

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

## КВАЛІФІКАЦІЙНА РОБОТА

магістр

Освітній рівень

Удосконалений метод машинного навчання для керування роботом

Назва теми

КВРАКІТ. 2022168.01.02.ПЗ

Галузь знань 15 «Автоматизація та приладобудування»

Шифр, назва

Спеціальність 151 «Автоматизація та комп'ютерно-інтегровані технології»

Шифр, назва

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

Назва

Виконав:

студент 2 курсу, група АКІТ<sub>м</sub>-22-1

  
Підпис

Андрій КОЛОС

Ім'я, ПРІЗВИЩЕ

Керівник:

д-р техн. наук, проф.

  
Підпис, дата

Валерій МАРТИНЮК

Ім'я, ПРІЗВИЩЕ

Нормоконтролер

  
Підпис, дата

Людмила КОРЕЦЬКА

Ім'я, ПРІЗВИЩЕ

До захисту допускаю:

зав. кафедри автоматизації,  
комп'ютерно-інтегрованих  
технологій та робототехніки

  
Підпис, дата

Валерій МАРТИНЮК

Ім'я, ПРІЗВИЩЕ

«21» грудня 2023 р.

Хмельницький 2023

Хмельницький національний університет

Факультет інформаційних технологій

Кафедра автоматизації, комп'ютерно-інтегрованих технологій та  
робототехніки

Освітній рівень другий (магістерський)

Галузь знань 15 – Автоматизація та приладобудування

Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології

Освітня-професійна програма Автоматизація та комп'ютерно-інтегровані  
технології

ЗАТВЕРДЖУЮ

Зав. кафедрою

АКТІТ.Р  
В. Меремин

« 01 » 09 2023 р.

## ЗАВДАННЯ НА ДИПЛОМНУ РОБОТУ

Колос Андрій Богданович

1 Тема роботи: Удосконалений метод машинного навчання для керування  
роботом

керівник роботи Мартинюк Валерій Володимирович, д.т.н, професор.

Затверджено наказом по університету від «15» 08 2023р. № 30.

2 Строк подання студентом роботи на кафедру: 02.12.2023р.

3 Вихідні дані (характеристика об'єкта, умов дослідження та ін.)

Мета роботи: розробка удосконаленого методу машинного навчання для  
керування роботом.

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити)

Вступ. Огляд літературних джерел та патентних даних. Математична модель  
процесу машинного навчання для керування роботом. Імітаційна модель  
машинного навчання для керування роботом. Експериментальні дослідження  
удосконаленого методу машинного навчання для керування роботом. Висновки.





Завдання отримав

АК

Науковий керівник

В.М.


## Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Антиплагіат	Микола ФЕДУЛА, доцент кафедри АКІТгаР		
Нормоконтроль	Людмила КОРЕЦЬКА, доц. кафедри АКІТгаР		


## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів (розділів) дипломної роботи	Строк виконання етапів дипломної роботи	Примітка
1	Вступ	10.09.2023р.	Виконано
2	Огляд літературних джерел та патентних даних	25.09.2023р.	Виконано
3	Математична модель процесу машинного навчання для керування роботом	15.10.2023р.	Виконано
4	Імітаційна модель машинного навчання для керування роботом	30.10.2023р.	Виконано
5	Експериментальні дослідження удосконаленого методу машинного навчання для керування роботом	10.11.2023р.	Виконано
6	Висновки	15.11.2023р.	Виконано
6	Оформлення пояснювальної записки до ДР	20.11.2023р.	Виконано
7	Оформлення презентаційних матеріалів	1.12.2023р.	Виконано

Студент

 Підпис Колос А.Б Ініціали, прізвище

Керівник роботи

 Підпис В. Мерзлюк Ініціали, прізвище

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Удосконалений метод машинного навчання для керування роботом».

Автор роботи: Колос Андрій Богданович.


Керівник роботи: Мартинюк Валерій Володимирович.

Пояснювальна записка: 75 с., 32 рис., 0 табл., 1 дод., 40 джерел.

Графічна частина: 16 презентаційних слайдів.

РОБОТ-МАНІПУЛЯТОР DOBOT MAGICIAN, ОДНОПЛАТНИЙ  
МІКРОКОМП'ЮТЕР RASPBERRY PI MODEL B+, БІБЛІОТЕКА ДЛЯ  
КОМП'ЮТЕРНОГО ЗОРУ OPENCV.

**Мета роботи:** розробка удосконаленого методу машинного навчання для керування роботом та підвищення його продуктивності роботи.

  
Підпис студента

10.12.2023р  
Дата

## ЗМІСТ

ВСТУП.....	3
1 ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ ТА ПАТЕНТНИХ ДАНИХ .....	5
1.1. Особливості процесу машинного навчання для керування роботом .....	5
1.2. Особливості машинного зору для керування роботом .....	10
1.3 Висновки до першого розділу.....	18
2 МАТЕМАТИЧНА МОДЕЛЬ ПРОЦЕСУ МАШИННОГО НАВЧАННЯ ДЛЯ КЕРУВАННЯ РОБОТОМ .....	20
2.1 Математична модель керування роботом.....	20
2.2 Математична модель ПД регулятора для керування роботом.....	31
2.3 Висновки до другого розділу.....	34
3 ІМІТАЦІЙНА МОДЕЛЬ МАШИННОГО НАВЧАННЯ ДЛЯ КЕРУВАННЯ РОБОТОМ.....	36
3.1 Імітаційна модель процесу керування вектором швидкості або положення .....	36
3.2 Імітаційна модель процесу керування роботом маніпулятором за допомогою ПД-регулятора.....	40
3.3 Висновки до третього розділу.....	48
4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ УДОСКОНАЛЕНОГО МЕТОДУ МАШИННОГО НАВЧАННЯ ДЛЯ КЕРУВАННЯ РОБОТОМ .....	50
4.1 Керування роботом в інструментальній рамці у певній конфігурації з шумом	50
4.2 Схема модифікованого методу управління.....	56
4.3 Розробка програмного забезпечення.....	59
4.4 Висновки до четвертого розділу.....	70
ВИСНОВКИ.....	71
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	72
ДОДАТОК А Стаття у фаховому журналі .....	76

## ВСТУП

**Актуальність теми.** Dobot Magician - це високоточна багатофункціональна роботизована рука, розроблена спеціально для настільного використання. Він розроблений таким чином, щоб додати різні кінцеві ефекти, які виконують письмо, малювання, 3D-друк, лазерне гравірування та захоплення блоків (вибір і розміщення).

Тому, підвищення ефективності, взаємодія з маніпулятором робота, вирішення прямих та зворотних кінематичних задач з використанням удосконаленого методу машинного навчання для керування роботом, є дійсно актуальними задачами.

**Мета роботи** – Метою дослідження є розробка удосконаленого методу машинного навчання для керування роботом та підвищення його продуктивності роботи.

Відповідно до поставленої мети необхідно вирішити **завдання**:

- виконати огляд літературних джерел та патентних даних про методи керування роботом маніпулятором;
- розробити математичну модель процесу машинного навчання для керування роботом;
- розробити алгоритм та програмну реалізацію удосконаленого методу машинного навчання для керування роботом;
- розробити імітаційну модель удосконаленого методу машинного навчання для керування роботом.

**Об'єкт дослідження** - процес машинного навчання для керування роботом.

**Предмет дослідження** – удосконалений метод машинного навчання для керування роботом та підвищення продуктивності його роботи.

**Методи досліджень.** При вирішенні поставлених завдань у роботі були використані методи фізики, теорії автоматичного керування, методи обчислювальної математики, а також методи алгоритмізації та програмування.

**Наукова новизна одержаних результатів.** У результаті проведеного дослідження:

Розроблено удосконалений метод машинного навчання для керування роботом та підвищення продуктивність його роботи.

## 1 ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ ТА ПАТЕНТНИХ ДАНИХ

### 1.1 Особливості процесу машинного навчання для керування роботом

Автономне маніпулювання об'єктами є центральним завданням у галузі робототехніки. Дослідження в цій галузі традиційно були зосереджені на побутових застосуваннях, таких як розвантаження посудомийної машини [1], маніпуляції з тканиною [2], застилання ліжка [3], автоматизоване приготування їжі [4, 5], сервісна робототехніка [6-9], безлад клінінг [10, 11] і загальне побутове захоплення [12-14].

За останнє десятиліття маніпуляції, орієнтовані на галузь, набули значної популярності, частково через зацікавленість гігантів галузі, а саме Amazon і Google. Amazon Picking Challenge (APC) [15], наприклад, поставив завдання «підібрати та укласти» в неструктурованих, захаращених середовищах, зосереджених навколо автоматизації складу.

Цей виклик має багато робіт щодо інтеграції розпізнавання об'єктів, оцінки пози об'єкта, захоплення та планування руху та виявлення та відновлення помилок [16-22]. Деякі інші приклади, коли автоматичне захоплення вивчалось для промислових цілей, показані на рисунку 1.1 і включають сортування сміття [23], додаток касового робота [24], сортування мінералів [25], автономну мікроманіпуляцію [26].

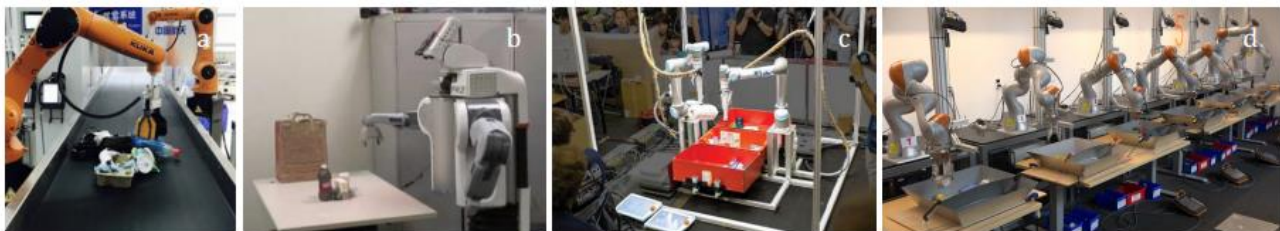


Рисунок 1.1 - Сучасні промислові роботи

Сучасні промислові роботи можуть виконувати багато складних завдань із маніпулювання та збирання за допомогою ретельно розроблених сценарієм

дій, починаючи від додатків підбору й розміщення [37, 38] до складання низьковольтних вимикачів [39].

Однак автономна взаємодія з об'єктами, невідомими апіорі, без явних інструкцій і інтелектуальним способом залишається відкритою проблемою. Автоматизувати це завдання надзвичайно складно, оскільки хороше розуміння пов'язане з формою об'єкта, розміром, вагою, розподілом ваги, властивостями поверхні, коефіцієнтами тертя, деформованістю об'єкта тощо, і на нього може сильно вплинути точність сприйняття та активації та шум [10-13].

Отримання такої інформації на практиці також не є тривіальним і може бути неможливим для деяких програм. Крім того, зв'язок між цими змінними та конкретною стратегією хапання, робототехнікою та захватом не завжди зрозумілий.

Захоплення нових об'єктів є добре вивченою сферою. Попередні роботи були зосереджені на універсальних захватах, призначених для зменшення важливості розміщення кінцевого ефектора. Odhner, наприклад, запропонував конструкцію захвату з двома пальцями, спрямовану на імітацію продуктивності для маленьких тонких предметів, таких як ключі чи монети, що є загальною проблемою для багатьох сучасних робіт. Конструкції захватів із недостатнім приводом були добре вивчені [15-17].

Також були запропоновані більш складні захвати, включаючи 3-пальці [18, 19], 4-пальці [10-12] і антропоморфні руки [13, 14]. Браун запропонував гранулярну конструкцію кінцевого ефектора, в якій маса притискається до об'єкта [55]. Усуваючи повітря всередині маси через вакуум, форма відповідає об'єкту.

Ця методологія продемонструвала чудову продуктивність для широкого діапазону об'єктів і значно зменшила складність апаратного та програмного забезпечення, але не вистачало сили захоплення круглих, плоских і маленьких об'єктів.

Складні проекти, як правило, перевершують дуже конкретні завдання на шкоду загальній продуктивності. Таким чином, промислові захвати та захвати,

які використовуються в цій галузі для досліджень, як правило, є переважно двопальцевими та з паралельними кінцевими ефекторами [24].

Нещодавно відомі та забезпечені ресурсами інститути, такі як Google, MIT, NASA, NVIDIA та Каліфорнійський університет у Берклі, фінансували нові дослідження з обробки об'єктів. Ши та Кунджул досліджували використання антропоморфного ручного планувальника захоплення для різноманітних загальних автомобільних роботів в NASA-JSC [28]. Робота Левіна, Пастора, Крижевського та Квіллена, пов'язана з Google, реалізувала безпрецедентну масштабу структуру створення набору даних [35].

Використовуючи 14 роботів-маніпуляторів, вони зібрали понад 800 000 зразків спроб захоплення протягом двох місяців, що дозволило їх для навчання складних згорткових нейронних мереж із самоконтролем (CNN) для прогнозування успіху.

Подібним чином Калашников використовував понад 580 000 зразків захоплення в реальному світі, щоб навчити структуру глибокого навчання з підкріпленням для виконання замкнутого захоплення об'єктів, які раніше не бачили [36].

Пов'язані з Массачусетським технологічним інститутом, Мануеллі, Гао, Флоренс і Тедрак запропонували КРАМ - нове представлення об'єктів для конкретного завдання, яке ігнорує нерелевантну для завдання геометрію [12].

Представляючи об'єкт у термінах семантичних ключових точок, вони досягли цілеспрямованого роботизованого маніпулювання новими об'єктами в межах відомої категорії.

Трамбле навели приклад робота, що підтримується NVIDIA [14]. Вони навчили оцінювати положення для 6 DOF для відомих об'єктів з одного зображення веб-камери RGB, використовуючи суміш 120 000 реальних і синтетичних даних.

Загалом, методології синтезу захоплення на основі машинного навчання (МН) стали надмірно представленими в літературі з обробки об'єктів [63, 64]. МН використовувалося для прогнозування захоплення різними способами у

роботах Саксена, Дрімейер і Кернса [29]. Вони передбачили місця захоплення безпосередньо з даних 3D-сенсора, використовуючи методику навчання під наглядом.

Пізніше Jiang, Moseson і Saxena [25] підійшли до синтезу захвату з двома пальцями, представивши хват у вигляді орієнтованого прямокутника, безпосередньо пов'язаного з розташуванням захоплення.

Відтоді у багатьох роботах використовувалися аналогічні підходи до вирішення задачі. Наприклад, Пінто та Гупта зібрали 50 000 точок даних у двовимірному прямокутному представленні [20]. Сунь, Ю, Лю та Гу використовували подібне представлення, використовуючи інформацію про глибину об'єкта [26].

Подальша робота Ленца, Лі та Саксени ефективно зменшила синтез захоплення до двоетапної задачі генерації та тестування, використовуючи їхнє попереднє представлення [13]. По-перше, неглибокий класифікатор використовувався для швидкого створення вікон для захоплення об'єктів.

Більш глибока оцінка була використана для вибору захоплення. Їхній набір даних - Cornell Grasp Detection Dataset [27] - з тих пір широко використовувався [20, 18, 26, 28-33]. Існують також інші зображення захвата.

Методології, не пов'язані з машинним навчанням, також досліджувалися Schwarz, які відображали сегментовані об'єкти за допомогою МН, але вибрані захоплення використовували евристично [17].

Militaru, Mezei і Tamas зіставили об'єкти-кандидати зі своїми відомими 3D-моделями [6], Harada апроксимував об'єкти за допомогою циліндрів [36], а С. Антон і Ф. Антон вивели захоплення шляхом апроксимації топології об'єкта [32].

Незважаючи на успіх багатьох таких робіт, машинне навчання суттєво переосмислило сучасний рівень як з точки зору обчислювальної ефективності, так і загальної продуктивності. Машинне навчання часто працює в режимі реального часу порівняно з не-машинним навчанням, що може зайняти набагато більше часу.

Редмон і Ангелова ілюструють це за допомогою однопрохідної регресії CNN, яка передбачає захоплення безпосередньо з даних датчиків [29]. Їхня методологія працювала зі швидкістю 13 кадрів на секунду (FPS) і досягла 88,0% точності класифікації за набором даних Cornell Grasp Detection Dataset.

Навпаки, система без МН, запропонована Arruda, Wyatt і Коріскі [28], була задокументована таким чином: 1,2 секунди для обчислення захоплення після того, як було зроблено принаймні два зображення об'єкта з різних точок зору. У середньому їх система вимагала 4,92 точки зору для планування захоплення та випробувань і досягнула 80,4% успіху захоплення.

В літературі про маніпулювання об'єктами ефективність системи або методології захоплення вимірюється різними способами. У багатьох роботах згадується швидкість, з якою їхня система була випробувана для успішного захоплення об'єктів, хоча визначення успішного захоплення, як правило, залежить від автора.

Деякі використовують датчики сили [15, 24, 29, 30], тоді як багато хто вважає спробу захоплення вдалою, якщо об'єкт можна підняти вище певної заздалегідь визначеної висоти [1, 16, 22, 31, 32, 39].

У деяких роботах перевіряється положення пальця після захоплення [35], а в інших використовується протокол струшування [21, 32]. Кілька робіт використовували захоплення предмету і вважали захоплення є успішним, якщо предмет можна скинути за командою, помістити в кошик або виконати якесь завдання [9, 12, 24, 36].

Morales, Chinellato, Fagg and del Pobil [33] класифікують результат через класифікацію на основі оціненої надійності. Загалом, більшість робіт просто оформляють випробування в термінах [10, 13, 21, 24, 28, 35] - без чіткого визначення того, що під ними мається на увазі.

Хоча двійкова метрика може бути корисною для порівняння та навчання, не було показано, що вона добре корелює з якістю захоплення об'єкта, якістю обробки або якістю розміщення. Це стає очевидним, коли досягаються хороші результати щодо швидкості захоплення, але об'єкт явно зміщується

самим процесом захоплення [1, 12, 25, 30, 38] - на шкоду маніпуляції та якості розміщення.

## 1.2 Особливості машинного зору для керування роботом

Автоматизований синтез захоплень зазвичай оформляється як задача пошуку - знайти оптимальне розуміння з потенційно нескінченної кількості захоплень - часто розглядається як завдання генерування та тестування [13, 31, 32, 35, 36, 38].

Шляхом вибірки робочого простору можна створити початкову підмножину осмислених гіпотез. Потім кожен зразок у цій підмножині може бути оцінений, щоб полегшити остаточний вибір.

Визначено нові показники, які вимірюють якість захоплення та більш повно описують результат захоплення, ніж традиційні класифікації. Запропоновані показники є безперервними, що корисно для навчання мережі. Крім того, такі показники кількісно визначають ефективність синтезу захоплення методологія - надання корисної базової лінії для об'єктивного порівняння.

Це дослідження вивчає масштабовану, агностичну методологію маніпулятора, яка сприяє як правильному захопленню, так і вдалому поводженню з невідомими об'єктами в промисловому контексті.

На відміну від багатьох систем, які використовують інформацію про глибину [6, 20], 3D-моделі об'єктів [31, 40] або датчики, встановлені на зап'ясті [29, 38], запропонована система працює на необробленому, монокулярному спостереженні за місцем події.

Пропонується 3-етапна структура навчання. На першому етапі здійснюється пошук і класифікація об'єктів у робочому просторі, ідентифікація відомих об'єктів і каталогізація невідомі предмети. Другий етап генерує потенційні зони захоплення виявлених об'єктів з точки зору орієнтації та пози.

На третьому етапі об'єднуються різні джерела інформації, які використовуються для ранжирування потенційних зон захоплення. Захоплення із найвищим рейтингом може бути передано роботу-маніпулятору для реалізації.

Щоб перевірити запропоновану методологію, було створено прототип. З 4000 випробувань кількісно було продемонстровано, що запропонована методологія може захоплювати маленькі об'єкти у 98,9% випадків.

Показано, що навчання для оптимізації запропонованих показників може покращити показники захоплення на 6,0% для відомих об'єктів і на 5,3% для невідомих об'єктів порівняно з базовою лінією. Крім того, вибір за запропонованими показниками покращив рівень охоплення на 2,7% порівняно з вибором за самим показником охоплення.

Аналіз набору даних третього етапу виявив міцний зв'язок між запропонованими показниками та традиційним поняттям кінцевого результату. Якісне тестування виявило тонку поведінку захоплення, демонстровану запропонованою методологією, яка покращила якість захоплення та результативність.

На даний момент у сфері роботизованих маніпуляцій бракує стандартизації, зокрема у формі спільних тестів і показників ефективності. Ця проблема ускладнює оцінку, порівняння або відтворення експериментальних результатів, і вона є відомою в літературі [21-33].

