

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

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

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)

(рівень вищої освіти)

Розроблення імітаційного моделювання схвату промислового маніпулятора
(тема)

Виконав: студент 2 курсу, гр. КТРСм 21-1
Русаков Віктор Вікторович
(прізвище, ініціали)

Спеціальність 151 автоматизація та
комп'ютерно-інтегровані технології
освітньої програми Комп'ютеризовані та
робототехнічні систем
(код і повна назва напрямку)

Тип програми освітньо-професійна
(повна назва освітньої програми)

Керівник проф. Максимова С.С.
(посада, прізвище, ініціали)

Допускається до захисту
зав. кафедри

(підпис)

Невлюдов І.Ш.

(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет	Автоматики і комп'ютеризованих технологій
Кафедра	Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки
Рівень вищої освіти	другий (магістерський)
Спеціальність	151 автоматизація та комп'ютерно-інтегровані технології
Тип програми	освітньо-професійна
Освітня програма	Комп'ютеризовані та робототехнічні системи (код і повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Русакову Віктору Вікторовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення імітаційного моделювання схвату промислового маніпулятора

затверджена наказом по університету від 07.11. 2022 р. № 1462 Ст

2. Термін подання студентом роботи до екзаменаційної комісії . . 2022 р.

3. Вихідні дані до роботи _____

3.1 Мобільний робот маніпулятор

3.2 Мова імітаційного моделювання на базі природньої мови

3.4 Переміщення у 3-х вимірному пространстві

3.5 Можливості програмування

4. Перелік питань, що потрібно опрацювати в роботі

4.1 Вступ

4.2 Аналіз сучасних методів роботи промислового маніпулятора в Індустрії 5.0

4.3 Розробка мови імітаційного моделювання

4.4 Розробка середовища імітаційного моделювання

4.5 Експериментальні дослідження та аналіз отриманих результатів

4.6 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційний матеріал представлений у форматі презентації PowerPoint (*.ppt) – 10 с., формату А4

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз актуальності роботи	10.09.2022 – 15.09.2022	Виконано
2	Аналіз аналогічних рішень	16.09.2022 – 10.10.2022	Виконано
3	Розробка мови імітаційного моделювання	11.10.2022 – 25.10.2022	Виконано
4	Розробка середовища імітаційного моделювання	25.10.2022 – 25.11.2022	Виконано
5	Оформлення пояснювальної записки	25.11.2022 – 12.12.2022	Виконано
6	Подання роботи на перевірку Інтернет-сервісом Unichек	13.12.2022	
7	Подання роботи на рецензію		
8	Подання роботи на підпис зав.кафедри		
9	Подання кваліфікаційної роботи в ЕК		

Дата видачі завдання 7 листопада 2022 р.

Студент

(підпис)

Керівник роботи

(підпис)

Русаков В.В.

(прізвище, ініціали)

проф. Максимова С.С.

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 74 с., 0 табл., 22 рис., 2 дод., 24 джерела.

ІНДУСТРІЯ 5.0, ПРОМИСЛОВИЙ МОБІЛЬНИЙ МАНІПУЛЯТОР,
КАЛАБОРТИВНІ РОБОТИ, ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ, РОЗРОБКА МОВИ

Об'єкт дослідження – системи імітаційного моделювання роботів.

Предмет дослідження – моделі, методи та мови імітаційного моделювання.

Мета дослідження – розробка системи імітаційного моделювання переміщення мобільного робота маніпулятора.

Методи дослідження – алгоритми, патерни та принципи програмування, мова програмування C#, фреймворк Unity.

У ході роботи було проведено аналіз сучасних методів моделювання роботи промислових роботів маніпуляторів в Індустрії 5.0, було проведено розробку мови імітаційного моделювання та розробку середовища імітаційного моделювання, після чого було проведено експеримент та розрахунки з охорони праці.

ABSTRACT

Explanatory note: 74 pp., 0 table, 22 fig., 2 appendix, 24 sources.

INDUSTRY 5.0, INDUSTRIAL MOBILE MANIPULATOR,
COLLABORATIVE WORK, SIMULATION, LANGUAGE DEVELOPMENT

The object of research is robot simulation modeling systems.

The subject of research is models, methods and languages of simulation modeling.

The purpose of the research is to develop a system for simulating the movement of a mobile robot manipulator.

Research methods – algorithms, patterns and principles of programming, C# programming language, Unity framework

In the course of the work, an analysis of modern methods of modeling the work of industrial manipulator robots in Industry 5.0 was carried out, a simulation modeling language was developed and a simulation environment was developed, after which an experiment and calculations on labor protection were carried out.

ЗМІСТ

Перелік скорочень	8
Вступ.....	9
1 Аналіз сучасних методів моделювання роботи промислового маніпулятора Індустрія 5.0.....	11
1.1 Аналіз поняття калабортивні роботи в рамках Промисловість 5.0	11
1.2 Аналіз конструкцій та параметрів промислового мобільного маніпулятора.....	15
1.3 Аналіз систем імітаційного моделювання роботи промислових маніпуляторів.....	22
1.3.1 Навчальна програма для імітаційного моделювання	22
1.3.2 Robot Operation System (ROS) з MoveIt	25
1.4 Дослідження методів розробки мов імітаційного моделювання.....	26
1.5 Постановка завдань дослідження	32
2 Розробка мови імітаційного моделювання.....	34
2.1 Розробка базових команд управління схвату	34
2.2 Розробка синтаксису мови	35
2.3 Висновки до 2 розділу	36
3. Розробка середовища імітаційного моделювання	37
3.1 Вибір середовища розробки.....	37
3.2 Розробка елементів 3D візуалізації промислового маніпулятора.....	38
3.3 Розробка алгоритмів роботи системи імітаційного моделювання.....	39
3.4 Розробка (цікавої функцій)	40
3.5 Розробка інтерфейсу користувача системи імітаційного моделювання.....	42

3.6 Висновки до 3 розділу	44
4 Експериментальні дослідження та аналіз отриманих результатів.....	45
4.1 Постановка завдання експерименту.....	45
4.2 Результати проведеного експерименту.....	45
4.3 Охорона праці.....	51
4.4 Висновки до 4 розділу	53
Висновки	54
Перелік джерел посилання	55
Додаток А Лістинг програми	57
Додаток Б Демонстраційний матеріал у вигляді презентації.....	63

ПЕРЕЛІК СКОРОЧЕНЬ

ДС – досліджуваної системи;

ЕОМ – електронній обчислювальній машині;

ЕРС – послідовно-керований технологічний ланцюжок;

НДЧ – науково-дослідна частина;

ПО – програмне забезпечення;

СМО – систем масового обслуговування;

ХНУРЕ – Харківський національний університет радіоелектроніки;

AML – мови алгебраїчного моделювання;

DSM – моделювання доменних імен.

EEML – мова розширеного корпоративного моделювання;

FMC – основи моделювання фундаментальних концепцій;

JSP – Jackson Structured Programming;

ORM – Object-Role Modeling;

ВСТУП

Глобальні зміни клімату, пандемія COVID-19 і геополітичні конфлікти серйозно змінюють світове бізнес-середовище та глобальні ланцюги вартості, змінюючи економічний ландшафт та створюючи підґрунтя для фундаментальних змін у політиці, промисловості, суспільстві та управлінні. Через це важливо не регресувати до нестабільної промислової парадигми (наприклад, Індустрія 3.0), а активно впроваджувати та користуватися можливостями Індустрії 5.0. Стрімкий розвиток нових технологій часто випереджає можливості організацій узгоджувати з ним свої пріоритети розвитку та ефективно використовувати переваги нових можливостей, що призводить до потреби вчасно орієнтуватися в мінливому бізнес-середовищі і виробляти релевантну стратегію розвитку. З огляду на велику кількість технологічних змін, мінливі вимоги до робочої сили, соціальні та регуляторні пріоритети є потреба у визначенні керівних принципів, які допоможуть встановити курс розвитку цифрової індустрії. Індустрія 5.0 стала основою для перегляду майбутнього енергетики, виробництва, мобільності ланцюгів поставок, які опираються на основи, закладені Індустрією 4.0.

Об'єкт дослідження – системи імітаційного моделювання роботів.

Предмет дослідження – моделі, методи та мови імітаційного моделювання.

Мета дослідження – розробка системи імітаційного моделювання переміщення мобільного робота маніпулятора.

Методи дослідження – алгоритми, патерни та принципи програмування, мова програмування C#, фреймворк Unity.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз сучасних систем імітаційного моделювання мобільних роботів;
- провести аналіз існуючих мов імітаційного моделювання;
- провести розробку базових команд для керування мобільним роботом схватом;

- провести розробку синтаксису мови імітаційного моделювання;
- обрати середовище та мову розробки;
- розробити візуальну модель промислового маніпулятора;
- розробити алгоритм роботи системи;
- розробити інтерфейс користувача;
- провести експеримент;
- оформити пояснювальну записку згідно з методичними вказівками [1], вимогами ДСТУ 3008:2015 [2].

1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ МОДЕЛЮВАННЯ РОБОТИ ПРОМИСЛОВОГО МАНІПУЛЯТОРА В ІНДУСТРІЯ 5.0

Промисловість 5.0 є неминучою. Основною рисою Індустрії 4.0 є максимально широке застосування у виробничих процесах роботів, BIG DATA та штучного інтелекту. Це з переходом на цифрові технології, дозволяють приймати рішення з урахуванням одержуваних у час даних про виробничі процеси. У свою чергу, це дозволяє знизити виробничі витрати, підвищити якість продукції, забезпечити швидкий випуск товарів, що відповідають поточним запитам ринку.

Навіть з урахуванням того, що багато професій зникнуть в результаті 4-ої промислової революції, у переробній промисловості спостерігається серйозна нестача кваліфікованої робочої сили. Дефіцит кваліфікованих робітників створює ідеальні умови для впровадження Індустрії 5.0 у виробництво [3].

Впровадження роботів у виробництво може спростити автоматизацію та створити справжню співпрацю між людиною та роботом. Коли кваліфікованого персоналу стає менше, така комбінація людина-робот може працювати ефективно та продуктивно, незважаючи на зміну чисельності робітників.

1.1 Аналіз поняття колаборативні (коботи) роботи у рамках Індустрії 5.0

Колаборативний робот (кобот) – це машина, яка важить від 4 до 40 кілограмів, здатна пересуватись в просторі за допомогою навігації. Кобот схильний навчатися – і під час пересування ідентифікує елементи будівлі, чітко визначає людей та інші об'єкти, адаптує маршрут у залежності від того, що наразі відбувається на виробництві. Максимальна вантажопідйомність таких роботів до 300 кг. Як правило, чим менша ціна, тим легший вантаж, який кобот може підняти.

Промислові роботи запрограмовані виконувати певні операції без урахування людей, які працюють поруч із ними, через що на виробництві вони можуть

створювати загрозу життю та здоров'ю людини. Відомі випадки загибелі людей через промислових роботів. Тому їх встановлюють у спеціально відведених місцях, монтують огороження в зоні дії робота і фарбують у яскраві кольори, щоб не наражати на небезпеку людей. За будь-якої фізичної взаємодії людини з промисловим роботом механізм необхідно попередньо відключити.

Колаборативні роботи оснащуються датчиками для контролю руха та позиції людини і не допускають заподіяння їй шкоди. Деякі моделі можна встановлювати безпосередньо на робочих місцях. Як правило, керування та програмування у коботів на порядок простіше, ніж у промислових роботів, і включає зокрема елементи ручного управління. Також ці роботи дешевші і не вимагають додаткових виробничих площ.

Більшість колаборативних роботів мають невеликі розміри (вага – від 15 кг до 20 кг, висота – близько 1,5 м). Промислові роботи значно більші. Так, KUKA має лише одну лінійку компактних роботів вагою близько 50 кг; моделі інших серій важать від 100 кг до 4600 кг [4].

Процес навчання кобота також відрізняється від стандартних роботів. Зазвичай, коботів не програмують на виконання специфічних дій, а замість цього, їх тренують люди: виконують деяку послідовність дій разом з коботами, показують рухи і кроки, після чого колаборативні роботи закріплюють успіх самостійно. Таким чином коботів гнавчають орієнтуватись у просторі.

Згідно зі звітом Business Wire, до кінця 2020 року ринок коботів досягне \$981 млн. Очікується, що ця цифра збільшиться до \$7,9 млрд до 2026 року. Таким чином, сукупний середньорічний темп зростання в розмірі 42% у найближчі п'ять-шість років прогнозує значну перспективність розвитку даного напрямку в робототехніці.

Значущі події в робототехніці відбулися в 1954 році, коли американський винахідник Джордж Девол запатентував механічну «руку», здатну піднімати і переміщати об'єкти. Прототип промислового робота відомий під назвою Unimate.

У 1962 році легендарна компанія General Motors взяла за основу винахід Девола і удосконаливши його, створила робота з шістьма осями для виконання найпростіших завдань на виробництві.

Наступна віха – 1981 рік: японський вчений-комп'ютерник Такео Канаде, відомий дослідженнями в галузі комп'ютерного зору, сконструював «руку» з вбудованим мотором, для регулювання швидкості та точності виконання поставлених завдань.

У 1990-х роках корпорація Yaskawa America розробила ускладнену електричну систему, що мала функцію віддаленого керування, за допомогою якої можна було змусити декількох роботів виконувати певні завдання.

Нарешті, в 1997 році Едвард Колгейт і Майкл Пешкін запатентували першого колаборативного робота.

Проїшли роки, і коботи стають невід'ємною частиною індустріального Інтернету речей.

Користь від впровадження коботів на підприємства – безсумнівна. Переглянемо наступну статистику:

– через підвищення продуктивності роботи на підприємствах завдяки коботам, кількість їх замовлень у нинішньому році збільшилася на 24 % (у сфері , переробки сировини, безпеки, харчової промисловості та електроніки);

– якщо зараз продажі колаборативних роботів складають 3 % від загальних продажів у сфері робототехніки, очікується, що до 2025 року цей показник зросте до 34 %;

– коботи стають дешевшими, а отже, їх можуть дозволити собі навіть компанії із середнім достатком: зокрема, в 2009 році коботи коштували в межах \$63 тис., а в 2018 році ціна знизилася до \$45 тис.

