

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)
Моделювання робота-маніпулятора та його функцій із використанням
програмного забезпечення CoppeliaSim
(тема)

Виконав:
здобувач 4 року навчання,
групи АКТАКІТ-21-1
Дарина НІНОВА
(власне ім'я, прізвище)

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(повна назва освітньої програми)

Керівник доц. Артем БРОННІКОВ
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри КІТАР

(підпис)

Ігор НЕВЛЮДОВ
(власне ім'я, прізвище)

2025 р.

Я, Нєнова Дарина Валеріївна, як здобувачка вищої освіти ХНУРЕ, розумію та підтримую політику закладу з академічної доброчесності. Я не надавала і не одержувала недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовувала штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«29» травня 2025 р.



(підпис)

Дарина НЄНОВА

Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
Рівень вищої освіти перший (бакалаврський)
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва)
Тип програми освітньо-професійна
Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«30» квітня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Неновій Дарині Валеріївні
(прізвище, ім'я, по батькові)

1. Тема роботи Моделювання робота-маніпулятора та його функцій із використанням програмного забезпечення CoppeliaSim

затверджена наказом університету від 19 травня 2025 р. № 390 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 09 червня 2025 р.

3. Вихідні дані до роботи Комп'ютерне моделювання в середовищі CoppeliaSim

Модель робота типу Pick-and-Place The Universal Robots UR5

Кінематичне моделювання переміщення

4. Перелік питань, що потрібно опрацювати в роботі Аналіз стану автоматизованих систем; вибір та обґрунтування технічних рішень для моделювання; розроблення моделі роботизованого виробничого процесу із використанням програмного забезпечення CoppeliaSim, яка включає проектування та написання програмного забезпечення для керування роботом-маніпулятором; оформлення пояснювальної записки кваліфікаційної роботи.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) слайди у форматі PowerPoint у кількості 15 слайдів з розширенням .pptx.

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз технічного завдання	30.04.2025	виконано
2	Аналіз літератури за темою	02.05.2025	виконано
3	Виконання розділу 1 Аналіз сучасного стану автоматизованих систем	10.05.2025	виконано
4	Виконання розділу 2 Вибір та обґрунтування технічних рішень для моделювання	18.05.2025	виконано
5	Виконання розділу 3 Розроблення моделі роботизованого виробничого процесу із використанням програмного забезпечення CoppeliaSim	20.05.2025	виконано
6	Оформлення пояснювальної записки	29.05.2025	виконано
7	Подання роботи на перевірку Інтернет-сервісом StrikePlagiarism	01.06.2025	виконано
8	Подання роботи на рецензію	03.06.2025	виконано
9	Подання роботи на підпис зав. кафедри	07.06.2025	виконано
10	Подання атестаційної роботи в ЕК	09.05.2025	виконано

Дата видачі завдання 30 квітня 2025 р.

Здобувач Дарина НСНОВА
(підпис)

Керівник роботи _____
(підпис)

доц. Артем БРОННІКОВ
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 75 с., 4 табл., 26 рис., 3 дод., 36 джерел.

РОБОТ-МАНІПУЛЯТОР, PICK-AND-PLACE, РОБОТ, COPPELIASIM,
ЗВОРОТНА (ІНВЕРСНА) КІНЕМАТИКА.

Об'єкт розробки – процес моделювання робототехнічних систем типу Pick-and-Place.

Предметом розробки є методи та засоби моделювання робота-маніпулятора для автоматизації процесів захоплення та переміщення об'єктів у віртуальному середовищі CoppeliaSim.

Мета роботи – моделювання робота-маніпулятора та його функцій у віртуальному середовищі CoppeliaSim для демонстрації автоматизованого процесу захоплення та переміщення об'єктів.

У першому розділі було проведено аналіз сучасного стану автоматизованих систем та огляд літератури за темою.

У другому розділі було обрано та обґрунтовано технічні рішення для моделювання. Проведено порівняльний аналіз переваг та недоліків обраного програмного забезпечення, а також проведено математичні розрахунки.

У третьому розділі було розроблено та змодельовано робот-маніпулятор типу Pick-and-Place, а також створено відповідне програмне забезпечення для керування рухами робота.

У результаті розроблений макет може використовуватись для створення ділянки виробничої лінії, яка вимагає автоматизованого процесу захоплення та переміщення об'єктів.

ABSTARCT

Explanatory note: 100 pages, 4 tables, 26 figures, 36 sources according to the list of references, 3 appendices.

ROBOT MANIPULATOR, PICK-AND-PLACE MANIPULATOR, COPPELIASIM, INVERSE KINEMATICS.

Object of the study is the process of modelling Pick-and-Place type robotic systems.

Subject of the study – methods and tools for modelling a manipulator robot for automating object grasping and relocation processes in the virtual environment CoppeliaSim.

The aim of the study is modelling a manipulator robot and its functions in the virtual environment CoppeliaSim to demonstrate the automated process of object grasping and relocation.

Chapter One provides an analysis of the current state of automated system and a review of the relevant literature on the topic.

Chapter Two presents the selection and justification of technical solutions for the modelling process. A comparative analysis of the advantages and disadvantages of the chosen software is provided, along with the necessary mathematical calculations.

Chapter Three describes the development and simulation of the Pick-and-Place robotic manipulator, as well as the creation of the corresponding control software for robot motion.

As a result, the developed prototype can be used to create a section of production line that requires an automated object gripping and relocation process.

ЗМІСТ

Перелік умовних скорочень.....	7
Вступ.....	8
1 Аналіз сучасного стану автоматизованих систем	10
1.1 Аналіз стану сучасного виробництва.....	10
1.2 Маніпуляційні роботи та їх використання на сучасному виробництві	13
1.3 Основні технологічні операції на виробництві.....	17
1.4 Цифрові двійники в епоху Industry 4.0 та 5.0	19
1.4 Види моделювання.....	22
2 Вибір та обґрунтування технічних рішень для моделювання	25
2.1 Програмне забезпечення CoppeliaSim, його переваги та недоліки.....	25
2.2 Розрахунок ПД-регулятора роботизованої системи	27
2.3 Задача прямої та зворотної кінематики.....	32
3 Розроблення моделі роботизованого виробничого процесу із використанням програмного забезпечення CoppeliaSim.....	42
3.1 Схема роботи модельованої системи	42
3.2 Створення моделі системи	43
3.2.1 Моделювання та програмування функцій дволанкового маніпулятора .	44
3.2.2 Моделювання робота-маніпулятора та його функцій.....	51
3.2.3 Створення та програмування тривимірної моделі робота	58
3.4 Програмне забезпечення для керування роботом з CoppeliaSim	62
3.5 Охорона праці	66
Висновки	68
Перелік джерел посилання	70
Додаток А Апробація результатів роботи.....	75
Додаток Б Код програми	85
Додаток В Демонстраційний матеріал	91

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АСУ ТП – автоматизована система управління технологічним процесом;

ІК – інверсна кінематика;

ЛАЧХ – логарифмічна амплітудно-фазова частотна характеристика;

ПЗ – програмне забезпечення;

САУ – система автоматичного управління;

СКЗ – системи комп'ютерного зору;

ТАУ – теорія автоматичного управління;

ТЗ – технічне завдання;

ТО – технологічна операція;

ТП – технологічний процес;

ШІ – штучний інтелект;

AGV – Automated Guided Vehicles;

FMS – Flexible Manufacturing System;

IoT – Internet of Things;

FSM – Finite State Machine;

PLM – Product Lifecycle Management;

SDGs – Sustainable Development Goals;

V-REP – Virtual Robot Experimentation Platform;

WMS – Warehouse Management System.

ВСТУП

В умовах сучасного розвитку промисловості одним із ключових напрямків стає інтеграція автоматизованих систем управління технологічним процесом, засобів автоматизації, робототехнічних систем та засобів штучного інтелекту (ШІ) у виробничі процеси [1]. Цей процес обумовлений швидким ростом вимог до якості продукції, точності, безпеки, ефективного використання ресурсів та зниження впливу людського фактора. Зокрема, одним з аспектів є реалізація Цілей сталого розвитку [2], що передбачають підвищення ефективності промислового виробництва та вдосконалення підготовки фахівців.

У відповідь на нові виклики, підприємства активно застосовують концепції Industry 4.0 та 5.0, впроваджуючи роботизовані системи, які здатні функціонувати автономно та адаптуватися до змін в умовах виробництва [1]. Однак, для ефективного застосування цих технологій необхідно створити нові підходи до управління виробничими процесами. Зокрема, важливим є розв'язання питання зниження витрат, підвищення точності, ефективності та гнучкості роботизованих систем, що є важливими для швидкої адаптації до нових умов [3].

Однак для ефективного застосування цих технологій необхідно створювати нові підходи до управління виробничими процесами, що є основним завданням для інженерів та керівників виробничих систем. Таким чином, актуальність розробки обумовлена необхідністю вирішення низки питань, пов'язаних зі зниженням витрат, підвищенням ефективності та гнучкості роботизованих систем у виробничих процесах. Це особливо важливо в умовах швидко змінюваних вимог ринку та необхідності швидкої адаптації до нових умов та технологій. Зокрема, використання маніпуляційних роботів для виконання різних технологічних операцій дозволяє знизити трудові витрати, підвищити точність виконання завдання, а також забезпечити безпеку праці в небезпечних для людської життєдіяльності умовах [3]. Водночас застосування новітніх методів моделювання дає можливість прогнозування роботи систем в реальних умовах, що сприяє

оперативному виявленню дефектів та усунення проблем із продуктивністю та виконання поставлених задач.

Отже, метою роботи стало моделювання робота-маніпулятора та його функцій у віртуальному середовищі CoppeliaSim для демонстрації автоматизованого процесу захоплення та переміщення об'єктів [1, 2].

Об'єктом роботи є процес моделювання робототехнічних систем типу Pick-and-Place.

Предметом розробки є методи та засоби моделювання робота-маніпулятора для автоматизації процесів захоплення та переміщення об'єктів у віртуальному середовищі CoppeliaSim.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз сучасного стану автоматизованих систем;
- провести підбір елементної бази та обґрунтувати технічні рішення для моделювання;
- розробити схему роботи модельованої системи;
- провести математичні розрахунки для втілення поставленої задачі;
- розробити модель робота у програмному середовищі CoppeliaSim;
- створити програмне забезпечення для керування роботом із використанням мов програмування Python та Lua;
- провести симуляцію створеної системи;
- оформити кваліфікаційну роботу.

Кваліфікаційна робота оформлена згідно з ДСТУ 3008:2015 [4], навчального посібника з кваліфікаційної роботи [5]. Також матеріали кваліфікаційної роботи були апробовані у [1-2]. Проведені дослідження відповідають цілям сталого розвитку (ЦСР), а саме: ЦСР 4, ЦСР 7, ЦСР 9, ЦСР 11 та ЦСР 12 [2].

1 АНАЛІЗ СУЧАСНОГО СТАНУ АВТОМАТИЗОВАНИХ СИСТЕМ

1.1 Аналіз стану сучасного виробництва

Сучасне промислове виробництво зазнало значних трансформацій під впливом цифрових технологій, автоматизації та розвитку штучного інтелекту [6]. Традиційні методи виробництва дедалі більше поступаються місцем гнучким, адаптивним та високотехнологічним системами, які поєднують фізичне обладнання із віртуальними цифровими копіями та програмним забезпеченням для управління.

Однією з основних тенденцій є поступовий перехід від ручної праці до повної або часткової автоматизації технологічних процесів. Автоматизовані системи управління технологічними процесами (АСУ ТП) дозволяють зменшити людський фактор, знизити енергоспоживання, підвищити точність виконання операцій та зменшити кількість дефектів під час виробництва. В умовах зростаючих умов до ефективності, безпеки та сталості виробництва автоматизація є не просто перевагою, а необхідною умовою конкурентоспроможності.

Центральним компонентом сучасної автоматизації стали робототехнічні системи. Роботи, зокрема маніпулятори, застосовуються на всіх етапах виробництва – від обробки матеріалів до складання, контролю якості та пакування. Роботи можна умовно поділити на два основні типи: стаціонарні та мобільні.

Стаціонарні роботи, зазвичай закріплені на одному місці, виконують специфічні завдання на виробничих лініях. До них належать промислові роботи-маніпулятори, які здійснюють операції зварювання, фарбування, пакування, збору компонентів тощо. Такі роботи характеризуються високою точністю, передбачуваністю дій і здатністю працювати безперервно в умовах, небезпечних або складних для людини. Стаціонарні роботи можуть бути як простими з обмеженою кількістю ступенів свободи, так і високоточними багатофункціональними комплексами, інтегрованими з системами комп'ютерного зору та інтелектуального керування [7].

На відміну від стаціонарних, мобільні роботи мають здатність до переміщення у просторі. Це дозволяє їм виконувати логістичні функції, доставлення компонентів між цехами, роботу в умовах змінного середовища або навіть у відкритому просторі. Вони оснащені сенсорами, LiDAR-системами, камерами машинного зору та навігаційними алгоритмами для побудови карт місцевості, виявлення перешкод і оптимального планування траєкторій [8, с. 5–9]. Прикладами таких роботів є автономні візки Automated Guided Vehicles (AGV) або безпілотні транспортні системи, які широко застосовуються в логістиці, складах, на заводах тощо.

Комбінація стаціонарних і мобільних роботів у межах однієї виробничої системи забезпечує максимальну гнучкість, швидкість реагування на зміну вимог і безперервність виконання завдань. Завдяки цьому формується нова парадигма – гнучкі виробничі системи (FMS), які можуть самостійно перелаштовуватись відповідно до змін обсягів виробництва чи типу продукції.

Крім цього, сьогодні важливою тенденцією є не лише механічна інтеграція роботів у виробничі процеси, а і їх поєднання із цифровими інструментами. Йдеться про впровадження цифрових двійників, мережевих технологій Інтернету речей, хмарних сервісів та систем управління життєвим циклом продукту, про які надалі буде йти мова у розділі [9, 10].

Інтеграція робототехнічних систем у виробничі процеси передбачає не лише механічне виконання завдань, а й активне використання сенсорів, інтелектуальних систем прийняття рішень та адаптивного керування. Таким чином, виробництво більше не буде розглядатися як статична лінія з жорстко закріпленими етапами, а перетворюється на динамічне середовища, де кожен елемент здатен самостійно реагувати на зміну умов, задач чи середовища.

Особливу роль у цьому процесі відіграє комп'ютерний зір. Системи комп'ютерного зору (СКЗ) дозволяють роботизованим установками сприймати візуальну інформацію, аналізувати її та на основі цього приймати рішення. У сучасному виробництві СКЗ використовуються для контролю якості, розпізнавання

об'єктів, сортування, позиціонування, і навіть для спільної роботи з людьми (у так званих коботах – collaborative robots) [11].

На відміну від традиційних датчиків, які передають лише прості сигнали (наприклад, чи є об'єкт на лінії), СКЗ дозволяють визначити форму, колір, розміри, орієнтацію об'єктів у просторі. Це робить можливим виконання складних завдань, таких як:

- точне захоплення предметів довільної форми;
- інспекція дефектів поверхні на етапі контролю якості;
- адаптація траєкторії маніпулятора залежно від розташування об'єкта;
- ідентифікація компонентів за маркуванням.

Наприклад у системах типу Pick-and-Place, які покладено в основу кваліфікаційної роботи, СКЗ буде використано для виявлення кольору та типу об'єкта, після чого робот адаптуватиме свої рухи. Така гнучкість дозволяє працювати з неперервним потоком неоднорідних об'єктів – те, що раніше вимагало втручання людини.

Крім візуальної інформації, сучасні робототехнічні системи активно використовують дані з сенсорів руху, тиску, температури, сили, а також аудіо- та відеопотоки. Вся ця інформація обробляється у реальному часі, що дає змогу реалізувати адаптивні алгоритми керування, зокрема:

- управління на основі зворотного зв'язку;
- інтелектуальне планування траєкторій;
- реакція на непередбачувані події (наприклад, зупинка при виявленні перешкоди).

Такі алгоритми широко реалізуються завдяки застосуванню ШІ, зокрема технологій машинного навчання та глибокого навчання (deep learning). Використання нейронних мереж дозволяє навчати роботів новим діям на основі даних, а не лише програмуванням жорстких сценаріїв. Це особливо актуально для роботів, які повинні працювати у змінному середовищі або в умовах високої варіативності продукції [10].

Таким чином, роботизоване виробництво нового покоління має бути не лише ефективним, а й адаптивним, безпечним для людини та навколишнього середовища. Це потребує появи нових фахівців, які володіють не лише інженерними навичками, а й знаннями у сфері цифрових технологій, програмування, аналізу даних та управління проектами.

1.2 Маніпуляційні роботи та їх використання на сучасному виробництві

Вперше маніпуляційні роботи почали використовуватися в промисловості ще у 60-х роках минулого сторіччя [11]. Так маніпулятори, як складники автоматизованих систем, стали важливою частиною сучасного виробництва завдяки своїй здатності виконувати точні й ефективні маніпуляції із різноманітними об'єктами.

Маніпуляційний робот (або маніпулятор) – це автоматизована механічна система, що здатна виконувати точні маніпуляції з об'єктами у визначеному середовищі. Маніпулятори призначені для операцій по зміні просторового положення будь-яких предметів, таких як інструменти, деталі чи вузли інших машин [11, 12].

Маніпуляційні роботи складаються з ряду з'єднаних між собою рухливих частин, які забезпечують необхідну свободу руху, подібну до людської руки, але з високою точністю та швидкістю на відміну від здатностей людини. Це дає змогу виконувати маніпуляції над об'єктами в умовах, де присутність людини є небажаною або неможливою, наприклад, у небезпечних середовищах, таких як атомні станції, хімічні лабораторії, космічні дослідження тощо. Через свою багатофункціональність маніпулятори набули широкої популярності саме на промислових виробництвах, оскільки саме там найпростіше забезпечити безпечні та ефективні умови для їхньої роботи. Завдяки цьому вони отримали назву «промислові роботи» [11].

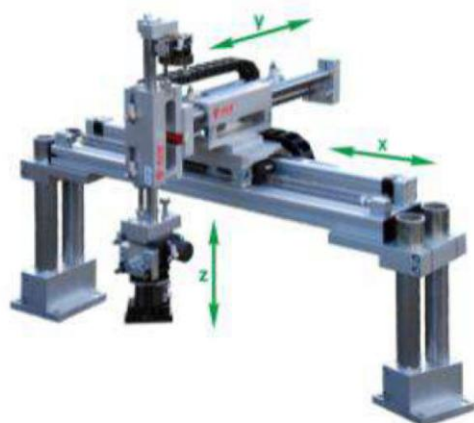
На рисунку 1.1 показано класифікацію роботів за типом механічної частини. З зазначеної на рисунку 1.1 інформації видно, що маніпуляційні роботи поділяються на:

- маніпулятори з прямокутною системою координат;
- маніпулятори з циліндричною системою координат;
- маніпулятори зі сферичною системою координат;
- маніпулятори з кутвою системою координат.