Малер припустив, що ця складність виникає, принаймні частково, через варіації в експериментальному протоколі, припущеннях і відмінностях у фізичному обладнанні, наприклад, датчики, освітлення, роботизовані руки, захвати та предмети [33].

Для усунення відсутності порівнянних результатів у дослідженні маніпуляцій були запропоновані різні тестові набори об'єктів. Наприклад, набір об'єктів складається з 75 об'єктів, класифікованих за корисністю, наприклад, продукти харчування, кухонні предмети, предмети інструментів, предмети форми та предмети завдань [34, 35].

Лейтнер запропонував еталонний тест відбору, що складається з 42 широко доступних об'єктів, на додаток до протоколів оцінки та запропонованих заходів [36, 37]. Хоча багато таких наборів є загальнодоступними, ця методологія не було прийнята ширшою спільнотою для стандартизованого використання. Це в основному пояснюється широкими методологічними відмінностями та апаратними обмеженнями, які обмежують застосовність об'єкта.

Через це багато дослідників розробляють власний тестовий набір об'єктів, часто зосереджений навколо звичайних побутових або звичайних лабораторних об'єктів [26, 34, 35]. Як правило, пул об'єктів, який використовується для тестування, добре задокументований. Також були запропоновані засоби моделювання, які створюють середовище для об'єктивного аналізу маніпуляцій, наприклад, OpenGRASP [29], GraspIt! [30] та VisGraB [31].

Спільні набори даних нещодавно набули популярності. Наприклад, набір даних Cornell Grasp Detection Dataset широко використовувався для навчання та тестування [20, 28, 36]. Деякі роботи навіть оцінюють свою методологію лише за продуктивністю набору даних [18, 20, 28, 38].

У таких випадках точність набору даних зазвичай інтерпретується як потенційний рівень захоплення - без проведення фізичних випробувань для перевірки. Редмон і Ангелова проілюстрували це.

Вони запропонували однопрохідний алгоритм, який значно покращив продуктивність розпізнавання захоплення з точки зору обчислювальної ефективності та продуктивності набору даних - досягнувши 88% точності класифікації зображень на наборі даних Корнелла [29]. Незважаючи на те, що вони поширили сучасні технології на реальні записані дані датчиків, їх метод не був фізично перевірений.

Продуктивність набору даних може стати чудовою платформою для порівняння, прикладом якого є успіх великомасштабних конкурентних наборів даних для розпізнавання об'єктів, наприклад, Pascal VOC [22, 23], ImageNet [24], COCO [25] тощо.

На жаль, застосування того самого підходу до дослідження маніпуляцій базується на припущенні, що продуктивність набору даних перетворюється на реальну продуктивність. Очевидно, це не так, оскільки багато робіт повідомляють про розбіжності між реальними показниками захоплення та точністю набору даних. Сунь Ю, Лю і Гу, наприклад, відзначили 11% розбіжності між точністю класифікації набору даних і результатом фізичного дослідження [26].

Уотсон, Хьюз і Ліда відтворили роботу Редмона і Ангелової за допомогою дослідницького робота Baxter [36]. Їхні випробування показали, що показник захоплення становить 62% - це на 26% зниження очікуваної продуктивності. Зазвичай набори даних у цьому полі анотуються вручну [12, 20, 35, 38].

Це проблематично, оскільки навчання налаштувань продуктивності набору даних а методологія розуміння, яку творці вважають оптимальною для реалізації в реальному світі. Крім того, кількісно оцінити ступінь, до якої такі анотовані захоплення відображаються у фізичній системі, надзвичайно складно, як показано в різних роботах, які повідомляють однакові характеристики з різними наборами даних.

Через проблеми, пов'язані з суб'єктивними наборами даних, у деяких роботах намагалися взагалі уникнути людської упередженості. Zeng та ін. продемонстрували, що глибоке навчання з підкріпленням без моделі можна використовувати для вивчення складної поведінки захоплення шляхом проб і помилок [36]. Подібним чином Пінто та Гупта використовували самонагляд протягом понад 700 робото-годин [40]. Сучасні підходи, які використовують ручні етикетки, все ще значно перевершують інші методології.

Щоб подолати людську упередженість, ця робота пропонує інститут об'єктивних показників ефективності захоплення. Вимірюючи помилку, внесену самим випробуванням розуміння, можна встановити показники для оцінки результату розуміння, кількісної оцінки якості розуміння та надання корисних критеріїв для навчання мережі.

Традиційно роботи в усій літературі зосереджені на понятті успішного захоплення, де фізична спроба захоплення. Показники помилок, викликаних захопленням, відходять від цього двійкового виміру, надаючи більш вичерпний безперервний опис результатів випробування.

За останнє десятиліття кількість роботизованих рук, які використовуються в промисловості, значно зросла, особливо в сферах точкового зварювання, фарбування розпиленням, електронного тестування, метрології, складання та обробки [38].

За оцінками Міжнародної федерації робототехніки (IFR), у 2018 році у світі було встановлено понад 400 000 нових роботів [339]. Порівняно з 2008 роком це означає збільшення кількості встановлень на 370%.

Крім того, IFR прогнозує, що до 2022 року щорічне глобальне встановлення перевищить 570 000 одиниць. Основна маса нових інсталяцій у 2018 році була пов'язана з автомобільною, електричною та електронною технікою (рис. 1.3), хоча найбільше зростання спостерігалось у інсталяціях, пов'язаних з харчовими продуктами, збільшившись на 32% між 2016 і 2018 роками [40].

Також зросла кількість медичних встановлено сервісних роботів, з 5100 встановлень у 2018 році порівняно з 7200 у 2019 році. Очікується, що до 2020 року ця кількість збільшиться до 19 700 одиниць встановлення.

Сучасна автоматизація виробництва відходить від масового виробництва через скорочення життєвих циклів і скорочення обсягів виробництва багатьох нових продуктів - під впливом споживачів. Ця тенденція була добре задокументована [28, 31-38].

Для поглибленого огляду сучасних технологій і тенденцій, що виникають у виробництві, читач може звернутися до огляду, наданого Dotoli, Fay, Miśkowiec і Seatzu [39]. Ця зміна під назвою «Індустрія 4.0» запроваджує інтелектуальні інформаційні та комунікаційні технології з метою підвищення гнучкості та масштабованості виробничих систем.

Маклін, Уокер і Брайт, наприклад, запропонували підхід до зменшення збурень для реконфігурованих виробничих систем на основі штучних нейронних мереж [40]. Вони продемонстрували, що штучний інтелект можна використовувати, щоб допомогти керувати такими збуреннями, як нестача матеріалу, поломка виробничих елементів і надмірний знос інструментів.

Ферраті, Нарді, Сеттімі, Маріно та Паллоттіно запропонували алгоритм розумного планування, який покращив автономність мобільних роботів у неструктурованому середовищі з кількома роботами [30]. Щоб задовольнити попит на адаптоване виробництво, багато виробників переходять до гнучких і реконфігурованих роботизованих систем.

Наприклад, компанія АББ розробила концепцію робота з подвійною рукою, призначену для співпраці з працівниками в гнучких виробничих середовищах, які регулярно змінюються [21, 22]. Їхня концепція була зосереджена на складанні дрібних деталей у стандартних ручних робочих місцях. Як правило, гнучкість керуючого програмного забезпечення таких робототехнічних концепцій не розглядається так само, як фізичний дизайн.

Щоб впоратися з невизначеністю, пов'язаною з асортиментом продукції, обсягом і тривалістю життя, виробничі системи, схильні до частих змін, прагнуть до ручної праці. У результаті ручне виробництво є поширеним у країнах з низьким рівнем оплати праці. Зараз Китай є чудовим прикладом у цьому контексті.

Враховуючи, що Китай є найбільшою у світі торговельною та виробничою нацією, на яку припадає 11,4% світових товарів, проданих у 2017 році [23], кількість промислових роботів на одного фабричного робітника є відносно низькою. Фонд інформаційних технологій та інновацій (ITIF) оцінив цю цифру приблизно в 100 промислових роботів на 10 000 працівників у 2017 році [24]. Корея була найбільшою за кількістю промислових роботів на одного працівника в 2017 році, оцінюється в 710 одиниць на 10 000 працівників.

Програми нових роботів, встановлених у Китаї, часто зосереджені на виконанні завдань, які складають майже чверть усіх встановлень окремих програм.

У той час як Китай демонструє найбільше зростання з точки зору автоматизації промислових роботів, багато інших країн поділяють подібну тенденцію. В Індії, наприклад, у 2018 році кількість встановлених установок зросла на 39% порівняно з попереднім роком [25].

Навіть у країнах, які вже мають високий рівень роботизованої автоматизації, кількість щорічних установок продовжує зростати. Наприклад, у Японії відтоді середній річний темп зростання становив 17%.

Виробнича промисловість стикається з постійною тенденцією до більш складних продуктів, збільшення асортименту продукції, менших розмірів партій і скорочення життєвих циклів [28, 31-38].

Попередні дослідження показують, що виробничі середовища з декількома продуктами часто досягають достатньої гнучкості, адаптивності та потенціалу реконфігурації за допомогою людей; також використовуються роботизовані системи, але вони вимагають частоті ручної роботи. Для ефективної роботизованої автоматизації в цьому контексті вирішальне значення мають надійні теорії та методології управління, здатні точно охоплювати нові об'єкти та маніпулювати ними.

Більш широке коло досліджень, з яким пов'язане це дослідження, стосується комплексу технологій, спрямованих на підвищення гнучкості та потенціалу налаштування промислових виробничих систем, які використовують робототехніку, штучний інтелект, машинне навчання, датчики та автоматизацію.

Дослідження інтелектуальної методології, яка базується на навчанні, для автономного керування новими об'єктами шляхом включення частково пов'язаної інформації та машинного зору, розглядається з промислової точки зору. Для проведення цих досліджень необхідно:

- дослідити поточний машинний зір, цифрову обробку зображень і виявлення об'єктів, щоб визначити потенційні методи обробки об'єктів;

- вивчати штучний інтелект, нейронні мережі та машинне навчання, що стосуються маніпуляції розумними роботами;
- оцінити потенціал машинного навчання для обробки нових деталей;
- дослідити та виявити фактори, що призводять до кращої якості захоплень;
- розробити нові методи/метрики для кількісної оцінки якості захоплення;
- дослідити вплив якості захоплення на показники захоплення;
- розробити методологію виявлення та захоплення частин на основі штучного інтелекту, яка забезпечить надійне, двома пальцями й автономне керування новими об'єктами, одночасно сприяючи майбутній масштабованості та легкості інтеграції.

У інформатиці штучний інтелект відноситься до вивчення методів, алгоритмів і статистичних моделей, які покладаються на шаблони та висновки для виконання конкретних завдань, на відміну від явних інструкцій. Цей термін був введений Джоном Маккарті в спільному дослідженні в 1955 році [33].

Багато реалізацій цієї технології стали знаковими у визначенні ШІ для громадськості. Наприклад, Деер Blue від ІВМ залишив незабутнє враження в 1997 році після перемоги в шаховому матчі-реванші з тодішнім чемпіоном світу Гарі Каспаровим [34, 35]. У 2011 році компанія ІВМ Watson брала участь у конкурсі Jeopardy і виграв у двох найкращих гравців гри [36].

З тих пір їх система використовується як доповнення до бази знань в охороні здоров'я [37, 38]. У 2016 році AlphaGo Lee від Google виграв чотири з п'яти ігор проти Korean Go Champion Лі Седол [39], а в 2017 році AlphaGo Master від Google переміг 60 професійних гравців у Го онлайн та виграв три матчі поспіль проти китайського чемпіона світу з Го Ке Джі [40].

Їхня система використовувала самостійне навчання з підкріпленням і не потребувала взаємодії з людиною протягом 40-денного періоду навчання. AI частіше використовується в комерційних програмах, наприклад, розпізнавання виразу обличчя [31], оцінка якості потокового відео в реальному часі [32],

розумні помічники, такі як Alexa та Hello Google [33], модерування небажаних гравців у відеоіграх [34], фільтрація спаму електронною поштою [35], технологія автономних автомобілів [36, 37], онлайн підтримка клієнтів [38], прогнозування поведінки споживачів [39], виявлення ботнетів в Інтернеті [40] і прогнозування курсів іноземних валют [31]. ШІ також займає важливе місце в дослідженнях, пов'язаних із зором, часто вивчається в сегментації [22, 32-37] і розпізнаванні об'єктів [35, 38].

Застосування штучного інтелекту набуло популярності в багатьох сферах за останні два десятиліття, чому сприяли значні досягнення в можливостях і зниження вартості графічних процесорів (GPU) [24, 25, 32, 33].

Наприклад, NVIDIA Riva TNT2 1999 року виконувала 0,25 мільярда (гіга) операцій із плаваючою комою за секунду (GFLOP). NVIDIA RTX 2060 Super 2019 року представляє еквівалентний за ціною графічний процесор із продуктивністю 7180 GFLOPs.

Ця тенденція зробила кілька аспектів машинного навчання, як-от глибоке навчання за допомогою штучних нейронних мереж і великих обсягів даних, більш практичними та доступними.

### 1.3 Висновки до першого розділу

1. Сучасні промислові роботи можуть виконувати багато складних завдань із маніпулювання та збирання за допомогою ретельно розроблених сценарієм дій, починаючи від додатків підбору й розміщення до складання низьковольтних вимикачів.

2. Однак автономна взаємодія з об'єктами, невідомими апріорі, без явних інструкцій і інтелектуальним способом залишається відкритою проблемою. Автоматизувати це завдання надзвичайно складно, оскільки хороше розуміння пов'язане з формою об'єкта, розміром, вагою, розподілом ваги, властивостями поверхні, коефіцієнтами тертя, деформованістю об'єкта тощо, і на нього може сильно вплинути точність сприйняття та активації та шум.

3. Отримання такої інформації на практиці також не є тривіальним і може бути неможливим для деяких програм. Крім того, зв'язок між цими змінними та конкретною стратегією хапання, робототехнікою та захватом не завжди зрозумілий.

4. Попередні дослідження показують, що виробничі середовища з декількома продуктами часто досягають достатньої гнучкості, адаптивності та потенціалу реконфігурації за допомогою людей; також використовуються роботизовані системи, але вони вимагають частішої ручної роботи. Для ефективної роботизованої автоматизації в цьому контексті вирішальне значення мають надійні теорії та методології управління, здатні точно охоплювати нові об'єкти та маніпулювати ними.

## 2 МАТЕМАТИЧНА МОДЕЛЬ ПРОЦЕСУ МАШИННОГО НАВЧАННЯ ДЛЯ КЕРУВАННЯ РОБОТОМ

### 2.1 Математична модель керування роботом

Робот - це машина, здатна автоматично виконувати складну серію дій. Іншими словами, робот - це адаптивний та універсальний пристрій. Адаптивний, тому що він здатний переналаштовуватися для виконання поставленого завдання. Універсальний - тому що може виконувати кілька типів завдань. Робот складається з 5 систем:

1. Механізм - фізична структура робота.
2. Здатність до сприйняття - відповідає за контроль навколишнього середовища та адаптацію робота до змін.
3. Контролер - генерує вхідні дані робота.
4. Комунікаційний інтерфейс - з'єднує користувачів з роботом.
5. Периферійні пристрої - формування робочого простору.

Будь-який робот складається з ланок, і ці ланки з'єднуються між собою за допомогою шарнірів, які надають роботу ще один ступінь свободи (СС).

Існує два типи з'єднань.

1. Револьверне з'єднання: це *обертання* між двома ланками через загальну вісь.
2. Призматичне з'єднання: це передача між двома ланками через спільну вісь.

Кожен суглоб має власний фрейм, тому існують способи зв'язати окремі суглоби за допомогою матричних обчислень. Всі ланки і суглоби, що утворюють роботизовану руку, називаються простором суглобів (Joint Space).

Крім того, останній простір називається простором завдань, оскільки

він є місцем розташування кінцевого пристрою, відповідального за виконання завдання, і цей пристрій зазвичай називається кінцевим ефектором або інструментом. На рисунку 2.1 зображено спільний простір та простір завдань.

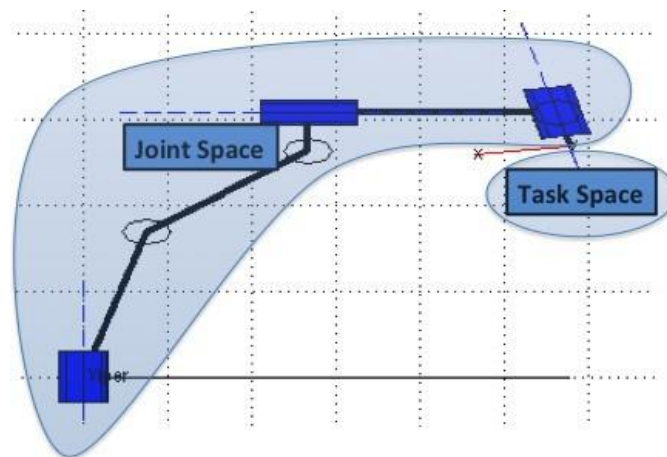


Рисунок 2.1 - Спільний простір та простір завдань

Теоретично за допомогою 6 СС можна досягти будь-якої позиції, але також є можливість використовувати більше 6. В такому випадку кажуть, що робот має надлишки.

Іншим цікавим поняттям є робототехнічна сингулярність - ситуація, в якій робот не може вирішити очікувану позиційну задачу. Існує два типи сингулярностей.

1. На межі робочого простору: це відбувається, коли робот досягає фізичної межі, або по висоті, або по довжині, логічно машина не може рухатися далі.

2. Всередині робочої області: Це відбувається, коли визначник якобінської матриці дорівнює 0, тому обернена якобінська матриця не має розв'язку.

$$J^{-1} = Jinv = \frac{adj(J^T)}{\det(J)}. \quad (2.1)$$



Рисунок 2.1 – Робот з 6 ступенями свободи

Послідовний робот складається з  $N + 1$  ланки та  $N$  з'єднань. З'єднання  $i$  з'єднує ланку  $i$  з ланкою  $i + 1$  і позначається символом  $q_i$ . Крім того, у кожному місці з'єднання визначається фрейм як  $F_i$ . Ці фрейми визначаються чотирма параметрами.

1.  $z_i$ - вісь вздовж осі суглоба  $i + 1$ .
2.  $x_i$  - відстань по нормалі між  $z_i$  та  $z_{i+1}$ .
3.  $y_i$  - координат, створена за правилом правої руки.
4. Початок координат  $O_i$  є точкою перетину осі  $z_i$  зі спільною нормаллю до осей  $z_{i-1}$  та  $z_i$ . (Командна нормаль - це пряма, що містить мінімальний відрізок відстані між двома прямими).

Крім того, створено матрицю переходу від базису  $F_{N-1}$  до базису  $F_N$  як функцію чотирьох параметрів D-H.

$a_i$  - кут між  $z_{i-1}$  і  $z_i$  вздовж  $x_{i-1}$ .

$d_i$  - відстань між  $z_{i-1}$  та  $z_i$  вздовж  $x_{i-1}$ .

$\theta_i$  - кут між  $x_{i-1}$  та  $x_i$  вздовж  $z_i$ .

$r_i$  - відстань між  $x_{i-1}$  та  $x_i$  вздовж  $z_i$ .

На рисунку 2.2 зображено D-H представлення робота маніпулятора.

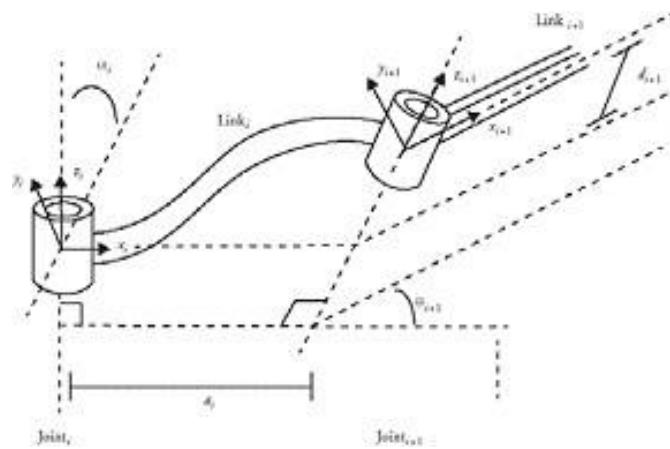


Рисунок 2.2 - Представлення D-H

На цьому етапі можна визначити значення **координати з'єднання** для кожного з'єднання. Загальна форма для цього параметра має вигляд:

$$q_i = \bar{\sigma}_i \cdot \theta_i + \sigma_i \cdot a_i \quad (2.2)$$

де  $\sigma_i = 0$  - для обертового шарніра;

$\sigma_i = 1$  - для призматичного з'єднання;

$\bar{\sigma}_i = 1 - \sigma_i$ .

