський В.П. Мехатроніка (мехатроніка та інтелектуальна автоматика): навчальний посібник / В.П. Хорольський, Ю.М. Коренець. – Кривий Ріг: ДонНУЕТ, 2023. – 342 с.
2. Стефанюк О. Аналіз концепції цифрових двійників / О. Стефанюк, О.В. Маціук // VI Міжнародна студентська науково - технічна конференція "ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ. АКТУАЛЬНІ ПИТАННЯ". – 2023. - с. 179-180
3. Цифрові двійники для промислового застосування. [Електронний ресурс]. – Режим доступу: https://atep.kpi.ua/wp-content/uploads/2021/12/iic_digital_twins_industrial_apps_white_paper_2020-02-18-ukr.pdf
4. Zorriassatine F. A survey of virtual prototyping techniques for mechanical product development / F. Zorriassatine, C. Wykes, R. Parkin, N. Gindy // Proceedings of the institution of mechanical engineers, Part B: Journal of engineering manufacture. – 2003. - vol. 217, no. 4. – p. 513-530
5. Franka Emika Panda Arm. [Електронний ресурс]. - Режим доступу: <https://robodk.com/robot/ru/Franka/Emika-Panda>
6. Data Sheet Robot Arm & Control. [Електронний ресурс]. - Режим доступу: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://pkj-robotics.dk/wp-content/uploads/2020/09/Franka-Emika_Brochure_EN_April20_PKJ.pdf
7. Створення роботів за допомогою ROS (Robot Operating System). [Електронний ресурс]. - Режим доступу: <https://dou.ua/lenta/articles/robots-in-containers/>
8. Quigley M. Programming Robots with ROS: A Practical Introduction to the Robot Operating System / M. Quigley, B. Gerkey, W.D. Smart – O'Reilly, 2015. – 448 с.
9. Gaz C. Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization / C. Gaz, M. Cognetti, A. Oliva, P. Robuffo Giordano, A. De Luca // IEEE Robotics and Automation Letters. - vol. 4, no. 4. – 2019. – с. 4147-4154
10. Unity at 10: For better – or worse – game development has never been easier. [Електронний ресурс]. - Режим доступу: <https://arstechnica.com/gaming/2016/09/unity-at-10-for-better-or-worse-game-development-has-never-been-easier/>
11. Kunz. Turning paths into trajectories using parabolic blends / T. Kunz, M. Stilman. - Georgia Institute of Technology. – 2011. -213 p.
12. Задачин В.М. Чисельні методи : навчальний посібник / В. М. Задачин, І. Г. Конюшенко. – Х. : Вид. ХНЕУ ім. С. Кузнеця, 2014. – 180 с.
13. Карбованець М.І. Теоретична механіка. Частина І. Метод Лагранжа: навчальний посібник / М.І. Карбованець, В.Ю. Лазур, Є.А. Нодь. – Ужгород: Видавництво УжНУ «Говерла», 2019. – 82 с.
14. Featherstone R. Inverse dynamics — the recursive Newton-Euler method / R. Featherstone // Robot Dynamics Algorithms. - Boston, MA: Springer US. – 1987. – pp. 65-77
15. Ang. М.Н. Encyclopedia of Robotics / М.Н. Ang, O. Khatib, B. Siciliano. – Living reference work. – 2020. – 623 p.

References

1. Khorolskyi V.P. Mechatronics (mechatronics and intelligent automation): a textbook / V.P. Khorolskyi, Y.M. Korenets - Kryvyi Rih: DonNUET, 2023. - 342 p.
2. Stefaniuk O. Analysis of the concept of digital twins / O. Stefaniuk, O. Matsiuk // VI International Student Scientific and Technical Conference "NATURAL AND HUMANITIES. ACTUAL ISSUES." - 2023. - с. 179-180
3. Digital twins for industrial applications. [Electronic resource]. – Access mode: https://atep.kpi.ua/wp-content/uploads/2021/12/iic_digital_twins_industrial_apps_white_paper_2020-02-18-ukr.pdf
4. Zorriassatine F. A survey of virtual prototyping techniques for mechanical product development / F. Zorriassatine, C. Wykes, R. Parkin, N. Gindy // Proceedings of the institution of mechanical engineers, Part B: Journal of engineering manufacture. – 2003. - vol. 217, no. 4. – p. 513-530
5. Franka Emika Panda Arm. [Electronic resource]. - Режим доступу: <https://robodk.com/robot/ru/Franka/Emika-Panda>
6. Data Sheet Robot Arm & Control. [Electronic resource]. - Режим доступу: <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://pkj-robotics.dk/wp-content/uploads/2020/09/Franka->

Emika_Brochure_EN_April20_PKJ.pdf

7. Creating robots using ROS (Robot Operating System). [Electronic resource]. - Access mode: <https://dou.ua/lenta/articles/robots-in-containers/>.

8. Quigley M. Programming Robots with ROS: A Practical Introduction to the Robot Operating System / M. Quigley, B. Gerkey, W.D. Smart – O'Reilly, 2015. – 448 p.