Рисунок 1.1 – Класифікація роботів за типом механічної частини [11]

Свою чергою, рисунок 1.2 [11] демонструє приклади конструкцій маніпуляційних роботів, що працюють у різних системах координат, що використовуються для керування положенням маніпулятора.



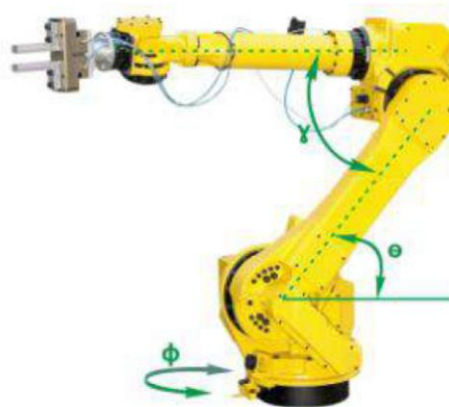
а)



б)



в)



г)

miro

- а) у прямокутній системі координат;
- б) у циліндричній системі координат;
- в) у сферичній системі координат;
- г) у кутовій системі координат

Рисунок 1.2 – Приклади конструкцій роботів-маніпуляторів, що працюють у різних системах координат

З моменту свого провадження, маніпулятори значно змінили промислове виробництво. Спочатку їх використовували для простих і повторюваних операцій, таких як транспортування і збірка деталей, однак з часом їхня функціональність значно розширилися на автомобільну та аерокосмічну промисловість, електроніку, робототехніку, медицину тощо.

Як і будь-яка технологія, використання маніпуляційних роботів на виробництві має свої переваги та недоліки. З переваг можна зазначити наступні:

- зменшення трудових витрат, адже роботи допомагають виконувати повторювані операції без втрат у якості та продуктивності;
- підвищення безпеки; як було зазначено, завдяки можливості виконання робіт в небезпечних умовах таких, як високі температури, токсичні гази, радіація тощо, використання маніпуляторів знижує ризики для життєдіяльності людини;
- покращення точності, адже маніпулятори здатні виконувати складні операції на виробництві з високою точністю на відміну від людини;
- гнучкість у налаштуванні: сучасні маніпуляційні роботи здатні адаптуватися під різні типи операцій, що робить їх універсальними для багатьох виробничих процесів.

Серед недоліків впровадження маніпуляторів на виробництві можна зазначити наступні:

- висока вартість устаткування та технічних засобів;
- необхідність кваліфікованого персоналу для обслуговування та програмування маніпуляторів;
- можливості технологічних збоїв: попри те, що використання маніпуляторів покращує точність, як і будь-яка автоматизована система, маніпулятори можуть зазнавати збоїв через апаратні чи програмні несправності, що може призвести до зупинки виробництва.

Отже, маніпуляційні роботи, як частини автоматизованих систем, за останні десятиліття стали незамінними інструментами в різних галузях промисловості завдяки своїй здатності виконувати точні, ефективні та безпечні для людської життєдіяльності дії.

Крім цього, їх використання не лише значно зменшило трудові витрати, але й забезпечило підвищену безпеку на виробництві, особливо в умовах, де присутність людини є неможливою або небезпечною. Попри певні недоліки їхнього активного впровадження в виробничі процеси, розвиток маніпуляторів продовжує набирати

обертів, що дозволяє покращувати ефективність і безпеку виробничих процесів у різних галузях виробництва.

1.3 Основні технологічні операції на виробництві

Сучасне виробництво (підприємство) являє собою технологічну систему, що складається із взаємопов'язаних по горизонталі й вертикалі підсистем [14]. Кожен етап (кожна підсистема) складається із взаємопов'язаних та структурованих компонентів – технологічних операцій (ТО). Отже, технологічні операції на виробництві є основними етапами, через які проходить виготовлення продукції, що містять різні процеси проєктування, збирання, обробки та тестування. Від ефективності кожної операції залежить загальна продуктивність виробничого процесу, якість продукції та витрати на її виготовлення. Усі технологічні операції взаємопов'язані, що дозволяє оптимізувати їх виконання та забезпечити сталість процесу [14].

Залежно від виду виробництва та специфікацій продукції, технологічні операції можуть відрізнятися, однак загалом можна виділити кілька основних категорій, що є важливими для більшості виробничих ліній. На рисунку 1.3 схематично наведено приклади основних ТО на сучасному виробництві.



Рисунок 1.3 – Основні ТО на виробництві

Обробка матеріалів є однією з найбільш поширених та важливих операцій у багатьох галузях промисловості. Вона включає різні методи змінювання фізичних, хімічних та механічних властивостей матеріалів. Прикладами механічні операції є, наприклад, токарні, фрезерні, шліфувальні операції, операції термічної обробки, складальні операції тощо. Під час виконання таких ТО відбуваються якісні зміни об'єкта, наприклад, зміна розміру, форми, зовнішнього вигляду або внутрішні властивості виробу.

Операції зі збірки та складання виробів є важливою складовою виробничого процесу, що полягає в об'єднанні окремих компонентів або підсистем у готові вироби чи технічні системи. Під час цієї операції часто використовуються промислові роботи для завантаження та розвантаження технологічних машин та верстатів, переміщення деталей, збірки електронних деталей, виробів та інше [14].

Операції контролю якості. Контроль якості є важливим етапом у ТП, оскільки цей етап забезпечує виготовлення продукції, що відповідає стандартам і вимогам замовника. Операції контролю якості можуть бути різними залежно від типу виробу та його складності. Такі типи контролю містять візуальний, механічний, електронний типи контролю. Під час кожного з типів контролю відбувається оцінка зовнішнього вигляду виробу на наявність дефектів, перевірка механічних характеристик деталей, наприклад, міцності та зносостійкості, перевірка електронних і електричних властивостей, таких як перевірка правильності роботи електричних схем чи вимірювання опору компонентів тощо.

Операції транспортування і зберігання, які складають важливу частину організації виробництва, оскільки без ефективної організації цих операцій неможливо забезпечити безперервність виробництва. Транспортування – це виробниче переміщення сировини, устаткування та готової продукції між різними ділянками виробництва або між складом та виробничою лінією. Використання автоматизованих систем транспортування, таких як конвеєри та робототехнічні установки, дозволяє значно знизити витрати часу і праці. Зберіганням, в свою чергу, називається організація складського обліку та зберігання матеріалів. Більшість

сучасних виробництв автоматизовані та використовують системи управління складом (WMS) [16] для відстеження руху виробів, щоб своєю чергою зменшити витрати та збільшити ефективність.

В умовах сталого виробництва важливим аспектом є також ефективне управління відходами та їх переробка, що активно використовується в умовах сучасного виробництва [17]. На сучасних підприємствах застосовують різноманітні методи утилізації відходів для зниження екологічного впливу та скорочення витрат.

Таким чином, технологічні операції на виробництві формують основу для організації та функціонування підприємства, а ефективність кожного етапу безпосередньо впливає на результативність всього процесу. Вдосконалення технологічних операцій, їх автоматизація та інтеграція з цифровими технологіями дозволяє підвищити конкурентоспроможність виробництва та забезпечити сталість його роботи та довгострокову перспективу.

1.4 Цифрові двійники в епоху Industry 4.0 та 5.0

Четверта промислова революція також відома як Industry 4.0 виникла внаслідок швидкого розвитку цифрових технологій, таких як Інтернет речей (Internet of Things або IoT) та кіберфізичні виробничі системи [12, 13]. Industry 4.0 здійснює позитивні зміни на підприємствах, впливаючи на структуру організацій. Своєю чергою, Industry 5.0 виникла як відповідь на виклики, які неможливо вирішити за допомогою автоматизації та робототехніки. Ця концепція більше зосереджена на співпраці людини та машини та нових соціальних потребах. Відповідно до цієї концепції, технології повинні працювати разом з людьми, підвищуючи їхні здібності, а не повністю замінюючи їх [18].

У таблиці 1.1, що побудована на основі даних [18], представлено основні характеристики, які відображають ключові відмінності між концепціями Industry 4.0 та Industry 5.0.

Таблиця 1.1 – Основні характеристики Industry 4.0 та Industry 5.0

Industry 4.0	Industry 4.0
1	2
<ul style="list-style-type: none"> – Збільшення використання роботів та автоматизованих систем для виконання рутинних завдань; – Створення віртуальних копій фізичних об'єктів, що дозволяє моделювати та контролювати їх функціонування; – Об'єднання пристроїв та систем в єдину мережу, яка дає змогу передавати та аналізувати дані в реальному часі; – Аналіз великих даних для прийняття рішень в автоматизованих системах; – Об'єднання всіх технологій для створення виробничих потужностей, які адаптуються до змін попиту, налаштувань та інших факторів. 	<ul style="list-style-type: none"> – Ключовий акцент робиться на людині, де технології використовуються для покращення умов праці, забезпечення більшої гнучкості та креативності; – Використання роботів, які працюють разом з людьми для підвищення ефективності та якості виробництва; – Більша увага до екологічних та етичних аспектів виробництва; – Технології Індустрії 5.0 дозволяють масове виробництво індивідуалізованих продуктів; – Індустрія орієнтована на екологічно сталий виробничий процес із мінімальним негативним впливом на навколишнє середовище.

Отже, поки Industry 4.0 зосереджена на автоматизації та використанні алгоритмів для оптимізації процесів, Industry 5.0 – відкриває нові можливості для розвитку людської творчості. У цій парадигмі роботи та ШІ виступають не як заміна людської праці, а як інструменти, які допомагають людям реалізувати нові ідеї. Такий підхід дозволяє розширити межі традиційного виробництва [11, 18]. Крім того, Industry 4.0 та Industry 5.0 має прямий зв'язок з ідеями, закладеними в списку цілей сталого розвитку (SDGs), а саме ці концепції сприяють зниженню шкідливого впливу на навколишнє середовище через використання ресурсоефективних рішень, зменшення викидів і покращення управління енергоспоживанням [2].

У контексті Industry 4.0 однією з важливих концепцій – є концепція використання цифрових двійників, яка бере на себе важливу роль у забезпеченні ефективності та гнучкості виробничих процесів.

Цифровий двійник (Digital Twin) – це програмний аналог фізичного пристрою, який здійснює моделювання внутрішніх процесів, технічних характеристик і поведінки реального об'єкта в умовах впливу перешкод та навколишнього середовища [19, 20].

Цифровий двійник застосовується на всіх стадіях життєвого циклу виробу, включаючи проєктування, виробництво, експлуатацію та утилізацію. На рисунку 1.4 показано склад цифрового двійника [19], який складається із даних, цифрових моделей та сервісних інтерфейсів.



Рисунок 1.4 – Структура цифрового двійника

Однією з ключових особливостей цифрового двійника є те, що для задання йому вхідних впливів використовується інформація з датчиків реального пристрою, який працює паралельно. Зокрема, у випадку цифрових двійників робота можлива як в онлайн, так і в офлайн режимах [19]. Отже, використання цифрових двійників дозволяє здійснювати моніторинг, аналіз та оптимізацію реальних систем у режимі реального часу.

Цифровий двійник виробу містить:

- геометричну в структурну модель об'єкта;
- набір розрахункових даних деталей, вузлів і виробів загалом;

- математичні моделі, які описують всі фізичні процеси, що відбуваються у виробі;
- інформацію про технологічні процеси виготовлення та збирання окремих елементів і виробу в цілому;
- систему керування життєвим циклом виробу [19].

Без створення цифрових двійників виробів неможливе впровадження сучасної технології PLM. IoT та PLM – є невіддільними атрибутами розумного виробництва (Smart Factory). Їх характерна риса – формування і використання цифрової моделі матеріальних потоків, тобто, цифрового двійника вже не окремого виробу, а виробничої системи [19]. Зокрема, відображення реальних об'єктів у цифровому вигляді дозволяє здійснювати детальну симуляцію різноманітних ситуацій, а також аналізувати поведінку об'єктів при різних умовах без необхідності фізичних експериментів. Це є потужним інструментом для підвищення точності прогнозування, оптимізації виробничих ліній і зниження ризиків, пов'язаних з несправністю обладнання чи збоєм процесів.

Отже, цифрові двійники є важливим інструментом для досягнення високої ефективності, безпеки та гнучкості в рамках Industry 4.0 та є основою для подальшого розвитку в напрямку Industry 5.0, де акцент зроблено на взаємодії між людиною та технологіями, зокрема, роботизованими системами та інтелектуальними системами управління.

1.4 Види моделювання

Сучасне гнучке виробництво неможливо уявити без промислового моделювання продукції [21]. Моделювання є важливим інструментом, оскільки дозволяє створювати цифрові аналоги фізичних об'єктів для вивчення їхніх характеристик, поведінки та взаємодії з іншими системами до етапу фактичного виробництва. Виробничі процеси, розробка нових виробів та їх оптимізація значно полегшуються завдяки використанню різних видів моделювання. Це дозволяє

зменшити витрати часу та ресурсів, підвищити точність та ефективність у проектуванні та виготовленні продукції.

Взагалі моделюванням заведено називати відтворення властивостей деякого об'єкта, який має схожі характеристики з реальним об'єктом і який використовують для дослідження й аналізу його поведінки в різних умовах [22]. Важливою характеристикою моделювання є здатність передбачати результат у реальних умовах на основі моделі, що дозволяє оптимізувати процеси ще до початку їх фізичної реалізації. У моделях відображаються процеси через взаємозв'язки показників, що дозволяє здійснювати оцінку характеристик (залежностей) та/або параметрів процесів досліджуваних систем.

Сучасні методи моделювання, що використовуються у виробничих процесах, можна поділити на кілька основних видів:

- фізичне моделювання, яке передбачає створення масштабованих або аналогічних моделей реальних об'єктів. Створення таких моделей дає змогу досліджувати фізичні властивості та поведінку матеріалів і конструкцій у реальних умовах. Наприклад, фізичні моделі використовуються для вивчення аеродинаміки, механічних властивостей матеріалів, теплових режимів тощо;

- математичне моделювання є важливим інструментом для передбачення поведінки складних систем за допомогою числових методів, методів кінцевих елементів (FEM – finite element method), методів чисельного інтегрування, моделювання потоків та багато інших [22]. Через необхідність математичного передбачення роботи системи на виробництві широкого застосування набули методи ідентифікації та моделювання об'єктів автоматизації;

- імітаційне моделювання, яке дозволяє створювати моделі виробничих процесів або технологічних систем, де можна оцінити їхню продуктивність, ефективність або інші параметри без реального втручання. Цей метод моделювання містить комплекс методів алгоритмізації функціонування об'єктів дослідження, програмної реалізації алгоритмічних описів, а також організацію, планування та виконання обчислювальних експериментів з математичними моделями на ЕОМ;

– як було зазначено раніше [19] моделювання із використанням цифрових двійників є однією з найбільш інноваційних форм моделювання в рамках Industry 4.0, адже цей підхід дає можливість здійснювати оптимізацію процесів, аналізувати можливі збої чи дефекти в системах, а також здійснювати контроль за станом обладнання без необхідності фізичних експериментів [20];

– моделювання зі створенням 2D або 3D-моделей. Такий вид моделювання передбачає створення дво- або тривимірних моделей, що застосовуються для проєктування та аналізу різних компонентів і механізмів.

Якщо говорити про 2D-моделі (плоскі моделі), то в основному вони використовуються в проєктування деталей, схем, планів, а також для створення технічної документації. 3D-моделі (тривимірні моделі) своєю чергою активно використовуються для створення моделей механічних, електричних та електронних компонентів, а також для проєктування виробничих ліній і систем. Зокрема, 3D-моделювання має три основні види: твердотільне, поверхневе та полігональне [21], використання яких залежить від окремих специфічних вимог до моделювання компонентів та/або виробничих систем.

Отже, кожен з описаних вище видів моделювання має своє специфічне застосування, яке на пряму залежить від типу виробництва, складності продукції та вимог до якості. Завдяки використанню сучасних комп'ютерних технологій та ПЗ, моделювання дозволяє значно прискорити процеси розробки та випробування нових продуктів та технологій. Таким чином, в результаті застосування різноманітних видів моделювання на виробництві можна досягти значної економії часу та ресурсів, зменшення кількості помилок, підвищення якості готової продукції та ефективності виробничих процесів, що є критично важливим в умовах сучасного виробництва в рамках Industry 4.0 та 5.0.

2 ВИБІР ТА ОБҐРУНТУВАННЯ ТЕХНІЧНИХ РІШЕНЬ ДЛЯ МОДЕЛЮВАННЯ

2.1 Програмне забезпечення CoppeliaSim, його переваги та недоліки

Для реалізації задач кваліфікаційної роботи було обрано програмне забезпечення CoppeliaSim (раніше, V-REP), яке є потужним стимулятором для моделювання, програмування та тестування роботизованих систем. CoppeliaSim дозволяє здійснювати симуляцію майже всіх типів роботів та відзначається широким набором інструментів, що значно спрощують процес моделювання порівняно з іншими подібними програмними засобами [23].

Для реалізації поставлених задач було використано версію CoppeliaSim для навчальних закладів (Education Version), що надає можливість безоплатного використання всіх функцій програмного забезпечення.

Основною мовою програмування в CoppeliaSim є Lua, що була застосована для втілення завдання практичної частини роботи. Однак, зазначене програмне забезпечення також підтримує використання мови програмування Python, що також було продемонстровано в процесі виконання технічного завдання. Можливість використання різних мов програмування, а також інтеграція з іншими програмними засобами, таким як MATLAB, є значною перевагою програмного забезпечення. Крім цього, слід підкреслити низку інших переваг обраного програмного засобу:

- гнучкість у налаштування симуляцій: як було зазначено CoppeliaSim дозволяє легко налаштовувати параметри моделей роботів завдяки зручному інтерфейсу та зрозумілій взаємодії між компонентами;

- підтримка різних типів роботів, маніпуляторів, додаткового устаткування тощо: широкий спектр мобільних (mobile) та немобільних (non-mobile) роботів, а також устаткування до них дозволяє моделювати різні системи в межах одного середовища;

– мультимовна підтримка: як було зазначено, в той час, як Lua є основною мовою для написання скриптів, CoppeliaSim підтримує також різні мови програмування, включаючи Python та MATLAB;

– власний API [24]: CoppeliaSim має багатий API, який дозволяє інженерам-програмістам в області робототехніки контролювати симуляцію на всіх етапах і використовувати сторонні бібліотеки. API включає підтримку графіки, фізики, сенсорів, управлінських алгоритмів та інше.