Таким чином, матриця D-H для робота з 6 DOF може бути представлена таблицею 2.1.

Робот-маніпулятор складається з 6 ротаційних суглобів (рисунок 2.3). Перші п'ять СС можуть оброблятися незалежно від шостого, який був дуже цікавим для позиції Control, як показано нижче.

Таблиця 2.1: Таблиця D-H параметрів для робота з 6 СС

$N$	$\sigma_N$	$\alpha_N$	$a_N$	$\theta_N$	$d_N$
1	$\sigma_1$	$\alpha_1$	$a_1$	$\theta_1$	$d_1$
2	$\sigma_2$	$\alpha_2$	$a_2$	$\theta_2$	$d_2$
3	$\sigma_3$	$\alpha_3$	$a_3$	$\theta_3$	$d_3$
4	$\sigma_4$	$\alpha_4$	$a_4$	$\theta_4$	$d_4$
5	$\sigma_5$	$\alpha_5$	$a_5$	$\theta_5$	$d_5$
6	$\sigma_6$	$\alpha_6$	$a_6$	$\theta_{16}$	$d_6$

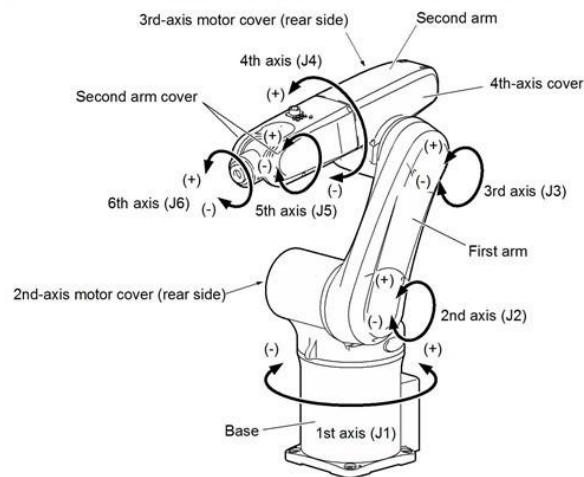


Рисунок 2.3 – Структура робота-маніпулятора

Додатково, параметри D-H, що визначають робот-маніпулятор, наведені в таблиці 2.2.

Інформація, надана виробником, пояснює, що можлива сингулярність під час роботи робота-маніпулятора.

Вирівнювання стиків  $q_1$  та  $q_6$ .

Вирівнювання стиків  $q_4$  та  $q_6$ .

Таблиця 2.2 - Таблиця D-H параметрів для робота-маніпулятора

$N$	$\sigma_N$	$\alpha_N$	$a_N$	$\theta_N$	$d_N$
1	0	$-\pi/2$	0.075	0	0.335
2	0	0	0.27	$-\pi/2$	0
3	0	$\pi/2$	-0.09	0	0
4	0	$-\pi/2$	0	0	0.295
5	0	$-\pi/2$	0	$\pi/2$	0
6	0	0	0	0	0.080

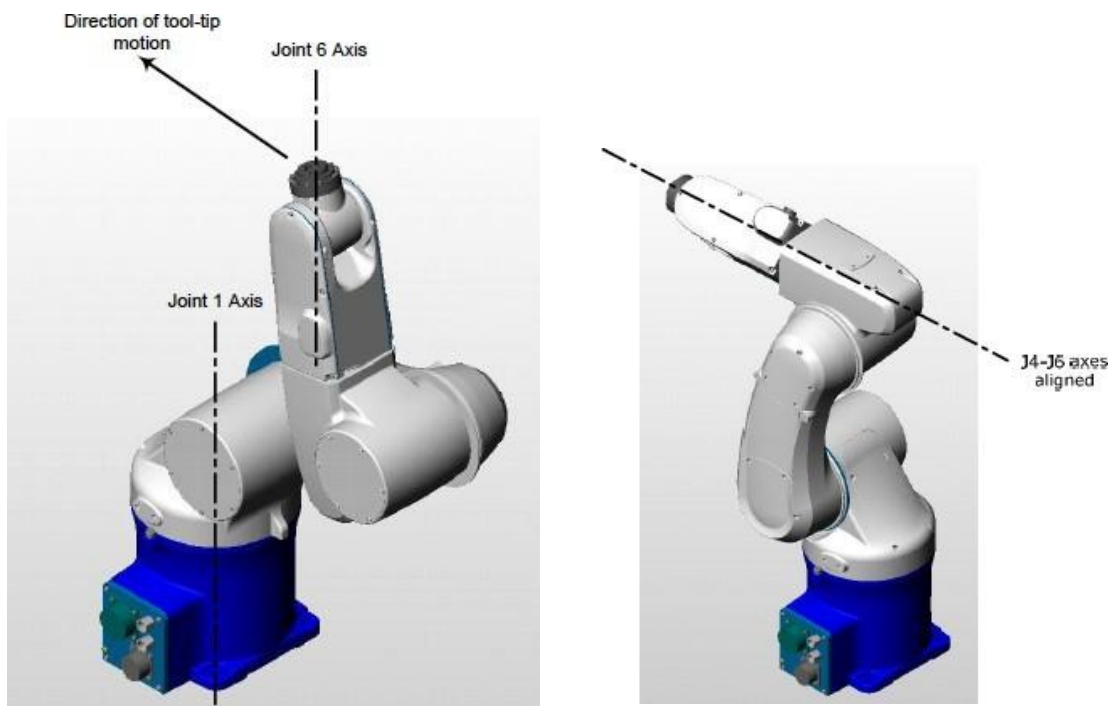


Рисунок 2.4 - Сингулярності робота-маніпулятора

Однорідна система є корисним математичним інструментом, який часто використовується в робототехніці. Вона використовується для координат та матриці перетворення.

Однорідні координати є альтернативою для представлення будь-якої точки відносно рамки. Ця точка визначається вектором

$[\rho P_x \quad \rho P_y \quad \rho P_z \quad \rho]$ , де  $\rho$  - масштабний коефіцієнт (у робототехніці він дорівнює 1).

Крім того, будь-який напрямок також формується за допомогою чотирикомпонентного вектору з 0 як четвертим числом,  $[e_x \quad e_y \quad e_z \quad 0]$ .

Однорідне перетворення - це матриця 4x4, яка пов'язує різні фрейми. Ця матриця складається з 3 векторів напрямку та одного положення. Нормальними символами для її представлення є T або H.

Наприклад, однорідна матриця перетворення між кадрами  $i$  та  $j$ , описується виразом (2.3).

$${}^i T_j = {}^i H_j = \begin{bmatrix} s_x & n_x & a_x & O_x \\ s_y & n_y & a_y & O_y \\ s_z & n_z & a_z & O_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.3)$$

де  $[\vec{s} \quad \vec{n} \quad \vec{a}]$  - одиничний вектор вздовж осі, який називається матрицею обертання.

$\vec{O}$ - початок координат фрейму  $j$ , виражений у координатах фрейму  $i$ ,  $a$ , які виражені у вигляді точок перекладу.

Геометричне моделювання має на меті пов'язати декартовий простір із суглобом за допомогою однорідної матриці перетворення.

Пряма геометрична модель дозволяє досягти декартового розташування інструмента, використовуючи відомі спільні значення. Зазвичай цей результат є унікальним.

$$P_{cartesian} = f(q_{articular}). \quad (2.4)$$

Використовуючи конвенцію D-H, робот може бути змодельований суглоб за суглобом за допомогою матриці перетворень (2.5).

$${}^{i-1}T_i = {}^{i-1}H_i = Tr(x, d_i) \cdot R(x, a_i) \cdot Tr(z, r_i) \cdot R(z, \theta_i) \quad (2.5)$$

$${}^{i-1}T_i = {}^{i-1}H_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & d_i \\ \cos(a_i)\sin(\theta_i) & \cos(a_i)\cos(\theta_i) & -\sin(a_i) & -r_i\sin(a_i) \\ \sin(a_i)\sin(\theta_i) & \sin(a_i)\cos(\theta_i) & \cos(a_i) & r_i\cos(a_i) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

де  $q_i$  - це спільний ідентифікатор;

$a_i$ ,  $\theta_i$ ,  $d_i$  та  $r_i$  - параметри D-H для  $i$ -го суглоба.

Все відношення отримується перемноженням цих матриць у порядку зростання.

$${}^0T_n = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot \dots \cdot {}^{N-1}T_N. \quad (2.6)$$

Інверсна геометрична модель - це зворотна операція до прямої геометричної моделі. Знаючи декартове положення інструмента, а значення, які потрібно отримати, - це градусні міри зчленування. Цей розв'язок рідко буває єдиним, оскільки для однієї позиції може існувати нескінченна кількість комбінацій шарнірів. Це може ускладнити досліди для автоматичної ітераційної роботи.

$$q = f^{-1}(P_{cartesian}). \quad (2.7)$$

Перетворення безпосередньо пов'язує декартовий вектор швидкості,  $\dot{x}$ , зі спільною швидкістю,  $\dot{q}$ , через якобіанську матрицю.

Матриця Якоба,  $J$ , - це математичний метод, який використовується в робототехніці для перетворення швидкості суглоба в декартову швидкість за наведеним нижче рівнянням.

$$\dot{x} = J \cdot \dot{q}, \quad (2.8)$$

$$\begin{bmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} J_{6 \times N} \end{bmatrix} \cdot \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \\ \vdots \\ \dot{q}_N \end{bmatrix}. \quad (2.9)$$

де  $\dot{x}$  - декартова швидкість, що складається з лінійної та кутової швидкостей.

$\dot{q}$  - спільна швидкість, утворена швидкостями для кожного суглоба.

$J$  - матриця Якоба розмірністю  $(6 \times N)$ , де  $N$  дорівнює кількості з'єднань.

Ця матриця обчислюється щоразу, коли робот змінює позицію, даючи унікальний розв'язок з визначником, відмінним від 0, уникаючи таким чином сингулярностей.

Найцікавішою властивістю є те, що працюючи з дуже малими вхідними даними,  $J$  можна використовувати у дослідях швидкості або положення.

$$\dot{x} = J \cdot \dot{q} \Rightarrow \frac{dx}{dt} = J \frac{dq}{dt} \Rightarrow \frac{x_i - x_{i-1}}{t} = J \cdot \left( \frac{q_i - q_{i-1}}{t} \right). \quad (2.10)$$

$$x_i = x_{i-1} + J \cdot (q_i - q_{i-1}). \quad (2.11)$$

Якобіан також може бути застосований для обчислення обернених операцій з оберненою Якобінською матрицею  $J_{inv}$ .

$$\dot{q} = J_{inv} \cdot \dot{x} \Rightarrow \frac{dq}{dt} = J_{inv} \frac{dx}{dt} \Rightarrow \frac{q_i - q_{i-1}}{t} = J_{inv} \left( \frac{x_i - x_{i-1}}{t} \right). \quad (2.12)$$

$$q_i = q_{i-1} + Jinv \cdot (x_i - x_{i-1}). \quad (2.13)$$

Матриці Якобі в прямій або оберненій формі постійно використовуються в розрахунках. На щастя, у використаних бібліотеках вже реалізовано ці функції, і працювати з ними досить інтуїтивно зрозуміло.

Ця робота спочатку мала на меті представити рішення, написане найпростішим способом з використанням базових обчислень, зрозуміле навіть без сильного досвіду в робототехнічному моделюванні. Але це виявилось складно, коли потрібно було обчислити одиничний вектор орієнтації  $\vec{U}$  і кут повороту  $\theta$ .

Обчислення  $\theta$  і  $\vec{U}$  через матрицю обертання  $R$  виконується за наведеними нижче рівняннями.

$$\theta = \cos^{-1}\left(\frac{\text{trace}(R)-1}{2}\right). \quad (2.14)$$

$$\hat{U} = \frac{1}{2\sin(\theta)} \begin{bmatrix} R(3,2) - R(2,3) \\ R(1,3) - R(3,1) \\ R(2,1) - R(1,2) \end{bmatrix}. \quad (2.15)$$

$$\vec{U} = \theta \cdot \hat{U}. \quad (2.16)$$

де  $R(\alpha, \beta)$  - відповідає позиції матриці, де  $\alpha$  - номер рядка, а  $\beta$  - номер стовпця.

Теорія стверджує, що з цим обчисленням немає проблем, якщо кут  $\theta$  ненульовий або кратний  $2\pi$ , але, на жаль, в одному напрямку руху  $\theta$  дорівнює нулю. Для вирішення цієї проблеми було зручно працювати у новій системі числення, системі кватерніонів.

Кватерніон - це число, що належить до системи числення  $\mathbb{H}$  і є

похідним від комплексних чисел. Основна відмінність полягає в тому, що система комплексних чисел породжує ще один вимір  $(1, i)$ , а система кватерніонів породжує ще три виміри  $(1, i, j, k)$ .

Математичний сенс цього запису полягає в тому, щоб перетворити жорстке тіло з тривимірними координатами в чотиривимірний базис, що представляє вісь, навколо якої буде відбуватися обертання, та величину обертання, яке буде відбуватися навколо цієї осі.

Отже, будь-який кватерніон,  $Q \in H$  однозначно записаний виразом.

$$Q = ar + b_x i + c_y i + d_z k = Qr + \vec{Q}. \quad (2.17)$$

Робота була виконана з розширенням *кватерніонів*, яке називається подвійний *кватерніон* (DQ). DQ також використовується для визначення кута зміщення твердого тіла навколо осі, що проходить через початок координат, але було продемонстровано, що кватерніон не може показати всі можливі 3D перетворення. DQ вирішує цю проблему більш ефективно, ніж простий  $Q$ , але є складнішим у реалізації.

DQ утворюється парою кватерніонів:

$$DQ = Q + \varepsilon \cdot Q_0. \quad (2.18)$$

Крім того, DQ можна розглядати як кватерніон, чотири члени якого є подвійними числами.

$$DQ = \tilde{Q} = \tilde{a}r + \tilde{b}_x i + \tilde{c}_y j + \tilde{d}_z k. \quad (2.19)$$

де  $Q$  - називається дійсною частиною  $DQ$ ;

$Q_0$  - називається подвійною частиною  $DQ$ ;

$\varepsilon$  - дуальний оператор з властивістю  $\varepsilon^2 = \varepsilon^3 = \dots \simeq 0$  ;

$r$  - дійсна частина,  $i, j, k$  уявна частина.

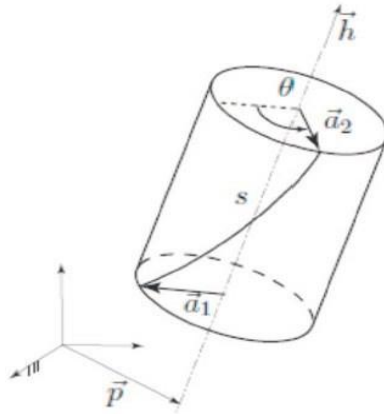


Рисунок 2.5 - Представлення кута повороту та вектора орієнтації

Проходячи через DQ Space, розрахунок кута тета і осі руху досягається за допомогою простих формул.

$$\theta = 2\cos^{-1}(ar) ; \quad (2.20)$$

$$\vec{h} = U_x = \frac{\tilde{b}_x}{\sin(\frac{\theta}{2})}, U_y = \frac{\tilde{c}_y}{\sin(\frac{\theta}{2})}, U_z = \frac{\tilde{d}_z}{\sin(\frac{\theta}{2})}. \quad (2.21)$$

Як видається, така ж проблема виникає і з тета-кутом, але кватерніон і подвійний кватерніон мають можливість впоратися з цією невизначеністю. Більше того, Matlab надає панель інструментів з декількома вже реалізованими функціями.

## 2.2 Математична модель ПІД регулятора для керування роботом

ПІД - це промислова система управління, заснована на механізмі зворотного зв'язку. Блок складається з трьох регуляторів: пропорційний, інтегруючий та похідний. Система працює з диференціалами, або помилками, в якості входів. Ці помилки генеруються між поточним станом системи та бажаним станом, і мета полягає в тому, щоб мінімізувати значення цієї помилки після декількох ітерацій.

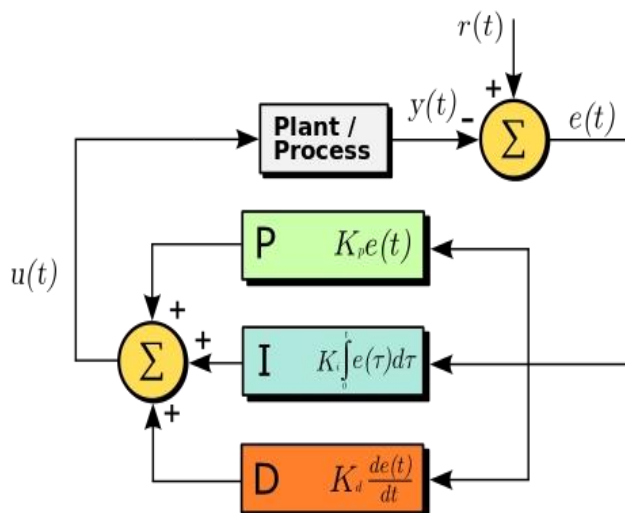


Рисунок 2.6 - Конфігурація ПІД

ПІД можна використовувати в повному вигляді (P+I+D) або тільки з використанням деяких блоків (P+D, P+I...). Математична формула ПІД:

$$e_q = e_c = Kp \cdot e_i + Ki \cdot t \cdot \sum_{N=1}^i e_N + \frac{Kd}{t} \cdot (e_i - e_{i-1}), \quad (2.22)$$

де  $e_q$  та  $e_c$  - це сумарна похибка в декартових або спільних координатах.

Природно, що  $e_q$  а  $e_c$  потрібно вказати всі помилки в координатах:

$t$  - час дискретизації.

$Kp$ ,  $Ki$  та  $Kd$  - це коефіцієнти підсилення;

$Kp$  - пряме значення;

$Ki$  - використовується з інтегруючим значенням;

$Kd$  - значення відхилення;

$e_i$  - відповідає поточній ітерації, отже,  $e_{i-1}$  представляє вхідний прецедент;

$\sum_{N=1}^i e_N$  - це сума всіх входів до поточного моменту.

Залежно від досліджуваної схеми, ПД обробляє декартові або спільні входи. З одного боку, спільний ПД легше реалізувати і він впливає більш безпосередньо на кожне з'єднання. З іншого боку, декартові ПД об'єднують набагато більше факторів, що робить їх складнішими для реалізації.

Основним інструментом для моделювання був Robotics Toolbox для Matlab. Цей інструментарій був створений Пітером Корком і дозволяє моделювати будь-які роботизовані маніпулятори. Усередині інструментарію вже передбачено багато корисних кінематичних функцій.

Моделювання базується на покроковому управлінні рухом, і для досягнення цієї мети необхідно завантажувати кінематичні параметри робота в кожній ітерації. Знову ж таки, пояснення було розділено на швидкість та положення.

Для досягнення задовільних результатів з використанням конфігурації *швидкості* потрібно було використовувати лише Robotics Toolbox. Однак, для цієї *позиції* також був використаний інший набір інструментів, DualQuaternion. Ці два набори інструментів добре працюють разом, і їх поєднання було найкращим способом отримати хороші результати.

Зокрема, використовуючи таблицю D-H і функцію "links.m" з Robotics Toolbox, можна змоделювати будь-яку роботизовану руку і побудувати її схему таким чином, щоб дуже легко було розпізнати цікаві частини (такі як суглоби, основа, кінцевий ефект...). також, "fkine.m" і "jacob0.m" є двома важливими функціями для тесту.

link.m - ця функція встановлює значення з'єднання за допомогою таблиці D-H.

fkine.m - ця функція обчислює матрицю H зі спільних координат.

Jacob0.m - ця функція обчислює матрицю Якоба зі спільних координат.

Для керування рухом необхідно було об'єднати Robotics Toolbox з dual-quaternion Toolbox. Утиліти DQ були використані для отримання значень

орієнтації.

Кілька функцій було використано з dual-quaternion Toolbox, "rotMatrix2dquat.m" та "dquat2rot.m". Натомість "T2DQ.m" - з власного коду. Крім того, для цього типу управління також знадобилася функція "ikine.m" з Robotics Toolbox.

rotMatrix2dquat.m - перетворення матрицю R матрицю у подвійний кватерніон координати.

T2DQ.m - перетворення матриці H у подвійні кватерніонні координати.

dquat2rot.m - ця функція повертає 2 результати, тета-кут та кут вектор орієнтації (x,y,z), від координат подвійного кватерніону.

ikine.m - цей код виконує обернену операцію до "fkine.m", обчислюючи спільні координати з матриці H.