9. Gaz C. Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization / C. Gaz, M. Cognetti, A. Oliva, P. Robuffo Giordano, A. De Luca // IEEE Robotics and Automation Letters. - vol. 4, no. 4. – 2019. – p. 4147-4154

10. Unity at 10: For better – or worse – game development has never been easier. [Electronic resource]. - Access mode: <https://arstechnica.com/gaming/2016/09/unity-at-10-for-better-or-worse-game-development-has-never-been-easier/>

11. Kunz. Turning paths into trajectories using parabolic blends / T. Kunz, M. Stilman. - Georgia Institute of Technology. – 2011. -213 p.

12. Zadachyn V.M. Numerical methods: a textbook / V.M. Zadachyn, I.G. Koniushenko - Kh.

13. Karbovanets M.I. Theoretical mechanics. Part I. Lagrange's method: a textbook / M.I. Karbovanets, V.Y. Lazur, E.A. Nod - Uzhhorod: UzhNU Publishing House “Hoverla”, 2019. - 82 p.

14. Featherstone R. Inverse dynamics — the recursive Newton-Euler method / R. Featherstone // Robot Dynamics Algorithms. - Boston, MA: Springer US. – 1987. – pp. 65-77

15. Ang. M.H. Encyclopedia of Robotics / M.H. Ang, O. Khatib, B. Siciliano. – Living reference work. – 2020. – 623 p.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Семенишен Андрій Леонідович

Тема: Метод оцінювання робота маніпулятора Panda Arm на основі імітаційного моделювання

Спеціальність: 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Обсяг кваліфікаційної роботи:

Кількість листів креслень _____ Кількість сторінок записки _____ 94 _____

1. Короткий зміст роботи та прийнятих рішень: розроблено метод оцінювання робота маніпулятора Panda Arm на основі імітаційного моделювання

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі було проведено огляд та аналіз підходів, які застосовуються при створенні віртуальних прототипів. В другому розділі виконано огляд функціональних можливостей методу оцінки робота-маніпулятора на основі імітаційного моделювання як системи вцілому, так і як кожного її окремого компонента. У третьому розділі наведено два конкретні приклади використання розробленого методу оцінки функціональності робота-маніпулятора, які демонструють можливість застосування методу. В четвертому розділі наведені результати моделювання траєкторій руху Panda Arm при підйомі об'єктів з коробок та їх подальшого переміщення на конвеєр. Ці траєкторії руху моделювались для двох можливих геометрій коробок. Створені траєкторії були проаналізовані з точки зору можливих зіткнень із іншими об'єктами. Такі хибні траєкторії були відсіяні та не приймали участь у подальшому аналізі.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: в роботі розглядається конкретний приклад застосування робота-маніпулятора, а не процес створення траєкторій руху для цього роботу незалежно від поставлених задач

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

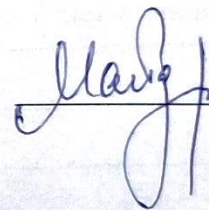
8. Інші зауваження: відсутні

9. Оцінка дипломної роботи: задов. (3,50/5)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Майдан Павло Сергійович, кандидат технічних наук, доцент кафедри МАЕЕС Хмельницького національного університету

“13” 12 2024 р.

 (підпис)

Завідувачу кафедри АКІТтаР
д-ру техн.наук, проф. Мартинюку В.В.

Семенишена А.Л.

ПІБ здобувача вищої освіти

ФІТ, 2 курс, групи АКІТРм-23-1

ЗАЯВА

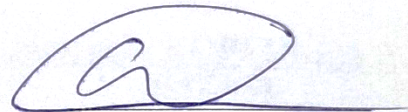
З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність плагіату ознайомлений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

02.12.24

дата



підпис

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Андрій СЕМЕНИШЕН

Співавтор:

Назва: МКР Семенишен

Науковий керівник: Денис МАКАРИШКІН

Підрозділ: Кафедра автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

Коефіцієнт подібності 1: 1.8%

Коефіцієнт подібності 2: 0%

Мікропробіли: 0

Заміна букв: 27

Інтервали: 0

Білі знаки: 2

Дата створення звіту: 2024-12-13 08:20:22.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

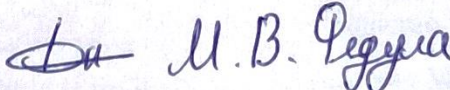
Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2024-12-13

Дата



експерт

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 9%**

ID: 158538 Назва: МКР Метод оцінювання робота маніпулятора Panda Arm на основі імітаційного Додано в БД: 2024-12-13 Автора: Андрій СЕМЕНИШЕН Керівники: Денис МАКАРИШКІН Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	98159	1372	3635 (4%)	54 (4%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ АВТОМАТИЗАЦІЇ, КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ ТА
РОБОТОТЕХНІКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод оцінювання роботи маніпулятора Panda Arm на основі імітаційного моделювання

Автор: Семенишен Андрій Леонідович

Спеціальність: 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Освітня програма: Освітньо-професійна програма «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

Науковий керівник: Макаришкін Денис Анатолійович, кандидат технічних наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у переліку джерел посилання;

2) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

3) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає 1,75% і адресується до 46 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи



Валерій МАРТИНЮК

Валерій МАРТИНЮК

Денис МАКАРИШКІН