Серед недоліків CoppeliaSim можна зазначити наступні особливості програмного забезпечення:

– продуктивність: при роботі з великими та складними системами, особливо при використанні в реальному часі симуляцій з великою кількістю об'єктів, система може дещо сповільнювати роботу, що ускладнює тестування складних рішень автоматизації;

– можливі проблеми з сумісністю: в окремих випадках можуть виникати проблеми при інтеграції з іншими програмними платформами або версіями бібліотек, що потребує додаткових налаштувань для коректної роботи.

Таким чином, обране програмне забезпечення CoppeliaSim забезпечує високу гнучкість у налаштуванні параметрів симуляції та підтримує інтеграцію з різноманітними програмними платформами й алгоритмами, що робить обране програмне забезпечення зручним засобом для реалізації інженерних задач у галузі автоматизації та робототехніки, а також при проєктуванні АСУ ТП.

Як було зазначено раніше, в процесі виконання завдання кваліфікаційної роботи було застосовано обидва можливі підходи програмування в середовищі CoppeliaSim. Зокрема, використано мову програмування Python, що продемонстровано під час реалізації дволанкового маніпулятора, який є першою частиною практичного втілення завдання. Крім того, застосовано мову програмування Lua, що є стандартною мовою середовища CoppeliaSim, для виконання додаткових функцій та налаштувань. Взагалі, Lua – це мова програмування, яку можна віднести до категорії скриптових мов. Код, написаний на Lua, складається з інструкцій або команд, які програма виконує по черзі, одна за

іншою. Lua підтримує кілька типів програмування, включаючи об'єктно-орієнтоване програмування, процедурне програмування, програмування на основі даних та функціональне програмування [25].

2.2 Розрахунок ПІД-регулятора роботизованої системи

У сучасних системах автоматичного управління (САУ), зокрема в задачах позиціонування маніпуляторів, важливим аспектом є точне та стабільне керування рухом. Загалом розрізняють чотири основні закони керування: пропорційний, інтегральний, пропорційно-інтегральний, пропорційно-інтегро-диференціальний.

На рисунку 2.1 наведено класичну схему САУ, яка є типовою схемою управління з одиничним негативним зворотним зв'язком.



Рисунок 2.1 – Типова схема САУ

Отже, для забезпечення необхідної точності та швидкості виконання завдань використовуються різноманітні методи управління, серед яких одним із найбільш ефективних є ПІД-регулятор, який приймає дані від різних елементів і генерує відповідний сигнал керування.

Цей регулятор базується на трьох складниках: пропорційній, інтегральній та диференціальній, що дозволяє оптимізувати процес регулювання і досягти бажаного результату з мінімальними коливаннями та помилками. У випадку пропорційно-інтегро-диференціального управління (ПІД) пристрій управління формує сигнал, що дорівнює сумі трьох складників: пропорційної похибки, інтегралу від похибки та похідної похибки.

Отже, для початку за формулою (2.1) визначимо передавальну функцію пристрою управління (ПД-регулятора):

$$W_{\text{ПД}}(s) = k_1 + \frac{k_2}{s} + k_3 s, \quad (2.1)$$

де k_1, k_2, k_3 – коефіцієнти пропорційної інтегральної та диференціальної складників відповідно.

Для того, щоб надалі робити налаштування ПД-регулятора, необхідно отримати передавальну функцію двигуна, яка пов'язує вхідний сигнал (керівну напругу) із вихідним сигналом (кутом повороту). Отже, передавальну функцію двигуна визначимо як систему другого порядку, що описує динаміку електромеханічного об'єкта, за формулою (2.2)

$$G(s) = \frac{K}{s(Js+B)} = \frac{K}{Js^2+Bs}, \quad (2.2)$$

де K – коефіцієнт підсилення двигуна;

J – момент інерції;

B – коефіцієнт в'язкого тертя;

s – оператор Лапласа.

Для моделювання та розрахунків було обрано двигун постійного струму з параметрами, характерними для промислових застосувань, а саме:

– момент інерції $J = 0,01 \text{ кг} \times \text{м}^2$;

– коефіцієнт в'язкого тертя: $B = 0,1 \text{ Н} \times \text{м} \times \text{с/рад}$;

– коефіцієнт підсилення: $K = 10,0$.

Ці значення взяті з дослідження, де моделювали двигун для системи керування швидкістю [21].

Отже, враховуючи обрані параметри, передавальна функція двигуна виглядатиме наступним чином:

$$G(s) = \frac{K}{Js^2 + Bs} = \frac{10,0}{0,01s^2 + 0,1s}$$

Наступним кроком проведемо налаштування ПІД-регулятора методом Циглера-Ніколса. За правилами Циглера-Ніколса було експериментально визначено критичні наступні значення, за яких система починає коливатися:

- критичний коефіцієнт підсилення: $K_u = 2,0$;
- критичний період коливань: $T_u = 0,5$ с.

Отже, параметри ПІД-регулятора обчислюються наступним чином:

$$\begin{aligned} K_p &= 0,6 \times K_u = 1,2, \\ T_i &= 0,5 \times T_u = 0,25, \\ T_d &= 0,125 \times T_u = 0,0625, \\ K_i &= \frac{K_p}{T_i} = 4,8, \\ K_d &= K_p \times T_d = 0,075, \end{aligned}$$

де K_p – пропорційний коефіцієнт;

T_i – інтегральний час;

T_d – диференціальний час.

Таким чином, було отримано наступні значення параметрів ПІД-регулятора: $K_p = 1,5$, $T_i = 0,2$, $T_d = 0,05$.

Отже, передавальна функція для розімкнутої системи буде мати наступний вигляд (2.3):

$$\begin{aligned} W_{\text{роз}}(s) &= \frac{10(1,2s + 4,8 + 0,075s^2)}{s(0,01s^2 + 0,1s)} = \\ &= \frac{0,75s^2 + 12s + 48}{0,01s^3 + 0,1s^2}. \end{aligned} \quad (2.3)$$

Передавальна функція для замкнутої системи має такий вигляд (2.4):

$$\begin{aligned} W_{\text{зам}}(s) &= \frac{0,75s^2 + 12s + 48}{0,01s^3 + 0,1s^2 + 0,75s^2 + 12s + 48} = \\ &= \frac{0,75s^2 + 12s + 48}{0,01s^3 + 0,85s^2 + 12s + 48}. \end{aligned} \quad (2.4)$$

Отже, за результатами розрахунків (2.3) та (2.4) маємо значення передавальних функцій розімкнутої та замкнутої систем:

$$\begin{aligned} W_{\text{роз}}(s) &= \frac{0,75s^2 + 12s + 48}{0,01s^3 + 0,1s^2}, \\ W_{\text{зам}}(s) &= \frac{0,75s^2 + 12s + 48}{0,01s^3 + 0,85s^2 + 12s + 48}. \end{aligned}$$

Для подальшого аналізу стабільності системи та оцінки її поведінки було побудовано характеристики амплітудно-частотної характеристики (ЛАЧХ) розімкнутої системи. Цей етап є важливим для визначення частотних характеристик системи та її здатності підтримувати стабільну роботу при різних робочих умовах.

Крім того, на основі ЛАЧХ можна надалі визначити межі стійкості та можливі коливання системи, що дозволить оптимізувати параметри ПД-регулятора для досягнення бажаних результатів.

За допомогою ЛАЧХ можна визначити, на яких частотах система може виявляти нестабільність або коливання, а також проаналізувати, чи не виникають резонансні явища, які можуть погіршити якість роботи системи.

На рисунку 2.2 зображено результат побудови ЛАЧХ розімкнутої системи, де видно амплітуду сигналу на різних частотах.

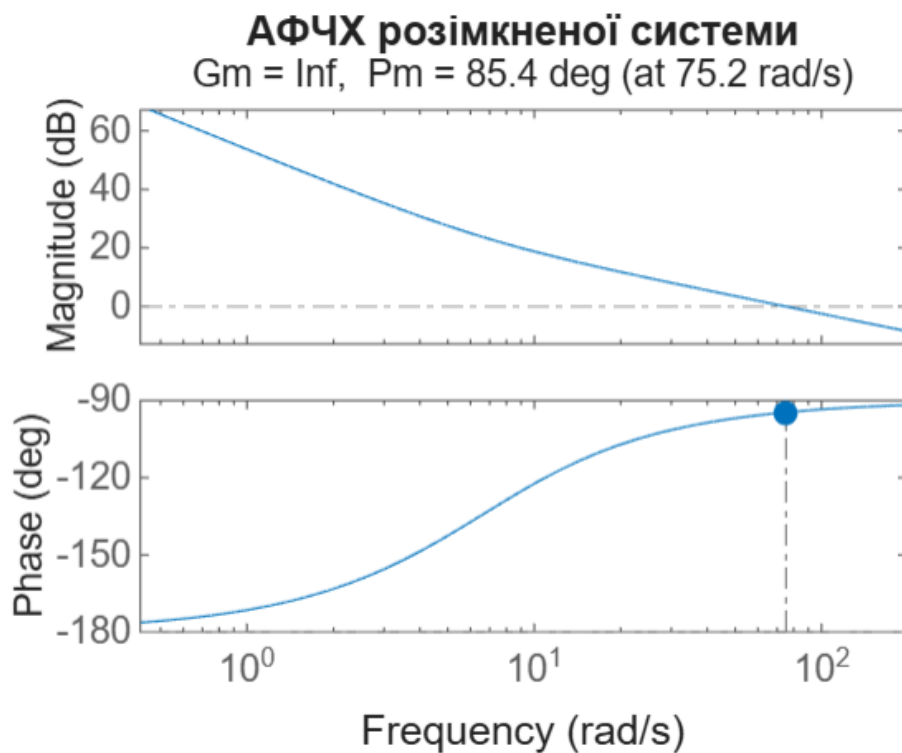


Рисунок 2.2 – АФЧХ розімкнутої системи

Для перевірки системи на стійкість було використано критерій Найквіста. Згідно із рисунком 2.3, отриманим у програмному засобі MATLAB, видно, що система є стійкою, оскільки вона не перетинає точку $(-1; j0)$.

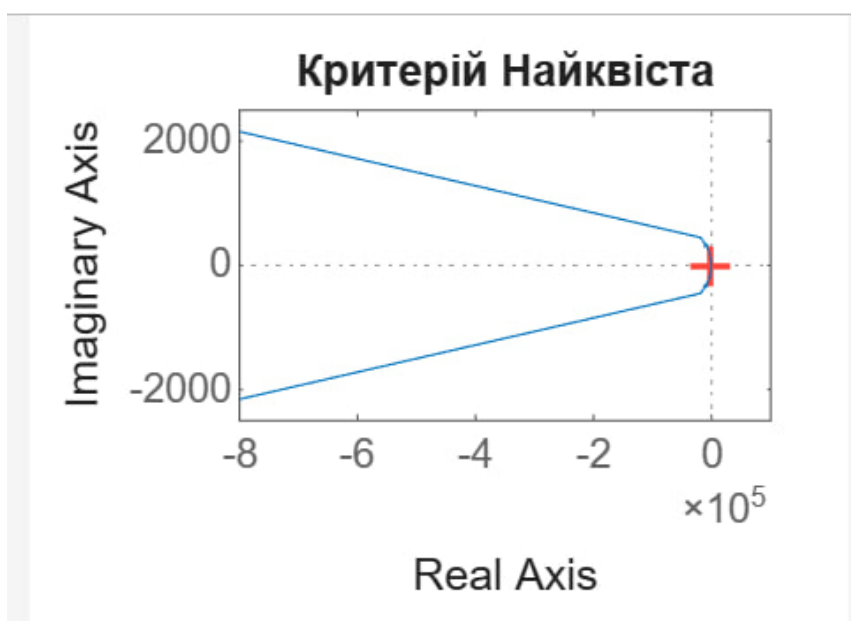


Рисунок 2.3 – Діаграма Найквіста

Таким чином, у програмному засобі MATLAB було отримано ЛАЧХ системи, а також перевірено систему на стійкість із використанням наступних функцій, а саме:

- для побудови ЛАЧХ було використано функцію (bode);
- для побудови діаграми Найквіста було використано функцію (nyquist).

За результатами розрахунків, які наведено на рисунках в ході роботи, видно, що отримана система є стійкою.

2.3 Задача прямої та зворотної кінематики

Для побудови дволанкового маніпулятора було використано принципи зворотної задачі кінематики. Далі розглянемо основні положення, які було враховано при практичній реалізації моделі.

Кінематика – це наука, що вивчає рух без врахування його причин, таких як сили та моменти сил. Інверсна (або зворотна) кінематика полягає у застосуванні кінематичних рівнянь для визначення руху робота, щоб досягти бажаної позиції.

Наприклад, для автоматизованого підбирання предметів зі стелажа роботизована рука, що використовується на виробничій лінії, повинна точно переміщатися з початкової точки в задану позицію між стелажми та виробничими машинами.

Робочий кінець роботизованої руки, який виконує захоплення, називається кінцевим ефектором (End Effector). Конфігурація робота – це набір положень кінцівок роботу, що знаходяться в межах допустимих значень для даної моделі робота і не порушують жодних обмежень, які накладаються на його рух [26].

На рисунку 2.4 [26, 27] наведено конфігурацію положень суглобів робота за допомогою прямої та інверсної кінематики.

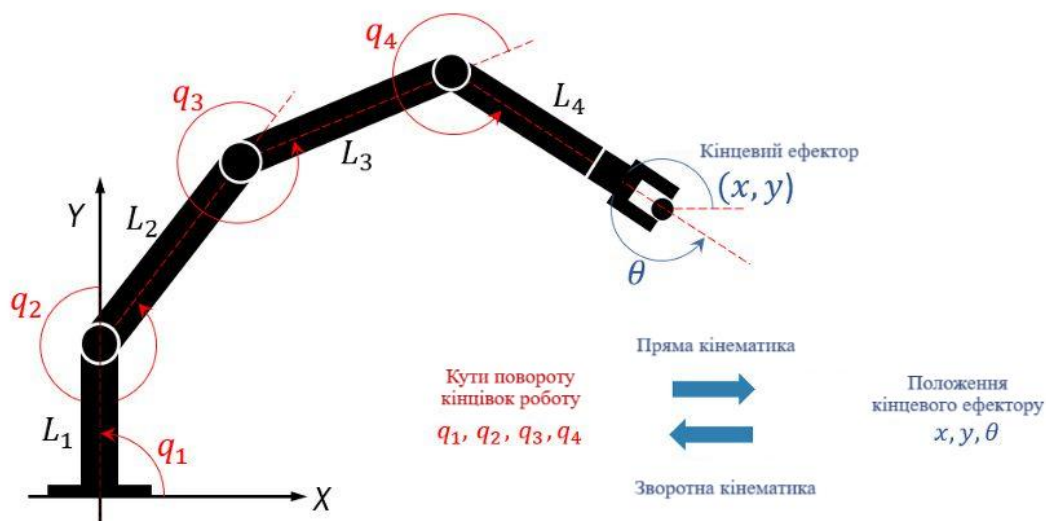


Рисунок 2.4 – Конфігурацію положень суглобів робота на прикладі прямої та інверсної кінематики

Згідно з рисунком 2.4 можна зробити висновок, що на відміну від прямої кінематики, яка обчислює координати робочого простору робота, виходячи з його конфігурацій як вхідних даних, інверсна кінематика (ІК) є по суті зворотною операцією: обчислення конфігурацій для досягнення бажаної координати в робочому просторі. Ця операція є важливою для багатьох задач робототехніки, таких як переміщення по заданому шляху, маніпулювання об'єктами та спостереження за процесом з бажаної позиції.

Для подальших розрахунків введемо таблицю вхідних значень.

Таблиця 2.1 – Вхідні параметри дволанкового маніпулятора

Параметр	Умовне позначення	Значення
1	2	3
Довжина ланки 1	l_1	1,0 м
Довжина ланки 2	l_2	0,8 м
Центр кола	(x_0, y_0)	(1,0 м, 0,5 м)
Радіус кола	r	0,2 м
Початкові кути	$\theta_1 = 0, \theta_2 = 0$	0

Для демонстрації принципу прямої кінематики розглянемо приклад з конкретними значеннями кутів суглобів, де необхідно знайти положення кінцевого ефектора (x, y) .

Для початку, переведемо кути в радіани для подальших розрахунків:

$$\theta_1 = \frac{\pi}{6} \approx 0,524 \text{ рад,}$$

$$\theta_2 = \frac{\pi}{4} \approx 0,785 \text{ рад.}$$

Отже, формули прямої кінематики матимуть вигляд (2.5) та (2.6):

$$x = l_1 \times \cos(\theta_1) + l_2 \times \cos(\theta_1 + \theta_2), \quad (2.5)$$

$$y = l_1 \times \sin(\theta_1) + l_2 \times \sin(\theta_1 + \theta_2). \quad (2.6)$$

Підставимо вхідні параметри системи до формул (2.5) та (2.6) для отримання положення кінцевого ефектора для заданих кутів:

$$x = 1,0 \times \cos(0,524) + 0,8 \times \cos(0,524 + 0,785) = 1,075,$$

$$y = 1,0 \times \sin(0,524) + 0,8 \times \sin(0,524 + 0,785) = 1,273,$$

Таким чином, положення ефектора для заданих кутів:

$$(x, y) \approx (1,075 \text{ м, } 1,273 \text{ м}).$$

Таким чином, використання прямої кінематики дозволяє однозначно визначити положення кінцевого ефектора на основі заданих значень кутів суглобів. Це дозволяє перевіряти правильність моделі та є основою для побудови симуляційних траєкторій у середовищі CoppeliaSim.

Проте, на відміну від прямої кінематики [28], інверсну кінематику не можна вирішити у вигляді замкнутої формули. У випадку ІК може не бути жодного

рішення, може бути кілька рішень або навіть нескінченна кількість рішень. Тому, важливо зрозуміти тонкощі ІК, щоб ефективно застосовувати її на практиці [29]. Отже, на відміну від прямої кінематики, роботи з кількома обертальними суглобами, як правило, мають кілька рішень для ІК, і було запропоновано різні методи залежно від мети. Загалом, їх класифікують на два основні типи: перший, що дає аналітичне рішення, – аналітичний метод, та другий, що використовує чисельні розрахунки, – чисельний метод [26].

Чисельні методи використовуються для наближення конфігурації робота, що відповідає заданим цілям і обмеженням. Чисельні рішення є більш загальними, але вимагають кількох кроків для збіжності до розв'язку через нелінійність системи. Вони дозволяються враховувати кінематичні обмеження робота та зовнішні умови, наприклад, для орієнтації камери на певну точку. Проте, чисельне рішення є більш універсальним і підходить для роботів з надлишковими ступенями свободи, але потребують більшого часу на розв'язок у порівнянні з аналітичним рішенням. Наприклад, таке рішення можна використовувати для планування траєкторії руху робота з кількома кінематичними обмеженнями, використовуючи чисельні методи для обчислення ІК.