### 2.3 Висновки до другого розділу

1. Робот - це машина, здатна автоматично виконувати складну серію дій. Іншими словами, робот - це адаптивний та універсальний пристрій. Адаптивний, тому що він здатний переналаштовуватися для виконання поставленого завдання. Універсальний - тому що може виконувати кілька типів завдань..

2. Кожен суглоб має власний фрейм, тому існують способи зв'язати окремі суглоби за допомогою матричних обчислень. Всі ланки і суглоби, що утворюють роботизовану руку, називаються простором суглобів (Joint Space).

3. ПД - це промислова система управління, заснована на механізмі зворотного зв'язку. Блок складається з трьох регуляторів: пропорційний, інтегруючий та похідний. Система працює з диференціалами, або помилками, в якості входів. Ці помилки генеруються між поточним станом системи та

бажаним станом, і мета полягає в тому, щоб мінімізувати значення цієї помилки після декількох ітерацій.

4. Однорідна система є корисним математичним інструментом, який часто використовується в робототехніці. Вона використовується для координат та матриці перетворення.

5. Кватерніон - це число, що належить до системи числення  $\mathbb{H}$  і є похідним від комплексних чисел. Основна відмінність полягає в тому, що система комплексних чисел породжує ще один вимір  $(1, i)$ , а система кватерніонів породжує ще три виміри  $(1, i, j, k)$ .

### 3 ІМІТАЦІЙНА МОДЕЛЬ ПРОЦЕСУ КЕРУВАННЯ РОБОТОМ-МАНІПУЛЯТОРОМ

#### 3.1 Імітаційна модель процесу керування вектором швидкості або положення

Вектор може бути швидкістю або положенням, а напрямок завжди встановлювався на осі z інструменту (рисунок 3.1).

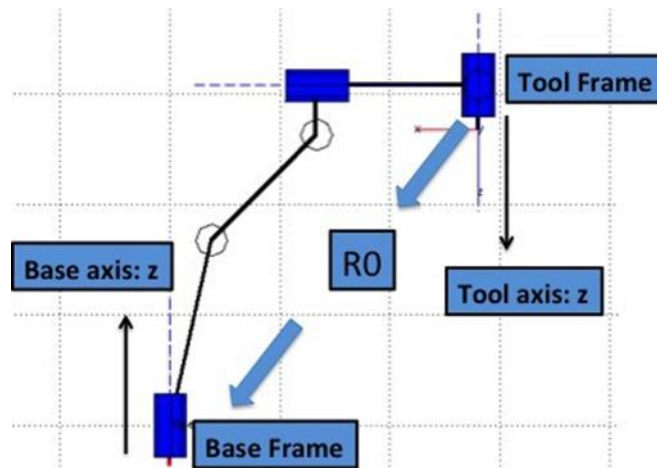


Рисунок 3.1 - Інструментальна рамка

Входи керування можна запускати з різних фреймів. Їх можна надсилати з Базового Кадру або з Початкового Кадру Інструменту. Технічно, ці два кадри безпосередньо пов'язані матрицею обертання  $R_0$ , але інтуїтивно зрозуміліше надсилати входи з Інструменту, оскільки вісь руху збігається з віссю Кадру.

Зв'язок є статичним протягом усієї процедури, тому матриця обертання зберігатиме ті самі значення. Це важливо, оскільки існують інші способи керувати рухом за допомогою оновлення матриці обертання, хоча для цього випадку це не є цікавим оскільки при використанні оновленої позиції інструменту відхилення або обчислення помилки буде розглядатися як

правильна ситуація, що змінює мету процедури. Однак ця методика може бути цікавою для експериментів, які потребують більшої адаптивності та універсальності.

Як згадувалося у вступі, вектор керування може бути налаштований за швидкістю або позицію.

З одного боку, у випадку швидкості використовується вектор вигляду [6x1].

$$V_{input} = [v_x \ v_y \ v_z \ w_x \ w_y \ w_z]^T. \quad (3.1)$$

З іншого боку, позиція генерується через вектор швидкості, залежно від значення ітерації для створення нової позиції (див. рівняння 3.2).

$$P_{input} = [v_x \ v_y \ v_z \ w_x \ w_y \ w_z]^T \cdot t \cdot i + P_0, \quad (3.2)$$

де  $t$  - час дискретизації;

$i$  - номер ітерації;

$P_0$  - початкова позиція.

Забезпечуючи плавність руху і контролюючи можливі відхилення вздовж різних осей, підпрограми обчислюють для кожної конфігурації кути Ейлера,  $\alpha$  і  $\beta$ , через матрицю  $H$  (рівняння 3.3). Ці значення позначають можливе відхилення вздовж декартової осі, яке може призвести до помилкового обчислення або недосягнення цілі.

Тому ці значення повинні знаходитися в невеликому діапазоні  $\epsilon$ , що розглядається як шум, навколо його початкового значення протягом усього

досліді. Обчислення цих значень здійснюється геометричними методами з використанням матриці обертання. (див. рівняння 3.3 та документ [24] для отримання додаткової інформації про ці обчислення).

$$\alpha = -\arcsi(H(3,1)); \beta = \arctan\left(\frac{H(3,2)}{H(3,3)}\right). \quad (3.3)$$

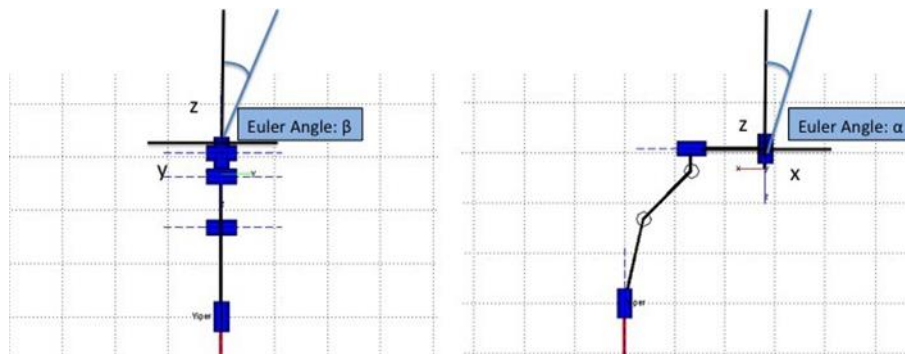


Рисунок 3.2 - Кути Ейлера

Крім того, всі процедури також проілюстровані вісьмома рисунками.

Положення і швидкість суглоба.

Декартове положення та швидкість.

Дві похибки (декартова та спільна), обчислені після ПІД.

Два кути Ейлера.

Зауважте, що, хоча робот керується у фреймі Tool Frame, накази формуються в Base, яка керує кінцевим ефектором. Тому результати на графіках віднесено до бази робота, а значення абсцис, не зазначені на графіках, є номерами семплів.

Як вже пояснювалося, інструментальна рамка зберігає ту ж саму орієнтацію, що й вісь руху, тому керувати роботом інтуїтивно зрозуміліше. У цьому випадку початкове положення суглоба задається вирівнюванням

інструментальної рамки відносно базової рамки, зі значенням повороту  $\pi$ , представленим через  $R_0$ . Значення суглобів такі.

$$q_{init} = [0 \quad -\pi/4 \quad 3\pi/4 \quad 0 \quad \pi/2 \quad 0]. \quad (3.4)$$

Ця конфігурація не була вибрана випадково. Як пояснювалося у вступі, вирівнювання з'єднань  $q_1$  і  $q_6$  може призвести до ситуації сингулярності. Але вона була обрана тому, що візуально дуже легко передбачити очікувані рухи. Якщо його характеристики будуть задовільними, можна буде екстраполювати інформацію і протестувати рух у загальній конфігурації.

Щоб керувати рухом з інструментальної рамки, команди повинні враховувати це обертання, тому вхідні вектори множаться на відношення між обома просторами інструменту та бази, тобто на матрицю обертання  $R_0$ . Таким чином, вектори набувають вигляду, наведеного нижче:

$$V_{Control} = R_0 \cdot [0 \quad 0 \quad v_z \quad 0 \quad 0 \quad w_z]^T; \quad (3.5)$$

$$P_{Control} = R_0 \cdot [0 \quad 0 \quad v_z \quad 0 \quad 0 \quad w_z]^T \cdot i \cdot t + P_0. \quad (3.6)$$

де -  $V$  - вектор швидкості, а  $P$  - вектор положення;

$P_0$  - початкова позиція робота;

$v_z$  та  $w_z$  - це лінійна та обертальна швидкості.

Для цього експерименту були встановлені значення:

-  $v_z = 0.01$  (м/с) (Matlab будує графіки значень з бази Robot'a, тому  $v_z$  протилежне:  $-0.01$  (м/с));

-  $w_z = \pm\pi/16$  (рад/с);

$i$  - значення ітерації;

$t$  - час дискретизації.

Використовуючи цей тип керування, було розроблено чотири різні схеми. Дві з них використовують швидкість, а дві інші - положення.

Очікувана швидкість на момент входу  $i$  в систему як  $V\_des_i$ . Це значення віднімається від поточної швидкості робота,  $V\_mes$ , генеруючи Error  $ev$ . Потім це значення проходить через різні математичні обчислення.

### 3.2 Імітаційна модель процесу керування роботом маніпулятором за допомогою ПД-регулятора

Декартовий ПД +  $J_{inv}$ .



Рисунок 3.3 – Швидкість - декартова процедура ПД

$J_{inv}$  + спільний ПД



Рисунок 3.4 – Швидкість - суглобова процедура ПД

У цьому випадку якобінська матриця працює з усіма з'єднаннями, тому її формою є матриця  $[6 \times 6]$ .

Нарешті, експонента перетворюється в приріст переміщення, який додається до поточного положення. Це обчислення позначено на діаграмах квадратиком "new q\_vel".

$$q_{i+1} = eqp \cdot t + q_i \quad (3.7)$$

На останньому кроці за допомогою похідної форми обчислюються швидкості суглобів, а потім конвертуються в декартовий простір.

$$qp_{mes} = \frac{q_{i+q} - q_i}{t}. \quad (3.8)$$

$$V_{mes_{i+1}} = J_{i+1} \cdot qp_{mes}. \quad (3.9)$$

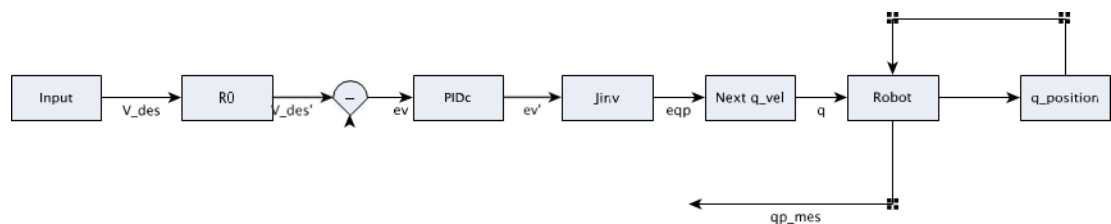


Рисунок 3.5 - Декартова швидкість і декартовий ПД

Для цієї конфігурації результати задовільні. Шарніри  $q_1$  та  $q_4$  не беруть участі в русі, а інші шарніри мають правильні тенденції. Швидкості суглобів задовільно збігаються до правильних значень і зменшуються через інерцію постійного руху. Значення похибок, як декартових ( $ev$ ), так і суглобових ( $eqp$ ), досить швидко збігаються до очікуваних значень.

Декартова швидкість ( $real-v$ ) дуже схожа на декартову похибку, м'яко досягаючи передбачуваних значень. Декартове положення кінцевого ефектора ( $real-pos$ ), як і очікувалося, слідує за лініями тренду з низхідним рухом по осі  $z$  і без змін по інших осях. Нарешті, кути Ейлера, помножені на 10–15, зберігають задовільні значення, обмежені нулем.

Ця варіація задовільно збігається з очікуваними результатами, як і попередня конфігурація, але прикметно, що на перших кроках спостерігаються незначні коливання. Ці варіації можуть зробити систему нестабільною для того, щоб отримати хорошу збіжність.

Наступні графіки показують ефективність цієї структурної схеми.

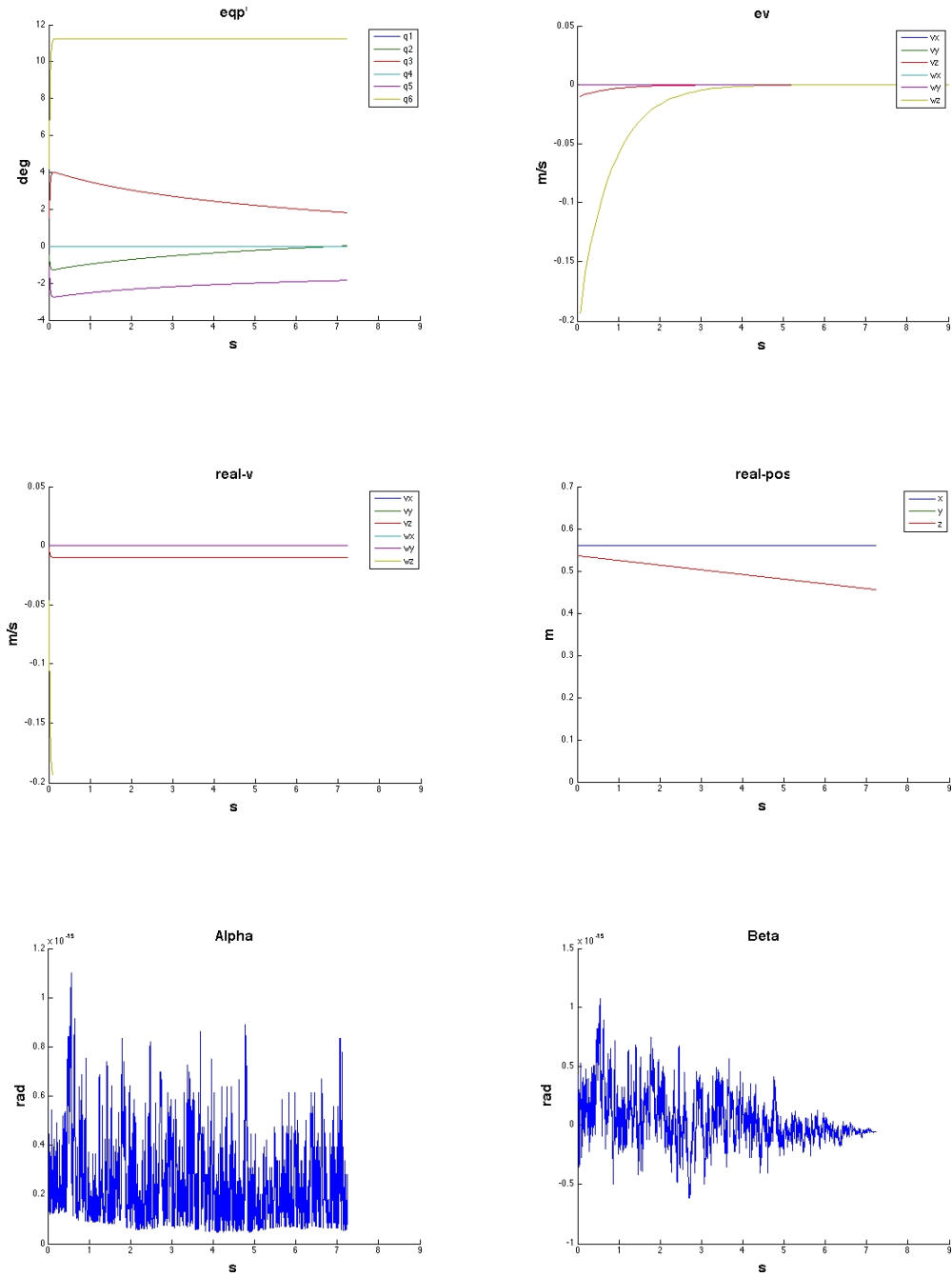


Рисунок 3.6 – Ефективність декартового ПД

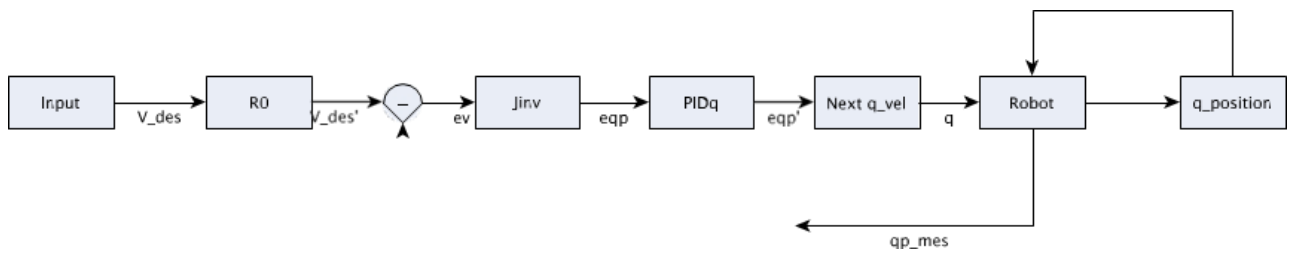


Рисунок 3.7 - Декартова швидкість і спільний ПД

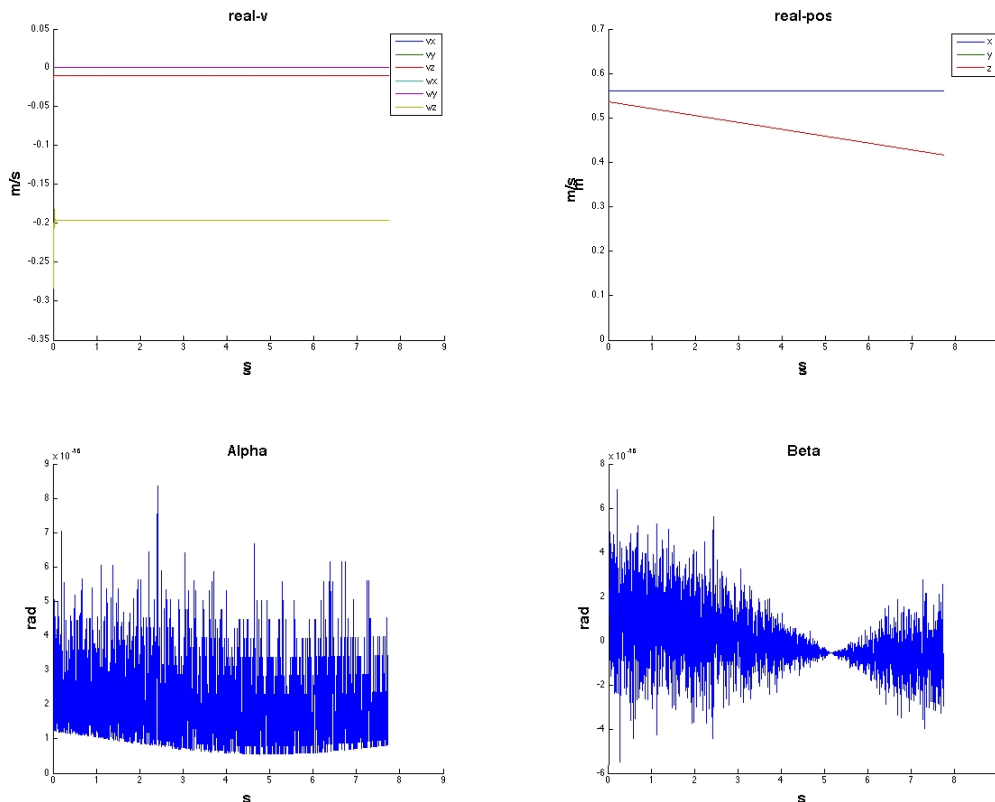


Рисунок 3.8 – Ефективність спільного ПД

Графік, що показує декартову швидкість (*real-v*), підтверджує коливання на початку експерименту. Однак, декартове переміщення вздовж осі та кути Ейлера зберігають задовільні значення протягом усієї процедури.

Різницю між обома схемами можна побачити на декартовій діаграмі похибки. Декартові ПД набагато м'якше сходяться до нуля порівняно зі спільною ПД-процедурою, яка коливається навколо нуля, перш ніж досягти збіжності.

Керування швидкістю дає хороші результати, але може бути проблематичним у неідеальній системі. Припустимо, що в реальному середовищі з шумом важливіше контролювати точність системи. Якщо робот керується постійною швидкістю, то після команди зупинки шум може змусити систему продовжувати рух, перевищуючи ціль або навіть межі безпеки в найгіршому випадку. Отже, ідея полягала в тому, щоб перетворити той самий вектор керування швидкістю у вектор керування положенням.

$$P_{Control} = R0 \cdot [0 \ 0 \ v_z \ 0 \ 0 \ w_z]^T \cdot i \cdot t + P_0. \quad (3.10)$$

Для того, щоб використовувати матрицю Якобіана, вхідні дані, що генеруються, повинні бути якомога ближчими до прецеденту, щоб генерувати невелику похибку.