Єдина проблема, яка «супроводжує» коботів, пов'язана з кібербезпекою. Як і будь-який інший девайс, кобот може бути зламаний. І, якщо зловмисники захочуть зупинити роботу на підприємстві, у них, на жаль, це вийде. Тому керівникам великих компаній варто подбати про те, щоб відкрити відділ, що буде

займатись кібербезпекою, а також навчити всіх співробітників основам безпечної роботи в мережі.

Наприклад, якщо керування роботами відбувається через комп'ютер у віддаленому режимі, необхідно захистити з'єднання, наприклад, встановивши VPN. Також потрібно подбати про надійні паролі для всіх девайсів, що перебувають на підприємстві. Очевидно, що промислових приладів виявиться чимало, тому для генерації паролів можна використати віртуальні платформи з великою кількістю пам'яті для обробки та збереження, такі, як, наприклад, Last Pass, або розробити власний подібний сервіс.

Через «незграбність» традиційних роботів, їм не можна було доручити такі завдання, що вимагають спілкування з персоналом. У випадку з роботами, ці машини можна без проблем використовувати для автомобільної та промислової сфери. Також, роботів можна використовувати у логістиці, формуванні ланцюжка постачання і збуту, в цехах, а також у роздрібній торгівлі і тих областях, які спрямовані на надання послуг клієнтам. Роботи зможуть виконувати «чорнову» роботу, залишивши простір людям для творчих рішень.

На даний момент роботів застосовують у таких сферах:

- промисловість – працюючи в умовах дуже низьких або високих температурах, з матеріалами, які є небезпечними для людини, ці роботи визначають продукцію з дефектами та усувають браковані товари, а також тестують якість доставленої продукції;

- охорона здоров'я – роботів використовують для проведення простих медичних обстежень пацієнта, що було доведено в період пандемії, коли роботи в Японії проводили тест на COVID перед посадкою в літак, тим самим заощаджуючи сили лікарів. Крім того, роботи були використані для асистування в роботі з дітьми з затримкою в розвитку, і як виявилось, взаємодія з роботами поліпшує розумові показники;

- будівництво – роботи здатні виконувати трудомісткі завдання, що займають у людей багато часу, наприклад, кладку цегли. Під час будівельних

робіт завдяки роботам знижується коефіцієнт смертності при виконанні завдань особливої складності;

– сільське господарство – роботів залучають до посадок зернових культур, тим самим заощаджуючи час працівників.

Коли Інтернет речей став частиною будь-якого промислового підприємства, розробники замислилися про те, як забезпечити високу продуктивність праці. Роботи були розроблені для виконання одного завдання, і взаємодіяти з ними стало досить складно. Тому нині експерти в галузі робототехніки концентруються на створенні колаборативних роботів, а головні відкриття в сфері спілкування людини і робота ще попереду [5].

1.2 Аналіз конструкцій та параметрів промислового мобільного маніпулятора

Декартові (картезіанські, лінійні та порталні) промислові роботи-маніпулятори порівняно прості за кінематикою та надійні в експлуатації. Дана конструкція роботів складається з кількох лінійних осей переміщення, розташованих перпендикулярно один до одного.

Картезіанська конструкція – найпростіший варіант стаціонарної системи. Така система забезпечує високу жорсткість конструкції, завдяки малій кількості шарнірів та зчленувань, що при точних роботах дає високі показники точності. Наприклад, роботи цього типу при довжині робочого ходу 2500 мм, здатні точно повторювати однотипні руху з відхиленням не більше 0,05 мм. Поступальні кінематичні пари забезпечують мінімальне навантаження на кінематичні елементи і, отже, забезпечують високу кінематичну стабільність, зносостійкість та довговічність. Простота конструкції зумовлює низьку вартість обладнання.

Мінуси даної конструкції – не дуже зручна форма простору, який обслуговується, низьке співвідношення обслуговуваного і займаного роботом просторів, істотно обмежені можливості орієнтування інструменту, і, як наслідок, неможливість виконання робіт у місцях з обмеженим доступом.

Цей тип роботів досить широко використовують з метою автоматизації устаткування переробки пластичних мас. Безпосередньо для застосування на ливарних машинах вони були розроблені на початку 1980-х рр., і на сьогодні їх електричні та пневматичні приводи значною мірою стандартизовані. Промислові роботи цього легко інтегруються у систему управління ливарної машиною.

Колінно-важільні роботи можна віднести до маніпуляторів у сферичній системі координат. Основні особливості робота цього типу:

- наявність тільки обертальних рухів від серводвигунів, що обумовлює сферичну просторову траєкторію переміщення схвата;
- робочий обсяг має сферичну форму;
- вищу (порівняно з лінійним роботом) можливу кількість ступенів свободи схвата;
- ускладнене обслуговування та програмування робота.

Задана траєкторія переміщення схвата досягається за рахунок шести обертальних рухів робочої руки на контрольовані кути повороту. При цьому кожен рух здійснюється за допомогою окремого автономного сервоприводу з високою швидкодією, через що забезпечується практично миттєва зупинка схвату у заданій позиції.

Необхідно враховувати, що керування рухом осей колінно-важільних роботів набагато складніше, ніж лінійних роботів, тому обслуговування та усунення проблем колінно-важільних роботів часто стає можливим лише за допомогою фахівців-виробників роботів. Вартість використання колінно-важільних роботів зростає через необхідну вищу кваліфікацію персоналу та значні витрати на програмування. Так, наприклад, витрати на програмування та адаптацію колінно-важільних роботів до ливарних машин приблизно втричі вищі, ніж такі ж витрати для лінійних роботів.

Для того, щоб колінно-важільний робот виконував рух схвата із збереженням просторового становища, всі його шість осей повинні регулюватися одночасно. Таким чином реалізується управління траєкторією, що вважається ідеальною при допусках на відхилення в міліметровому діапазоні. Така траєкторія руху вимагає

від колінно-важільного робота значно більшої продуктивності порівняно з лінійним роботом.

До цієї групи належать шарнірні (антропометричні) промислові маніпулятори. За принципом дії, цей тип маніпулятора нагадує людську руку. У його кінематичній системі присутні не менше трьох поворотних з'єднань, які утворюють полярну систему координат (рис. 2.1). Основні шарніри забезпечують поворот маніпулятора, нахил у плечовому з'єднанні та згинання у лікті (осі 1-3); ще 3 поворотні осі (4-6) і одне призматичне зчленування забезпечують додаткові ступені свободи (крен, тангаж, нишпорення), що дозволяє руці вибрати будь-які напрями і вільно витягуватися по радіусу в будь-якому напрямку.

Відмінність даної конструкції - дуже висока гнучкість, що дозволяє роботу обходити будь-які перешкоди. Рука може прийняти будь-яке положення всередині зони, що обслуговується. Крім того, робот цього типу досить компактний, і зона, що обслуговується, набагато більше, ніж займане роботом місце. Як правило, типовий шарнірний маніпулятор має 5 або 6 програмованих ланок. Незважаючи на очевидні переваги конструкції, керувати таким роботом досить складно.

Серед кінематичних схем маніпуляторів у кутовій системі координат в останні роки широкого поширення набули два типи шарнірних роботів: PUMA та SKARA.

Абревіатура PUMA у перекладі розшифровується як «програмований універсальний маніпулятор для складання», SCARA розшифровується як «складальна рука з виборчою гнучкістю».

Компонувальна схема шарнірного промислового робота представлена на рис.

1.1.

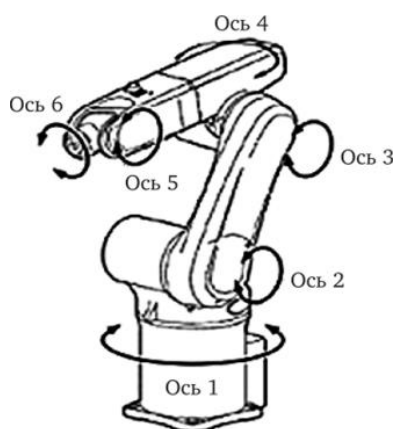


Рисунок 1.1 – Компонувальна схема шарнірного промислового робота

У схемі типу PUMA відбувається рух усіх зв'язок «руки» у вертикальній площині щодо горизонтальних осей і загальний поворот всієї «руки» навколо вертикальної осі (рис. 1.2).

Схеми маніпулятора промислового робота PUMA представлені на рис. 1.2.

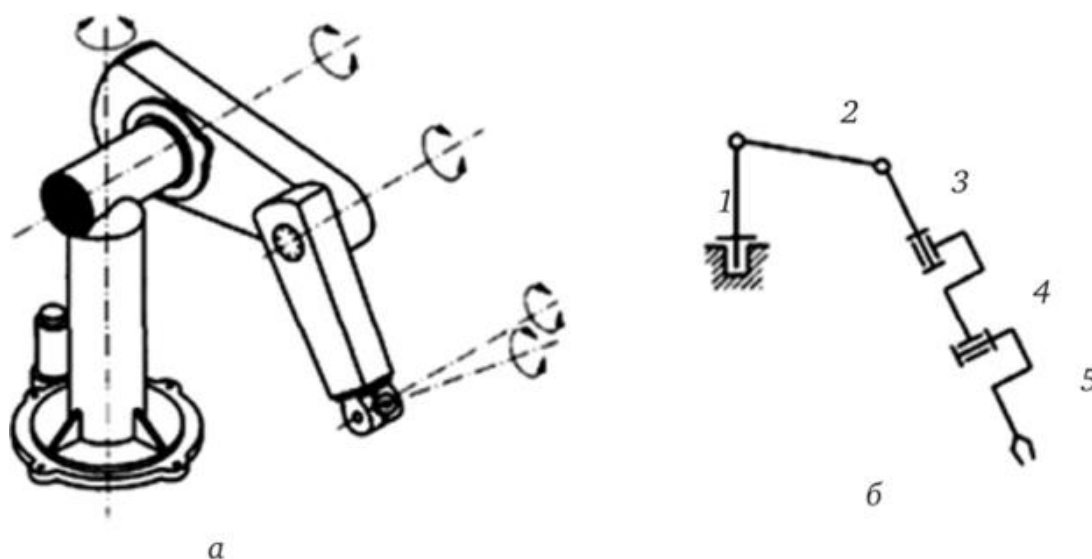


Рисунок 1.2 – Схеми маніпулятора промислового робота PUMA:

а – компонувальна; б – кінематична

У схемі типу SCARA усі ланки маніпулятора («руки») розташовані в горизонтальній площині і обертаються навколо вертикальних осей, а остання ланка зі схопленням (або інструментом) має вертикальне поступальне переміщення (рис. 1.3).

Головною перевагою даної конструкції є паралельне з'єднання зчленувань маніпулятора, внаслідок чого «рука» може вільно рухатися по горизонталі, зберігаючи при цьому вертикальну жорсткість. Важливо, що конструкція, що складається з двох ланок, може витягнутися, розпрямивши «лікоть», а може звернутися, звільнивши простір, що займає. Це зручно особливо в робочій зоні з обмеженим простором та під час роботи, коли деталі переміщуються з одного виробничого модуля до іншого. Зона обслуговування такого робота має вигляд сегмента порожнистого криволінійного циліндра. Промислові роботи цього типу мають низку переваг: велика швидкість пересування по горизонталі за рахунок підсумовування щодо вертикальних осей; висока точність позиціонування; особлива жорсткість конструкції щодо вертикальної осі; велика зона обслуговування; компактність [6].

Схема маніпулятора промислового робота SCARA представлена на рис. 1.3.

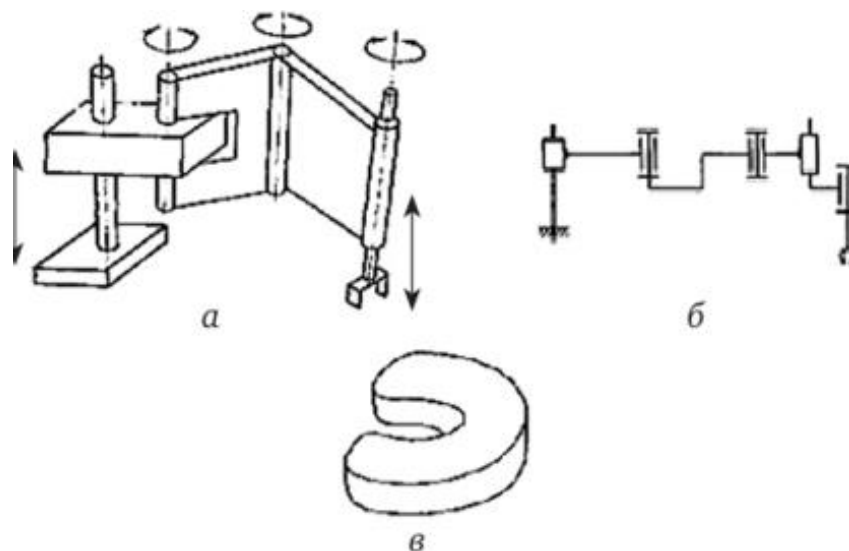


Рисунок 1.3 – Схема маніпулятора промислового робота SCARA:

а – компоновальна; б – кінематична; в – зона обслуговування

На виробництві виробів із пластмас промислові роботи такого типу використовуються переважно для транспортування рулонів плівки масою від 5 кг до 25 кг, труб діаметром до 500 мм, листів шаруватих пластиків розміром до 2000 мм на 4000 мм, листів жорсткого полівінілхлориду, рулонів лінолеумів тощо заключні операції. Відомий приклад застосування промислового робота типу PUMA для механічної обробки прес-виробів відповідального призначення з волокна.