Аналітичне рішення використовує математичні формули для обчислення кутів суглобів на основі положення кінцевого ефектора. ІК шукає рішення у вигляді функції довжини ланок і початкової позиції, що підходить для роботів з низьким числом ступенів свободи. Аналітичне рішення є більш ефективним для простих задач, де можна отримати замкнуті вирази для кутів суглобів, але вони не масштабуються для складніших або надлишкових систем. Наприклад, таке рішення може використовуватися для виведення і застосування ІК для дволанкової роботизованої системи з обчисленням кутів суглобів для заданої пози кінцевого ефектора.

В ході побудови робота-маніпулятора проблема полягала в тому, що функція є нелінійною функцією кутів, а отже знайти аналітичне рішення було неможливо, тому було використано чисельні методи для обчислення.

Розглянемо функцію f , яка складається з кількох функцій $(f_1, f_2, f_3, \dots, f_n)$, які є функціями Q , де Q – незалежна змінна, що складається з $(x_1, x_2, x_3, \dots, x_n)$.

Для прямої кінематики дволанкового маніпулятора положення у світовій системі координат, визначається за кутами суглобів, використовуючи формули (2.7) та (2.8):

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2), \quad (2.7)$$

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2), \quad (2.8)$$

де l_1 – довжина першої ланки;

l_2 – довжина другої ланки;

θ_1, θ_2 – кути суглобів маніпулятора;

(x, y) – положення кінцевого ефектора.

Для демонстрації керування задається траєкторія по колу (2.9) та (2.10):

$$x_{ref}(t) = x_c + r \cos(2\pi/T), \quad (2.9)$$

$$y_{ref}(t) = y_c + r \sin(2\pi/T), \quad (2.10)$$

де (x_c, y_c) – центр кола (визначається на основі початкового положення ефектора);

r – радіус кола;

T – період обертання по колу.

Зворотна кінематика, своєю чергою, ставить завдання: знаючи цільові координати (x_{ref}, y_{ref}) кінцевого ефектора, знайти потрібні кути θ_1, θ_2 . Для цього використаємо ітеративний метод з Якобіаном.

Першим кроком, обчислимо поточну позицію кінцевого ефектора (x, y) та знайдемо вектор помилки за формулою (2.11):

$$dr = \begin{bmatrix} x_{ref} - x \\ y_{ref} - y \end{bmatrix}, \quad (2.11)$$

Далі, знайдемо положення точки Q , тобто положення кінцевого ефектора. За умови для знаходження позиції кінцевого ефектора, використовуємо наступне математичне твердження (2.12):

$$r_Q = f(q), \quad (2.12)$$

де $f(q)$ – деяка функція Q .

Таким чином, за формулами (2.11) та (2.12) знайдемо значення положення кінцевого ефектора дволанкового маніпулятора:

$$r_Q = \begin{bmatrix} l \sin(\theta_1 + \theta_2) + l \sin(\theta_1) \\ -l \cos(\theta_1 + \theta_2) - l \cos(\theta_1) \end{bmatrix}^T, \\ q = [\theta_1, \theta_2].$$

Далі, знайдемо матрицю Якобі, тобто похідні кінцевих координат за кутами, яка виглядатиме наступним чином (2.13):

$$J = \begin{bmatrix} l(\cos\theta_1 + \cos(\theta_1 + \theta_2)) & l \cos(\theta_1 + \theta_2) \\ l(\sin\theta_1 + \sin(\theta_1 + \theta_2)) & l \sin(\theta_1 + \theta_2) \end{bmatrix}, \quad (2.13)$$

Отже, за формулою (2.7) було знайдено необхідну матрицю Якобі, яка описує залежність лінійної швидкості ефектора від кутових швидкостей суглобів. Для наближення до цільового положення обчислюється різниця між бажаними та поточними координатами ефектора за формулою (2.14):

$$\Delta x = \begin{bmatrix} x_{ref} - x \\ y_{ref} - y \end{bmatrix}, \quad (2.14)$$

Своєю чергою вектор змін кутів суглобів отримується через зворотну матрицю Якобі (2.15):

$$\Delta\theta = J^{-1}\Delta x, \quad (2.15)$$

Таким чином, оновленні значення кутів суглобів будуть дорівнювати рівнянню (2.16):

$$\theta_1 \leftarrow \theta_1 + \Delta\theta_1, \theta_2 \leftarrow \theta_2 + \Delta\theta_2, \quad (2.16)$$

На рисунку 2.5 наведено схематичне зображення руху кінцевого ефектора відповідно до проведених розрахунків розв'язання задачі зворотної (інверсної) кінематики.

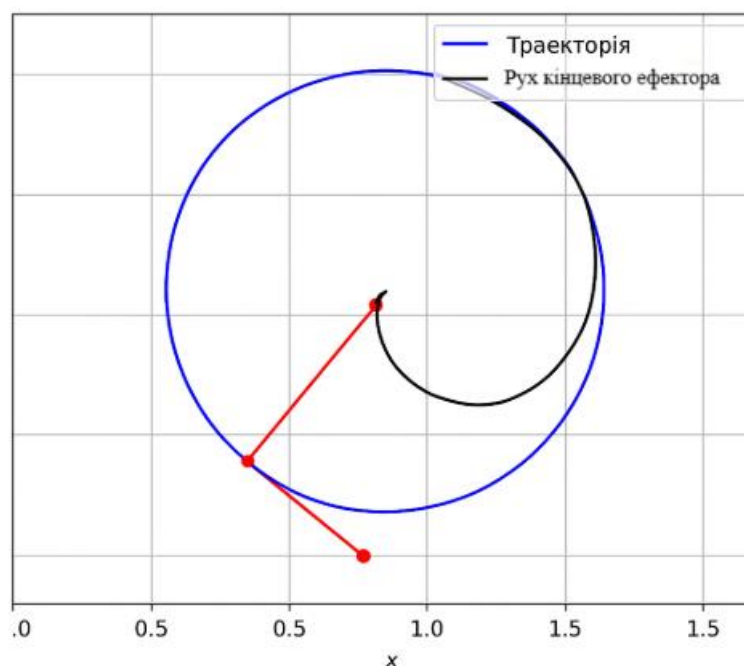


Рисунок 2.5 – Траекторія та рух кінцевого ефектора

Для перевірки працездатності чисельного методу зворотної кінематики на практиці було задано початкові (вхідні) умови, зазначені в таблиці 2.1.

Мета розрахунків – знайти значення кутів θ_1, θ_2 , які забезпечать наближення кінцевого ефектора до точки (x_t, y_t) за допомогою методу зворотної матриці Якобі.

Першим кроком зробимо початкове наближення. Припустимо, що початкові кути мають наступні значення: $\theta_1^{(0)} = 0,0$ рад, $\theta_2^{(0)} = 0,0$ рад. Отже початкове

положення кінцевого ефектора розрахуємо за формулами (2.7) та (2.8). Маємо наступні рівняння:

$$x^{(0)} = l_1 \times \cos(0) + l_2 \times \cos(0) = 1,0 + 0,8 = 1,8,$$

$$y^{(0)} = l_1 \times \sin(0) + l_2 \times \sin(0) = 0,0 + 0,0 = 0,0.$$

Далі, проведемо розрахунок помилки (2.17, 2.18):

$$e_x = x_t - x^{(0)}, \quad (2.17)$$

$$e_y = y_t - y^{(0)}, \quad (2.18)$$

Таким чином, отримуємо наступні значення:

$$e_x = 1,2 - 1,8 = -0,6,$$

$$e_y = 0,5 - 0,0 = 0,5.$$

Після отримання необхідних даних, побудуємо матрицю Якобі, спираючись на формулу отриману у розрахунках (2.13). Таким чином, маємо рівняння (2.19):

$$J = \begin{bmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos(\theta_1) - l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}, \quad (2.19)$$

При $\theta_1 = 0, \theta_2 = 0$, маємо наступний вигляд матриці Якобі для зворотної кінематики системи:

$$J = \begin{bmatrix} 0 & 0 \\ 1,8 & 0,8 \end{bmatrix},$$

Оскільки верхній рядок нульовий – використовуємо регуляризацію або псевдообернену матрицю. Це зумовлено геометрією положення ефектора на осі x ,

коли ланки витягнуті горизонтально. Така конфігурація є сингулярною, тому для розв'язку необхідно використати псевдообернену матрицю Якобі, використовуючи команду `pinv()` у MATLAB. Порахуємо оновлені значення кутів з розрахунку ІК системи:

$$\theta_1^{(1)} = \theta_1^{(0)} + \Delta\theta_1 = 0,0 - 0,28 = -0,28 \text{ рад,}$$

$$\theta_2^{(1)} = \theta_2^{(0)} + \Delta\theta_2 = 0,0 - 0,72 = 0,72 \text{ рад.}$$

На першій ітерації чисельний метод зворотної кінематики вже дозволяє значно зменшити відхилення кінцевого ефектора від цільової точки. Повторюючи ітерації з новими значеннями кутів, можна досягти точного позиціонування з бажаною точністю. Після досягнення точного результату, описаний крок повторюється декілька разів, поки значення вектора помилки Δx не стане менше за задану точність ε . Як правило, для типових задач дволанкової кінематики достатньо 3-5 ітерацій при правильному підборі початкових значень.

В таблиці 2.2 наведено результати ітераційного наближення значень кутів суглобів.

Таблиця 2.2 – Результати ітераційного наближення

Ітерація	θ_1 , рад	θ_2 , рад	x , м	y , м	Помилка Δx
1	2	3	4	5	6
1	-0,28	0,72	1,31	0,41	0,13
2	-0,32	0,65	1,25	0,47	0,06
3	-0,30	0,63	1,21	0,49	0,02
4	-0,295	0,62	1,20	0,50	0,003

Як видно з таблиці 2.2, після четвертої ітерації значення помилки стає незначним, що свідчить про збіжність чисельного методу.

Таким чином, згідно із розрахунками (2.7-2.16) на кожному кроці симуляції обчислюється, наскільки кінцевий ефектор відхиляється від бажаного положення. Це відхилення переводиться у відповідні зміни кутів суглобів за допомогою зворотної матриці Якобі. Отримані зміни кутів додаються до поточних значень кутів, тим самим оновлюючи положення маніпулятора. Таким чином, такий підхід дозволяє ефектору поступово наближатися до цілі, рухаючись уздовж заданої траєкторії. Це забезпечує плавність та точність слідування.

3 РОЗРОБЛЕННЯ МОДЕЛІ РОБОТИЗОВАНОГО ВИРОБНИЧОГО ПРОЦЕСУ ІЗ ВИКОРИСТАННЯМ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ CORPELIASIM

3.1 Схема роботи модельованої системи

ТЗ кваліфікаційної роботи включає три логічно взаємопов'язані частини.

Перша частина роботи передбачає розробку дволанкового маніпулятора, в основі якого лежать принципи та алгоритми, що використовуються у другій частині завдання. Друга частина передбачає створення імітаційної моделі робота типу Pick-and-Place у програмному середовищі CoppeliaSim, де реалізуються алгоритми та методи управління, розроблені в попередньому етапі [1, С.7-10]. Обидва перші етапи реалізовано у двовимірному вигляді для більш детальної демонстрації логічної та математичної сутності моделі. У такий спосіб, обидва етапи роботи тісно пов'язані між собою, забезпечуючи комплексний підхід до дослідження та реалізації робототехнічних систем.

Третім етапом моделювання є створення тривимірної моделі робота-маніпулятора із використанням вбудованих функцій програмного середовища CoppeliaSim. Створення моделі передбачає використання запропонованого програмним забезпеченням мобільного роботу, проектування та моделювання бажаного виробничого процесу та створення програмного забезпечення для керування рухами промисловим роботом.

Основна технологічна операція, покладена в основу третього етапу моделювання, – програмування функцій робота, а саме:

- створення програмного забезпечення для ідентифікації необхідного об'єкту за кольором;
- процес захоплення роботом (операція pick) необхідного об'єкту та переміщення його в бажану позицію;
- позиціонування об'єкта (операція place) залежно від кольору в задане місце.

Розробка програмного забезпечення для реалізації завдання кваліфікаційної роботи здійснюється за допомогою двох мов програмування: Python, яка обирається для демонстрації потенційних можливостей цієї мови у відповідному програмному середовищі, та Lua, що використовується як мова програмування за замовчуванням в програмному засобі CoppeliaSim.

На рисунку 3.1 наведено схематичне зображення триланкового маніпулятора типу Pick-and-Place, модельований макет якого було реалізовано в ході виконання практичної частини завдання кваліфікаційної роботи.

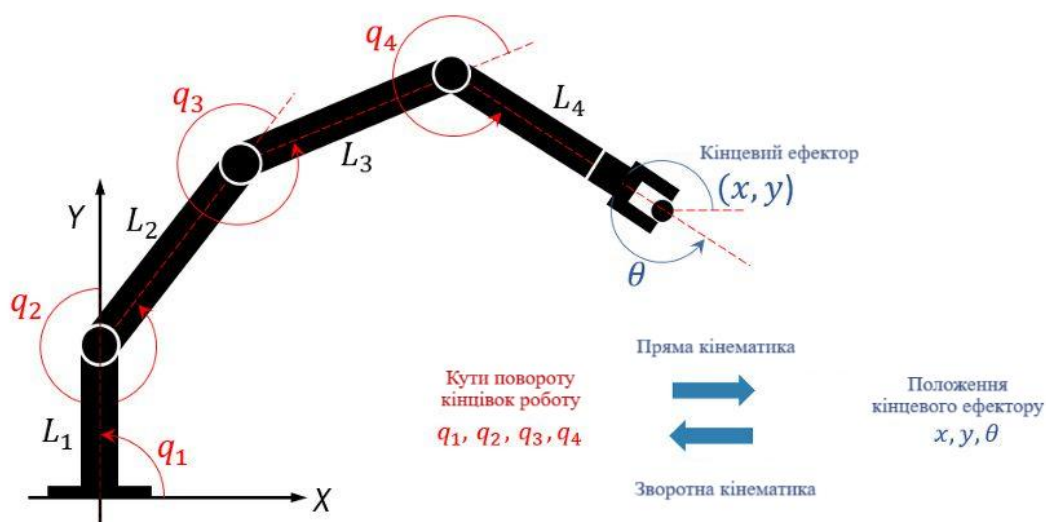


Рисунок 3.1 – Схематичне представлення маніпулятора

Надалі для кращого представлення всіх етапів роботи в ході опису створення моделі системи буде показано відповідні схематичні представлення модельованої системи.

3.2 Створення моделі системи

Для виконання поставленої задачі кваліфікаційної роботи, а саме моделювання робота-маніпулятора типу Pick-and-Place та його функцій, практичну частину завдання було запропоновано поділити на два логічно поєднаних етапи: перший етап включає побудову дволанкового маніпулятора, що буде основою

побудови промислового робота. Друга частина включає побудову безпосередньо робота та програмування необхідних функцій робота в програмному середовищі CoppeliaSim.

3.2.1 Моделювання та програмування функцій дволанкового маніпулятора

Першою частиною виконання практичної частини завдання є побудова дволанкового маніпулятора для побудови майбутнього робота-маніпулятора.

На рисунку 3.2 наведено схематичне зображення бажаної системи, яку необхідно змоделювати в програмному середовищі CoppeliaSim.

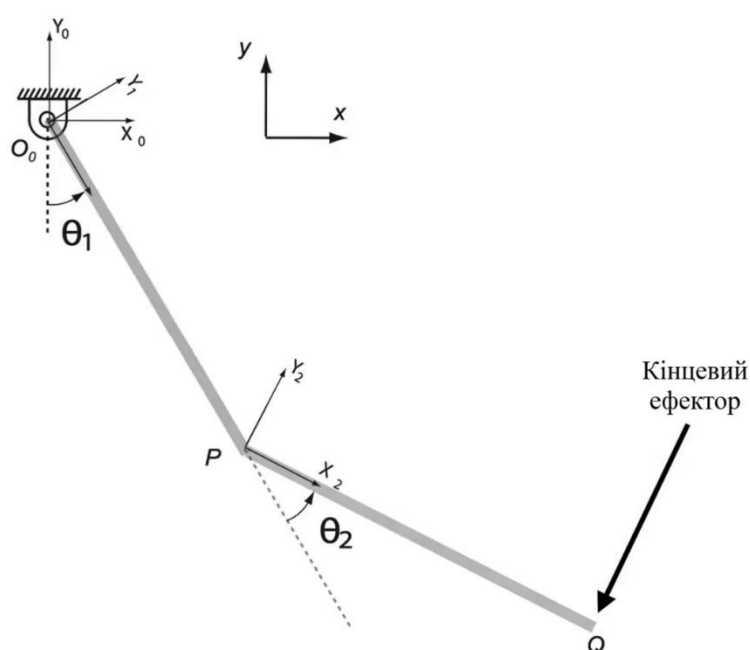


Рисунок 3.2 – Схематичне представлення дволанкового маніпулятора

Згідно з рисунком 3.2 маніпулятор складається з наступних елементів:

O_0 – фіксована, нерухома в положенні основа маніпулятора;

P – з'єднання між двома ланками (так званий, суглоб маніпулятора);

Q – кінцева точка маніпулятора (так званий, кінцевий ефектор), яка має досягати певного положення у просторі.

Крім цього, у склад системи входить також дві ланки: перше ланка з'єднує нерухому основу (O_0) із точкою (P), своєю чергою друга ланка – точку (P) з кінцевим ефектором (Q).

Кут θ_1 – є кутом повороту першої ланки відносно вертикальної осі (Y_0). Отже, це кут обертання першого суглобу. Кут θ_2 – є кутом повороту другої ланки відносно першої ланки (тобто відносно осі x_1 , прив'язаної до першої ланки).

Згідно зі схематичним зображенням, представленим на рисунку 3.2, система спирається на три системи координат, а саме:

- (x_0, y_0) – глобальна система координат. Система координат, що лишається нерухомою в ході роботи;
- (x_1, y_1) – локальна система координат, що рухається разом із першою ланкою;
- (x_2, y_2) – локальна система координат, що рухається із другою ланкою.

Таким чином, маніпулятор повинен змінювати кути повороту θ_1 та θ_2 , щоб керувати положенням кінцевого ефектора (Q) у площині, тобто у координатах (x, y) .

Для того, щоб реалізувати подану схему в програмному середовищі необхідно побудувати та змінити параметри наступних елементів, що наведено на рисунку 3.3.