Відокремлення обертання шостого суглоба від решти суглобів може стати цікавою альтернативою для керування рухом робота. Тепер перші п'ять суглобів відповідають за рух робота, а обертанням, незалежно від них, керує шостий суглоб. Як наслідок, яacobінська матриця змінює свою форму з [6x6] на [5x5]. Робота з цією гіпотезою дозволяє уникнути можливих проблем, пов'язаних з обчисленням орієнтацій, і спростити рядки коду, що призводить до легкого в обчислювальному плані рішення.

Обчислення нового положення  $q$  відрізняється від заданого для швидкості. Тепер розрахунок базується на положенні (поле діаграми: "new q pos") за цією формулою.

$$q_{i+1} = eq + q_i \quad (3.11)$$

Похибка отримана шляхом віднімання  $[[P\_des]]$  і поточного положення інструменту,  $[[P\_mes]]$ . Після цього ці значення проходять через дві різні системи, засновані на наступних прикладах швидкостей.

Значення ПД повністю змінюються в системах швидкостей. Використовуючи конфігурацію положення, згенеровану помилку буде набагато складніше анулювати. У цьому випадку система більше не шукає вирівнювання швидкості між виміряною та бажаною, а намагається згенерувати різницю між поточним та бажаним положенням. Таким чином, помилка завжди відмінна від нуля, і обмірковування помилки інтегрування та похідної в ПД повинно бути набагато меншим для входів положення.

На відміну від прикладів зі швидкістю, з використанням  $J$  та  $J_{inv}$ , тепер обчислення виміряного стану системи відбувається через DGM. У Robotics Toolbox ("fkine.m") цей розрахунок може спричинити деякі проблеми зі значеннями орієнтації. Для цього  $H$  було перетворено у простір DQ за допомогою рутини "T2DQ.m" з DQ Toolbox. Потім значення орієнтації,  $\theta$  і  $\hat{U}$ , були обчислені за допомогою "dquat2rot.m".

$$P_{mes} = \begin{bmatrix} fkine.m_{(3x1)} \\ \theta \cdot \hat{U}_{(3x1)} \end{bmatrix}. \quad (3.12)$$

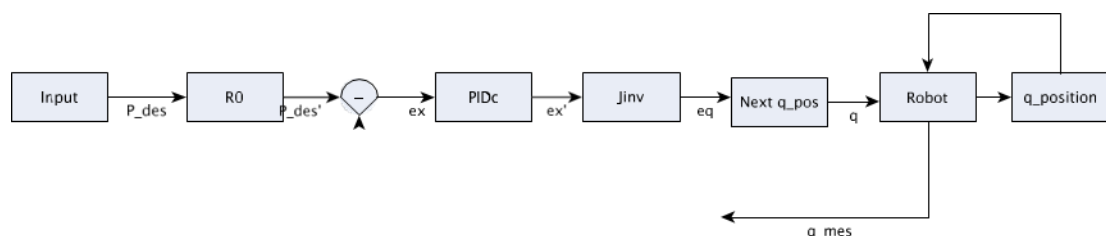


Рисунок 3.9 - Декартове положення з декартовим ПД

Значення суглобів генеруються правильно, викликаючи рух тільки в тих суглобах, які потрібні для руху. Дуже цікаво помітити, як криві, побудовані для швидкостей суглобів, сходяться, позначаючи м'яку траєкторію, що вказує на плавність процедури. Те саме відбувається для декартової швидкості (ex) та похибки суглоба (eq). Їхні криві сходяться, утворюючи м'який графік і досягаючи очікуваних значень.

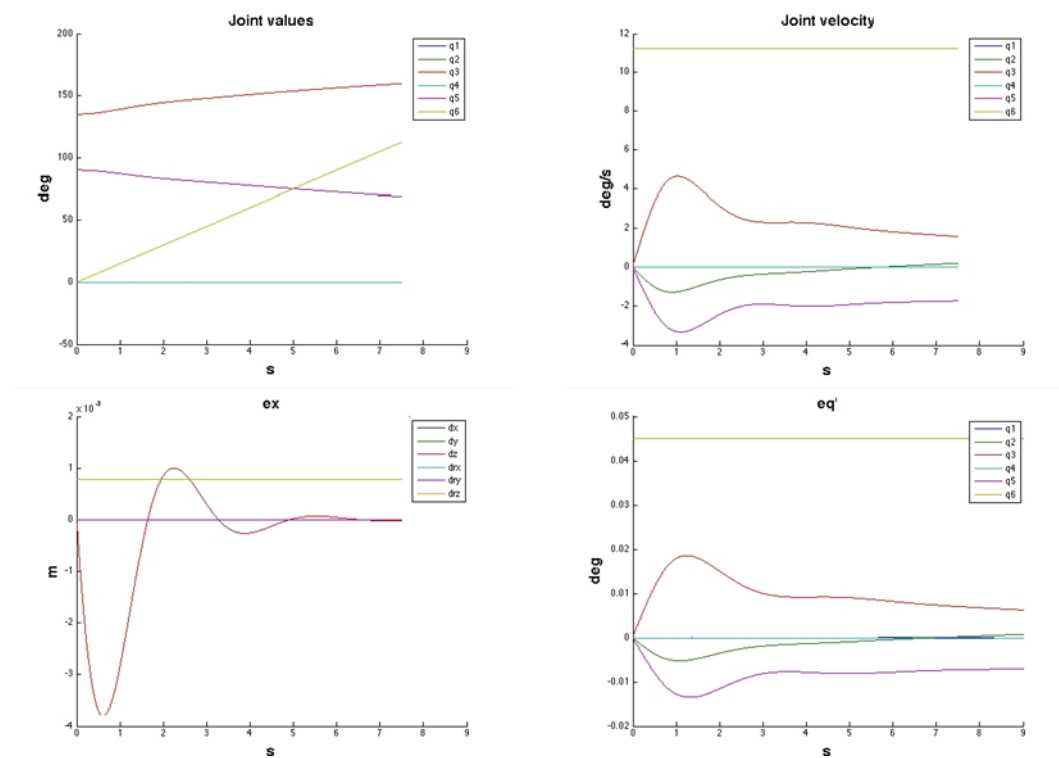


Рисунок 3.10 – Результати моделювання схеми екартове положення з декартовим

ПД

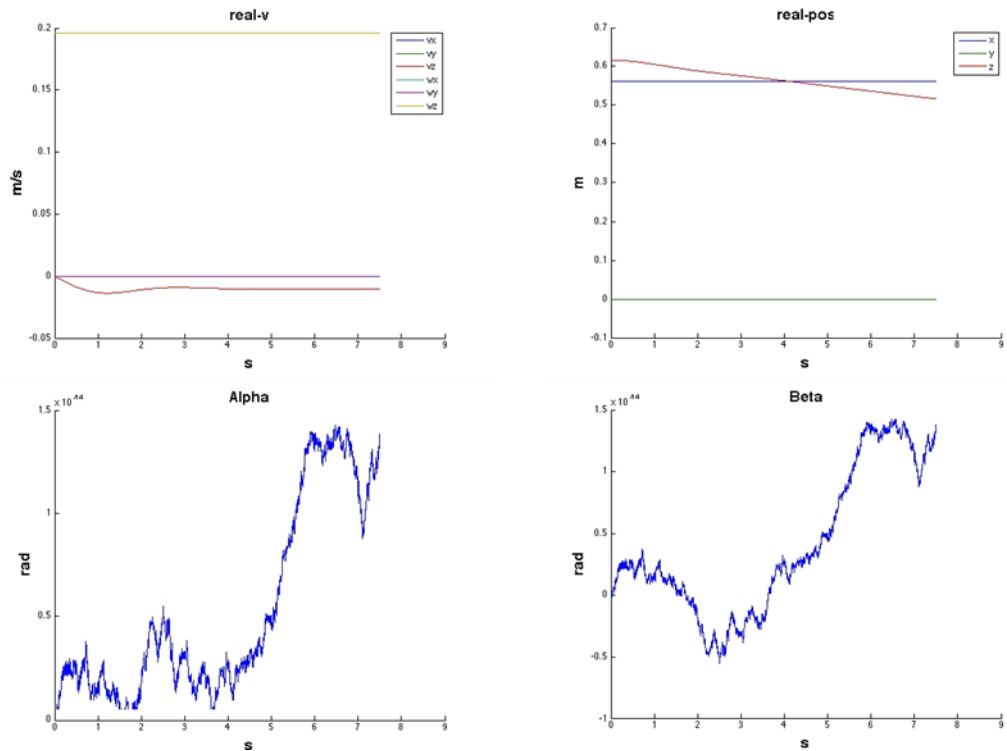


Рисунок 3.11 - Ідеальний випадок

Як видно з рисунка 3.11, декартова швидкість (real-v) правильно відповідає очікуванням. Обертання інструмента постійно зберігає одне й те саме значення, а лінійна швидкість вставки ідеально збігається з очікуваним значенням. Декартові значення слідує передбачуваним тенденціям лінії, а кути Ейлера, помножені на  $10^{-14}$ , правильно зберігають обмежені значення.

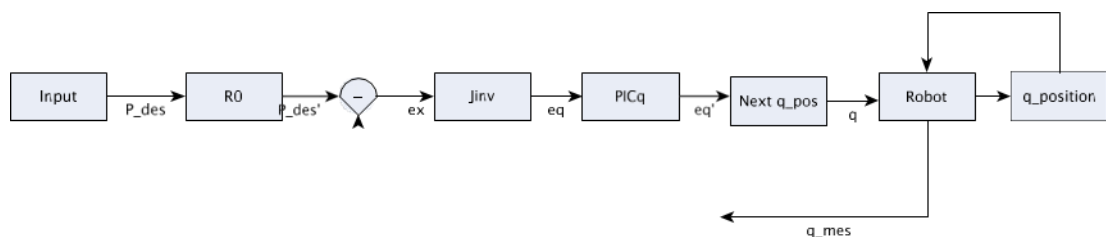


Рисунок 3.12 - Декартове положення та спільний ПІД

Показники для цього дослідження є задовільними. Дуже цікаво бачити, що коливання, які з'являлися у випадку входів швидкості зі спільним ПІД-регулятором, зникають. Лінії тренду слідує за плавним рухом,

генеруючи очікувані криві, і, як згадувалося раніше, декартова похибка змінюється, не досягаючи нульової похибки при використанні більшої кількості відліків в якості часової характеристики.

Показники, показані на наступних малюнках, знаходяться в межах хороших параметрів, що в цілому є дуже хорошим рішенням для нашої проблеми.

Ідеальні випадки дають дуже задовільні результати. Обидва випадки, швидкість і положення, дають хороші результати, але в положенні використання спільного ПД не досягає нульової помилки. На наступному кроці до експериментів було додано деякий шум, щоб спостерігати, як ці збурення впливають на результати.

### 3.3 Висновки до третього розділу

1. Забезпечуючи плавність руху і контролюючи можливі відхилення вздовж різних осей, підпрограми обчислюють для кожної конфігурації кути Ейлера,  $\alpha$  і  $\beta$ , через матрицю  $H$ . Ці значення позначають можливе відхилення вздовж декартової осі, яке може призвести до помилкового обчислення або недосягнення цілі. Тому ці значення повинні знаходитися в невеликому діапазоні  $\epsilon$ , що розглядається як шум, навколо його початкового значення протягом усього дослідження. Обчислення цих значень здійснюється геометричними методами з використанням матриці обертання.

2. Значення ПД повністю змінюються в системах швидкостей. Використовуючи конфігурацію положення, згенеровану помилку буде набагато складніше анулювати. У цьому випадку система більше не шукає вирівнювання швидкості між вимірною та бажаною, а намагається згенерувати різницю між поточним та бажаним положенням. Таким чином, помилка завжди відмінна від нуля, і обмірковування помилки інтегрування та

похідної в ПДД повинно бути набагато меншим для входів положення.

3. На відміну від прикладів зі швидкістю, з використанням  $J$  та  $J_{inv}$ , тепер обчислення виміряного стану системи відбувається через DGM. У Robotics Toolbox ("fkine.m") цей розрахунок може спричинити деякі проблеми зі значеннями орієнтації. Для цього  $H$  було перетворено у простір DQ за допомогою рутини "T2DQ.m" з DQ Toolbox. Потім значення орієнтації,  $\theta$  і  $U^{\wedge}$ , були обчислені за допомогою "dquat2rot.m".

## 4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ УДОСКОНАЛЕНОГО МЕТОДУ МАШИННОГО НАВЧАННЯ ДЛЯ КЕРУВАННЯ РОБОТОМ

### 4.1 Керування роботом в інструментальній рамці у певній конфігурації з шумом

Наступні схеми працюють так само, як і попередні, але з додаванням деякого спільного шуму при обчисленні спільних позицій.

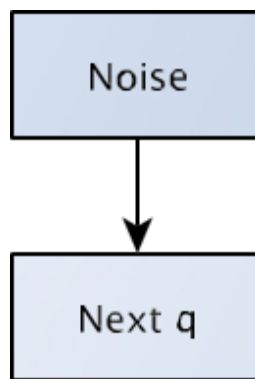


Рисунок 4.1 – Додавання шуму

Висновки, зроблені для ідеального випадку, можна екстраполювати на більшу частину цієї системи, але декартова похибка дає досить проблематичні результати, не дозволяючи розрізнити вхідну криву. Натомість, інші результати побудовані задовільним чином і дають цікаві результати.

Декартова швидкість по осях, яка повинна залишатися нульовою, має багато коливань навколо нуля. Крім того, цю інформацію можна доповнити кутами Ейлера. Значення кутів залишаються близькими до нуля з коефіцієнтом  $10^{-14}$ . Отже, значення гірші, ніж в ідеальному випадку, але все гарантувати плавність процесу.

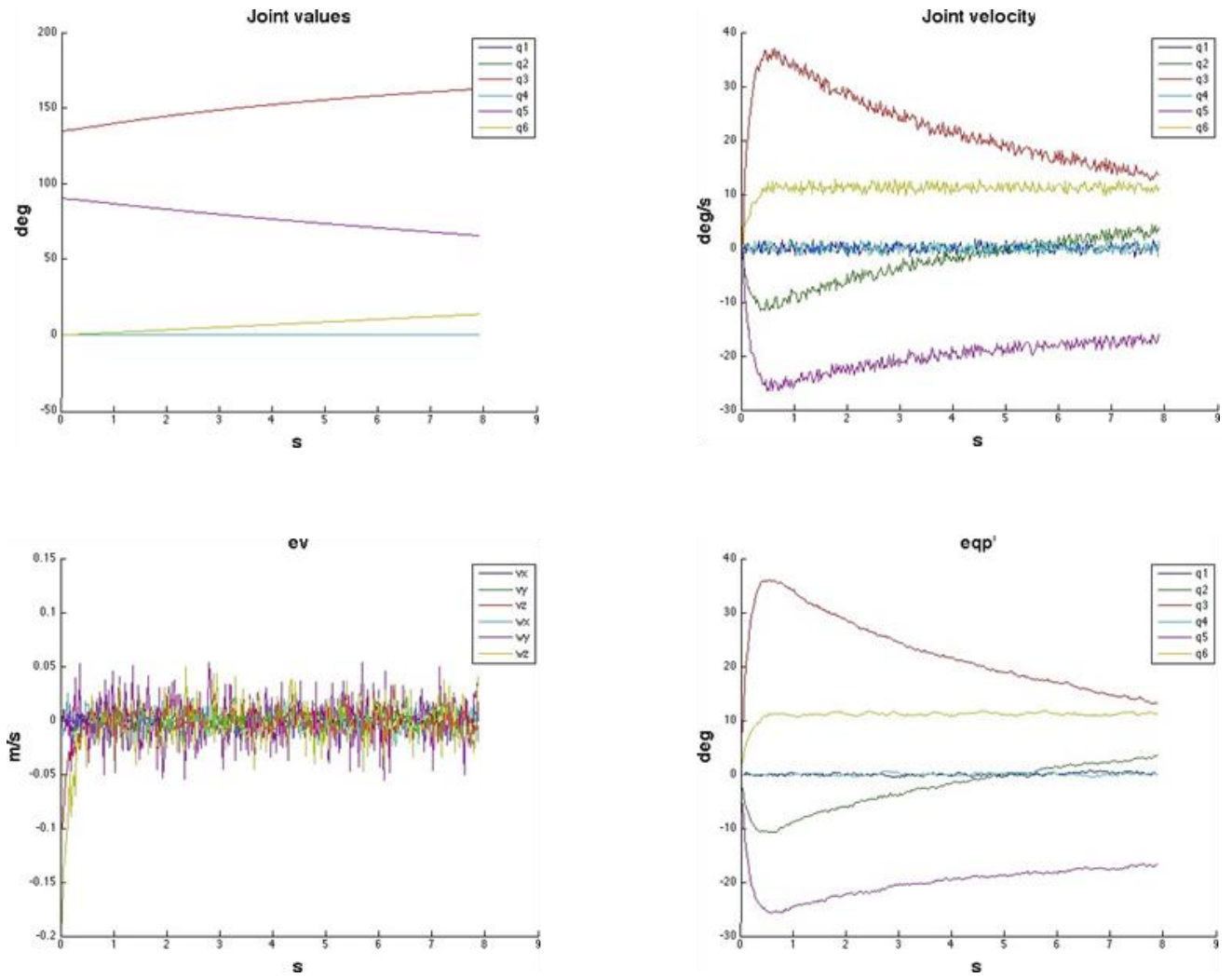


Рисунок 4.1 – Результати експериментального дослідження із шумом

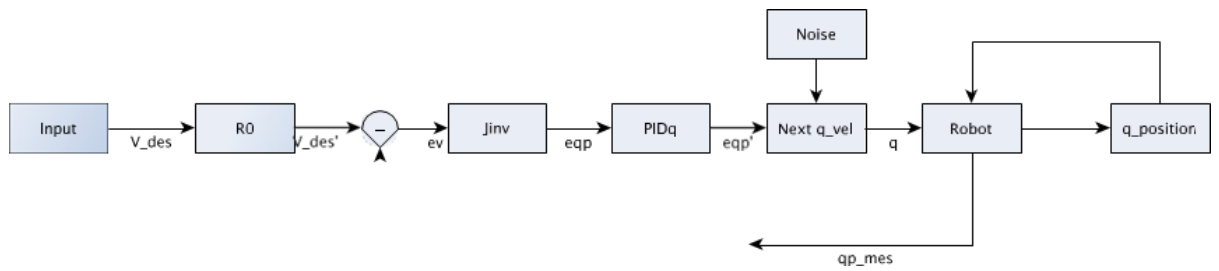


Рисунок 4.2 – Структурна схема ПД регулятора із шумом

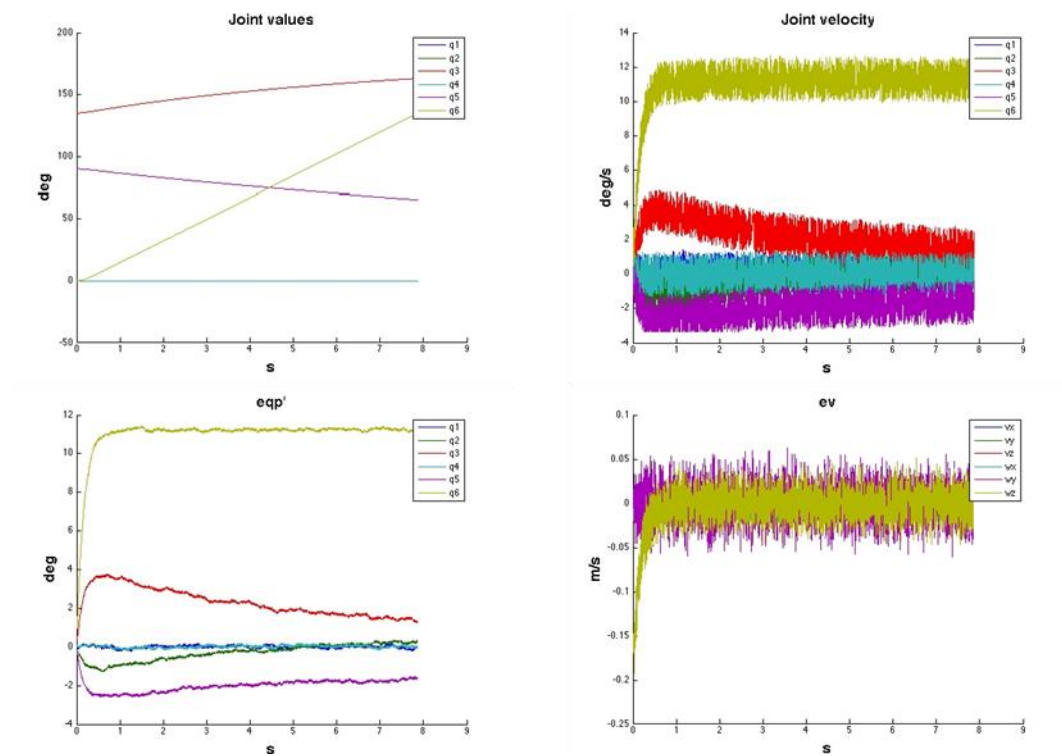


Рисунок 4.3 – Результати експериментального дослідження структурної схеми зі спільним ПІД

Сумарні значення та помилки слідують за очікуваними лініями тренду з деяким шумом. Симуляція руху робота дала правильний результат. Однак, суглобові швидкості та декартовий графік похибок викликають сумніви щодо результату в реальному русі з роботом.