У промисловості часто зустрічаються операції, де робот повинен взяти деталь, перенести в потрібному напрямку та опустити в інше місце (витягти виріб з форми і перенести його в накопичувач). Для виконання таких операцій було розроблено паралельні стрижневі роботи (рис. 1.4).

Суть їх конструкції така, що до базової установки кріпляться чотири «руки» що були з'єднані паралельно, утворивши кінематичну схему паралелограма (це така схема, при якій два паралельно з'єднані важелі рухаються синхронно, коригуючи напрямок один одного; приклад такої системи – спиця парасольки). Особливістю цієї системи є те, що вихідна ланка пов'язана з базою кількома кінематичними ланцюгами, які паралельно повідомляють інструмент рух. У результаті кожного кінематичного ланцюга є вільні від приводів зчленування, на які можна встановити різні датчики, а також додаткові приводи. Такий механізм дозволяє одночасно керувати швидкістю та зусиллям вихідної ланки по одній координаті. Паралельна структура має високу точність, що дозволяє використовувати маніпулятор для виконання операцій, що вимагають високої точності (вимірювальні роботи). Маніпулятор має малу вагу, тому здатний працювати на надшвидких швидкостях.

Схема паралельного промислового робота представлена на рис. 1.4.

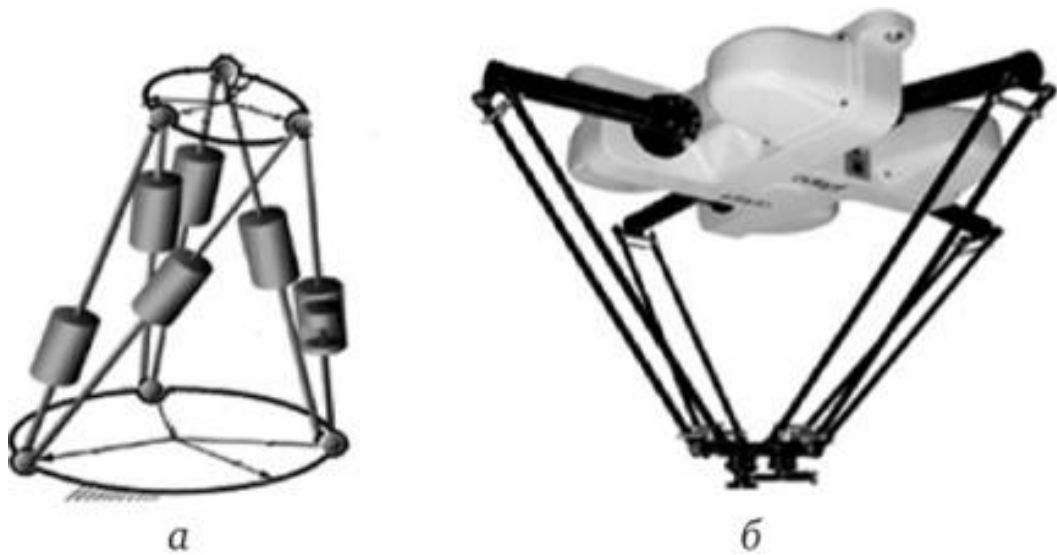


Рисунок 1.4 – Схема паралельного промислового робота:

а – компоувальна; б – структурна

В даний час широко поширені апарати, об'єднані під загальною назвою «дельта-роботи», які являють собою типову паралельну конструкцію: три маніпулятори, що рухаються карданною передачею, кріпляться до бази, розташованої вгорі у вигляді підвісної конструкції; маніпулятори, що сходяться внизу, з'єднані невеликою трикутною платформою, яка під час роботи зміщується по осях X , Y або Z , а центральний, четвертий важіль дає додатковий ступінь свободи – обертальний. Завдяки тому, що приводи встановлені в основі базової конструкції, а «руки» виконані з легкого композитного матеріалу, робот може здійснювати до 150 захватів за хвилину. Найчастіше такі роботи використовуються для пакування та сортування виробів [7].

Нині класифікація промислових роботів за типами конструкції стає дедалі умовною. Все частіше різні конструкції компонуються одна з одною – шарнірні маніпулятори ставляться на декартові конструкції, SCARA-роботи доповнюються шарнірними маніпуляторами, а роботи декартів обзаводяться додатковими ступенями свободи [8], [9].

1.3 Аналіз систем імітаційного моделювання роботи промислових маніпуляторів

1.3.1 Навчальна програма для імітаційного моделювання

Програму написано в середовищі візуального програмування Delphi 3.0.

Зовнішній вигляд головного вікна програми приведено на рис. 1.5.

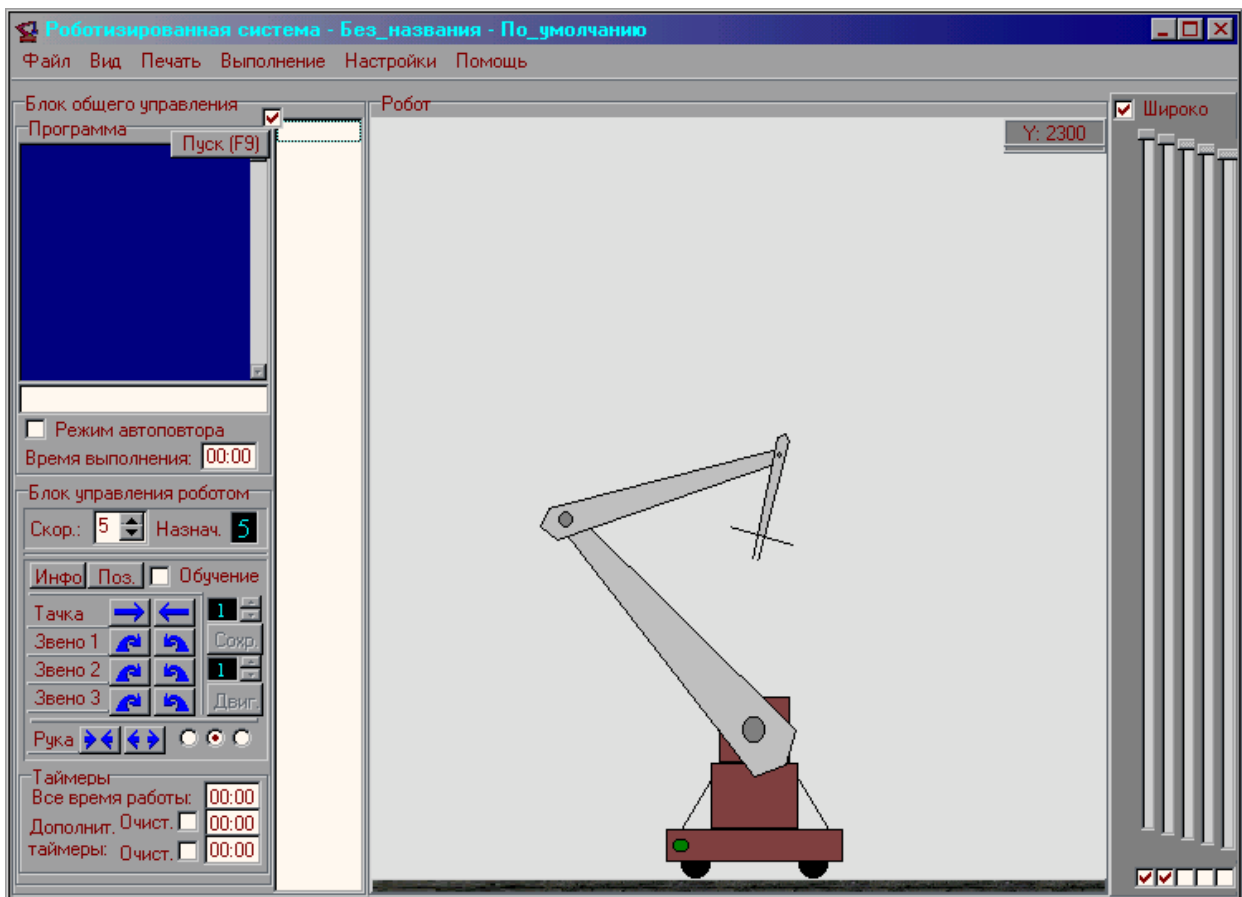


Рисунок 1.5 – Зовнішній вигляд головного вікна програми

Головне меню програми містить наступні пункти:

– файл – запис та зчитування параметрів робота, текстів програм і параметрів системи, а також закодування поточної програми у процедуру і вихід із програми. Більшість підпунктів цього пункту має “гарячі” клавіші;

- вид – керує видимістю допоміжних форм (Список процедур – список доступних процедур;

- окно кода – вікно виводу закодованої програми, його видимістю керує також перемикач зліва зверху вікна;

- інформація робота – вікно виводу стану маніпуляторів робота, використовується при програмуванні для спрощення описання потрібної траєкторії;

- широкие підставки – керує виглядом підставок: вони можуть бути широкими, з елементами керування положенням підставок, і вузькими, без елементів керування);

- выполнить – містить підпункти для запуску, зупинки виконання, а також для компіляції програми;

- настройки – настроювання параметрів ланок робота, гама кольорів в області введення програми (в редакторі програм), а також параметрами виведення зображення;

- помощь – коротко про програму.

Програми пишуться мовою RCL (Robot's Communication Language), що не є реальною мовою програмування промислових роботів. Синтаксис мови полягає в тому, що:

- оператори пишуться латинськими літерами, пробіли на початку рядку і після оператора ігноруються;

- ігнорується і будь-який інший текст у кінці рядка після оператора, відокремлений від останнього пробілом;

- кожен оператор розміщується в новому рядку;

- після оператору зразу ж слідує дужка із списком параметрів через кому при відсутності будь-яких інших розділових знаків між параметрами, а також між параметрами і дужками;

- коментарі пишуться після крапки з комою в новому рядку, чи ж після оператора, відокремлені від останнього пробілом.

У мові RCL є наступні оператори:

– CARMOVE – має один параметр: координату X. Пересуває “возик” робота в точку з заданою координатою;

– ROTATE – має два параметри, перший вказує номер ланки, що має обертатися, другий – відносний кут, на який має обернутися ланка. Нумерація ланок починається з одиниці і з найбільшої ланки. Якщо кут повороту ланки перевищує допустиме відхилення, ланка стає у відповідне крайнє положення;

– CHANGE – має три параметри, відповідає трьом операторам ROTATE, що виконуються одночасно. В параметрах записуються відносні кути обертання відповідних ланок;

– SPEED – має один параметр, вказує швидкість пересування робота (в межах 1-9, інакше – ігнорується);

– PROC – використовується для виклику доступної на цей час процедури (усі доступні процедури зображені в допоміжному вікні *Список процедур*);

– SETPOS – має чотири параметри, першим з яких задається позиція “возика” робота, іншими тьома – прямі значення локальних кутів повороту кожної ланки відповідно. Робот займає положення, вказане параметрами цієї команди. Усі переміщення виконуються одночасно;

– SAVEPOS – має один параметр – номер запису у чарунці пам’яті, де зберігатимуться значення поточних локальних кутів повороту усіх трьох ланок робота. Номер запису може змінюватись в межах від 1 до 30. Використовується разом з оператором LOADPOS;

– LOADPOS – має один параметр, що вказує на номер запису у пам’яті, де зберігаються значення локальних кутів повороту усіх трьох ланок робота, що їх буде мати кожна ланка в результаті виконання оператора. Використовується разом з оператором SAVEPOS. Усі переміщення виконуються одночасно;

– PALMDIST – має один параметр: відстань між пальцями схвату робота. Його величина може змінюватись у межах від 2 до 100;

– PALMANGLE – має один параметр, що вказує номер жорстко заданого в системі керування роботом номера кута, на який може бути повернутим схват

робота. Є наступні можливі значення цього параметру: - 1, 0 і 1. вони відповідають локальним кутам - 90°, 0° і 90° відповідно.

1.3.2 Robot Operation System (ROS) з MoveIt

Проаналізувавши статті деяких авторів, було виявлено що для моделювання імітації роботи роботів пропонується використовувати програми Robot Operation System (ROS) та MoveIt, зсилаючись на швидку та модульну архітектуру, яка включає весь процес конфігурації системного моделювання, кінематики, керування та візуалізації [10].

Ця програма імітаційного моделювання пропонує:

- 3D інтерактивний візуалізатор.

Готові візуальні демонстрації в Rviz дозволяють новим користувачам експериментувати з різними алгоритмами планування навколо перешкод [11];

- симуляція альтанки.

Прискорений цикли розробки та тестування, використовуючи повний фізичний симулятор із MoveIt. Поєднання Gazebo, ROS Control і MoveIt для потужної платформи для розробки робототехніки [12];

- конструктор завдань [13], [14].

Гнучкий і прозорий спосіб визначення та планування дій, які складаються з кількох взаємозалежних підзавдань [15].

Але ці аналоги мають свої недоліки та складності у використанні, наприклад відсутність 3D простору у першого та складність використання другого.

Пропонує розробка програми імітаційного моделювання мобільного робота схвата з простим та зрозумілим інтерфейсом з використанням сучасних платформ та мови програмування. Буде використовуватися мова програмування C# а також фреймворк Unity 2021.

На відміну від аналогів дана система імітаційного моделювання буде використовувати 3D простір для переміщення мобільного робота схвата, набір вже

готових команд управління робота, за допомогою візуальних кнопок екрані програми [16].

Можна побачити, що систем для імітаційного моделювання насправді не так вже й багато, що лише підкреслює актуальність обраної теми.

1.4 Дослідження методів розробки мов імітаційного моделювання

Мова імітаційного моделювання – це спеціалізований мова програмування для розробки моделей і проведення імітаційних експериментів на ЕОМ. В даний час мови імітаційного моделювання, як правило, вбудовуються в програмні середовища, призначені для розробки і прогону імітаційних моделей [17].