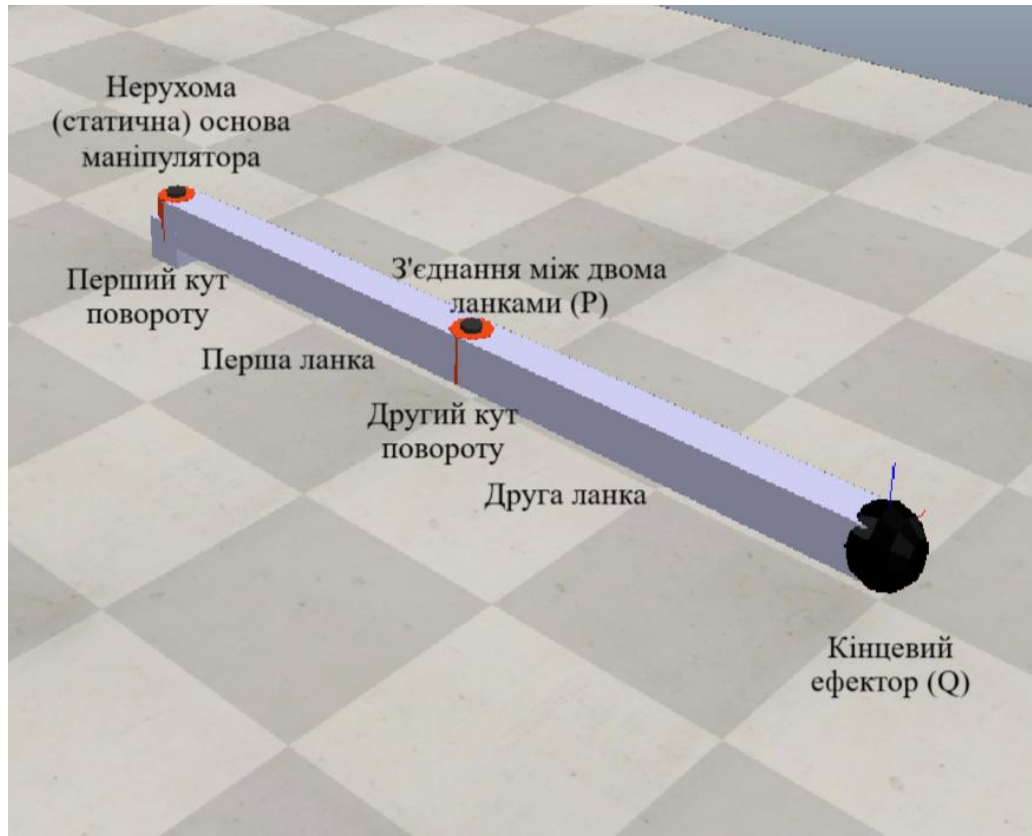


Рисунок 3.3 – Зображення компонентів системи

Таким чином, згідно з рисунком 3.3 побудова дволанкового маніпулятора в програмному середовищі CoppeliaSim містить створення статичної (нерухомої впродовж всього часу) основи, двох рухомих ланок та кінцевого ефектора.

Для того, щоб уможливити рух необхідних компонентів було використано мову програмування Python, яку попередньо було встановлено основною мовою програмування створеного проєкту. Для цього етапу також було використано команди з API програмного середовища CoppeliaSim [24].

На початковому етапі виконувалась ініціалізація об'єктів: ідентифікація двох обергальних суглобів маніпулятора, кінцевого ефектора, а також створення об'єкта для візуалізації траєкторії його руху. Це дозволяло надалі не лише керувати положенням маніпулятора, але й спостерігати за переміщенням кінцевої точки у просторі в реальному часі. Під час симуляції кутові положення обох суглобів змінювались як функції часу. Кожному суглобу було задано лінійно висхідний кут зі сталою швидкістю: перший суглоб обертався зі швидкістю, пропорційною $\pi/6$

від часу симуляції, а другий – зі швидкістю $\pi/3$. Таке завдання дозволило сформувати різноманітні траєкторії кінцевого ефектора залежно від співвідношення швидкостей обертання обох ланок.

На рисунку 3.4 зображено очікуваний результат роботи створеної системи.

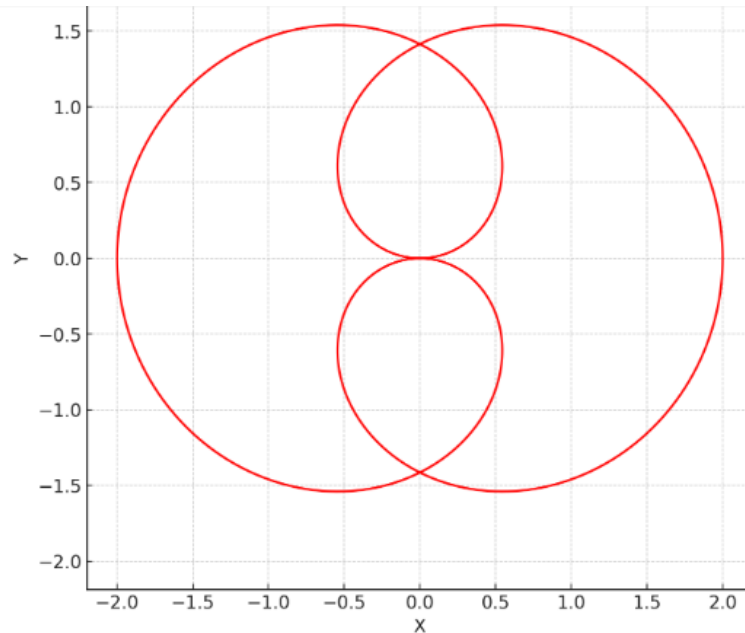


Рисунок 3.4 – Траєкторія руху кінцевого ефектора

Згідно з рисунком 3.4 результатом роботи є розв’язання задачі прямої кінематики, тобто обчислення просторового положення кінцевого ефектора за заданими кутами обертання суглобів. Вхідними даними є значення кутів обох суглобів у кожен момент часу, тоді як вихідними даними є координати кінцевої точки маніпулятора.

На рисунку 3.5 зображено результат роботи створеного дволанкового маніпулятора у програмному засобі CoppeliaSim.

Згідно з рисунком 3.5 можна зробити висновок, що створена початкова система реалізує задачу прямої кінематики руху системи, в якій кінцевий ефектор (End Effector) реалізує рух за проведеними до цього розрахунками.

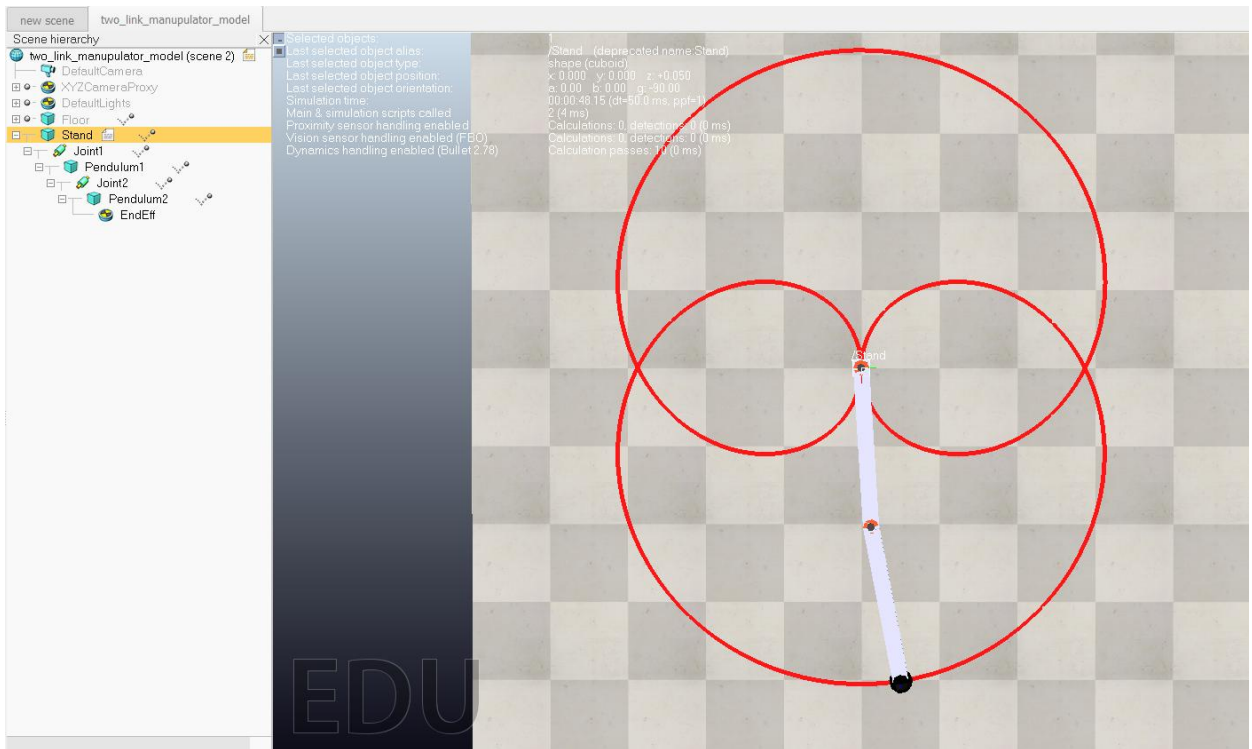


Рисунок 3.5 – Траєкторія руху кінцевого ефектора у середовищі CoppeliaSim

Таким чином, отримана траєкторія дає змогу оцінити динамічні характеристики системи, виявити можливі обмеження у досяжності та планувати подальші кроки у розробці складніших задач, таких як зворотна кінематика.

Розв’язання задачі зворотної кінематики для дволанкового маніпулятора є складною задачею, оскільки для кожної бажаної позиції кінцевого ефектора можуть існувати кілька можливих варіантів розв’язків, або взагалі їх не бути.

Зокрема для маніпуляторів з двома ланками може бути два рішення, що відповідають різним траєкторіям руху кінцевого ефектора в залежності від співвідношення кутів суглобів. Тому важливим етапом є розробка алгоритму, який би в реальному часі оцінював можливість досягнення певної точки і вибирав найбільш оптимальний шлях.

Таким чином,, наступним кроком є розробка необхідного скрипту для розв’язання задачі зворотної кінематики. На рисунку 3.6 наведено схематичне зображення бажаної траєкторії руху кінцевого ефектора маніпулятора.

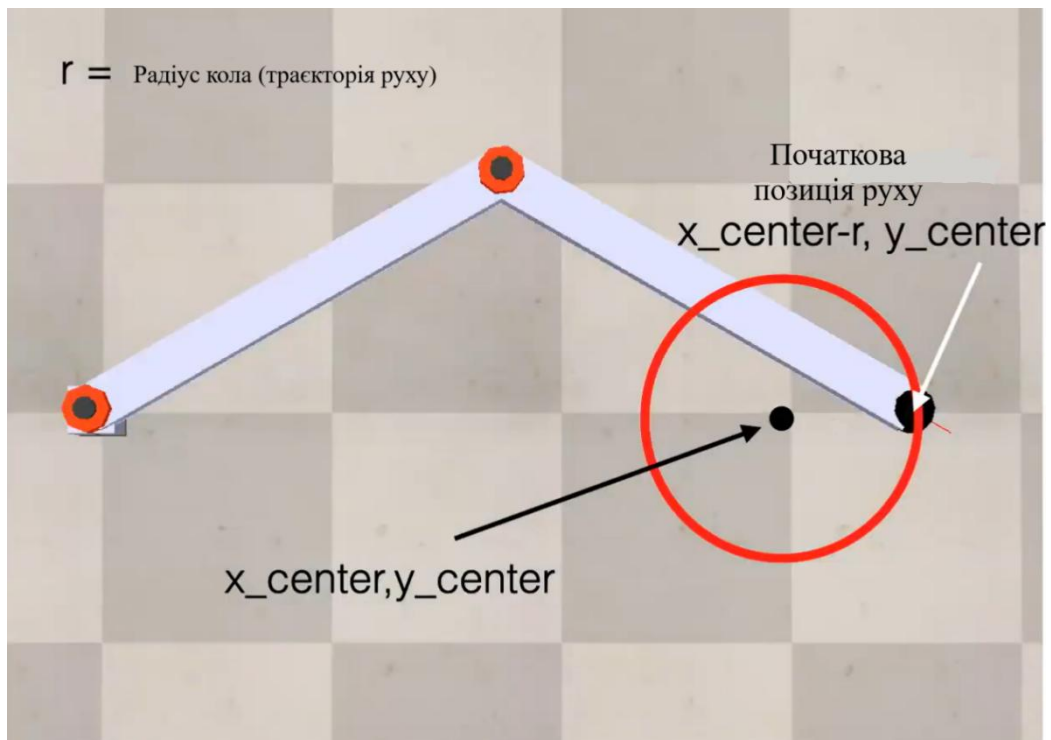


Рисунок 3.6 – Схематичне зображення бажаної траєкторії руху кінцевого ефектору

Для втілення поставленої задачі у програмному середовищі CoppeliaSim було реалізовано алгоритм керування дволанковим маніпулятором. Отже, метою даної реалізації є змусити маніпулятор слідувати заданій траєкторії.

Рух кінцевої ланки маніпулятора (ефектора) забезпечується за допомогою інверсної кінематики, принцип та основні розрахунки якої наведено в другому розділі роботи. На кожному кроці симуляції визначається поточне положення ефектора, розраховується різниця між ним та цільовим положенням, і на цій основі визначається, як саме потрібно скоригувати кути обертання ланок, щоб наблизити кінцевий ефектор до цілі.

Для цього в ході програмної реалізації було втілено наступні кроки:

- визначено початкове положення ефектора, від якого обчислюється центр кола;
- задано функцію, що залежно від часу формує координати точки на колі, яка є поточним цільовим положенням; центр кола визначається відносно початкової позиції кінцевого ефектора;

- в основному циклі керування розраховується, як змінити кути з'єднань, щоб ефектор рухався у напрямку до цільової точки;
- увімкнено візуалізацію траєкторії, що дозволяє наочно спостерігати шлях, яким рухався маніпулятор.

На рисунку 3.7 наведено результат, отриманий в ході запуску написаного коду програми.

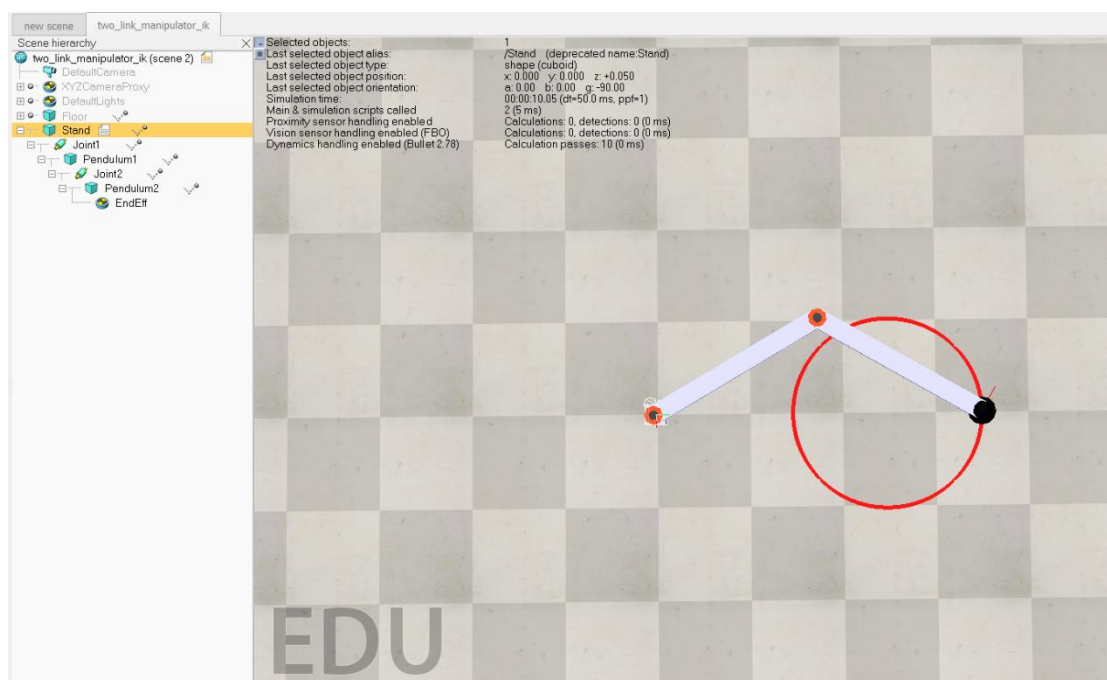


Рисунок 3.7 – Траєкторія руху ефектора у середовищі CoppeliaSim

Отже, у результаті виконання симуляції дволанковий маніпулятор слідує за коловою траєкторією із заданим радіусом $r = 0,5$ навколо свого початкового положення. Рух забезпечується шляхом обчислення зворотної кінематики в кожен момент часу, що дозволяє точно регулювати положення кінцевого ефектора.

У середовищі CoppeliaSim траєкторія візуалізується у вигляді червоної лінії, яка підтверджує правильність слідування за заданим шляхом, що наведено на рисунку 3.7.

Побудова дволанкового маніпулятора та наведення математичного обґрунтування розв'язання задачі зворотної кінематики є основою для побудови бажаної роботизованої системи, яку буде описано далі в ході розділу.

3.2.2 Моделювання робота-маніпулятора та його функцій

Наступним кроком є побудова основної частини практичної роботи – моделювання робота-маніпулятора типу Pick-and-Place у програмному середовищі CoppeliaSim.

Для пояснення роботи та представлення роботи з різних боків, побудову робота-маніпулятора запропоновано розділити на дві логічні частини:

– перша частина стосується побудови маніпулятора типу Pick-and-Place у 2D-виді для пояснення логіки роботи запропонованої роботи в умовах заданого механізму з подоланням перешкод. Такий тип моделювання обрано для більш детального пояснення логіко-математичної частини змодельованої системи;

– другою частиною роботи є представлення робота у 3D-виді із завданням сортування кольорів переміщуваних та розміщуваних об'єктів. Даний вигляд програми надасть краще візуальне розуміння роботи програми, а також дозволить проаналізувати ефективність обраного типу маніпулятора для подальшого використання на виробничих лініях.

В основу роботи, що побудовано у 2D виді у програмному засобі покладено триланковий маніпулятор, основні правила та принципи побудови якого наведено раніше в ході виконання першої частини практичної роботи.

За умовою завдання поданий маніпулятор виконує дії за заданим сюжетом виконання завдань, а саме позиціонування кінцевого ефектора до необхідної позиції, захоплення об'єкта, який необхідно перемістити в цільову позицію, переміщення об'єкта з подоланням перешкоди у вигляді стороннього об'єкту, який не є частиною виробничої лінії.

Таким чином, першим кроком роботи є побудова триланкового маніпулятора, який показано на рисунку 3.8. Він складається з трьох поєднаних між собою ланок та кінцевого ефектора, який буде використано для захоплення та переміщення об'єктів. Рух кінцевого ефектора реалізується на основі проведених у другому розділі розрахунків задачі зворотної (інверсної) кінематики.