Для декартової швидкості неможливо розрізнити лінійні швидкості вздовж різних осей. Це погана ситуація, тому що тільки швидкість обертання в напрямку до нуля може бути чітко визначена. В іншому випадку кути Ейлера, помножені на  $10^{-3}$  обмежені в правильних діапазонах.

Попередня гіпотеза для експерименту з позиційними входами з шумом полягає в тому, що в робочій позиції великі флуктуації, які з'являються у випадку швидкості, повинні бути набагато м'якшими або навіть нульовими.

Перша схема використовує декартові входи в поєднанні з декартовим ПІД з шумом.

Як показано нижче на рисунках. Примітно, що на даний момент результати узгоджуються з вищезгаданою гіпотезою. Лінії тренду такі ж самі, як

і при використанні вхідних даних про швидкість, однак крива має набагато кращий діапазон значень.

Фактично, рисунки показують покращення порівняно з тією ж конфігурацією з входами швидкості. Декартове положення і швидкість відповідають очікуваним результатам, з деякими варіаціями шуму. Також значення кутів Ейлера близькі до нуля.

На наступній схемі представлено декартове положення, спільний ПД та деякий шум.

На наведених нижче рисунках видно, що покращення є очевидними порівняно з тією ж конфігурацією, але з використанням входів зі швидкістю. Швидкість зчленування страждає від покращення для цієї продуктивності. Зчленування представлені правильно з деякими варіаціями через шум. Похибки показують передбачувані результати для цього досвіду, так само як і положення суглоба.

Наступний крок був зосереджений на екстраполяції висновків, зроблених для конкретного випадку, на загальний випадок. Тепер фрейм інструменту більше не вирівняно з базовим фреймом. До рами інструменту було застосовано поворот на  $\phi$  градусів відносно вертикальної осі основи робота.

Змодельовані дослідження повинні підтвердити масштабованість вектора керування, запропонованого після нової осі інструментальної вставки. Процедура відповідає тій самій гіпотезі про множення вектора керування на початкову матрицю обертання нової конфігурації рисунок 4.4.

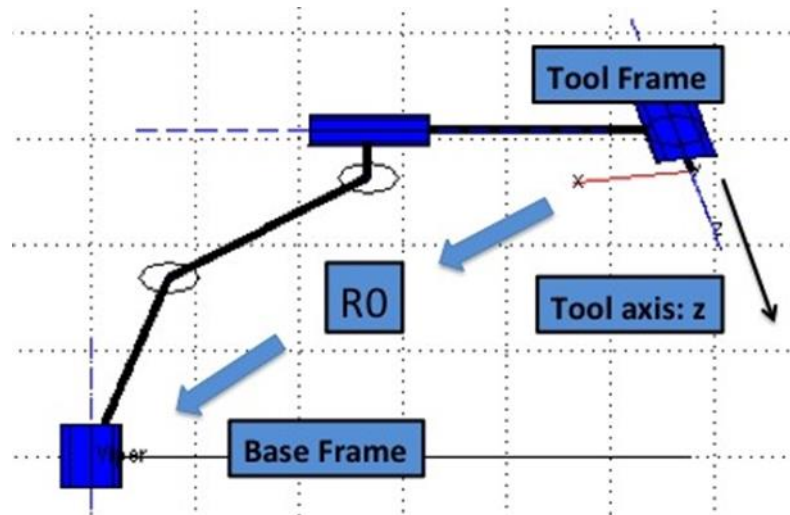


Рисунок 4.4 - Множення вектора керування на початкову матрицю обертання

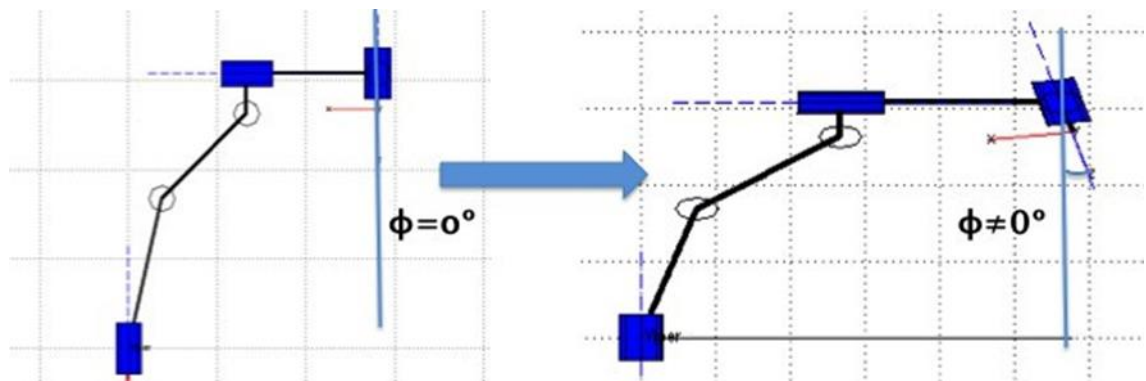


Рисунок 4.5 Початкове зміщення кута інструменту

$$q_{init} = \left[ 0 \quad -\frac{\pi}{4} \quad \frac{3\pi}{4} \quad 0 \quad \frac{\pi}{2} + \phi \quad 0 \right]. \quad (4.1)$$

Схеми, які ми випробували, були такими ж, як і в попередніх випадках. Однак вони були протестовані безпосередньо з шумом. Зауважте, що тепер рух відбувається у площині  $xz$  з постійним значенням  $y$ .

Отже, графіки, на які завжди посилаються відносно бази робота, як очікується, будуть на осях  $z$  та  $x$ . Примітно, що обчислення кута Ейлера  $\alpha$  більше не буде обмеженим навколо нуля відносно осі  $x$ , тому що в даний момент інструмент нахилений на  $\phi^\Omega$ . Однак,  $\beta$  все ще буде обмежений всередині невеликого діапазону навколо нуля.

Як наслідок, відносно бази робота рух тепер розкладається у двох напрямках. Отже, на рисунках буде показано цю декомпозицію для декартових швидкостей і положень. Для наведених нижче експериментів було обрано значення  $\phi = -\pi/16$ .

$$z_{base} = \cos\phi \cdot z_{tool}; \quad (4.2)$$

$$x_{base} = \sin\phi \cdot z_{tool}. \quad (4.3)$$

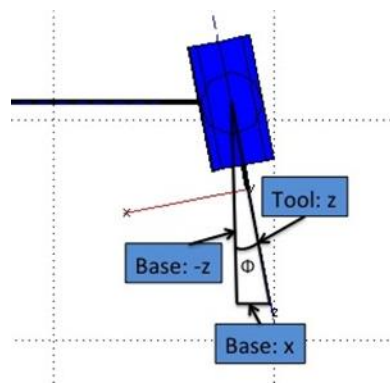


Рисунок 4.6 - Загальний випадок - тригонометричне відношення

Значення занадто сильно коливаються навколо очікуваних значень. Швидкості суглобів, що відповідають за рух робота, неможливо відрізнити одна від одної, і це може призвести до несподіваних рухів у реальній процедурі.

Декартова швидкість демонструє ті самі проблеми, що й швидкість суглоба. Більше того, обертання по осі у має несподіване значення, більше за нуль. На наступних рисунках підтверджується, що ця конфігурація не задовольнила очікувань. Коливання декартових швидкостей не є обмеженими, і їх значення не можна розрізнити. Більше того, кути Ейлера дають незадовільні результати, відхиляючись від очікуваного діапазону, втрачаючи цільову траєкторію.

Результати, отримані за допомогою цієї конфігурації, покращують результати, отримані раніше. Швидкості суглобів рухаються по м'яких траєкторіях, що коливаються лише під впливом шуму.

Результати, отримані за допомогою цієї конфігурації, покращують

результати, отримані раніше. Швидкості суглобів рухаються по м'яких траєкторіях, що коливаються лише під впливом шуму. Шарніри можуть бути ідеально дискриміновані. І спільні значення, і згенеровані помилки правильно відповідають очікуваним значенням. Зауважте, що декартові помилки мають чотири відмінні від нуля рядки:  $w_z$ ,  $v_x w_z$  і  $v_z$ .

#### 4.2 Схема модифікованого методу управління

Для наступних дослідів вхідні дані були введені в термінах положення. Важливо зазначити, що обертання контролювалося незалежно від руху. Тому лінійна швидкість виглядає розкладеною по осях  $x$  і  $z$ , але не по обертанню.

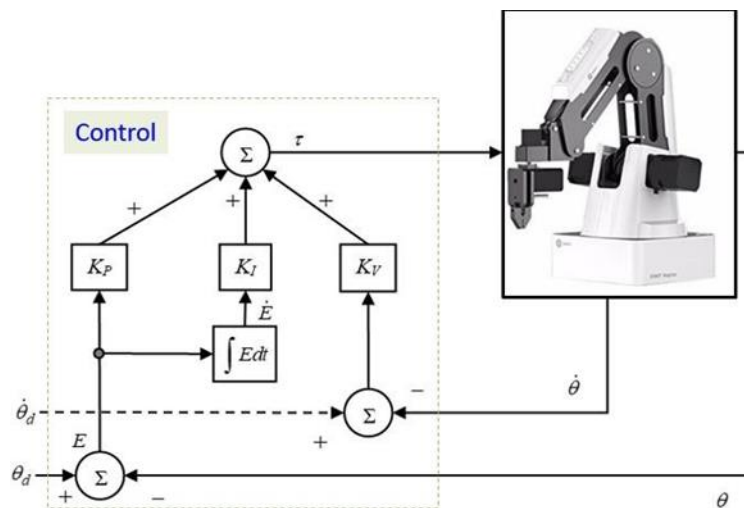


Рисунок 4.7 - Принципова схема ПІД-регулювання

Загальну схему ПІД-регулятора зображено на рисунку 4.7. Метод ПІД-регулювання є найпоширенішим методом регулювання для промислових застосувань (Крейг, 2014). Він простий за конструкцією, ефективний в обчисленнях і не вимагає динамічної моделі системи. Як показано на схемі на рисунку 4.7, за допомогою ПІД-регулятора команди спільного крутного моменту для керування роботом можна виразити наступним рівнянням:

$$\tau = K_p(\theta_d - \theta) + K_v(\dot{\theta}_d - \dot{\theta}) + K_I \int (\theta_d - \theta) dt \quad (4.4)$$

де  $\theta_d$ ,  $\theta \in \mathbb{R}^4$  - вектори шуканого та виміряного кутів з'єднання відповідно,  $\dot{\theta}_d$ ,  $\dot{\theta} \in \mathbb{R}^4$  вектори шуканої та виміряної швидкостей суглобів відповідно:

$K_p, K_v, K_I$  - діагональні матриці додатних визначених коефіцієнтів підсилення, а  $\tau \in \mathbb{R}^4$  - узагальнений вектор крутного моменту.

Нехай вектор похибки  $E$  та її похідна виражаються виразом (4.5).

$$E = \theta_d - \theta; \dot{E} = \dot{\theta}_d - \dot{\theta}. \quad (4.5)$$

Тому рівняння (4.4) можна переформулювати як рівняння похибки (4.6).

$$\tau = K_p E + K_v \dot{E} + K_I \int E dt. \quad (4.6)$$

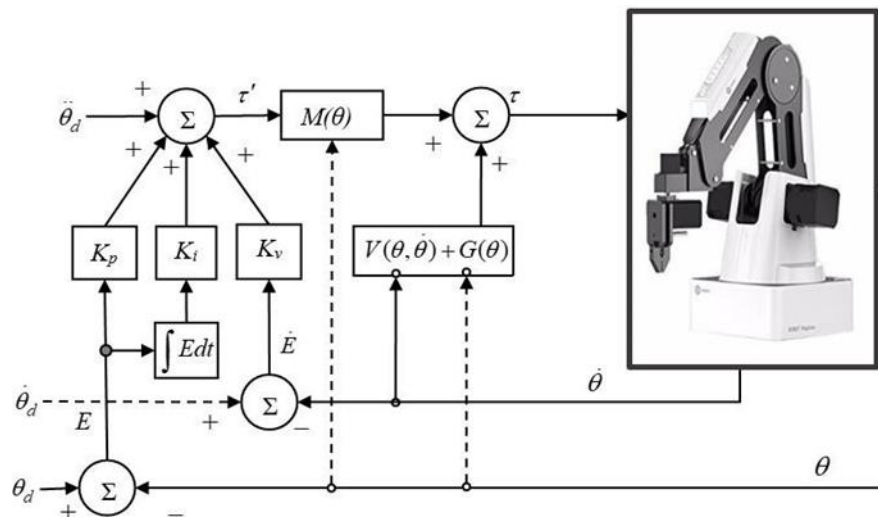


Рисунок 4.8 - Принципова схема обчислювального керування крутним моментом

Обчислювальне керування крутним моментом включає в себе динамічну модель системи, тому за допомогою СТС можна досягти кращої продуктивності

відстеження, якщо динамічна модель є точною. Динамічну поведінку Dobot можна виразити відомим рівнянням динаміки жорсткого тіла:

$$M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta)\tau. \quad (4.7)$$

де  $\theta \in \mathbb{R}^4$  - вектор спільних змінних;

$\tau$  - узагальнений вектор крутного моменту;

$M(\theta) \in \mathbb{R}^{4 \times 4}$  - матриця інерції;

$V(\theta, \dot{\theta}) \in \mathbb{R}^4$  - вектор Кориоліса/відцентровий вектор;

$G(\theta) \in \mathbb{R}^4$  - вектор гравітації.

$$\ddot{\theta} = -M^{-1}(\theta)[V(\theta, \dot{\theta}) + G(\theta)] + M^{-1}(\theta)\tau, \quad (4.8)$$

де  $M^{-1}(\theta)$  завжди існує, оскільки  $M(\theta)$  симетрична і додатньо-визначена.

Схема модифікованого методу управління обчислюваним моментом зображена на рисунку 4.8. На відміну від звичайного підходу до керування обчислюваним моментом, тут ми додали інтегральний член, щоб мати кращу продуктивність відстеження і компенсувати помилку відстеження траєкторії, яка зазвичай виникає через недосконалість динамічного моделювання, оцінки параметрів, а також через зовнішні збурення.

Загальна схема архітектури керування для "декартової траєкторії з відстеженням на основі суглобів control" показано на рисунку 4.9, де  $\theta_d, \dot{\theta}_d, \ddot{\theta}_d$  представляють бажане положення суглоба, швидкість та прискорення відповідно, а для декартових координат -  $x_d, \dot{x}_d, \ddot{x}_d$  відповідно.

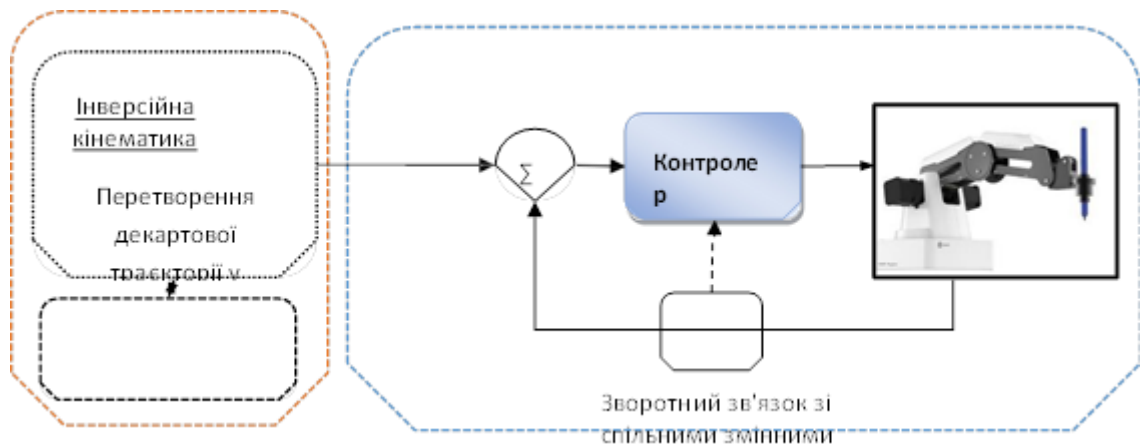


Рисунок 4.9 - Відстеження декартової траєкторії зі спільним керуванням

### 4.3 Розробка програмного забезпечення

Dobot Magician використовує програмне забезпечення під назвою Dobot Studio, інтерфейс якого зображений на рисунку 4.10.



Рисунок 4.10 - Інтерфейс програмного забезпечення Dobot Studio

Програмне забезпечення Dobot Studio працює на основі графічного інтерфейсу користувача (ГІК). Графічний інтерфейс користувача має такі

функції, як надання команд Dobot для виконання таких операцій, як писання та малювання, лазерне гравірування, 3D-друк і керування роботом за допомогою жестів рукою. Графічний інтерфейс користувача має ще дві функції блок і скрипт для графічного та скриптового програмування відповідно.

Виробництво Dobot Magician надає користувачам SDK на різних мовах програмування для додаткової розробки Dobot Magician. У цьому дослідженні Dobotdll.dll і C# SDK використовуються для отримання доступу до даних датчиків і виконання переміщення від точки до точки кінцевого ефектора Dobot і ковзної рейки. C# SDK Kinect V2 також використовується в цьому проекті для доступу до функції мовлення сенсора Kinect.

DobotDll.dll - це підтримуваний файл DLL, необхідний для отримання доступу до MCU материнської плати Dobot Magician. Повною формою DLL є динамічна бібліотека, яка містить набір даних і код для виконання певних операцій в операційній системі Windows. Попередньо скомпільований вихідний код файлу DLL вивчається інтуїтивно, щоб зрозуміти принцип роботи Dobot Magician.

Інтерфейс прикладного програмування (API) у комп'ютерному програмуванні - це набір протоколів зв'язку, визначень підпрограм та інструментів для розробки програмного забезпечення. Контролер Dobot підтримує два типи команд: негайна команда та команда черги. У розробці програмного забезпечення Dobot C# обидві команди реалізуються. Негайна команда виконується контролером Dobot відразу після отримання команди незалежно від того, обробляються інші команди чи ні. З іншого боку, команди в черзі виконуються контролером Dobot, зберігаючи послідовність отримання команди.

Протокол зв'язку Dobot використовується для побудови зв'язку з Dobot. Зв'язок може бути встановлений через послідовний порт USB, послідовний порт рівня TTL, Wi-Fi (UDP), де фізичний рівень отримує необроблені дані довжиною вісім бітів для визначення початку та кінця даних, а також для перевірки точності даних. Протокол зв'язку - це комбінація заголовка пакета, завантаження

пакета та контрольної суми для забезпечення правильного зв'язку.

C# - потужна мова програмування для розробки додатків у середовищі .NET. Це сучасна мова програмування, яка має такі особливості, як можливість об'єктно-орієнтованого програмування, багата бібліотека та зручне середовище кодування, що робить її кращою для розробників програм.

Операційна система роботів (ОСР) - це набір інструментів, бібліотек і погоджень. ОСР є потужною основою для програмування програмного забезпечення роботів, а мобільні роботизовані системи на основі ROS стають все більш важливими в житті людини.

Основна мета ОСР полягає в тому, щоб надати відкриту, конфігуровану та уніфіковану структуру програмування, яка використовується для керування роботами в різних симульованих середовищах і навіть у реальному світі. ОСР зараз оснащена набором інструментів, бібліотек і угод і є потужною структурою для програмування програмного забезпечення роботів.

Системи мобільних роботів на основі ОСР стають все більш важливими в різних аспектах життя людини. ОСР не тільки спрощує місії зі створення складної та надійної поведінки роботів на широкому спектрі роботизованих платформ, але також забезпечує недороге рішення для студентів у ситуаціях, коли необхідно виконати різноманітні симуляції, щоб завершити проект у прискореному темпі.

ОСР містить кілька компонентів, включаючи вузли, головний сервер, сервер параметрів, повідомлення, теми та служби. Вузли є основною функціональною одиницею для ОСР і вони підтримують мови C++ та Python. Майстер - це центральний процес, який дозволяє різним вузлам знаходити один одного, щоб спілкуватися. Зв'язок здійснюється шляхом надсилання та отримання даних до або з теми.