На всіх інших рівнях, крім системного, задана структура об'єкта однозначно дає її математичну модель, яка виходить автоматично за моделями компонентів і топологічних зв'язків. Однак реалізація тих чи інших зв'язків на системному рівні залежить від програмного забезпечення, яке на структурній схемі ніяк не відображається. Таким чином, необхідно перетворювати структуру на кожному кроці її функціонування так, щоб відображалися тільки активізовані зв'язки між пристроями, тобто можна говорити про змінну структуру досліджуваної системи (ДС).

Це значно ускладнює підготовку завдання для моделювання на електронній обчислювальній машині (ЕОМ) і на відміну від інших рівнів робить її двоетапною:

- на першому етапі розробляється математична модель ДС, яка зазвичай не є формальною і виконується людиною (схема, алгоритм);

- другий етап полягає в поданні результатів першого етапу на мові систем імітаційного моделювання.

До теперішнього часу виділяють мови безперервного і дискретного моделювання:

- алгоритмічні мови.

Переваги:

- необхідно знання тільки однієї мови алгоритмічного;

- необхідно лише стандартне програмне забезпечення (ПО);
- більша гнучкість розробки програм, а також налагодження і використання моделі.

Недоліки: незручність описів імітаційних моделей, відсутність лаконічних засобів для операцій з чергами, управління часом, синхронізації процесів в моделі, трудомісткість програмування, складність внесення змін.

- розширення алгоритмічних мов (SIMSCRIPT, SIMULA, СЛЕНГ).

(Алгоритмічні мови, розширені бібліотеками підпрограм для роботи з типовими операціями імітаційного моделювання).

- загальноцільові мови імітаційного моделювання (GPSS, НЕДИС).

Переваги: спеціально орієнтовані на імітаційне моделювання та зручність, мова відповідає визначенням і термінам систем масового обслуговування (СМО), це не просто високорівневі мови – це системи моделювання.

Недоліки: кілька великі обчислювальні витрати в порівнянні з компіляторами алгоритмічних мов, необхідність супроводу на ЕОМ додаткової системи, необхідність вивчення додаткової мови, часто з менш ретельно відпрацьованою документацією, спеціалізовані мови (МПЛ / ВС, CSS, OASIS).

Метою розробки цього класу є подальше підвищення ефективності імітаційного моделювання спеціально для ДС.

Переваги: орієнтація не так на СМО, а на ДС, включають ПО як частина ДС, служать і для налагодження алгоритмів операційних систем ЕОМ.

Недолік: застосування цих мов обмежена тільки завданнями моделювання ДС [18].

Мови графічного моделювання використовують діаграмну техніку з іменованими символами, які представляють концепції та лінії, що зв'язують символи та надають співвідношення і різні інші графічні позначення, для подання обмежень.

Мови текстового моделювання, можуть використовувати стандартизовані ключові слова, супроводжувані параметрами або термінами та фразами природною мовою, щоб утворити вирази, які визначаються комп'ютером.

Прикладом мови графічного моделювання та відповідної мови текстового моделювання, є EXPRESS.

Не всі мови моделювання є виконуваними, і для тих, хто їх використовує, це не обов'язково означає, що програмісти більше не потрібні. Навпаки, виконувані мови моделювання, призначено для посилення продуктивності кваліфікованих програмістів, аби вони могли розв'язувати складніші проблеми, такі як паралельні обчислення та розподілені системи;

– графічні типи.

Приклади мов графічного моделювання у галузі інформатики, керування проектами та системної інженерії:

Поведінкове Дерево – це формальна, графічна мова моделювання, що використовується в основному у системах та програмній інженерії. Зазвичай застосовується, щоб однозначно представляти сотні або навіть, тисячі вимог до природних мов, які зазвичай вживаються для вираження потреб зацікавлених сторін, щодо всеохопної інтегрованої системи програмного забезпечення.

Позначення моделювання бізнес-процесів (BPMN та XML форма BPML) – це приклад мови моделювання процесу.

Послідовно-керований технологічний ланцюжок (EPC) – це блок-схема для моделювання бізнес-процесів. EPC можна використовувати для налаштування системи планування ресурсів підприємства і поліпшення бізнес-процесів.

Теорія С-К – складається з мови моделювання для процесів проектування.

DRAKON – це мова алгоритмічного моделювання загального призначення для визначення програмно-інтенсивних систем, схематичного представлення алгоритму або поетапного процесу, та сімейство мов програмування.

EXPRESS та EXPRESS-G (ISO 10303-11) – це міжнародна стандартна мова моделювання даних загального призначення.

Мова розширеного корпоративного моделювання (EEML) – зазвичай використовується для моделювання бізнес-процесів на декількох рівнях.

Блок-схема – являє собою схематичне зображення алгоритму або поетапного процесу.

Основи моделювання фундаментальних концепцій (FMC) для програмно-інтенсивних систем.

IDEF – це сімейство мов моделювання, яке містить IDEF0 для функціонального моделювання, IDEF1X для інформаційного моделювання, IDEF3 для моделювання бізнес-процесів, IDEF4 для об'єктно-орієнтованого програмування та IDEF5 для моделювання онтологій.

Jackson Structured Programming (JSP) – це спосіб структурного програмування на основі відповідності структури потоку даних і структури програми.

LePUS3 – це об'єктно-орієнтований візуальний дизайн опису мови та формальна мова специфікації, яка підходить, перш за все, для моделювання великих об'єктно-орієнтованих (Java, C ++, C #) програм і шаблонів проектування.

Object-Role Modeling (ORM) у галузі розробки програмного забезпечення – це метод концептуального моделювання і може бути використаний як інструмент аналізу інформації та правил.

Мережі Петрі – використовують варіації лише на одному методі діаграм і топології, а саме на двочастковому графі. Простота його базового інтерфейсу користувача, легко забезпечувала широку підтримку інструментів протягом багатьох років, особливо стосовно перевірки моделі, графічного моделювання та дослідження програмного забезпечення.

Southbeach Notation – це мова візуального моделювання, що використовується для опису ситуацій з точки зору агентів, які вважаються корисними чи шкідливими з боку модельєра. Позначення показують, як агенти співдіють один з одним і чи ця взаємодія покращує або погіршує становище.

Мова специфікації та опису (SDL) – це мова специфікації, спрямована на однозначну специфікацію та опис поведінки реактивних і розподілених систем.

SysML – це мова моделювання специфічна для домену системотехніки, яка визначається як профіль UML (налаштування).

Уніфікована мова моделювання (UML) – це мова моделювання загального призначення, яка є галузевим стандартом для визначення програмно-інтенсивних

систем. UML 2.0, поточна версія, підтримує тринадцять різних діаграмних способів і має всеохопну підтримку інструментів.

Сервісно-орієнтована модельна структура (SOMF) – це цілісна мова для проектування моделей архітектури корпоративного рівня та рівня додатків у галузі архітектури підприємств, віртуалізації, сервіс-орієнтованої архітектури (SOA), хмарних обчислень тощо

Мова опису архітектури (ADL) – це мова, яка використовується для опису та представлення устрою архітектури системи.

AADL (AADL) – це мова моделювання, яка підтримує ранні та повторювані аналізи архітектури системи, відносно критично важливих властивостей, за допомогою розширюваних позначень, структур інструментів та точно визначеної семантики.

Приклади мов графічного моделювання в інших галузях науки.

EAST-ADL – це спеціальна мова моделювання, присвячена проектуванню автомобільних систем.

Мова енергетичних систем (ESL) – мова, яка спрямована на моделювання відновлюваної енергетики та світової економіки;

– текстові види.

Інформаційні моделі також, може бути виражено формалізованими природними мовами, такими як Gellish. Gellish має варіанти з природними мовами, такими як формальна англійська Gellish і формальна голландська Gellish (Gellish Formeel Nederlands) тощо. Gellish Formal English – це мова представлення інформації, або мова семантичного моделювання, яка визначається в англійському словнику-таксономії Gellish (Gellish English Dictionary-Taxonomy), що має форму таксономії-онтології (аналогічно для голландців). Gellish Formal English, не лише підходить для виразу знань, вимог і словників таксономій та онтологій, а також для інформації про окремі речі. Всі ці дані виражаються однією мовою і тому можуть бути інтегровані, незалежно від того, чи зберігаються вони у центральних, розподілених або у федеративних базах даних. Інформаційні моделі у Gellish

Formal English, складено з набору виразів Gellish Formal English, які використовують терміни з природною мовою та формалізовані вирази.

В галузі інформатики згодом з'явилися більш конкретні типи мов моделювання:

– алгебраїчна.

Мови алгебраїчного моделювання (AML) – це мови програмування високого рівня для опису та вирішення завдань високої складності для громіздких математичних обчислень. Однією з особливих переваг AML, таких як: AIMMS, AMPL, GAMS, LPL, MPL, OPL і OptimJ, є подібність їх синтаксису з математичним позначенням завдань оптимізації. Це дозволяє отримати дуже стисле та зрозуміле визначення проблем у галузі оптимізації, яке підтримується деякими мовними елементами, такими як набори, показники, алгебраїчні вирази, потужні розріджені змінні індексів і обробки даних, обмеження з довільними іменами. Алгебраїчне формулювання моделі не містить ніяких підказок, як його обробляти;

– поведінкова.

Поведінкові мови призначені для опису спостережуваної поведінки складних систем, що збираються зі складових, які виконуються одночасно. Ці мови зосереджено на описі ключових визначень, таких як: паралелізм, недетермінізм, синхронізація та зв'язок. Семантичні основи поведінкових мов – це обчислення або алгебра процесів;

– предметно-специфічна.

Мова моделювання, заснована на дисципліні (DspM), орієнтована на підсумки, пов'язані з проміжком життєвого циклу певної розробки програмного забезпечення. Таким чином, ця мова пропонує відмінну лексику, синтаксис та позначення для кожного етапу, такі як: виявлення, аналіз, дизайн, архітектура, стиснення тощо. Наприклад, щодо етапу аналізу проекту, модельєр використовує специфічну нотацію аналізу для доправлення діаграми аналітичної пропозиції. Однак, на етапі проектування, логічне позначення дизайну, використовується для опису взаємодії між об'єктами програмного забезпечення. Крім того, практичні

поради з мови моделювання, специфічні для конкретної дисципліни, не перешкоджають практикам поєднувати різні позначення на одній діаграмі;

– предметно-орієнтовані.

Моделювання доменних імен (DSM) – це методологія розробки програмного забезпечення для проєктування і створення систем – найчастіше це ІТ-системи, такі як комп'ютерне програмне забезпечення. Це передбачає систематичне використання графічної мови, специфічної для домену (DSL), щоби представляти різні грані системи. Мови DSM, мають властивість підтримувати абстракції вищого рівня, ніж мови моделювання загального призначення, тому вони потребують менше зусиль та деталей низького рівня, для позначення даної системи [19].

1.5 Постановка завдань дослідження

У ході проведення досліджень були виявлені недоліки існуючих систем, такі як велика вартість системи, роздрібленість систем або їх застарілість. В наслідок чого пропонується створити нову доступну та сучасну систему імітаційного моделювання та нову мову імітаційного моделювання на основі мови С# та фреймворку Unity.

Об'єкт дослідження – системи імітаційного моделювання роботів.

Предмет дослідження – моделі, методи та мови імітаційного моделювання.

Мета дослідження – розробка системи імітаційного моделювання переміщення мобільного робота маніпулятора.

Методи дослідження – алгоритми, патерни та принципи програмування, мова програмування С#, фреймворк Unity.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз сучасних систем імітаційного моделювання мобільних роботів;
- провести аналіз існуючих мов імітаційного моделювання;
- провести розробку базових команд для керування мобільним роботом схватом.

2 РАЗРОБКА МОВИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ

2.1 Розробка базових команд управління схвата

Для керування імітаційною моделлю робота схвата слід розробити базові команди керування схватом:

- Команда `Move _coords(x, y)` – задає координати руху мобільного робота схвата, де `_coords(x,y)` – координати кінцевої точки руху;
- команда `Rotate _angle` – задає кут оберту мобільного робота схвата;
- команда `RotateX _armLink _angle` – задає кут обертання ланки руки робота по осі X, де `_armLink` – номер звена, `_angle` – кут оберту;
- команда `RotateY _armLink _angle` – задає кут обертання ланки руки робота по осі Y, де `_armLink` – номер звена, `_angle` – кут оберту;
- команда `CloseTheGrip` – закриває схват;
- команда `OpenTheGrip` – розкриває схват.

Це перелік необхідних базових команд для повноцінного імітаційного моделювання мобільного робота схвата. Для використання цих команд далі буде розроблено 2 варіанти використання. Перший варіант буде виглядати як вікно вводу тексту для написання коду. Синтаксис мови програмування для імітаційного моделювання мобільного робота схвата буде розроблено далі.

Також буде розроблено можливість завантажувати код програми за допомогою простого текстового файлу. Другий варіант буде виглядати як набір віртуальних кнопок на інтерфейсі користувача, для спрощення моделювання без використання кода. Усі базові команди будуть реалізовані на інтерфейсі користувача.

2.2 Розробка синтаксису мови

Визначемо поняття мовної моделі (ММ) – це декларативна (не процедурна) мова, призначенням якої є визначення та опис термінології.

В данній магістерській роботі запропонована наступна специфікація мови моделей даних:

– дозволені буквено-цифрові символи: - , : ; пробіл;

– зарезервовані назви функцій: Move, Rotate, RotateX, RotateY, OpenTheGrip, CloseTheGrip

У ході написання коду назву функції та параметри треба відмежовувати за допомогою символу :, наприклад назва_функції: параметри_функції, а закінчувати строку слід символом ; (рисунок 2.1).

```
Move: -12, -12;  
Rotate: 60;  
RotateX: 1, 20;  
RotateY: 1, 20;  
OpenTheGrip: 0;  
CloseTheGrip: 0;
```

Рисунок 2.1 – Приклад написаної програми

На рисунку 2.1 зображено приклад написаної програми, що виконує наступні дії:

- рухає робота до координат -12 по X та -12 по Y;
- розвертає робота на 60 °;
- розвертає першу ланку робота на 20 ° по вісі X;
- розвертає першу ланку робота на 20 ° по вісі Y;

- відкриває схват;
- закриває схват.