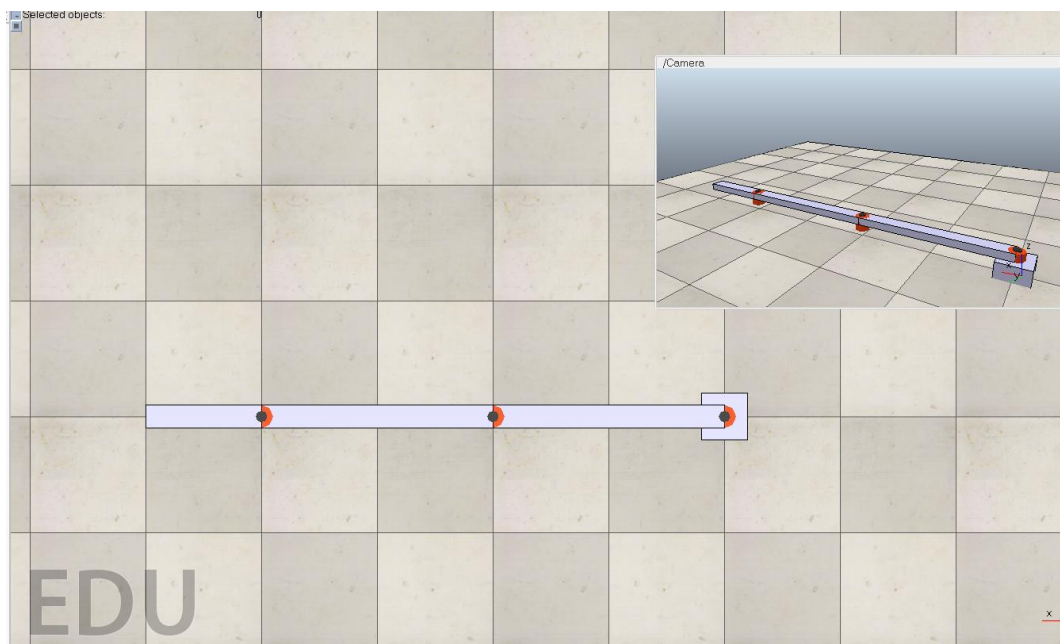


Рисунок 3.8 – Триланковий маніпулятор

Далі, за допомогою вікна налаштувань (Scene Object Properties) змінюємо налаштування окремих компонентів.

Кожна з трьох ланок маніпулятора поєднана між собою за допомогою обертових з'єднань (revolute joints), отже у вікні властивостей кожного з'єднання було налаштовано початковий кут повороту, діапазон обертання та цільове значення. Наступний елемент, основа маніпулятора, являє собою нерухому частину, до якої прикріплено першу ланку, для правильної взаємодії з фізичним середовищем було вимкнено динаміку, що означає, що впродовж всієї роботи ця частина буде залишатися статичною. Для кінцевого ефектора було встановлено детектори зіткнення (Collision Detection), щоб ефектор міг відчувати контакт з об'єктом, а також додано сенсор типу proximity sensor, який дозволяє визначити наявність об'єкта в зоні досяжності ефектора.

Отже, в ході налаштування компонентів було змінено наступні налаштування: кут між ланками, динамічні характеристики основи маніпулятора, а також налаштування кінцевого ефектора. На рисунку 3.9 наведено зображення зовнішнього вигляду триланкового маніпулятора після проведення необхідних початкових налаштувань.

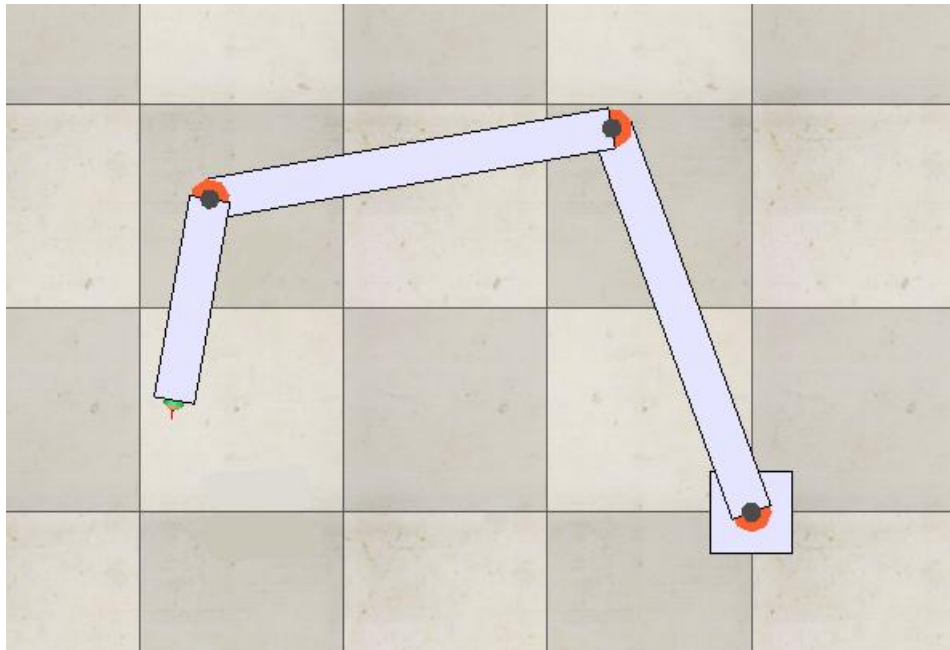


Рисунок 3.9 – Налаштування елементів системи

За умови роботи призначення робота є автоматичне захоплення об'єкта з початкової позиції, його транспортування через проміжну точку (обхід перешкоди або підняття над об'єктом) та переміщення у цільову позицію, що наведено на рисунку 3.10.

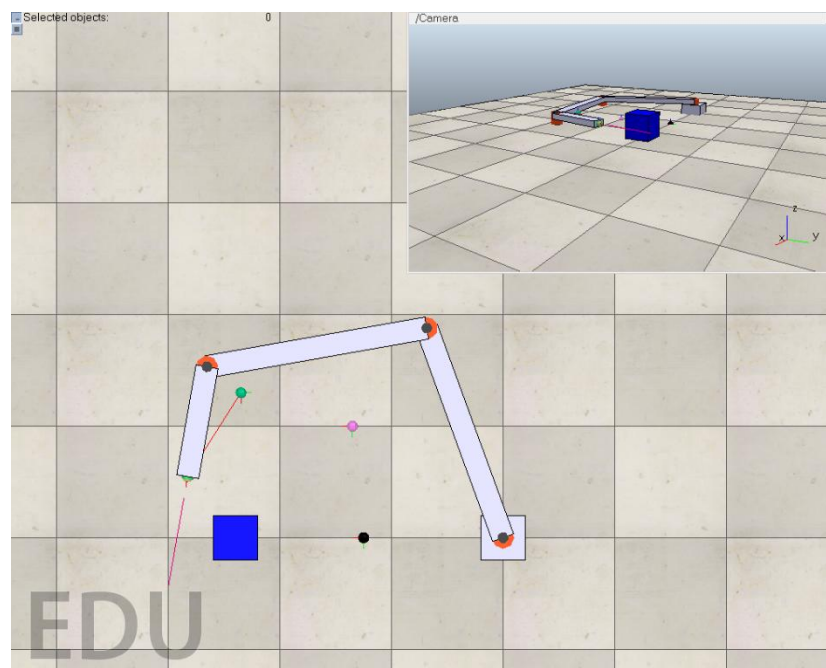


Рисунок 3.10 – Налаштування елементів системи

Отже, на рисунку 3.10 показано побудовані додаткові елементи для проходження роботом заданого шляху та виконання поставленого завдання згідно з умовою технічного завдання роботи.

Таким чином, основними елементами системи для виконання поставленої задачі є:

- триланковий маніпулятор, який складається з трьох ланок, з'єднані обертовими з'єднаннями, рухи яких здійснюються шляхом зміни позиції референсної точки, яка задає положення кінцевого ефектора;

- кінцевий ефектор, який відповідає за захоплення та транспортування об'єкта; його логічним доповненням є Connector, який фізично з'єднується з об'єктом при захопленні;

- proximity sensor, який використовується для виявлення об'єкта перед захопленням; якщо об'єкт потрапляє в зону дії сенсора, відбувається операція захоплення;

- об'єкти сцени: box – об'єкт, який потрібно перемістити, goal – цільова позиція для об'єкта та via – проміжна позиція, що використовується для уникнення зіткнень з іншими об'єктами.

Наступним кроком є написання необхідного скрипту мовою Lua для виконання поставленої задачі. По-перше, для цього було проведено ініціалізацію системи, яка включає зчитування координат ключових об'єктів сцени: об'єкта для захоплення (об'єкт «Box»), кінцевої точки розміщення (об'єкт «Goal»), проміжної точки (об'єкт «Via»), референсної точки позиціонування (об'єкт «Reference»), ефектора (об'єкт «End Effector»), з'єднувача (об'єкт «Connector Proximity»). Крім цього, далі було створено необхідні елементи скрипту для:

- позиціонування ефектора до об'єкта, узгоджене із розв'язаною раніше задачею інверсної кінематики;

- захоплення об'єкта за допомогою сенсора та з'єднання;

- переміщення захопленого об'єкта до проміжної точки, щоб уникнути зіткнення з перешкодою; переміщення захопленого об'єкта до цільової позиції та відпускання позиції;

– повернення маніпулятора через проміжну позицію до вихідної точки.

На рисунку 3.11 наведено зображення проміжного результату (в моменті переміщення маніпулятором об'єкта) виконання роботи без внесення додаткових налаштувань системи.

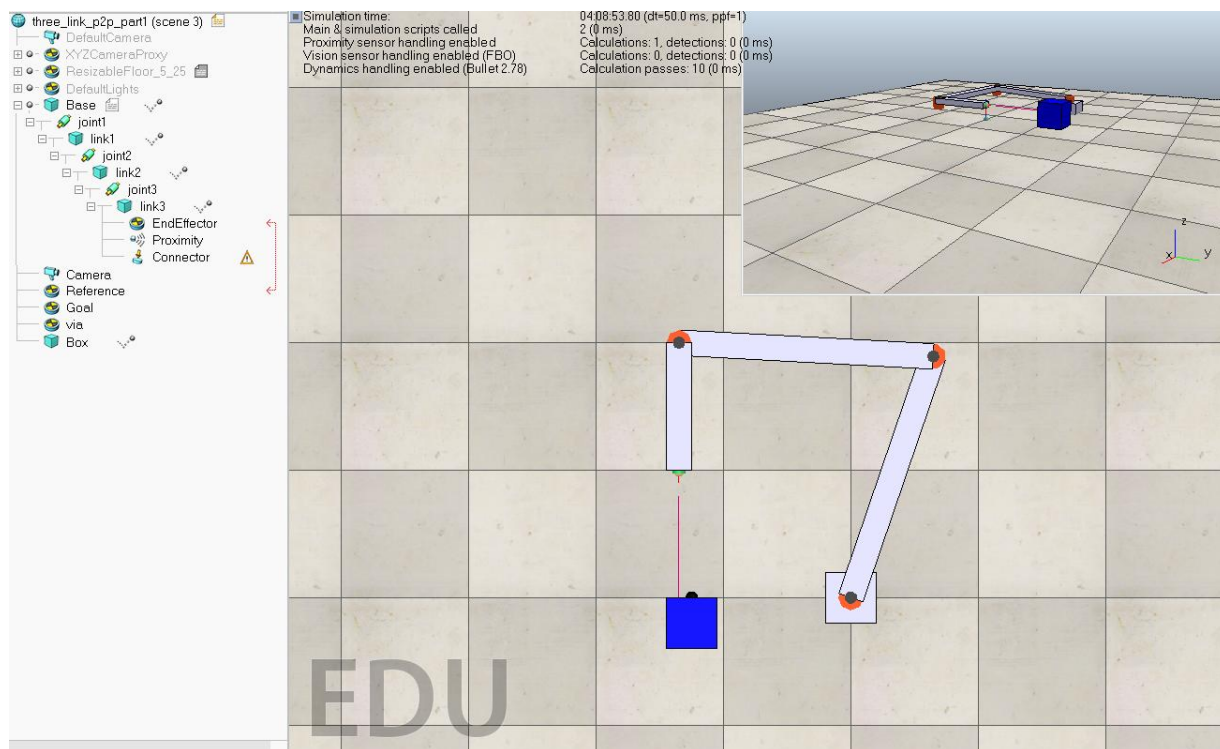


Рисунок 3.11 – Проміжний результат роботи системи

Таким чином, згідно з рисунком 3.10 маніпулятор працює за часовими інтервалами:

- 3-6 секунда – перехід до об'єкта та захоплення;
- 6-9 секунда – переміщення через проміжну точку;
- 9-11 секунда – переміщення (доставлення) об'єкта до цільової позиції;
- 11-12 секунда – відпускання об'єкта;
- 12-15 секунда – повернення маніпулятор у проміжну точку.

Наступним кроком є створення об'єктів-перешкод та створення функцій коду для написання алгоритму проходження маніпулятором перешкод. Основною задачею цього етапу є розширення можливостей роботи побудованого маніпулятора, який буде здатен автоматично переміщати декілька об'єктів у задану

позицію, використовуючи проміжну точку для уникнення перешкод або оптимізації траєкторії руху. Для виконання цієї операції було запрограмовано автоматичне циклічне переміщення кількох об'єктів з використанням кінцевого автомата (Finite State Machine), який послідовно виконує наступні стани:

- стартова ініціалізація;
- переміщення з проміжної точки до об'єкта (via → pick);
- захоплення об'єкта;
- повернення до проміжної точки (pick → via);
- доставлення до цільової точки (via → goal);
- завершення одного циклу та перехід до наступного об'єкта.

На рисунку 3.12 зображено змінену для виконання зазначеного завдання систему.

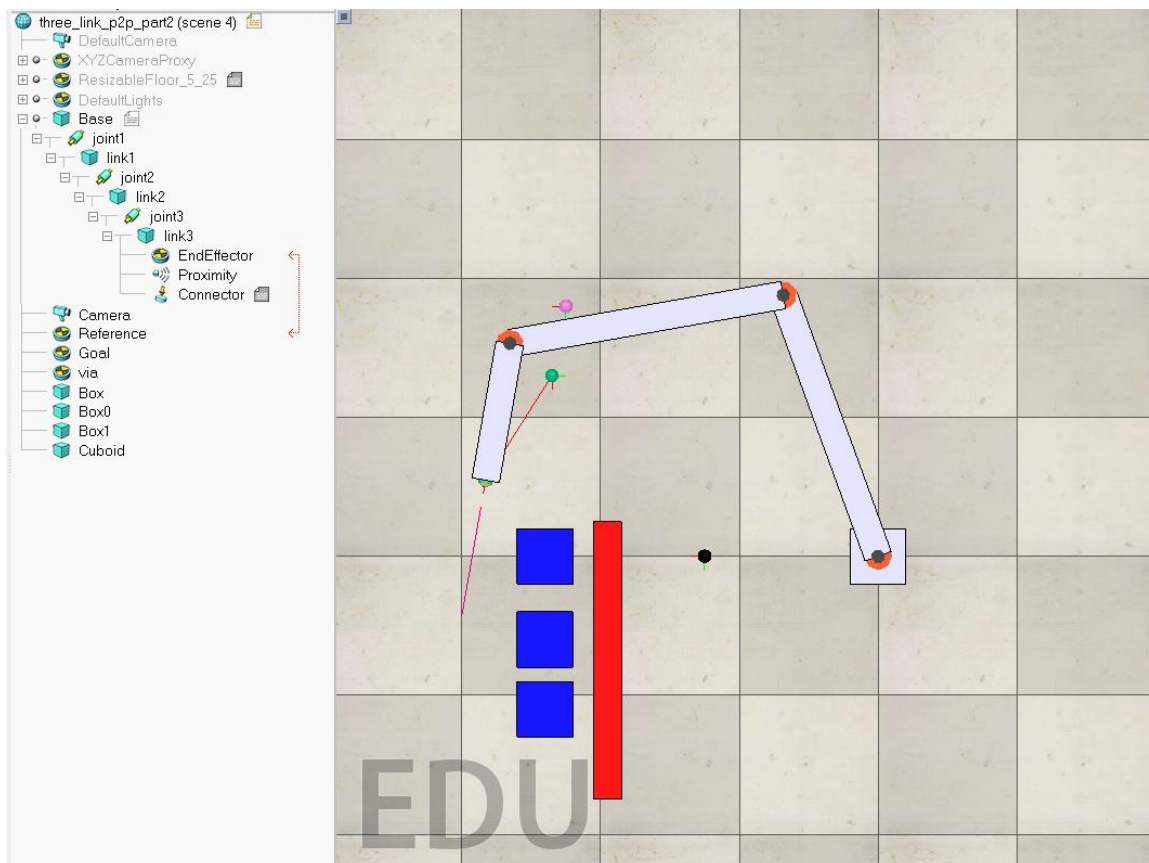


Рисунок 3.12 – Система після проведення додаткових налаштувань

У створеному скрипті реалізовано плавне переміщення між точками за допомогою інтерполяційної функції `moveIK()`, яка розраховує положення кінцевої точки у кожен момент часу та задає її у сцені, а також створено операції захоплення та відпускання об'єктів, які було реалізовано за допомогою функцій `graspObject()` та `releaseObject()`, які використовують сенсор для виявлення об'єкта в зоні досяжності та виконують зміну батьківського об'єкта через `sim.setObjectParent()`.

На рисунку 3.13 зображено результат виконання маніпулятором поставленої задачі з захоплення та переміщення об'єктів з униканням перешкоди на шляху.