У нашій програмі зв'язок реалізується двома типами вузлів, зокрема вузлом публікації та вузлом підписки. Вузли, які виконують публікацію, публікуватимуть дані для тем позиції об'єкта та оцінки позиції, а потім вузли підписки підписуються на ці теми та використовують ці дані для керування

роботом. Для програми керування роботом існують взаємодії запиту та відповіді, які можна реалізувати через службу ОСР як DobotServer.

Для програми керування роботом растосовуються чотири модулі розширення: захват, повітряна присоска, фотоелектричний перемикач і датчик кольору. Захват встановлюється шестигранним ключем на 2,5 мм. Після встановлення сигнальна лінія підключається до GP3, який є клемою передпліччя.

Блок повітряного насоса пневматично керує таким обладнанням, як присоска та захоплювач. Його встановлюють шляхом підключення повітряної труби до обладнання та встановлення сигнальних і керуючих ліній до GP1 і SW1 на інтерфейсі введення/виведення на задній частині основи.

Модуль конвеєрної стрічки приводить в рух конвеєрну стрічку через кроковий двигун, а двигун приводиться в рух через чотири штирі крокового механізму. Фотоелектричний перемикач встановлений поруч із конвеєрною стрічкою, а сигнальна лінія встановлена на GP2 на базовому інтерфейсі введення/виведення.

Мова Python використовується як платформа розробки та застосовується перетворення Гафа (ПГ), щоб знайти кут повороту кожного рухомого об'єкта на конвеєрі. Ця інформація дозволяє кінцевому шарніру маніпулятора обертатися під тим же кутом, щоб захопити об'єкт.

ПГ вперше було запропоновано Полом Гафом у 1962 році та популяризовано Дудою та Хартом у 1972 році. Це технологія виділення ознак для обробки зображень, яка розпізнає геометричні форми. Класичний ПГ використовується для виявлення прямих ліній, хоча його можна розширити для розпізнавання кіл, еліпсів і навіть неправильних форм.

На практиці прямокутні системи координат не можна вибрати в просторі зображення, оскільки для будь-якої вертикальної лінії, наприклад  $x=c$ , нахил є нескінченним. Приймаємо полярні координати  $(\rho, \theta)$  у нормальній формі Гессе, як показано на рисунку 4.11.

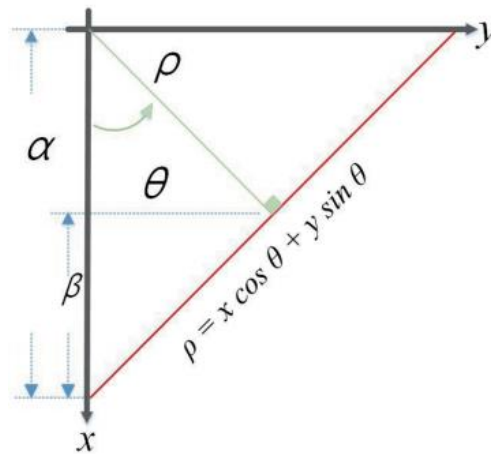


Рисунок 4.11 - Полярні координати  $(\rho, \theta)$  у нормальній формі Гессе

На рисунку 4.11  $\rho$  - це вертикальна відстань від початку координат до точки, а  $\theta$  - це кут між віссю  $x$  і вертикальною лінією. Тоді можна отримати рівняння:

$$\rho = \alpha \cos \theta = (x + \beta) \cos \theta = (x + y \tan \theta) \cos \theta = x \cos \theta + y \sin \theta. \quad (4.9)$$

У виразі (4.9)  $\rho$  та  $\theta$  полярні координати, які визначають рівняння лінії в просторі зображення, де  $\rho$  - вертикальна відстань від початку координат до лінії, а  $\theta$  - кут між віссю та вертикальною лінією. Подальші інструкції наведені в полярних координатах  $(\rho, \theta)$ .

Етапи виконання ПГ.

Етап 1. Встановлення та ініціалізація накопичувача 2D-масиву. Встановлення для всіх елементів масиву значення 0. Для полярних координат горизонтальна вісь -  $\theta$ , а вертикальна -  $\rho$ .

Етап 2. Визначення розміру масиву, який безпосередньо впливає на точність. Якщо  $\theta$  має точність до  $1^\circ$ , потрібно від 0 до 180 ліній, тоді абсциса може досягати 3,14 рад. Звідси випливає, що  $n=181$ . Діагональна відстань  $m$  для  $\rho$  повинна використовуватися як кількість стовпців. Іншими словами, розмір масиву дорівнює  $M=R^{m \times n}$ .

Етап 3. Припустимо, що на зображенні є пряма лінія, а її кінці мають

піксельні координати  $A=(a_1, b_1)$  та  $B=(a_N, b_N)$ . Припустимо, що  $N$  пікселів скануються камерою з точки  $A$  до точки  $B$  і ці пікселі визначаються від  $P_1$  до  $P_N$ . Вибір першої точки на лінії  $P_1=A=(a_1, b_1)$  і підстановка цих координат  $x$  і  $y$  в рівняння (4.9), в результаті отримаємо:

$$\rho = a_1 \cos\theta + b_1 y \sin\theta. \quad (4.10)$$

Етап 4. Заміна цілих чисел від  $0^\circ$  до  $180^\circ$  для  $\theta$  в рівнянні (4.10), тому що  $\theta$  має точність  $1^\circ$ . В результаті отримаємо числові пари  $(\rho_0, 0)$ ,  $(\rho_1, 1)$ ,  $(\rho_2, 2)$ , ...,  $(\rho_{180}, 180)$ . Ці 181 координати відповідають масиву голосування накопичувача  $M$ .

Далі додаємо по одній до кожної відповідної позиції. Якщо є точка в серії числових пар  $(\rho^*, \theta^*)$ , то голосування накопичувача масиву встановлено значення  $M(\rho^*, \theta^*)=1$ , де  $\rho^*$  - кількість пікселів, а  $\theta^*$  - кількість градусів.

Етап 5. Вибір усіх точки на лінії  $P_1=(a_1, b_1) - P_N=(a_N, b_N)$  і повторення етапів 3–4, щоб оновити накопичувальний масив голосування  $M$ . Після того, як точка на прямій буде вибрана, одиниця додається до вмісту 181 позиції накопичувального масиву.

Етап 6. Вибір двох кінцевих точок як піксельних координат  $A=(a_1, b_1)$  та  $B=(a_N, b_N)$ . Процес голосування здійснюється за наступним алгоритмом.

1. Загальна кількість голосів дорівнює  $n \times N$ , де  $n=181$ .
2. Загальна кількість кандидатів (розмір накопичувального масиву) дорівнює  $m \times n$ .
3. Розрахунок накопичувального голосуючого масиву та знаходження максимального значення у відповідності до виразу (4.11).

$$(\rho^*, \theta^*) = \arg \max(M(\rho, \theta)), \quad (4.11)$$

де  $0 \leq \rho \leq m - 1$ ;  $0 \leq \theta \leq n - 1$ .

4. Якщо  $M(\rho^*, \theta^*)$  перевищує певний поріг, то у просторі зображення існує пряма лінія. Крім того, припустимо, що координати пікового значення  $(\rho^*,$

$\theta^*)=(71, 45)$ , що означає, що відстань від прямої до початку координат становить 71 піксель, а кут між її вертикальною лінією та осі x дорівнює  $45^\circ$ .

Система автоматичного керування роботом побудована на основі мікрокомп'ютера RaspberryPI Model B+, який передає сигнали керування електродвигунами для обертання робота. Сигнали керування формуються мікрокомп'ютером на основі зображення, отриманого з відеокамери.

Значення кутів повороту передаються до мікрокомп'ютера RaspberryPI Model B+ двома енкодерами, що закріплені, відповідно, на осях обертання. Мікрокомп'ютер може здійснювати обмін даними з ПК та іншими пристроями через провідну мережу (Ethernet) та безпроводну мережу (WiFi), що необхідно для дистанційного керування через веб-інтерфейс.

Мікрокомп'ютер RaspberryPI Model B+, що використовується для автоматизованого керування роботом, зображено на рисунку 4.11.

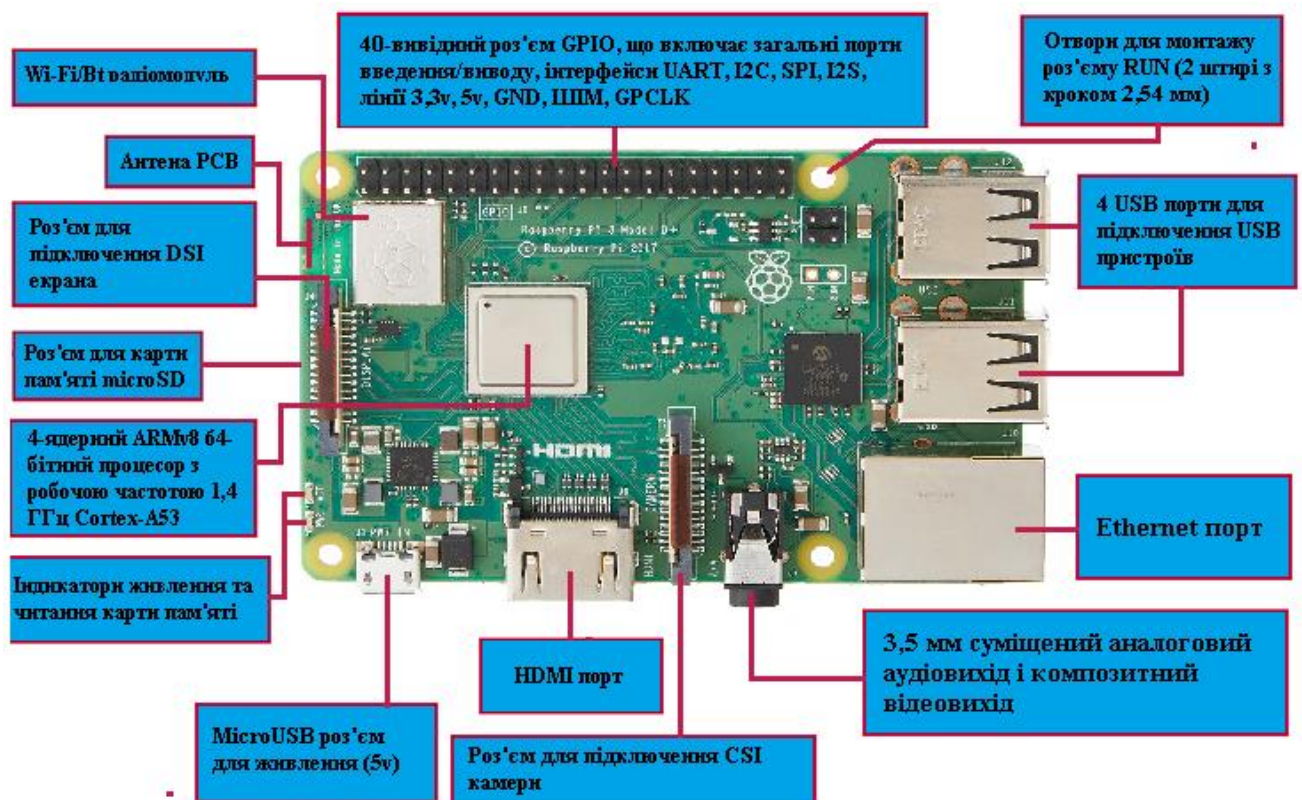


Рисунок 4.11 - Одноплатний мікрокомп'ютер RaspberryPI Model B+

RaspberryPI Model B+ є одноплатним комп'ютером на основі SoC

BCM2837B0, що побудована на базі чотириядерного 64-розрядного процесора ARMv8 з тактовою частотою 1,4ГГц та достатньо продуктивним відеопроцесором.

Характеристики мікрокомп'ютера RaspberryPI Model B+:

- SoC: Broadcom BCM2837B0 (CPU, GPU, DSP, and RAM);
- ЦП: ARMv8 Cortex-A53 такт. частота 1,4 ГГц, 64 розр., 4 ядра;
- ГП: VideoCore IV 3D;
- оперативна пам'ять: 1 Гб LPDDR2 (900 МГц).

Порти мікрокомп'ютера:

- USB: USB 2.0 - 4 шт;
- відео та аудіо (аналоговий порт): 3,5 мм jack (4 pin);
- відео та аудіо (цифровий порт): HDMI;
- відеокамера: MIPI Camera Serial Interface (CSI-2);
- провідна мережа інтернет: 10/100 Ethernet;
- дисплей: Display Serial Interface (DSI);
- додатковий накопичувач: microSD;
- порти введення та виведення: 40 pin GPIO.

Безпроводні інтерфейси:

- Bluetooth 4.2 Classic і Low Energy (BLE)
- WiFi: двосмуговий 2,4 ГГц та 5 ГГц IEEE 802.11.b / g / n / ac 2,4 ГГц.

Для визначення положення робота використовується відеокамера Canyon CNR-WCAM820, яка зображена на рисунку 4.12.

Відеокамера фіксує зображення та передає відеосигнал до мікрокомп'ютера RaspberryPI Model B+ з метою аналізу та формування команд повороту робота.

Технічні характеристики відеокамери:

- матриця: 2 млн пікселів., CMOS;
- роздільна здатність (відео): 1600x1200;
- максимальна частота кадрів: 30 Гц;
- формат зображення: 640x480 при 30 Гц 1600x1200 при 15 Гц;

- глибина кольору: 24-розрядний;
- підключення: USB 2.0;
- мікрофон: вбудований;
- кут огляду об'єктива :70 град;
- розміри (ШхВхГ): 72x42x65 мм;
- вага: 96 гр.



Рисунок 4.12 - Відеокамера Canyon CNR-WCAM820

Спочатку виконується завантаження бібліотек та ініціалізація відеокамери. Встановлюються параметри відеозображення та початкові умови для процесу автоматизованого керування роботом. Виконується ініціалізація бази даних Redis.

Згідно з даними відеокамери далі виконується захоплення зображення з камери. Здійснюється фільтрація зображення для знаходження області деталі, яку повинен взяти робот. Виконується аналіз розмірів та округлості контуру

знайденої області.

Якщо довжина та округлість контуру відповідають відображенню деталі, яку повинен взяти робот, то відбувається визначення центру контуру області. Розраховується зміщення центру контуру відносно центру зображення. Величина зміщення центру зберігається у базі даних Redis (TTL 5с). Отримані дані передаються для формування команді керування електродвигунами з метою мінімізації зміщення центру.

Положення деталі, яку повинен взяти робот, може візуалізуватися у веб-інтерфейсі програмного забезпечення робота. Якщо довжина та округлість отриманого контуру деталі, яку повинен взяти робот мають складну форму та/або велику площу, то виконується розширений аналіз з використанням додаткових даних. На основі результатів аналізу передаються дані для формування команд керування електродвигунами робота.

На початковому етапі виконується завантаження часу, бібліотек та встановлюється зв'язок з базою даних Redis. Формується список приєднаних клієнтів. При приєднанні нового клієнта викликається функція `async def on_connect`, яка керує очікуванням та обробкою команд від клієнта за допомогою функції `async def process_command`. Далі виконуються дії відповідно до отриманих команд та подаються керуючі сигнали на виводи GPIO мікрокомп'ютера Raspberry Pi Model B+.

Далі перевіряється існування та виконується перетворення даних у базі Redis. Відповідь формується у форматі JSON, і відправляється клієнту. Підпрограма виконує оброблення даних та команд, що надходять із різних пристроїв та дозволяє керувати параметрами робота.

Для початку на плату Raspberry Pi подається живлення, після чого можна запуснути написаний код. Також важливо щоб до плати була під'єднана камера, з якої саме буде йти відео-потік і йде перевірка на працездатність камери. Після обробки, кадр переходить до аналізу за допомогою бібліотеки для комп'ютерного зору OpenCV. Алгоритм програмного забезпечення зображено на рисунку 4.13.



Рисунок 4.13 - Алгоритм програмного забезпечення

Картинка конвертується у градації сірого кольору і починається визначення об'єктів з потокового відео. Йде кожна перевірка на відповідний об'єкт, де до кожного встановлений свій, відповідний класифікатор.

Якщо об'єкт не знайдений, процес повторюється. Ця операція проводиться для кожного встановленого класифікатора. Після обробки кадру, картинка нормалізується і на вихід вже подається відео відповідного формату, з обробкою об'єктів і їх визначенням.

#### 4.4 Висновки до четвертого розділу

1. Висновки, зроблені для ідеального випадку, можна екстраполювати на більшу частину цієї системи, але декартова похибка дає досить проблематичні результати, не дозволяючи розрізнити вхідну криву. Натомість, інші результати побудовані задовільним чином і дають цікаві результати.

2. Декартова швидкість по осях, яка повинна залишатися нульовою, має багато коливань навколо нуля. Крім того, цю інформацію можна доповнити кутами Ейлера. Значення кутів залишаються близькими до нуля. Отже, значення гірші, ніж в ідеальному випадку, але вони гарантують плавність процесу.

3. Сумарні значення та помилки слідує за очікуваними лініями тренду з деяким шумом. Симуляція руху робота дала правильний результат. Однак, суглобові швидкості та декартовий графік похибок викликають сумніви щодо результату в реальному русі з роботом.

4. Для декартової швидкості неможливо розрізнити лінійні швидкості вздовж різних осей. Це погана ситуація, тому що тільки швидкість обертання в напрямку до нуля може бути чітко визначена. В іншому випадку кути Ейлера, помножені на  $10^{-3}$  обмежені в правильних діапазонах.

5. Попередня гіпотеза для експерименту з позиційними входами з шумом полягає в тому, що в робочій позиції великі флуктуації, які з'являються у випадку швидкості, повинні бути набагато м'якшими або навіть нульовими.

6. На відміну від звичайного підходу до керування обчислюваним моментом, тут ми додали інтегральний член, щоб мати кращу продуктивність відстеження і компенсувати помилку відстеження траєкторії, яка зазвичай виникає через недосконалість динамічного моделювання, оцінки параметрів, а також через зовнішні збурення.

## ВИСНОВКИ

1. Комерційно доступний робот-маніпулятор із чотирма ступенями свободи Dobot Magician використовується, щоб продемонструвати перспективу розміщення більшої функціональності в одному роботі.

2. Додавання додаткових функцій у Dobot отримано з використанням модифікованих параметрів Denavit-Hartenberg. Було досліджено пропорційно-інтегрально-похідне (PID) керування, обчислене керування крутним моментом (СТС) і декартове відстеження траєкторії з керуванням на основі суглобів.

3. Багатофункціональний програмний інтерфейс для Dobot розроблено з використанням мови програмування C#. Для розробки програмного забезпечення використовуються Dobot API, Kinect V2 API і Speech SDK.

4. З безперервним розвитком глибокого навчання з підкріпленням і технології віртуальної реальності розробка контенту, що поєднує стратегії глибокого навчання з підкріпленням і технологію віртуальної реальності, стає все більш простою. Інструмент агента глибокого підкріплення навчання Unity3D може допомогти розробникам досліджень у складних сценаріях навчання. Ефективні навчальні агенти значно підвищили ефективність розробки. Порівняно з існуючими рішеннями система віртуальної лабораторії вдосконалена з точки зору зручності та інтерактивності та може бути застосована для викладання курсів механічної спеціальності.

5. Роботизоване є таким же природним і масштабованим, як навчання, що виконується людьми та тваринами, де будь-яка поведінка може бути набута, за умови, що завдання є спеціальним визначений точно, фізично можливий і не створює необґрунтованих проблем для дослідження.

## ПЕРЕЛІК ДЖЕРЕЛІ ПОСИЛАНЬ

1. A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157-173, 2008.
2. A. Ramisa, G. Alenya, F. Moreno-Noguer, and C. Torras, "Using depth and appearance features for informed robot grasping of highly wrinkled clothes," in *2012 IEEE International Conference on Robotics and Automation*, 2012: IEEE, pp. 1703-1708.
3. D. Seita et al., "Robot bed-making: Deep transfer learning using depth sensing of deformable fabric," *arXiv preprint arXiv:1809.09810*, 2018.
4. M. Beetz et al., "Robotic roommates making pancakes," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*, 2011: IEEE, pp. 529-536.
5. Moley Robotics, "The Moley Robotic Kitchen - Mission & Goals," ed, 2015.
6. C. Militaru, A. D. Mezei, and L. Tamas, "Object handling in cluttered indoor environment with a mobile manipulator," in *2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, 19-21 May 2016 2016, pp. 1-6, doi: 10.1109/AQTR.2016.7501382.
7. K. Wu, R. Ranasinghe, and G. Dissanayake, "Active recognition and pose estimation of household objects in clutter," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015: IEEE, pp. 4230-4237.
8. W. Miyazaki and J. Miura, "Object placement estimation with occlusions and planning of robotic handling strategies," in *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 3-7 July 2017 2017, pp. 602-607, doi: 10.1109/AIM.2017.8014083.
9. S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," *arXiv preprint arXiv:1501.05611*, 2015.
10. M. Gualtieri, A. t. Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *2016 IEEE/RSJ International Conference on Intelligent*

Robots and Systems (IROS), 9-14 Oct. 2016 2016, pp. 598-605, doi: 10.1109/IROS.2016.7759114.