2.3 Висновки до 2-го розділу

Після проведеної розробки базового функціоналу мови імітаційного моделювання можна зробити висновок, що дана реалізація підходить для моделювання простих процесів на виробництві за участю мобільних роботів маніпуляторів.

3 РОЗРОБКА СЕРЕДОВИЩА ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ

3.1 Вибір середовища розробки

Проаналізувавши сучасні технології, мови програмування та середовища розробки було прийнято рішення обрати як мову програмування мову C#, як середовище розробки JetBrains Rider та як фреймворк для 3-D рішень Unity 2022.

C# – об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтанутом та Пітером Гольде під егідою Microsoft Research (належить Microsoft).

Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Переїнявши багато від своїх попередників – мов C++, Object Pascal, Модула і Smalltalk – C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад, мова C#, на відміну від C++, не передбачає множинне успадкування класів [20].

Rider – це крос-платформна IDE для .NET-розробників, заснована на платформі IntelliJ та ReSharper.

Підтримка різних .NET-проектів

Rider підтримує .NET Framework, нову платформу .NET Core та проекти на основі Mono. IDE дозволяє розробляти десктопні програми, .NET-сервіси та бібліотеки, ігри на движку Unity, мобільні програми Xamarin, веб-додатки ASP.NET та ASP.NET Core.

Швидкість та функціональність

Rider надає понад 2200 інспекцій коду, сотні контекстних дій та рефакторингів, запозичених із ReSharper, у поєднанні з просунутою

функціональністю середовищ розробки на основі платформи IntelliJ. Незважаючи на великий набір функцій, Rider – швидка та чуйна IDE.

Крос-платформна IDE

Крім уміння запускати та налагоджувати різні програми в різних операційних системах, Rider сам по собі теж підтримує крос-платформність і працює на Windows, macOS та Linux [21].

Unity – багатоплатформний інструмент для розробки відеоігор і застосунків, і рушій, на якому вони працюють. Створені за допомогою Unity програми працюють на настільних комп'ютерних системах, мобільних пристроях та гральних консолях у дво- та тривимірній графіці, та на пристроях віртуальної чи доповненої реальності. Застосунки, створені за допомогою Unity, підтримують DirectX та OpenGL.

3.2 Розробка елементів 3D візуалізації промислового маніпулятора

3D візуалізація промислового маніпулятора була розроблена за допомогою базових 3D блоків, що доступні для використання у середі UNITY одразу після запуску, а саме за допомогою примітивних кубів, сфер та циліндрів (рис. 3.1).



Рисунок 3.1 – 3D візуалізація промислового робота маніпулятора

3.3 Розробка алгоритмів роботи системи імітаційного моделювання

Було розроблена алгоритм роботи програми як показано на рисунку 3.2.

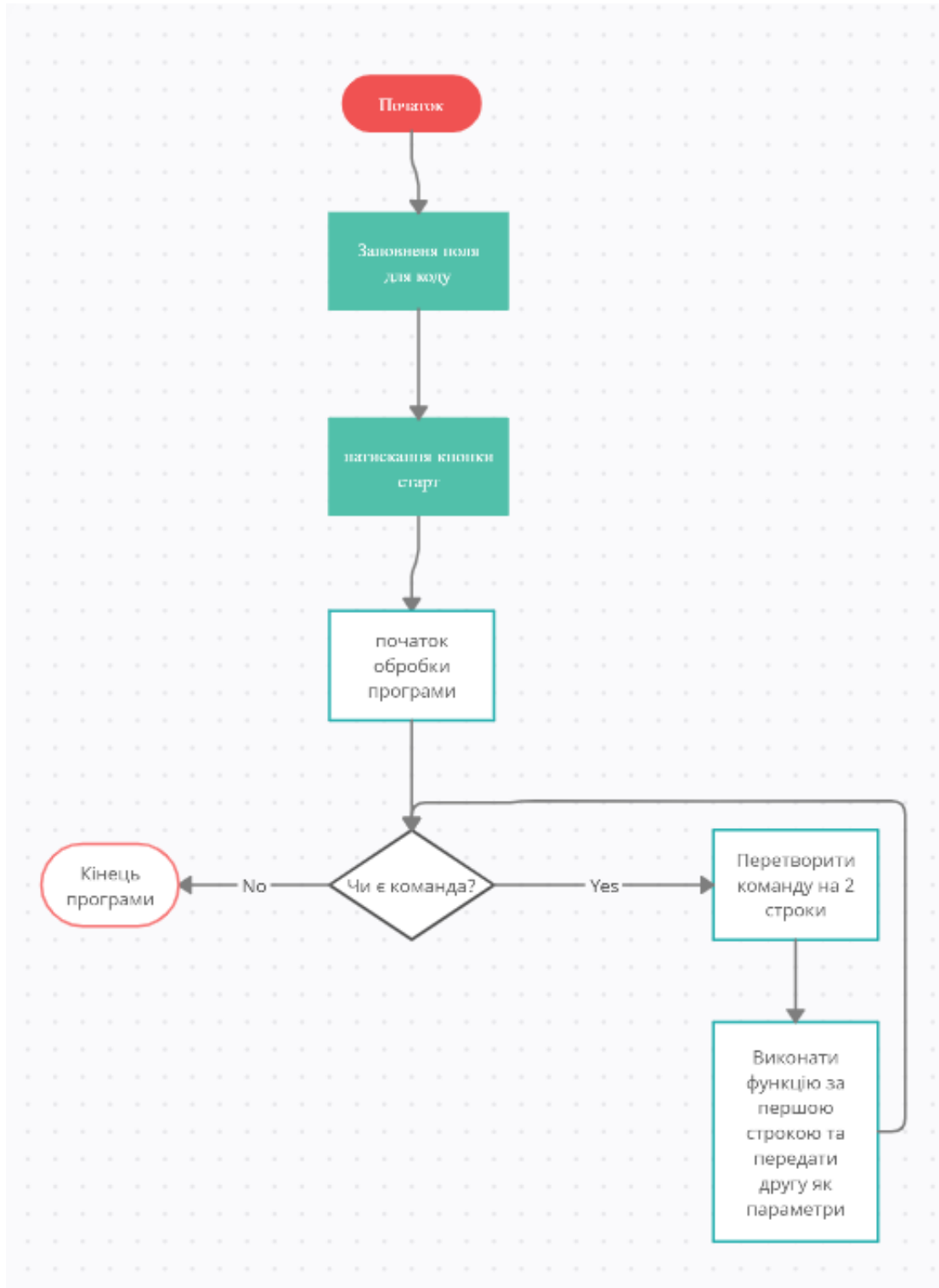


Рисунок 3.2 – Алгоритм роботи програми

3.4 Розробка функцій керування роботом

У ході роботи було розроблено наступні команди керування, що приведені на рисунках 3.3 – 3.5, описання цих команд можна побачити у розділі 2.1:

```
public void Move(string param)
{
    var q = param.Split(',');
    platform.transform.DOMove(new Vector3(Convert.ToInt32(q[0].Trim(' ')),
        platform.transform.position.y ,Convert.ToInt32(q[1].Trim(' '))), 3);
    Debug.Log(Convert.ToInt32(q[0]));
    Debug.Log(Convert.ToInt32(q[1]));
}
```

Рисунок 3.3 – Команда Move

```
public void Rotate(string param)
{
    var rotation = platform.transform.rotation;
    platform.transform.DORotate(new Vector3(rotation.x,
        rotation.y + Convert.ToInt32(param.Trim(' ')), rotation.y), 3);
    Debug.Log(Convert.ToInt32(param.Trim(' ')));
}
```

Рисунок 3.4 – Команда Rotate

```
public void RotateX(string param)
{
    var q = param.Split(',');
    var a = arms[Convert.ToInt32(q[0].Trim(' '))-1];
    var rotation = a.transform.rotation;
    a.transform.DOLocalRotate(new Vector3(rotation.x + Convert.ToInt32(q[1].Trim(' ')),
        rotation.y , rotation.z), 3);

    Debug.Log(Convert.ToInt32(q[0]));
    Debug.Log(Convert.ToInt32(q[1]));
}
```

Рисунок 3.5 – Команда RotateX

Розроблені команди керування, приведені на рисунках 3.6 – 3.8, описання цих команд можна побачити у розділі 2.1:

```
public void RotateY(string param)
{
    var q = param.Split(',');
    var a = arms[Convert.ToInt32(q[0].Trim(' '))-1];
    var rotation = a.transform.rotation;
    a.transform.DOLocalRotate(new Vector3(rotation.x ,
        rotation.y + Convert.ToInt32(q[1].Trim(' ')), rotation.z), 3);

    Debug.Log(Convert.ToInt32(q[0]));
    Debug.Log(Convert.ToInt32(q[1]));
}
```

Рисунок 3.6 – Команда RotateY

```
public void OpenTheGrip(object obj)
{
    grips[0].transform.DOLocalMove(new Vector3(1, 1.3f, 0), 1);
    grips[1].transform.DOLocalMove(new Vector3(-1, 1.3f, 0), 1);
}
```

Рисунок 3.7 – Команда OpenTheGrip

```
public void CloseTheGrip(object obj)
{
    grips[0].transform.DOLocalMove(new Vector3(0.5f, 1.3f, 0), 1);
    grips[1].transform.DOLocalMove(new Vector3(-0.5f, 1.3f, 0), 1);
}
```

Рисунок 3.8 – Команда CloseTheGrip

3.5 Розробка інтерфейсу користувача системи імітаційного моделювання

У ході роботи було розроблено програму для імітаційного моделювання та додано графічний інтерфейс як показано на рисунку 3.9.

A screenshot of a code input field. The field is a white rectangle with a thin black border, set against a background with a blue gradient on the right side. The text inside the field is as follows:

```
Move: -12, -12;  
Rotate: 60;  
RotateX: 1, 20;  
RotateY: 1, 20;  
OpenTheGrip: 0;  
CloseTheGrip: 0;
```

Рисунок 3.9 – Поле для вводу кода

Розроблено програму для імітаційного моделювання та додано графічний інтерфейс як показано на рисунках 3.10 – 3.11.



Рисунок 3.10 – Кнопка початку обробки коду



Рисунок 3.11 – Інформація о координатах положення робота

Розроблено програму для імітаційного моделювання та додано графічний інтерфейс як показано на рисунках 3.12.

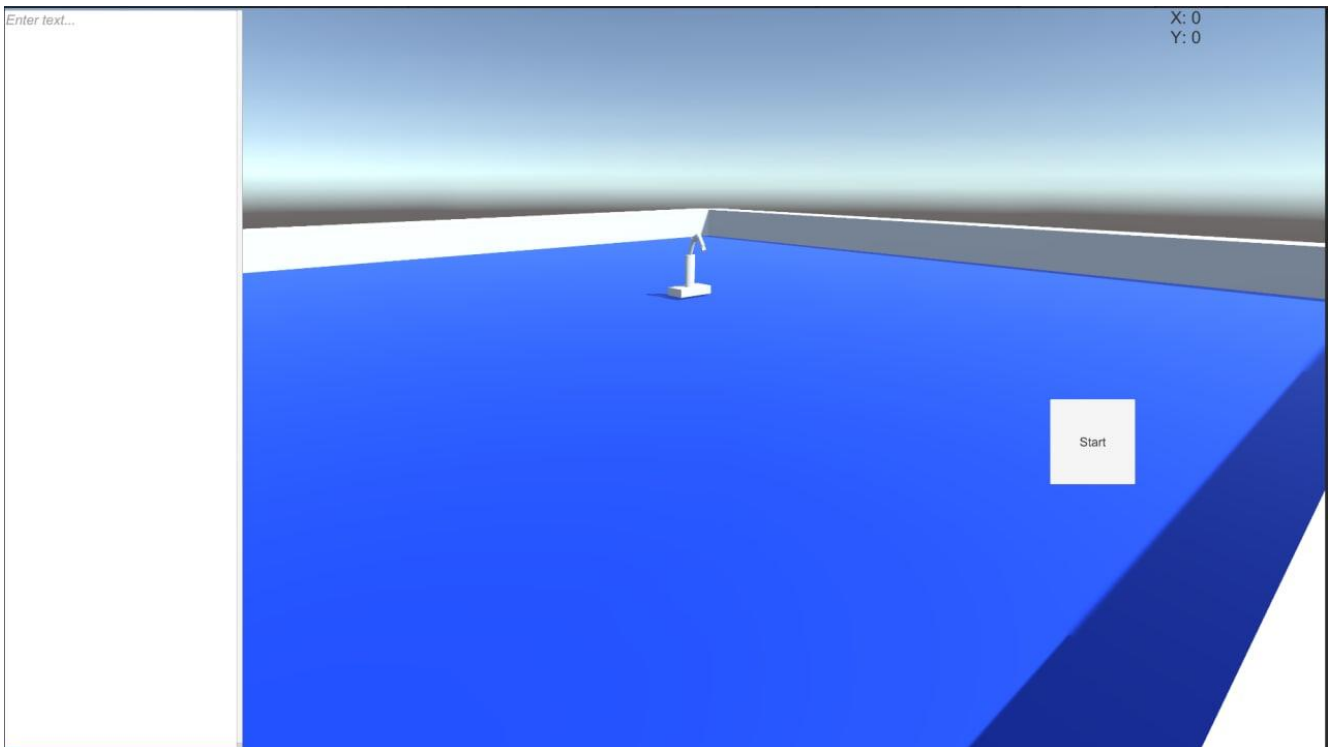


Рисунок 3.12 – Повний інтерфейс програми

3.6 Висновки до 3 розділу

У ході роботи було розроблено програму імітаційного моделювання промислового робота схвата та розроблення мови для проведення імітаційного моделювання.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Постановка завдання експерименту

Проведемо тестування системи та проведемо імітаційне моделювання мобільного робота маніпулятора у розробленому середовищі. Робот рухається у межах платформи розміром 50 на 50 умовних одиниць. Будемо проводити моделювання з використання спочатку простих команд по одній, після чого розробимо програму, за якою буде рухатись робот. Результати експерименту наведено у пункті 4.2.