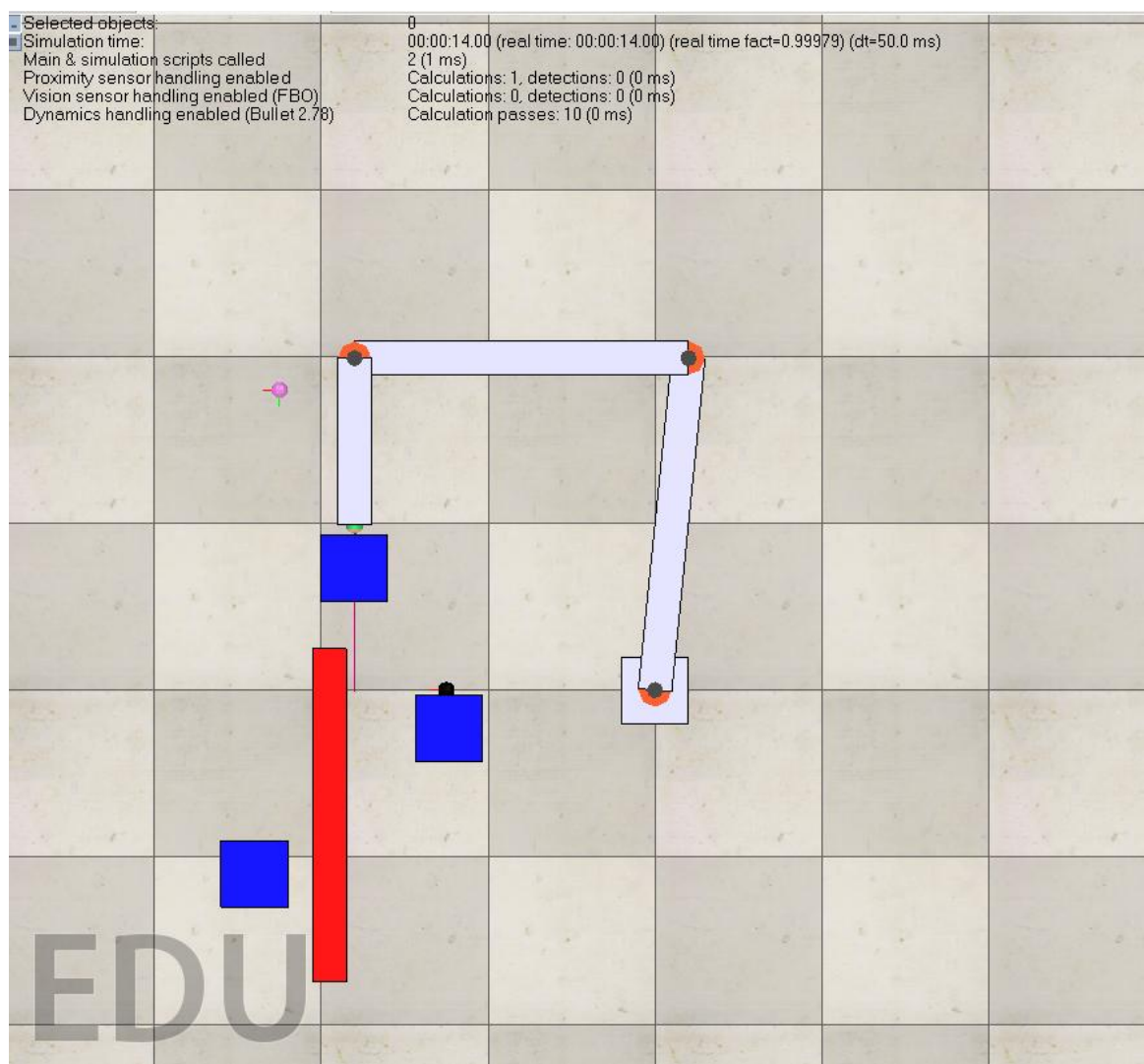


Рисунок 3.13 – Результат роботи системи

Таким чином, увесь процес організовано як послідовність станів, кожен з яких відповідає за окрему дію маніпулятора. Кожен об'єкт переміщується окремо,

при цьому маніпулятор автоматично переходить до наступного об'єкта після завершення переміщення попереднього. Завдяки гнучкій FSM, систему легко розширити на сортування за кольорами, розмірами або іншими критеріями. Крім того, створений скрипт не потребує ручного задання об'єктів та працює з будь-якою кількістю динамічних елементів у сцені.

Також, в коді було реалізовано принципи зворотної кінематики для управління роботом. Зокрема, функція `moveIK()` обчислює переміщення маніпулятора (виходячи з початкової й кінцевої позиції) за допомогою інтерполяції між двома точками. Дана функція обчислює проміжні координати для кожної осі (x, y, z) між початковою та кінцевою позицією. Це відбувається через лінійну інтерполяцію між початковим і кінцевим значеннями для кожної осі.

3.2.3 Створення та програмування тривимірної моделі робота

Останнім кроком виконання практичної частини кваліфікаційної роботи є створення частини виробничого процесу у тривимірному вигляді та написання програмного забезпечення для керування рухами робота-маніпулятора.

Для КР було обрано промисловий робот типу Pick-and-Place типу UR5 (The Universal Robots UR5) [32], зовнішній вигляд якого наведено на рисунку 3.14.

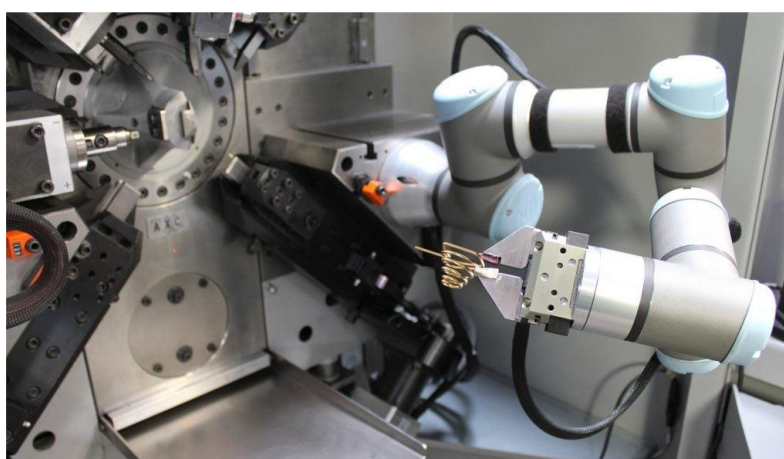


Рисунок 3.14 – Pick-and-Place робот типу UR5

Цей промисловий робот, призначений для автоматизації повторюваних завдань. Він має вантажність до 5 кг, радіус дії до 850 мм, вагу 18,5 кг і 6 ступенів свободи [32].

Перевагами даної моделі можна виділити наступні:

- гнучкість та легкість, адже ця модель ідеально підходить для завдань, які потребують точної та повторюваної роботи, такі як пакування, збирання та тестування;

- відносна простота використання, адже програмується цей робот за допомогою 3D-візуалізації на сенсорному планшеті;

- безпека використання, адже цей робот може працювати поряд із людьми без захисних бар'єрів (відповідно до норм);

- мобільність, що дозволяє легко переміщати робота між робочими зонами.

З погляду на недоліки основними можна зазначити наступні:

- обмежена вантажність, адже обмеження в 5 кг не підходить для важких завдань;

- обмеження в радіусі дії, що також зменшує коло використання цього роботу;

- дана модель не підходить для екстремальних умов середовища.

Отже, обраний промисловий робот особливо підходить для середніх підприємств, які прагнуть автоматизувати процеси ефективно, швидко та без значних витрат.

Описану вище модель мобільного робота було обрано в бібліотеці мобільних роботів програмного забезпечення CoppeliaSim для подальшого моделювання та програмування функцій. Для реалізації поставленої задачі було створено програмне забезпечення для керування рухами промислового робота та виконання ним заданого сценарію. Сценарій роботи робота передбачає наступні кроки:

- ідентифікація роботом об'єкта для захоплення;

- коректне виконання pick-операції (операції захоплення);

- ідентифікація кольору об'єкта для подальшої операції сортування;

- залежно від виявленого кольору об'єкта виконання операції перенесення об'єкта в задану позицію (place-операція).

На рисунку 3.15 наведено початкову сцену автоматизованого виробничого процесу.

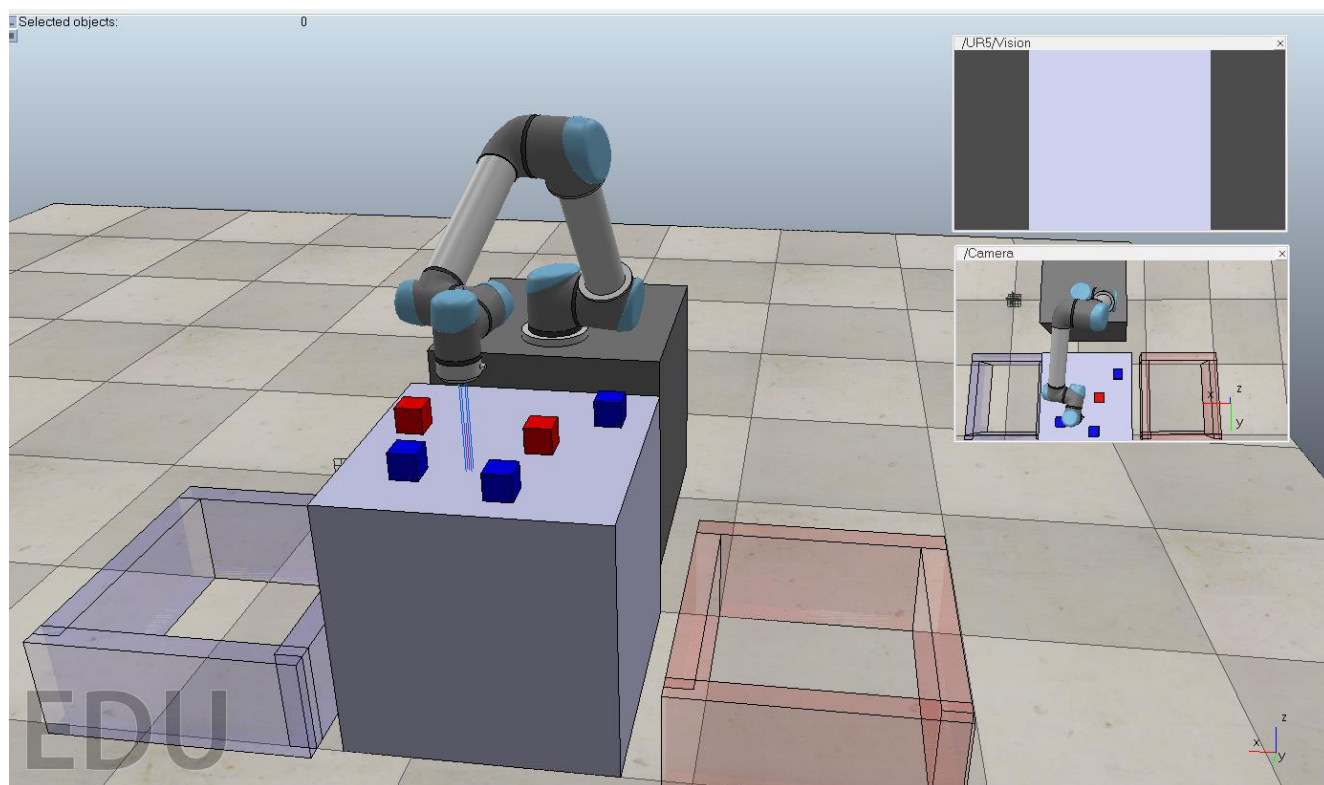


Рисунок 3.15 – Робот типу UR5 у програмному середовищі CoppeliaSim

Відповідно до зображення маємо частину виробничого процесу, яка включає наступні об'єкти: робот-маніпулятор для виконання завдань, об'єкти різного кольору (для прикладу взято об'єкти червоного та синього кольорів) та два бокси, які є фінальною (цільовою) позицію переміщення кінцевим ефектором зазначених об'єктів.

Наступним кроком було створення програмного забезпечення для керування рухами роботом-маніпулятором бажаних операцій, основні положення якого детально наведено у наступному підрозділі практичного розділу кваліфікаційної роботи.

Робот-маніпулятор виконує автоматизовану послідовність дій:

- захоплення коробки (box_1, box_2, ..., box_N) з початкової позиції;
- ідентифікація кольору об'єкта;
- підйом до безпечної висоти та переміщення до точки старту

– переміщення об'єкта до відповідного контейнеру залежно від кольору захопленого об'єкта (наприкладі синього та червоного об'єктів) та розміщення об'єкта в контейнер;

– повернення до стартової позиції, повторення для наступного об'єкта до моменту, поки робот не припинить бачити об'єкти для захоплення.

На рисунку 3.16 наведено процес роботи робота.

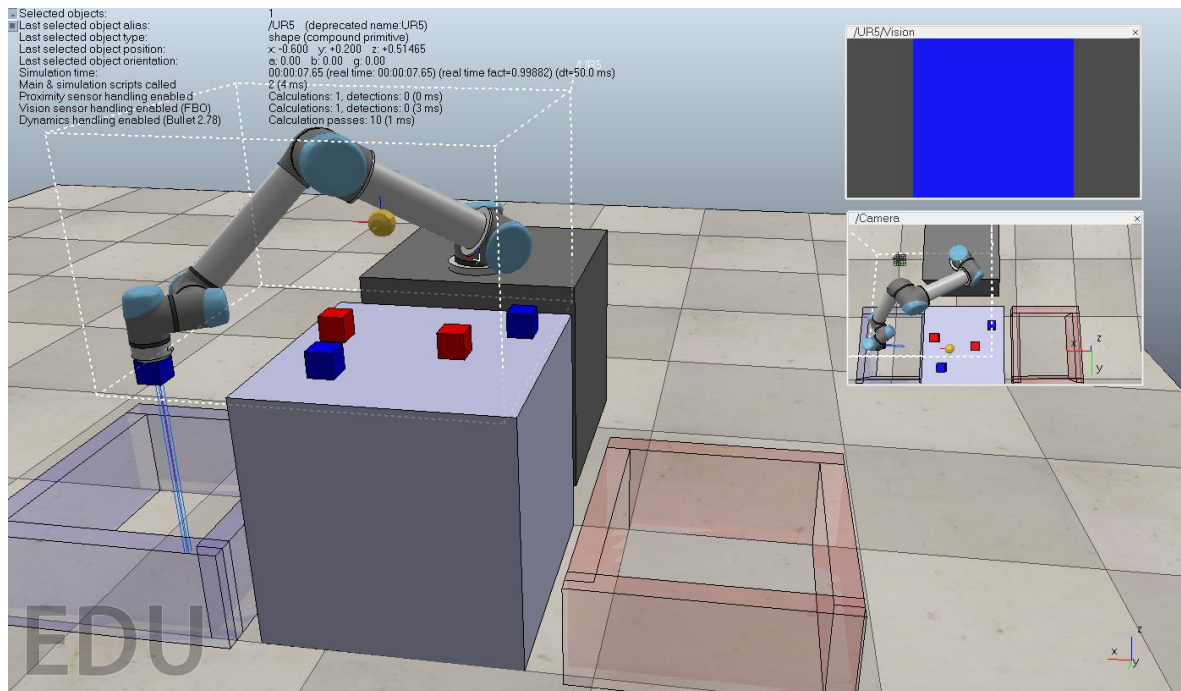


Рисунок 3.16 – Процес роботи в середовищі CoppeliaSim

Згідно з рисунком 3.16 видно ідентифікацію кольору об'єкта на екрані Vision, що спирається на дані, отримані із датчика, вигляд зверху зі встановленої камери та фронтальний вигляд процесу роботи.

На рисунку 3.17 показано результат завершення роботом роботи із захоплення, ідентифікації та розміщення об'єктів у робочому просторі.

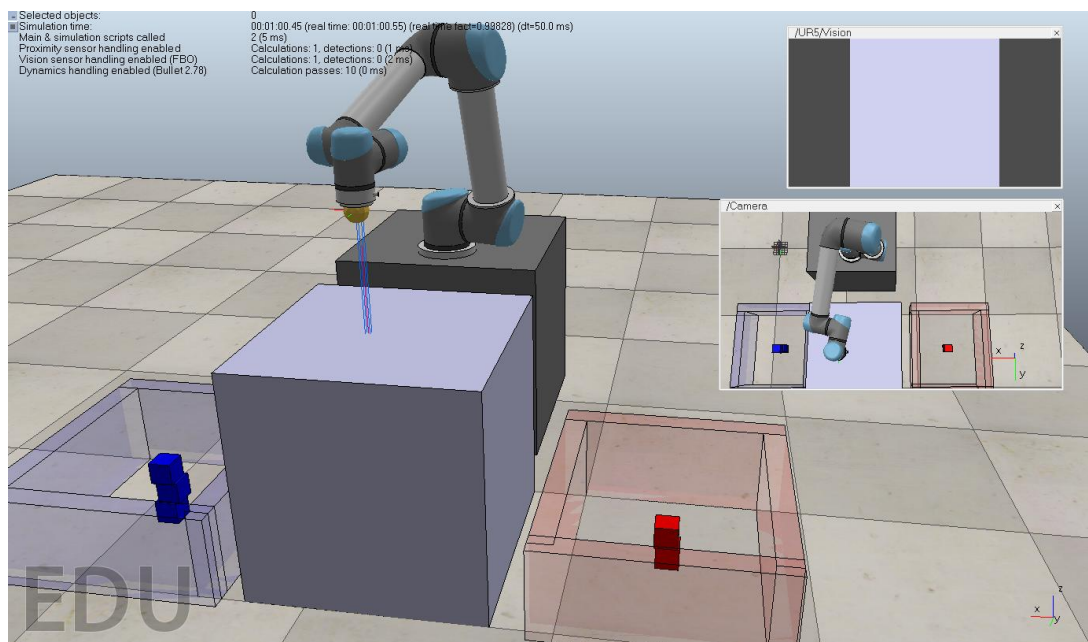


Рисунок 3.17 – Результат виконання роботом ТО

Процес реалізовано за допомогою кінцевого автомата, а переміщення – за принципами зворотної кінематики, розрахунок якої наведено в попередньому розділі. Вся логіка реалізована мовою програмування Lua в програмному середовищі CoppeliaSim.

3.4 Програмне забезпечення для керування роботом з CoppeliaSim

Створення програмного забезпечення для керування рухами промислового робота написано на мовою програмування Lua, що є мовою програмування за замовчуванням в обраному програмному середовищі. Програма реалізована як сукупність функцій, що обробляються програмним забезпеченням у ключові моменти симуляції:

- `sysCall_init()` – виконується один раз при запуску симуляції та ініціалізує всі об'єкти та змінні;
- `sysCall_actuation()` – виконується на кожному кроці симуляції та запускає цикл керування.

Ключовим елементом логіки керування є кінцевий автомат станів (FSM), реалізований у функції `moveSimpleManipulator()`. Кожна дія маніпулятора – це

окремий стан, між якими він переходить у логічному порядку відповідно до таймера та успішності попередніх операцій. У таблиці 3.1 наведено опис станів, через які проходить робот.

Таблиця 3.1 – Опис станів FSM

Стан	Опис
1	2
state_start	Початковий стан. Підготовка до захоплення
state_start2pick	Переміщення до місця захоплення об'єкта
state_pick	Спроба захопити об'єкт
state_lift	Підйом об'єкта до безпечної висоти
state_lift2start	Повернення до стартової позиції з об'єктом
state_star2bin	Переміщення до відповідного контейнеру
state_lower	Опускання об'єкта
state_place	Відпускання об'єкта
state_lift_after_place	Підйом маніпулятора після розміщення
state_bin2start	Повернення до старту
state_stop	Кінець поточного циклу, підготовка до наступного.

Згідно з таблицею 3.1 перехід між станами відбувається на основі часових відміток (змінні t_i , t_f) та позицій.

Рух маніпулятора між двома точками здійснюється за допомогою лінійної інтерполяції в тривимірному просторі. Функція `moveK()` поступово змінює положення контрольної точки маніпулятора (`RefPosition`) на основі поточного часу симуляції.

Функція `graspObject()` реалізує кілька способів захоплення:

- перевірка, чи об'єкт не є статичним;
- перевірка з використанням сенсора наближення (`Proximity`);
- примусове захоплення (на випадок відмови сенсора).

Захоплення відбувається шляхом встановлення об'єкта як дочірнього до з'єднувального елемента (Connector) маніпуляційного робота UR5.

Функція `releaseObject()` скидає об'єкт з маніпулятора, від'єднуючи його від з'єднувального елемента (Connector), тим самим завершуючи маніпуляцію.

Функція `getBoxDestination` виконує аналіз кольору активного об'єкта за допомогою візуального сенсора (Vision). Загальний алгоритм функції працює наступним чином:

- а) зчитування зображення з камери (`getVisionSensorCharImage`) та обхід усіх пікселів зображення, підрахунок сумарного значення каналів R, G та B;
- б) обчислення середнього кольору зображення;
- в) нормалізація значень, щоб врахувати зміну освітлення;
- г) перевірка умовної логіки:
 - 1) якщо синій канал домінує, об'єкт вважається синім;
 - 2) якщо червоний канал домінує, об'єкт вважається червоним;
 - 3) якщо не визначено, повертається «unknown».

Надалі, пропонується більш детальний аналіз даної функції.

Першим кроком функції є зчитування зображення з сенсора Vision. Зображення повертається у вигляді одновимірного рядка байтів, де кожні 3 байти відповідають одному пікселю (RGB: червоний, зелений, синій). Також зчитуються розміри зображення `resX` та `resY`.

```
function getBoxDestination(box_handle)
    local image, resX, resY = sim.getVisionSensorCharImage(Vision)
    local r_sum, g_sum, b_sum = 0, 0, 0
    local pixel_count = resX * resY
```

Далі, ініціалізуються змінні для накопичення сумарних значень кожного кольорового каналу. `Pixel_count` визначає загальну кількість пікселів у зображенні.

```
for i = 0, pixel_count - 1 do
    r = string.byte(image, i * 3 + 1)
    g = string.byte(image, i * 3 + 2)
    b = string.byte(image, i * 3 + 3)
```

```

r_sum = r_sum + r
g_sum = g_sum + g
b_sum = b_sum + b
end

```

В наступній частині коду, виконується обхід кожного пікселя зображення. За допомогою функції `string.byte` витягуються значення каналів R, G, B, які додаються до відповідних сумарних змінних.

```

local r_avg = r_sum / pixel_count
local g_avg = g_sum / pixel_count
local b_avg = b_sum / pixel_count

```

Після обходу пікселів обчислюються середні значення кольорових каналів. Це дає уявлення про домінуючий колір на всьому зображенні.

```

local max_val = math.max(r_avg, g_avg, b_avg)
if max_val == 0 then max_val = 1 end
local r_n = r_avg / max_val
local g_n = g_avg / max_val
local b_n = b_avg / max_val

```

Виконується нормалізація значень каналів. Це необхідно для того, щоб усунути вплив освітлення. Наприклад, світлі або темні зони можуть викривати абсолютні значення каналів. Нормалізація дозволяє порівнювати співвідношення між кольорами незалежно від яскравості.

```

if b_n > 0.6 and r_n < 0.4 and g_n < 0.4 then
    return "blue"
elseif r_n > 0.6 and g_n < 0.4 and b_n < 0.4 then
    return "red"
else
    return "unknown"
end
end

```

Фінальний етап – класифікація кольору. Якщо домінує синій канал (значення понад 0,6), то об'єкт визначається як «blue», аналогічно для «red». Якщо жоден колір не домінує – функція повертає «unknow», що дозволяє враховувати неоднозначні або темні об'єкти.

Таким чином, рух маніпулятор визначається шляхом інтерполяції між початковою і кінцевою точкою переміщення (`moveIK()`), з урахуванням поточного часу симуляції (`sim.getSimulationTime()`). У такий спосіб досягається плавність рухів і точність позиціонування.

Умови захоплення перевіряються за допомогою сенсора близькості (`Proximity`) та додатково перевіряються на рухомість (не є статичними). Якщо об'єкт успішно захоплено, він тимчасово прикріплюється до спеціального конектора (`Connector`), що імітує хват робота.

Отже, у створеному програмному забезпеченні використано наступні принципи та технології:

- принцип ІК: положення кінцевого ефектора змінюється залежно від координат задачі, а не від прямих кутів обертання суглобів;
- часова інтерполяція, яка дозволяє плавно анімувати рухи робота, не створюючи різких переходів;
- автоматизація через FSM, яка дозволяє чітко структурувати дії робота та легко розширювати логіку;
- вбудовані функції з API `CoppeliaSim` [24].

Отже, ПЗ реалізує симуляцію `Pick-and-Place` маніпулятора, який здійснює автоматизоване захоплення, сортування та переміщення об'єктів залежно від кольору захопленого об'єкта

3.5 Охорона праці

Забезпечення охорони праці при моделюванні та експлуатації робототехнічних систем є обов'язковим етапом у проєктуванні будь-якої автоматизованої або комп'ютеризованої технології. Згідно з Законом України «Про

охорону праці» №2694-ХІІ від 14.10.1992. (зі змінами) [33], роботодавець зобов'язаний створити безпечні та нешкідливі умови праці, а також забезпечити дотримання правил техніки безпеки, пожежної безпеки та санітарно-гігієнічних норм.

У процесі моделювання робототехнічних систем основні ризики пов'язані з електробезпекою (при роботі з апаратною частиною), ергономікою робочого місця, психофізіологічними навантаженнями, а також впливом електромагнітного випромінювання від комп'ютерної техніки.

Для забезпечення належних умов праці розробника, відповідно до ДСанПіН 3.3.2-007-98, робоче місце має бути організоване згідно з вимогами ергономіки та охорони зору, а саме:

- рівень освітлення робочої зони має становити не менше 300 лк;
- відстань від очей до екрана монітора має бути 50-70 см;
- тривалість безперервної роботи за ПК не повинна перевищувати 2 години із подальшою перервою 10-15 хв;
- наявність захисного екрана або належна якість дисплея є обов'язковою.

Окрім цього, відповідно до ДСТУ ISO «Основні принципи ергономіки для проектування систем», необхідно враховувати ергономічні характеристики програмного забезпечення, зокрема інтерфейсу програмного середовища CorreliaSim, для зменшення когнітивного навантаження на користувача.

У разі використання електронних компонентів потрібно дотримуватися ПУЕ та Правил технічної експлуатації електроустановок споживачів. Відповідно до НПАОА 0.00-1.28-10 необхідні заземлення, ізоляція, сигнальне маркування та використання захисних засобів для забезпечення безпечних умов праці [36]. В лабораторії повинні бути вогнегасники (типу ВП-5 або ВВК-2), план евакуації, інструкція з пожежної безпеки, що регламентується НПАОП 0.00-5.18-07 [36].

Таким чином, дотримання вимог чинного законодавства України у сфері охорони праці, норм ергономіки та техніки безпеки дозволяє забезпечити безпечні та ефективні умови для роботи над створенням, моделюванням та тестуванням робототехнічних систем у рамках освітньої чи дослідницької діяльності.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було досліджено, розроблено та реалізовано модель робота-маніпулятора типу Pick-and-Place з використанням програмного забезпечення CoppeliaSim. Робота охоплює весь цикл технічного проектування: від аналізу сучасного стану автоматизованих систем до перевірки функціональності створеної моделі в середовищі імітаційного моделювання.

Аналіз сучасного стану промислової автоматизації показав, що інтеграція роботизованих систем у виробництво є ключовим напрямком розвитку індустрії в умовах концепцій Industry 4.0 та 5.0. Зокрема, маніпуляційні роботи, як частина автоматизованих систем, забезпечують високу точність, повторюваність і безпечність виконання технологічних операцій, особливо в умовах, небезпечних або складних для людини.

У роботі було обґрунтовано вибір програмного забезпечення для моделювання системи, враховуючи його широкі функціональні можливості, підтримку скриптових мов, а також здатність інтеграції з іншими платформами. Проведено порівняльний аналіз переваг та недоліків середовища, що дозволило обрати найефективніший шлях реалізації поставленого завдання.

Практична частина роботи включала реалізацію кількох ключових етапів:

- розробку та програмування дволанкового маніпулятора;
- реалізацію алгоритмів прямої та зворотної кінематики на основі проведених розрахунків;
- побудову тривимірної моделі роботи типу Pick-and-Place;
- програмування основних функцій, а саме: ідентифікація об'єкта та визначення кольору об'єкта під час pick-операції, транспортування та розміщення під час place-операції;
- створення додаткових параметрів скрипту для проходження перешкод під час виконання роботом поставленої задачі.

Значну увагу було приділено математичному моделюванню системи, що включає проведення розрахунків ПД-регулятора для забезпечення точності управління, а також розв'язання задачі зворотної (інверсної) кінематики чисельним методом із використанням матриці Якобі.

Результати симуляцій підтвердили коректність роботи системи, візуалізація рухів у середовищі CoppeliaSim дозволила оцінити точність та ефективність запрограмованих сценаріїв. Крім цього, було визначено основні вимоги до умов роботи програміста та інженера при роботі з автоматизованими системами, сформульовано заходи безпеки при використанні електроустаткування та тестуванні робототехнічних систем.

Таким чином, в ході виконання кваліфікаційної роботи було досягнуто мети виконання роботи. Отриманні результати можуть бути використані в навчальному процесі, а також як основа для подальших досліджень і розширення функціональності робототехнічних систем. Побудована модель робота-маніпулятора може бути адаптована для складніших сценаріїв керування або інтегрована в симуляції виробничих ліній.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ненова Д. В. Моделювання робота-маніпулятора у програмному забезпеченні CoppeliaSim. 29-й Міжнародний молодіжний форум «Радіoeлектроніка та молодь у ХХІ столітті». Зб. матеріалів форуму. Т. 2. – Харків : ХНУРЕ. 2025. – С.8-10 с.
2. United Nations. The 17 Sustainable Development Goals. United Nations, 2025, Режим доступу: sdgs.un.org/goals (дата звернення: 18.03.2025).
3. Nienova D. Inverse Kinematics in Robotics: Case of Pick-and-Place Manipulators. «Automation and Development of Electronic Devices» ADED-2025 [Електронний ресурс] : збірник студентських наукових статей – Харків : ХНУРЕ, 2025. – Вип. 1. С.111-115.
4. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки та техніки. Структура та правила оформлювання. Введ. 2015-06-22. К. Держстандарт України, 2017. – 29 с.
5. Навчальний посібник з підготовки кваліфікаційної роботи першого (бакалаврського) рівня вищої освіти за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «АКІТ» : Навчальний посібник / Упоряд.: І. Ш. Невлюдов, О. І. Филипенко, О. В. Токарева, С. П. Новоселов, О. В. Сичова. – Харків : Видавництво Іванченка І. С., 2023. – 151 с.
6. Rathour, Kunal & Sikarwar, Basant & Phanden, Rakesh & Jacsó, Ádám & Singh, Rajeev. (2024). Smart Manufacturing: Issues and Challenges. 10.1007/978-981-97-3173-2_15. (дата звернення: 26.05.2025).
7. Конспект лекцій з дисципліни «Основи комп'ютерно-інтегрованого управління» для студентів спеціальності 151 [Електронне видання] / Упор.: А. І. Бронніков. – Харків: ХНУРЕ, 2023. – 358 с.
8. Nienova D. Programmable Providing of Data on Functional Dependencies of Material Characteristics. «Automation and Development of Electronic Devices» ADED-2023. [Електронний ресурс] : збірник студентських наукових статей / Харківський

національний університет радіоелектроніки ; [редкол.: І.Ш. Невлюдов та ін.]. – Харків : ХНУРЕ, 2023. – Вип. 2. – С. 143-146.

9. Nevliudov, Igor & Tsybal, Oleksandr & Bronnikov, Artem & Mordyk, Olexandr. (2020). INTERNET OF THINGS FOR ROBOTIC PROJECTS. Innovative Technologies and Scientific Solutions for Industries. 58-64. 10.30837/ITSSI.2020.13.058. (дата звернення: 23.04.2025).

10. Chala, Olena & Bronnikov, Artem & Nevliudov, Igor & Mospan, Denis. (2022). The Use of Neural Networks for the Technological Objects Recognition Tasks in Computer-Integrated Manufacturing. 1-5. 10.1109/MEES58014.2022.10005750. (дата звернення: 20.05.2025)

11. Невлюдов І. Ш., Андрусевич А. О., Євсєєв В. В., Новоселов С. П., Демська Н. П. Проєктування мобільних маніпуляційних роботів : Монографія. – Х. : 2022. – 427 с.

12. Невлюдов І. Ш., Андрусевич А. О., Євсєєв В. В., Максимова С. С. Моделі та методи кіберфізичних виробничих систем в концепції Industry 4.0 : Монографія. – Харків : 2023. – 321 с.

13. Невлюдов І. Ш., Андрусевич А. О., Євсєєв В. В. Проєктування мобільних роботів на базі одноплатних комп'ютерів (Raspberry Pi и мови Python 3.6) : Підручник – Харків : 2020. – 257 с.

14. Невлюдов І. Ш., Стародубцев М. Г. Конспект лекцій з дисципліни «Автоматизоване управління технологічними процесами» для студентів спеціальності 174. – Харків: ХНУРЕ, 2023. – 220 с.

15. Промислові роботи [Електронний ресурс]: Використання роботів на промислових підприємствах. Режим доступу: https://kaf-av.tntu.edu.ua/index.php/mn-abiturient/mn-articles/676-art-industrial-robots#rob_prom (дата звернення: 24.03.2025).

16. Shanmugamani, K., Fazeeda M. The Implementation of Warehouse Management System (WMS) to Improve Warehouse Performance in Business to Business (B2B). International Journal of Industrial Management. 17 : 2023. – P. 231-239. DOI: 10.15282/ijim.17.4.2023.10091 (дата звернення: 26.03.2025).

17. Suleiman, Z., Shaikholla, S., Dikhanbayeva, D., Shehab, E., Turkyilmaz, A. Industry 4.0: Clustering of concepts and characteristics. Cogent Engineering, 9(1) : 2022. Режим доступу: <https://doi.org/10.1080/23311916.2022.2034264> (дата звернення: 18.03.2025).

18. Gurin D. Key features and differences between Industry 5.0 and Industry 4.0.: Тези доповідей. Запоріжжя : XII Міжнародна науково-практичної конференція, 2024. Режим доступу: <https://openarchive.nure.ua/handle/document/29571> (дата звернення: 22.03.2025).

19. Невлюдов І. Ш., Новоселов С. П., Сичова О. В. Застосування цифрових двійників в технічних засобах автоматизації для розроблення програмно-технічних комплексів АСУТП : навч. посіб. Харків : Видавництво Іванченка І.С., 2023. – 267 с.

20. Novoselov S., Sychova O. Technology of Using Digital Twins in the Control of Industrial Equipment. Information systems in project and program management: Збірник монографій: 2023. Режим доступу: <https://doi.org/10.30837/MMP.2023.281> (дата звернення: 24.03.2025).

21. Демська Н. П., Конспект лекцій з дисципліни «Комп'ютерні системи інженерної графіки та 3D-моделювання» для студентів спеціальності 151. Упоряд.: Н.П. Демська. – Електронне видання. – Харків: ХНУРЕ, 2024. – 240 с.

22. Аллахверанов Р. Ю., Альохіна С. В., Невлюдов І. Ш., Ромашов Ю. В. Основи моделювання та ідентифікація об'єктів автоматизації: навч.посібник. – Харків: ХНУРЕ, 2023. – 446 с.

23. CoppeliaSim. Robotics simulation platform. Режим доступу: https://www.mathworks.com/products/connections/product_detail/coppeliasim.html (дата звернення: 05.03.2025).

24. Regular API reference. CoppeliaSim User Manual. Режим доступу: <https://manual.coppeliarobotics.com/en/apiFunctions.htm> (дата звернення: 05.03.2025).

25. What Is Lua Programming and What Is It Used For?. Coursera. Режим доступу: <https://www.coursera.org/articles/lua-programming> (дата звернення: 28.03.2025).

26. MathWorks. (n.d.). What Is Inverse Kinematics? Режим доступу: <https://www.mathworks.com/discovery/inverse-kinematics.html> (дата звернення: 27.04.2025).

27. Modeling Inverse Kinematics in a Robotic Arm. MATLAB & Simulink. Режим доступу: <https://www.mathworks.com/help/fuzzy/modeling-inverse-kinematics-in-a-robotic-arm.html> (дата звернення: 27.04.2025).

28. Forward Kinematics, ROS Robotics. (n.d.). Home. Режим доступу: <https://www.rosroboticslearning.com/forward-kinematics> (дата звернення: 27.04.2025).

29. InverseKinematics. Section II Modelling Motion.cs.illinois.edu, Режим доступу: motion.cs.illinois.edu/RoboticSystems/InverseKinematics.html (дата звернення: 27.04.2025).

30. Owen-Hill, A. Inverse Kinematics in Robotics: What You Need to Know. RoboDK Blog. Режим доступу: <https://robodk.com/blog/inverse-kinematics-in-robotics-what-you-need-to-know/> (дата звернення: 27.04.2025).

31. Ghazi, Yasir & Abdul Hussein, Ahmed. (2021). Implementing optimization of PID controller for DC motor speed control. Indonesian Journal of Electrical Engineering and Computer Science. 23. 657-664. 10.11591/ijeecs.v23.i2.pp657-664. (дата звернення: 07.05.2025).

32. “Universal Robots UR5 | Tough, Flexible Collaborative Robot Arm.” WiredWorkers, Режим доступу: www.wiredworkers.io/cobot/brands/universal-robots/ur5/ (дата звернення: 12.05.2025).

33. Закон України «Про охорону праці». Офіційний вебпортал парламенту України. Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12#Text> (дата звернення: 25.05.2025).

34. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Офіційний веб портал парламенту України. Режим доступу: <https://zakon.rada.gov.ua/rada/show/v0007282-98#Text> (дата звернення: 25.05.2025).

35. ДСТУ ISO 6385:2005 «Ергономічне проектування робочих систем. Основні принципи» Online - нормативні документи України. Режим доступу:

https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=53017 (дата звернення: 25.05.2025).

36. Про затвердження Правил охорони праці під час експлуатації електронно-обчислювальних машин. Офіційний вебпортал парламенту України. Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0293-10#Text> (дата звернення: 25.05.2025).