4.2 Результати проведеного експерименту

На рисунку 4.1 представлено початкове положення мобільного робота.

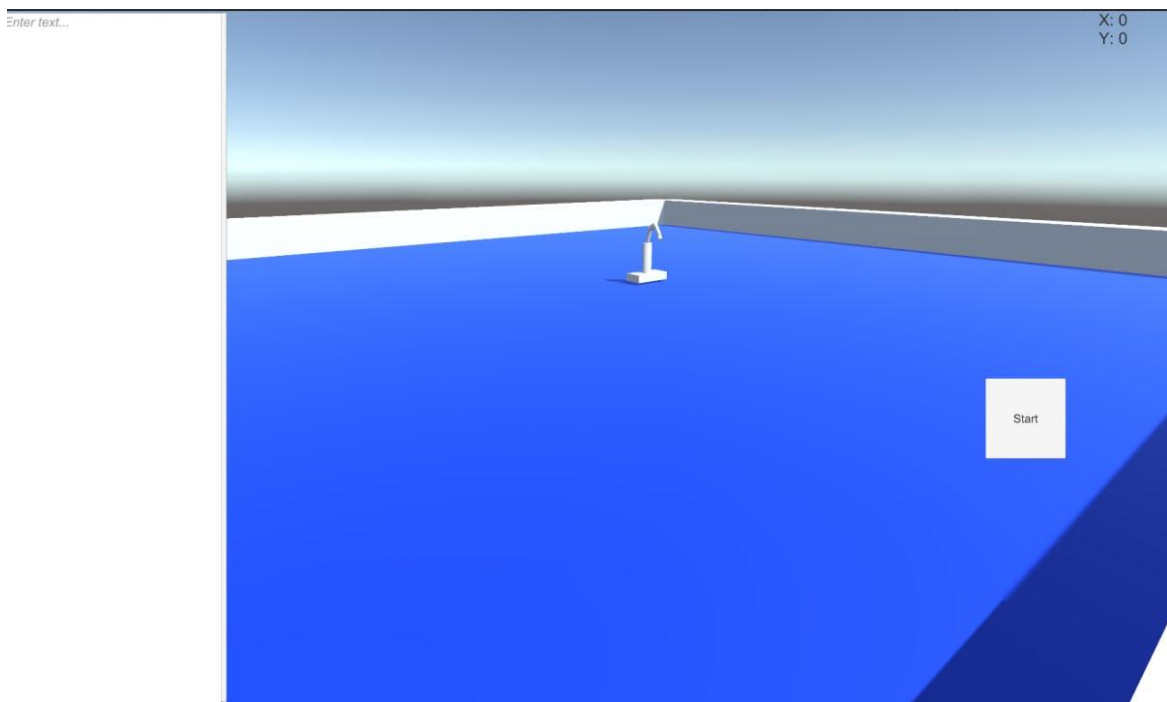


Рисунок 4.1 – Початкове положення мобільного робота

На рисунку 4.2 та 4.3 представлений результат виконання команди Move та команди Rotate.

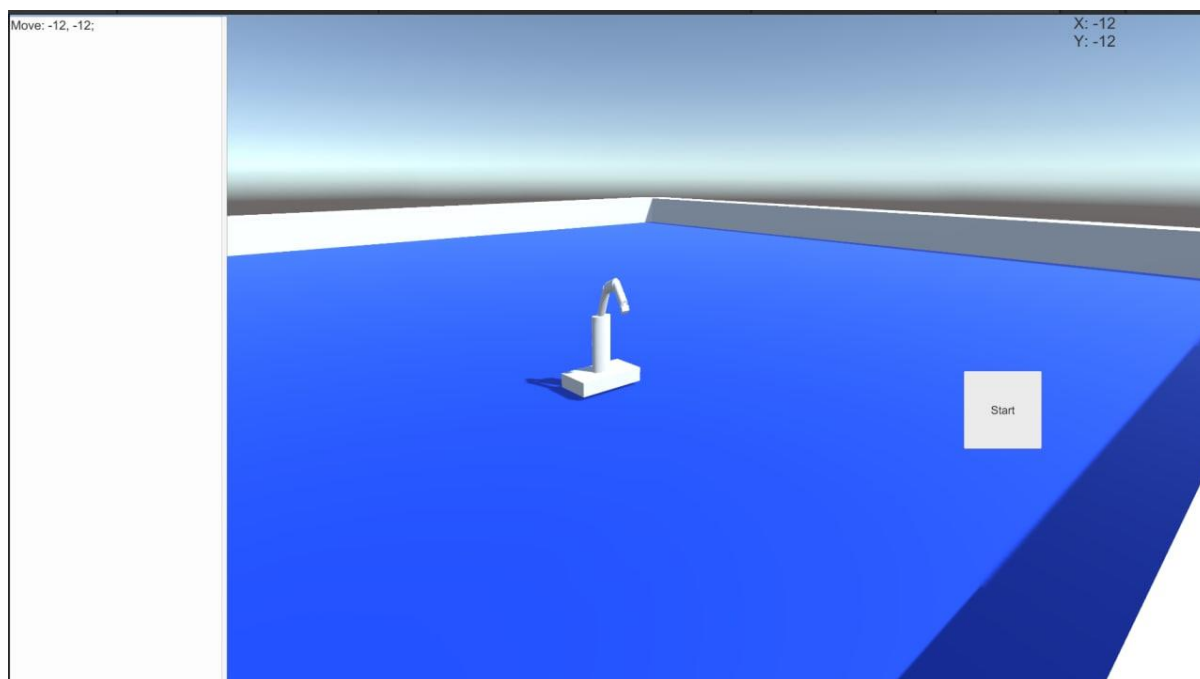


Рисунок 4.2 – Результат виконання команди Move

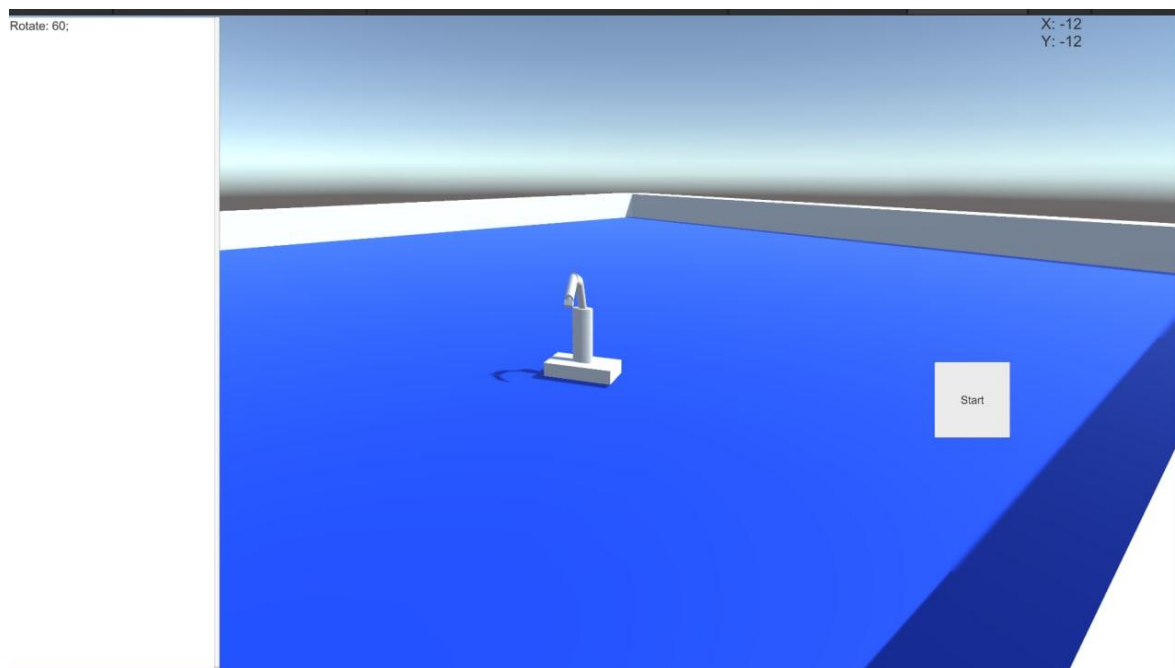


Рисунок 4.3 – Результат виконання команди Rotate

На рисунку 4.4 та 4.5 представлений результат виконання команди RotateX на першому звені та команди RotateY на першому звені.

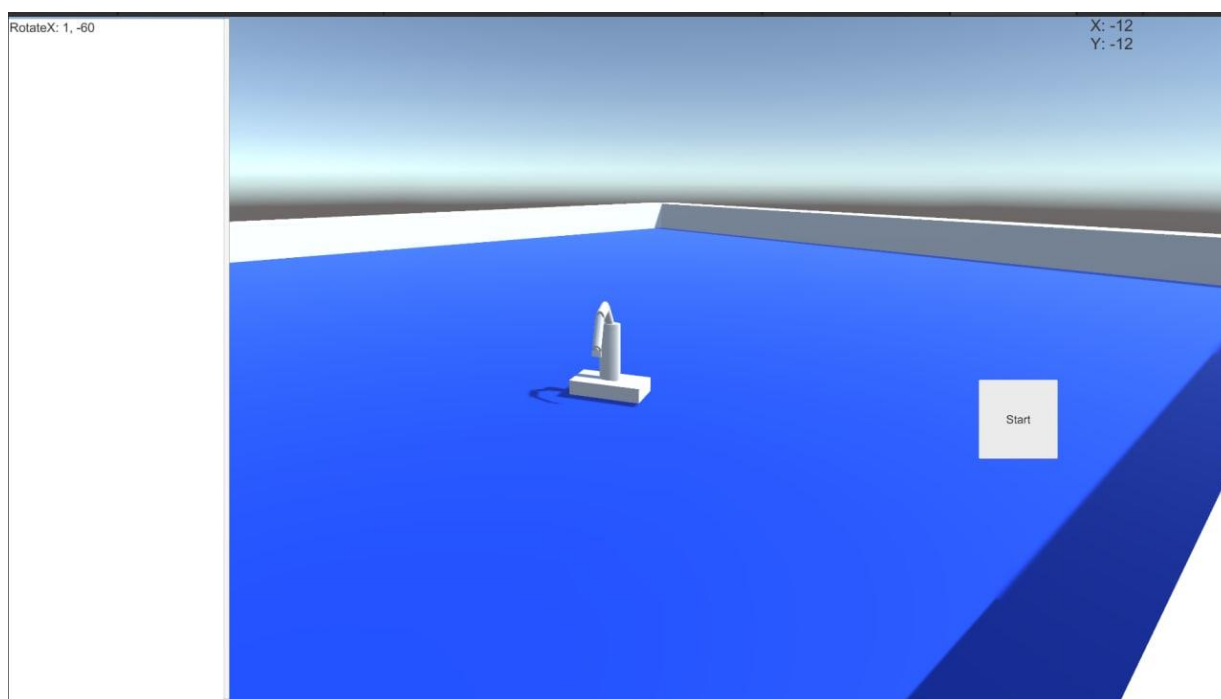


Рисунок 4.4 – Результат виконання команди RotateX на першому звені

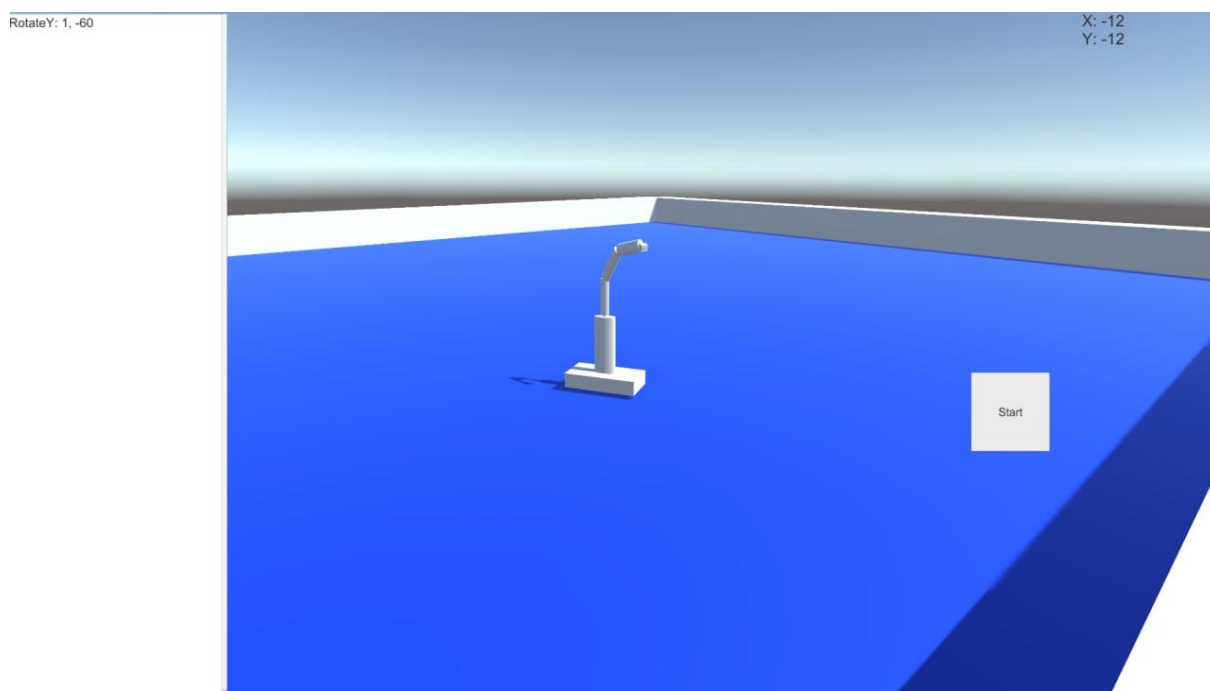


Рисунок 4.5 – Результат виконання команди RotateY на першому звені

На рисунку 4.6 та 4.7 представлений результат виконання команди OpenTheGrip та команди CloseTheGrip.

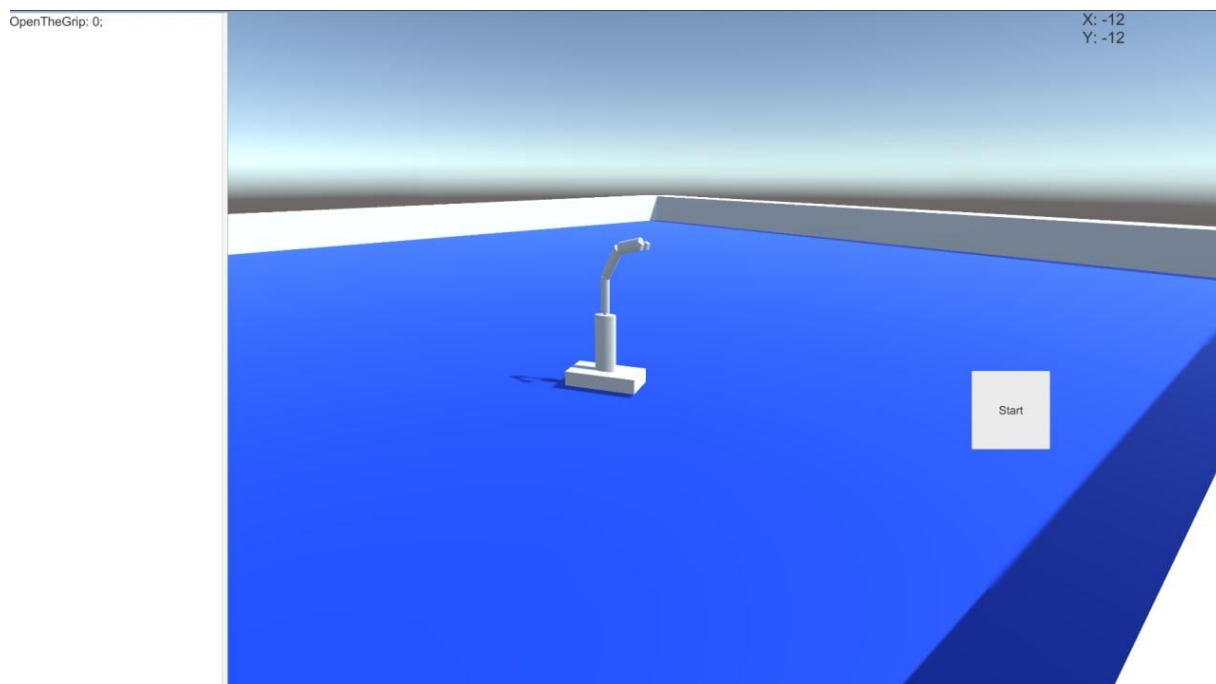


Рисунок 4.6 – Результат виконання команди OpenTheGrip

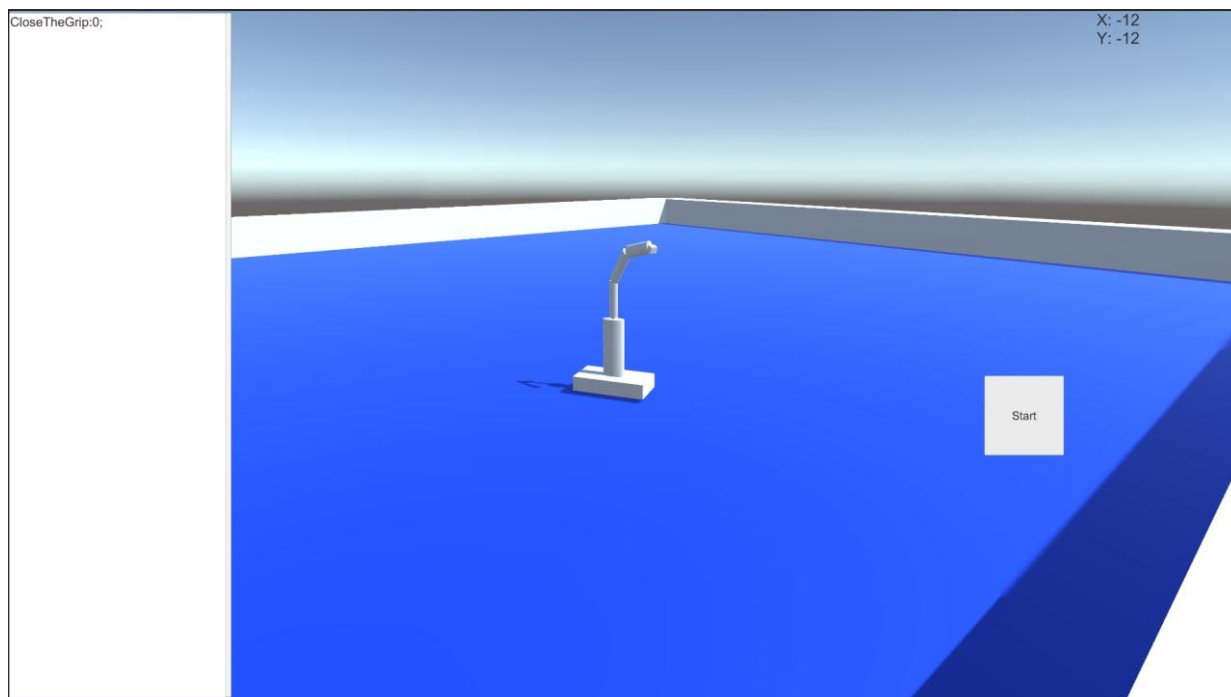


Рисунок 4.7 – Результат виконання команди CloseTheGrip

На рисунку 4.8 представлена написана програма роботи мобільного робота. На рисунку 4.9 представлений результат виконання написаної імітаційної програми.

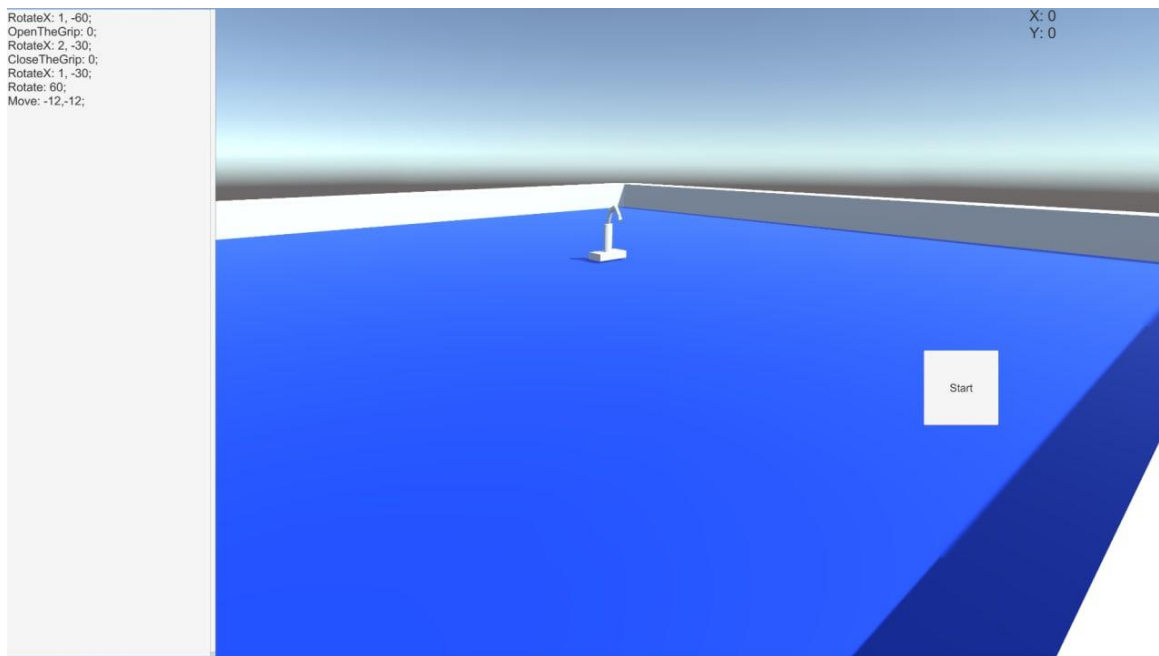


Рисунок 4.8 – Написана програма роботи мобільного робота

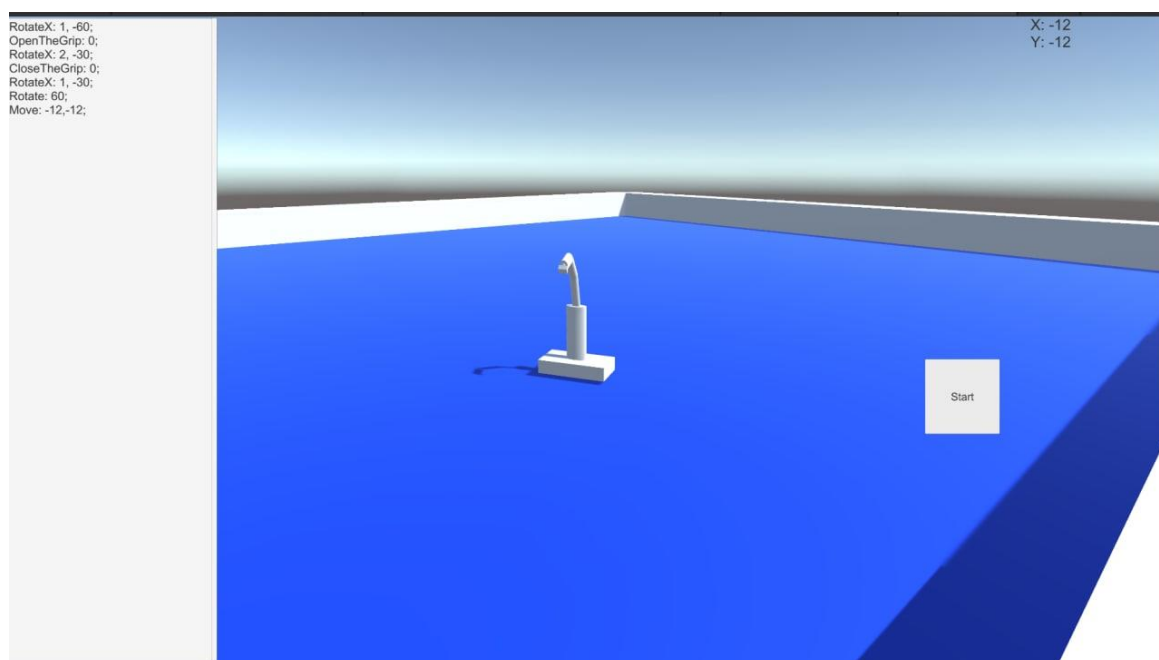


Рисунок 4.9 – Результат виконання написаної імітаційної програми

Тепер проведемо експеримент по енергозатратам на рух по різних траєкторіях, минаючи перешкоду (рис.4.10).

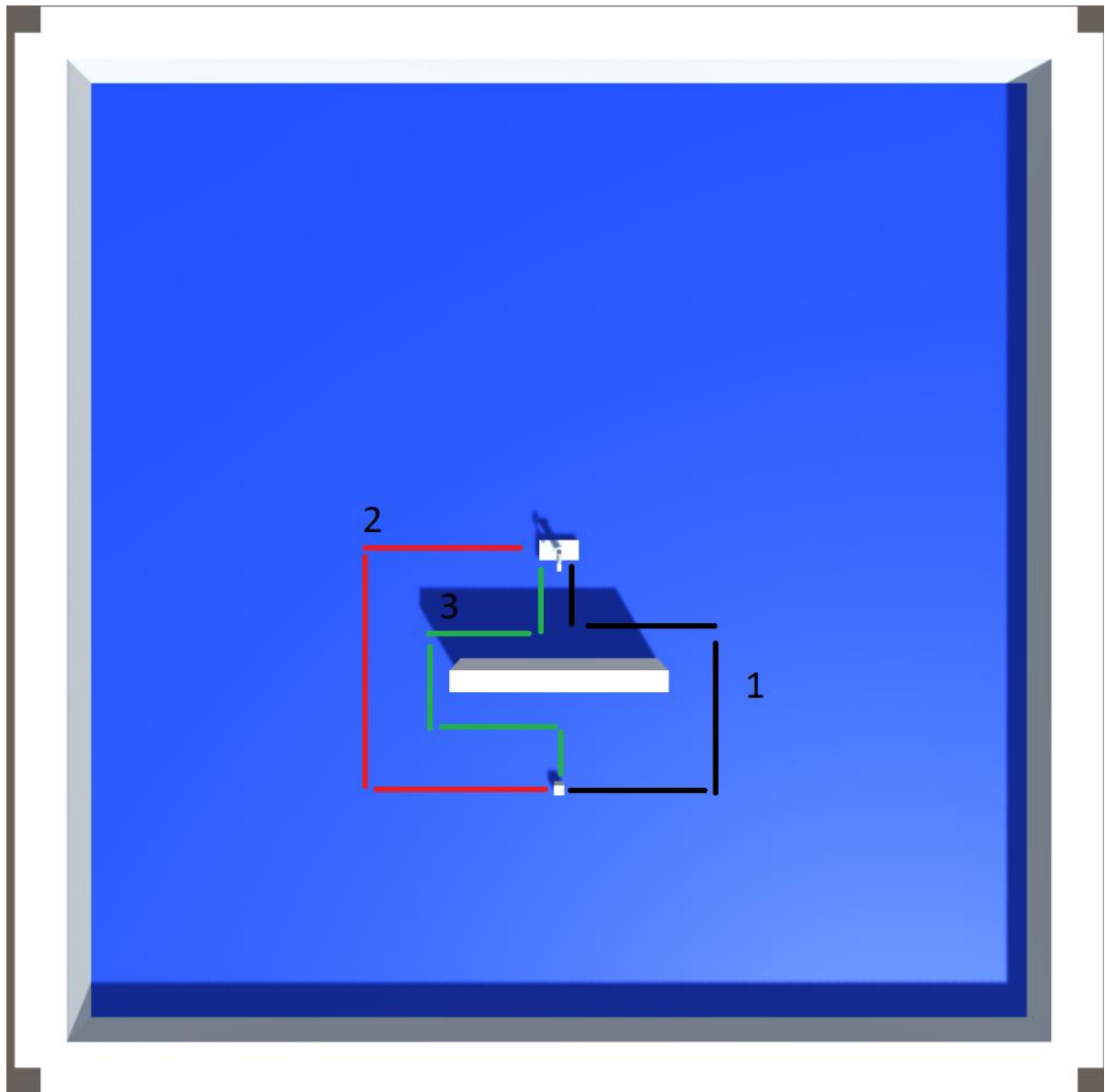


Рисунок 4.10 – Траєкторії руху робота повз перешкоду

Розрахунок та аналіз енергозатрат робота на рух по різних траекторіях у різний момент часу представлено на діаграмі на рисунку 4.11.

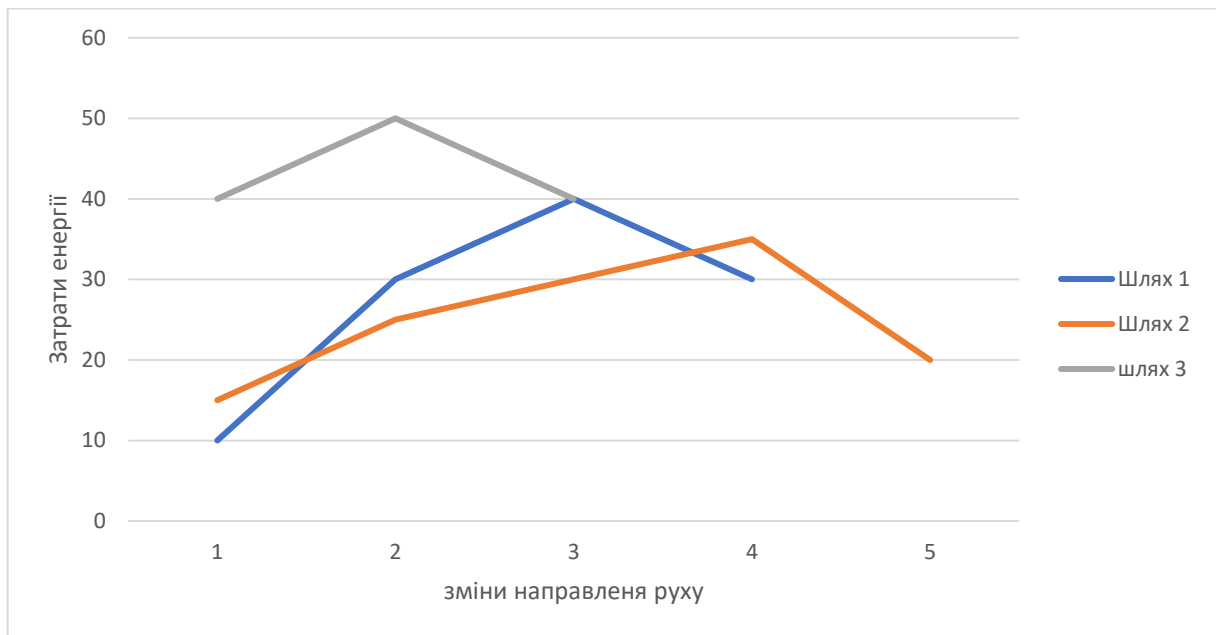


Рисунок 4.11 – Графік затрат енергії за отриманими результатами

Провівши дослідження та аналіз, можемо побачити, що найбільш ефективним шляхом виявився шлях номер три.

4.3 Охорона праці

Роботи в лабораторії можна віднести до робіт категорії 1б – легка фізична робота, вироблена сидячи, стоячи, пов'язана з ходьбою і супроводжується деяким фізичним напруженням.

Оптимальні норми мікроклімату згідно з ДСН 3.3.6.042-99

– температура: від 22 °С до 24 °С (при температурі зовнішнього повітря нижче + 10 °С) і від 23 °С до 25 °С (при температурі зовнішнього повітря вище + 10 °С);

– відносна вологість: від 40 % до 60 %;

– швидкість руху повітря: не більше 0,1 м / с.

Для підтримки параметрів мікроклімату в межах норм в теплий період застосовується кондиціонування повітря, в холодний період - опалення.

Так як в приміщенні проводиться робота з персональними комп'ютерами, природне світло повинне проникати через бічні світлові прорізи, зорієнтовані на північ. Вікна мають регулюють пристосування для відкривання, жалюзі.

Для розрахунку природного освітлення необхідно визначити площу віконних прорізів, яка забезпечить нормоване значення КПО. Потім необхідно порівняти розраховану площу з реальною. Умови праці вважаються нормальними, якщо виконується співвідношення (4.1):

$$S_{ок} \leq S_{реал}, \quad (4.1)$$

де $S_{ок}$ – площа вікон;

$S_{реал}$ – реальне значення площі вікон в приміщенні.

Необхідну площу світлового прорізу, при бічному природному освітленні обчислюється за формулою (4.2):

$$S_{окон} = \frac{e_n \cdot \eta_0 \cdot K_{зл} \cdot K_3 \cdot S_{пола}}{100 \cdot \tau_{общ} \cdot r_1}, \quad (4.2)$$

де $S_{окон}$ – площа світлових прорізів, яка забезпечить нормоване значення КПО в приміщенні;

$S_{пола}$ – площа підлоги приміщення;

e_n – нормоване значення КПО для 4-го кліматичного світлового поясу, $e_n = 1,35$;

η_0 – світлова характеристика вікна;

$K_{зл}$ – коефіцієнт, що враховує затінення вікон протистоящими будинками, $K = 1$;

K_3 – коефіцієнт запасу (залежить від концентрації пилу в приміщенні і від періодичності очищення стекол), $K_3 = 1,4$;

$\tau_{\text{общ}}$ – загальний коефіцієнт світлопропускання, що визначається залежно від коефіцієнта пропускання скла, втрат світла в оправках вікна, шару його забруднення, наявності сонцезахисних конструкцій, $\tau_{\text{общ}} = 0,94$;

r_1 – коефіцієнт, що враховує підвищення КПО за рахунок відбиття.

Підставивши значення параметрів знайдемо площа світлових прорізів, що визначаються співвідношеннями (4.3), (4.4):

$$S_{\text{окон}} = \frac{1,35 \cdot 11 \cdot 1 \cdot 1,4 \cdot 27,6}{100 \cdot 0,94 \cdot 1,2} = 5,08, \quad (4.3)$$

$$S_{\text{ок}} \leq S_{\text{реал}}; 5,08 > 5., \quad (4.4)$$

Виходячи з розрахунків робимо висновок, що співвідношення (4.1) майже виконується. Для даного виробничого приміщення рівень освітленості можна вважати достатнім.

4.4 Висновки до 4 розділу

Провівши експеримент можна зробити висновок, що дана програма імітаційного моделювання виконує свої базові функції та може бути об'єктом для навчання студентів керування та процесами, пов'язаними з мобільними роботами маніпуляторами. З отриманих результатів, можемо побачити що найефективнішим шляхом є шлях номер три.

ВИСНОВКИ

Дана робота присвячена розробці мови та середовища імітаційного моделювання мобільного робота схвата у рамках Індустрії 5.0.

У першому розділі було проведено аналіз сучасних методів моделювання роботи промислового маніпулятора в Індустрії 5.0. Було проведено аналіз поняття колаборативні роботи в рамках Індустрії 5.0, аналіз конструкцій та параметрів промислового мобільного маніпулятора, аналіз систем імітаційного моделювання роботів промислових маніпуляторів, проведено дослідження методів розробки мов імітаційного моделювання.

У другому розділі було проведено розробку мови імітаційного моделювання. Було проведено розробку базових команд управління схвата та розробка синтаксису мови імітаційного моделювання.

У третьому розділі було проведено розробку середовища імітаційного моделювання. Було обрано середовище розробки та мова програмування, розроблено елементи 3D візуалізації промислового маніпулятора, розроблено алгоритм роботи системи імітаційного моделювання, розроблено можливість керування за командами, розроблено інтерфейс користувача системи імітаційного моделювання.

У четвертому розділі було проведення експеримент у ході якого було протестовано усі розроблені команди та проведено розрахунок освітлення робочого приміщення. Побудували графіки за отриманими даними та провели їх аналіз.

Весь проект написаний на мові програмування C# у середовищі та фрейворці UNITY, що у майбутньому спрощує завдання розширення функціоналу програми. Увесь код написано згідно з існуючими принципами [22] та положеннями [23], [24] програмування, наскільки це можливо у даній реалізації, що спрощує читання та редагування коду.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки з підготовки та захисту кваліфікаційної роботи [електронний ресурс]. URL: https://drive.google.com/file/d/1K1Dt-_16lnvNkU6P0McC7MxореXSfgIB/view (дата звернення: 1.11.2022).
2. Структура та правила оформлювання, ДСТУ 3008:2015 [Електронний ресурс]. URL: <https://drive.google.com/file/d/1714Epz74lVhgBKSvMobd9g2Rdt0MMDzp/view> (дата звернення: 1.11.2022).
3. Що таке Індустрія 5.0 [Електронний ресурс]. URL: <https://ik555.ru/statyi/chto-takoe-industriya-5-0-i-kak-ona-izmenit-proizvodstvo/> (дата звернення: 1.11.2022).
4. Промислові роботи KUKA ROBOTICS [Електронний ресурс]. URL: <https://web.archive.org/web/20171010131008/http://vektor-grupp.ru/shop/promyshlennye-roboty/promyshlennye-roboty-kuka-robotics/> (дата звернення: 1.11.2022).
5. Хто такі коботи, навіщо вони на виробництві та скільки коштують? [Електронний ресурс]. URL: <https://biz.nv.ua/ukr/experts/hto-taki-koboti-shcho-vmiyut-robiti-i-skilki-koshtuyut-ostanni-novini-50122600.html> (дата звернення: 1.11.2022).
6. Z Yi та S. Wenshi, "Dynamics Analysis of 6-DOF Robot based on ADAMS [C]", 2017 Міжнародна конференція з мехатронних систем і техніки керування, стор. 6-9, 2017. (дата звернення: 2.11.2022).
7. KUKA.WorkVisual [Електронний ресурс]. URL: https://www.kuka.com/en-us/products/robotics-systems/software/system-software/kuka_systemsoftware/kuka-work-visual (дата звернення: 2.11.2022).
8. Деякі типові конструкції промислових роботів [Електронний ресурс]. URL: https://studme.org/125177/tehnika/nekotorye_tipovye_konstruktsii_promyshlennyh_robotov (дата звернення: 2.11.2022).

9. Д. А. Крішнамурті, Моделювання та симуляція робота з бортовим керуванням Pioneer 3AT [J], 2017. (дата звернення: 3.11.2022).
10. SolidWorks to URDF Exporter [Електронний ресурс]. URL: http://wiki.ros.org/sw_urdf_exporter (дата звернення: 4.11.2022).
11. The Open Motion Planning Library [Електронний ресурс]. URL: <http://ompl.kavrakilab.org/> (дата звернення: 4.11.2022).
12. H Deng, Z Xia, S Weng et al., "A motion sensing-based framework for robotic manipulation [J]", Robotics and Biomimetics, vol. 3, № 1, С. 23, 2016. (дата звернення: 5.11.2022).
13. Mobile manipulation task simulation using ROS with MoveIt [Електронний ресурс]. URL: <https://ieeexplore.ieee.org/abstract/document/8311930> (дата звернення: 6.11.2022).
14. Robot Operation System [Електронний ресурс]. URL: <https://moveit.ros.org/> (дата звернення: 6.11.2022).
15. ROS/Concepts [Електронний ресурс]. URL: <http://wiki.ros.org/ROS/Concepts> (дата звернення: 7.11.2022).
16. Віктор Русаков. Аналіз систем імітаційного моделювання // Manufacturing & Mechatronic Systems 2022: Proceedings of VIst International Conference, Kharkiv, October 21-22, 2022: Theses of Reports / [Ed. I.Sh. Nevlyudov (chief editor).] - Kharkiv .: [electronic version], 2022. - 136 p. with. P. 127-129. (дата звернення: 7.11.2022).
17. Особливості мов імітаційного моделювання та імітаційне програмування [Електронний ресурс]. URL: <http://um.co.ua/4/4-8/4-89183.html> (дата звернення: 8.11.2022).
18. Мови імітаційного моделювання [Електронний ресурс]. URL: <http://um.co.ua/8/8-13/8-130442.html> (дата звернення: 9.11.2022).
19. Імітаційне моделювання [Електронний ресурс]. URL: https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8Fч (дата звернення: 10.11.2022).

20. C Sharp [Електронний ресурс]. URL: https://uk.wikipedia.org/wiki/C_Sharp (дата звернення: 10.11.2022).
21. Fast & powerful cross-platform .NET IDE [Електронний ресурс]. URL: <https://www.jetbrains.com/idea/> (дата звернення: 11.11.2022).
22. Об'єктно-орієнтоване програмування (C#) [Електронний ресурс]. URL: <https://learn.microsoft.com/en-en/dotnet/csharp/fundamentals/tutorials/oop1> (дата звернення: 12.11.2022).
23. Угоди про написання коду на C# [Електронний ресурс]. URL: <https://learn.microsoft.com/en-en/dotnet/csharp/fundamentals/coding-style/coding-conventions> (дата звернення: 12.11.2022).
24. Правила та угоди про іменування ідентифікаторів C# [Електронний ресурс]. URL: <https://learn.microsoft.com/en-en/dotnet/csharp/fundamentals/coding-style/identifier-names> (дата звернення: 13.11.2022).