11. M. Gualtieri, A. t. Pas, and R. Platt, "Pick and Place Without Geometric Object Models," in 2018 IEEE International Conference on Robotics and Automation (ICRA), 21-25 May 2018 2018, pp. 7433-7440, doi: 10.1109/ICRA.2018.8460553.

12. L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kPAM: KeyPoint Affordances for Category-Level Robotic Manipulation," arXiv preprint arXiv:1903.06684, 2019.

13. I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705-724, 2015.

14. J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," arXiv preprint arXiv:1809.10790, 2018.

15. Amazon Robotics LLC. "2017 Amazon Robotics Challenge Official Rules." <https://www.amazonrobotics.com/site/binaries/content/assets/amazonrobotics/arc/2017-amazon-robotics-challenge-rules-v3.pdf> (accessed 24 June, 2019).

16. Höfer S. On Decomposability in Robot Reinforcement Learning. 2017.

17. Belousov B., Neumann G., Rothkopf CA., Peters JR. Catching heuristics are optimal control policies. *Advances in Neural Information Processing Systems* 29. 2016.

18. Corke PI. High-Performance Visual Closed-Loop Robot Control By. 1994.

19. Buttazzo GC., Allotta B., Fanizza FP. Mousebuster: a robot for real-time catching. *IEEE Control Systems*. February 1994; 14(1): 49–56.

20. Fernandes DG., Lima PU. A testbed for robotic visual servoing and catching of moving objects. 1998 IEEE International Conference on Electronics, Circuits and Systems. Surfing the Waves of Science and Technology (Cat. No.98EX196). IEEE; pp. 475–478.

21. Hong W., Slotine J-JE. Experiments in hand-eye coordination using active vision. *Experimental Robotics IV*. 2005; 130–139.

22. Mohamed Elgendy. Deep Learning for Vision Systems, Volume: 1, 2020, p. 480.
23. Joseph Howse. Learning OpenCV 4 Computer Vision with Python 3, Volume: 3, 2020, p. 372.
24. Leonard Eddison. Coding: Raspberry Pi & Python, Volume: 1, 2018, p. 184.
25. Ashwin Pajankar. Raspberry Pi Computer Vision Programming. Volume: 2, 2020, p. 306.
26. Ngurah Desnanjaya I. G. M., Arsana I. N. A. Home security monitoring system with IoT-based Raspberry Pi. Indonesian Journal of Electrical Engineering and Computer Science. 2021. Вип. 22, № 3.
27. Bowman R. W., Vodenicharski B., Collins J. T., Stirling J. Flat-Field and Colour Correction for the Raspberry Pi Camera Module. Journal of Open Hardware. 2020. Вип. 4, № 1.
28. Computer Vision / Y. Wang et al. KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event CA USA. New York, NY, USA, 2020.
29. Haidi Jaber, Dr. Luc Jaulin, "Robot Path Planning using Interval Analysis", ENSTA Bretagne/OSM, 2012.
30. B Abhishek; S Gautham; D Varun Rufus Raj Samuel; K Keshav; U.P. Vignesh; Shyam R Nair, - ROS based Stereo vision system for autonomous vehicle, Robotics and Automation (ICRA), 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPSI).
31. Vittoria, S.; Lahlou, G.; Torres, R.; Daoudi, H.; Mosnier, I.; Mazalaigue, S.; Sterkers, O. Robot-based assistance in middle ear surgery and cochlear implantation: First clinical report. Eur. Arch. Otorhinolaryngol. 2021, 278, 77–85.
32. Baskaran, S.; Niaki, F.A.; Tomaszewski, M.; Gill, J.S.; Chen, Y.; Jia, Y.; Mears, L.; Krovi, V. Digital Human and Robot Simulation in Automotive Assembly using Siemens Process Simulate: A Feasibility Study. Procedia Manuf. 2019, 34, 986–994.

33. Jahanshahi, H.; Jafarzadeh, M.; Sari, N.N.; Pham, V.-T.; Huynh, V.V.; Nguyen, X.Q. Robot Motion Planning in an Unknown Environment with Danger Space. *Electronics* 2019, 8, 201.
34. Bandala, M.; West, C.; Monk, S.; Montazeri, A.; Taylor, C.J. Vision-Based Assisted Tele-Operation of a Dual-Arm Hydraulically Actuated Robot for Pipe Cutting and Grasping in Nuclear Environments. *Robotics* 2019, 8, 42.
35. Ichter, B.; Pavone, M. Robot Motion Planning in Learned Latent Spaces. *IEEE Robot. Autom. Lett.* 2019 4, 2407–2414.
36. Arm, P.; Zenkl, R.; Barton, P.; Beglinger, L.; Dietsche, A.; Ferrazzini, L.; Hutter, M. SpaceBok: A Dynamic Legged Robot for Space Exploration. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, 20–24 May 2019; pp. 6288–6294.
37. Grigore L.S, Oncioiu I., Priescu I., Joit D. Development and Evaluation of the Traction Characteristics of a Crawler EOD Robot. *Appl. Sci.* 2021, 11, 3757.
38. Paxton, C.; Ratliff, N.; Eppner, C.; Fox, D. Representing Robot Task Plans as Robust Logical-Dynamical Systems. In *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, 4–8 November 2019; pp. 5588-5595.
39. Rafael Verano, M.; Jose Caceres, S.; Abel Arenas, H.; Andres Montoya, A.; Joseph Guevara, M.; Jarelh Galdos, B.; Jesus Talavera, S. Development of a Low-Cost Teleoperated Explorer Robot (TXRob). *Int. J. Adv. Comput. Sci. Appl. (IJACSA)* 2022, 13.
40. Vilcapaza Goyzueta, D.; Guevara Mamani, J.; Sulla Espinoza, E.; Supo Colquehuanca, E.; Silva Vidal, Y.; Pinto, P.P. Evaluation of a NUI Interface for an Explosives Deactivator Robotic Arm to Improve the User Experience. In *HCI International 2021- Late Breaking Posters. HCII 2021*; Stephanidis, C., Antona, M., Ntoa, S., Eds.; *Communications in Computer and Information Science*; Springer: Cham, Switzerland, 2021; Volume 1498.

## Додаток А

Стаття у фаховому журналі (подана в 1 номер ВОТТП, 2024р.)

МАРТИНЮК Валерій  
Хмельницький національний університет  
ORCID ID: 0000-0001-5758-4244  
e-mail: [martynyuk.valeriy@gmail.com](mailto:martynyuk.valeriy@gmail.com)  
КОЛОС Андрій  
Хмельницький національний університет  
e-mail: [andrewkolos520@gmail.com](mailto:andrewkolos520@gmail.com)

**УДОСКОНАЛЕНИЙ МЕТОД МАШИННОГО НАВЧАННЯ ДЛЯ КЕРУВАННЯ РОБОТОМ**

*У статті розглянуто комерційно доступний робот-маніпулятор DOBOT Magician із чотирма ступенями свободи використовується, щоб продемонструвати перспективу розміщення більшої функціональності в одному роботі.*

*Система машинного зору для робота-маніпулятора DOBOT Magician є дорогою, тому запропоновано використовувати одноплатний мікрокомп'ютер RaspberryPI Model B+ та відеокамеру Canyon CNR-WCAM820 разом із бібліотекою для комп'ютерного зору OpenCV для удосконаленого методу машинного навчання для керування роботом із використанням машинного зору.*

*Розроблено алгоритм та програмне забезпечення для удосконаленого методу машинного навчання для керування роботом-маніпулятором DOBOT Magician, який базується на розпізнаванні деталей та формуванню команд роботу-маніпулятору DOBOT Magician.*

**Ключові слова:** *робот-маніпулятор DOBOT Magician, одноплатний мікрокомп'ютер RaspberryPI Model B+, бібліотека для комп'ютерного зору OpenCV.*

MARTYNYUK Valeriy, KOLOS Andriy  
Khmelnitsky national university, Ukraine

**ADVANCED MACHINE LEARNING METHOD FOR ROBOT CONTROL**

*In the paper, a commercially available robot manipulator DOBOT Magician with four degrees of freedom is used to demonstrate the prospect of accommodating more functionality in a single robot.*

*The machine vision system for the DOBOT Magician manipulator robot is expensive, so it is proposed to use a RaspberryPI Model B+ single-board microcomputer and a Canyon CNR-WCAM820 video camera together with the OpenCV computer vision library for an advanced machine learning method to control the robot using machine vision.*

*Developed an algorithm and software for an advanced machine learning method to control the DOBOT Magician robot manipulator based on detail recognition and command generation of the DOBOT Magician robot manipulator.*

**Keywords:** *DOBOT Magician manipulator robot, RaspberryPI Model B+ single-board microcomputer, OpenCV computer vision library.*

**Постановка проблеми**

На кафедрі автоматизації, комп'ютерно-інтегрованих технологій та робототехніки придбано робот-маніпулятор DOBOT Magician, який зображено на рис. 1. DOBOT Magician - це багатофункціональна настільна роботизована рука для практичного навчання. Робот додатково оснащується різними інструментами та може реалізовувати цікаві функції, такі як тривимірний друк, лазерне гравіювання, написання та малювання.

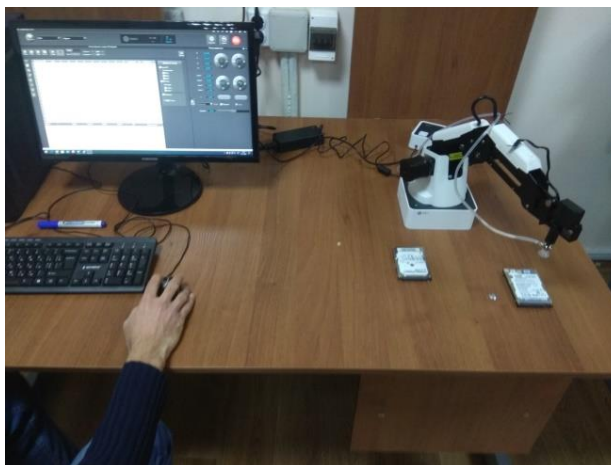


Рис. 1. Робот-маніпулятор DOBOT Magician

DOBOT Magician гнучкий в управлінні - програмувати роботом-маніпулятором і керувати ним в режимі реального часу можна за допомогою комп'ютера, смартфона або планшета. Як канал зв'язку використовується як Wi-Fi або Bluetooth, так і проводове USB-з'єднання. У придбаній версії робота-маніпулятора відсутня система машинного зору, тому підвищення ефективності, взаємодія з маніпулятором робота, вирішення прямих та зворотних кінематичних задач з використанням удосконаленого методу машинного навчання для керування роботом, є дійсно актуальними задачами.

#### Аналіз останніх джерел

Робот-маніпулятор DOBOT Magician програмується за допомогою наступних мов програмування: Google Blockly, Python, Java (Android), C/C++/C#, LabView та Objective-C (iOS, MacOS) [1]. Користувачі робота-маніпулятора DOBOT Magician використовують програмне забезпечення Dobot Studio для програмування DOBOT Magician.

Виробник робота-маніпулятора DOBOT Magician надає інтерфейс прикладного програмування (англійською мовою - Application Programming Interface (API)) та набір для розробки програмного забезпечення (англійською мовою - Software Development Kit (SDK)) для різних платформ [2].

Робот-маніпулятор DOBOT Magician може використовувати автоматизовані методи переміщення предметів з одного місця в інше [3]. Додатково до робота-маніпулятора DOBOT Magician можна придбати систему машинного зору, але така система є дуже дорогою, тому розробка удосконаленого методу машинного навчання для керування роботом із використанням машинного зору є актуальною задачею.

#### Виклад основного матеріалу

Система автоматичного керування роботом-маніпулятором DOBOT Magician побудована на основі мікрокомп'ютера RaspberryPI Model B+, який передає сигнали керування електродвигунами для обертання робота. Сигнали керування формуються мікрокомп'ютером на основі зображення, отриманого з відеокамери.

Значення кутів повороту передаються до мікрокомп'ютера RaspberryPI Model B+ двома енкодерами, що закріплені, відповідно, на осях обертання. Мікрокомп'ютер може здійснювати обмін даними з ПК та іншими пристроями через провідну мережу (Ethernet) та безпроводну мережу (WiFi), що необхідно для дистанційного керування через веб-інтерфейс.

Мікрокомп'ютер RaspberryPI Model B+, що використовується для автоматизованого керування роботом, зображено на рис. 2 [4].

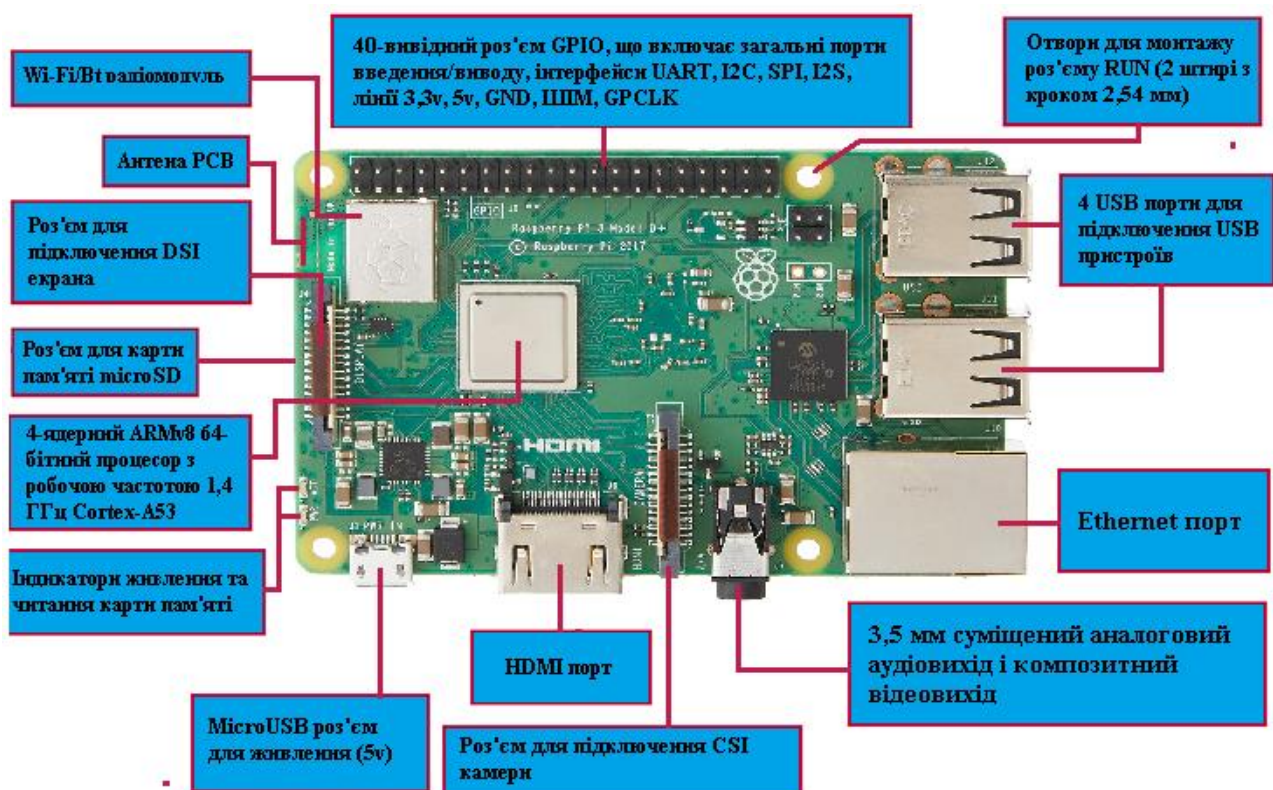


Рис. 2. Одноплатний мікрокомп'ютер RaspberryPI Model B+ [4]

Для визначення положення робота використовується відеокамера Canyon CNR-WCAM820, яка зображена на рис. 3.

Відеокамера фіксує зображення та передає відеосигнал до мікрокомп'ютера RaspberryPI Model B+ з метою аналізу та формування команд повороту робота.

Технічні характеристики відеокамери:

- матриця: 2 млн пікселів., CMOS;
- роздільна здатність (відео): 1600x1200;
- максимальна частота кадрів: 30 Гц;
- формат зображення: 640x480 при 30 Гц 1600x1200 при 15 Гц;
- глибина кольору: 24-розрядний;
- підключення: USB 2.0;
- мікрофон: вбудований;
- кут огляду об'єктива :70 град;
- розміри (ШxВxГ): 72x42x65 мм;
- вага: 96 гр.



**Рис. 3. Відеокамера Canyon CNR-WCAM820**

Спочатку виконується завантаження бібліотек та ініціалізація відеокамери. Встановлюються параметри відеозображення та початкові умови для процесу автоматизованого керування роботом. Виконується ініціалізація бази даних Redis.

Згідно з даними відеокамери далі виконується захоплення зображення з камери. Здійснюється фільтрація зображення для знаходження області деталі, яку повинен взяти робот. Виконується аналіз розмірів та округлості контуру знайденої області.

Якщо довжина та округлість контуру відповідають відображенню деталі, яку повинен взяти робот, то відбувається визначення центру контуру області. Розраховується зміщення центру контуру відносно центру зображення. Величина зміщення центру зберігається у базі даних Redis (TTL 5с). Отримані дані передаються для формування команді керування електродвигунами з метою мінімізації зміщення центру.

Положення деталі, яку повинен взяти робот, може візуалізуватися у веб-інтерфейсі програмного забезпечення робота. Якщо довжина та округлість отриманого контуру деталі, яку повинен взяти робот мають складну форму та/або велику площу, то виконується розширений аналіз з використанням додаткових даних. На основі результатів аналізу передаються дані для формування команд керування електродвигунами робота.

На початковому етапі виконується завантаження часу, бібліотек та встановлюється зв'язок з базою даних Redis. Формується список приєднаних клієнтів. При приєднанні нового клієнта викликається функція `async def on_connect`, яка керує очікуванням та обробкою команд від клієнта за допомогою функції `async def process_command`. Далі виконуються дії відповідно до отриманих команд та подаються керуючі сигнали на виводи GPIO мікрокомп'ютера Raspberry PI Model B+.

Далі перевіряється існування та виконується перетворення даних у базі Redis. Відповідь формується у форматі JSON, і відправляється клієнту. Підпрограма виконує оброблення даних та команд, що надходять із різних пристроїв та дозволяє керувати параметрами робота.

Для початку на плату Raspberry Pi подається живлення, після чого можна запустити написаний код. Також важливо щоб до плати була під'єднана камера, з якої саме буде йти відео-потік і йде перевірка на працездатність камери. Після обробки, кадр переходить до аналізу за допомогою бібліотеки для комп'ютерного зору OpenCV. Алгоритм програмного забезпечення зображено на рис. 4.

Картинка конвертується у градації сірого кольору і починається визначення об'єктів з потокового відео. Йде кожна перевірка на відповідний об'єкт, де до кожного встановлений свій, відповідний класифікатор.

Якщо об'єкт не знайдений, процес повторюється. Ця операція проводиться для кожного встановленого класифікатора. Після обробки кадру, картинка нормалізується і на вихід вже подається відео відповідного формату, з обробкою об'єктів і їх визначенням.



Рис. 4. Алгоритм програмного забезпечення

#### Висновки

Комерційно доступний робот-маніпулятор DOBOT Magician із чотирма ступенями свободи використовується, щоб продемонструвати перспективу розміщення більшої функціональності в одному роботі.

Система машинного зору для робота-маніпулятора DOBOT Magician є дорогою, тому запропоновано використовувати одноплатний мікрокомп'ютер RaspberryPI Model B+ та відеокамеру Canyon CNR-WCAM820 разом із бібліотекою для комп'ютерного зору OpenCV для удосконаленого методу машинного навчання для керування роботом із використанням машинного зору.

Розроблено алгоритм та програмне забезпечення для удосконаленого методу машинного навчання для керування роботом-маніпулятором DOBOT Magician, який базується на розпізнаванні деталей та формуванню команд роботу-маніпулятору DOBOT Magician.

#### Література:

1. Dobot Magician. User Manual. - Shenzhen Yuejiang Technology Co.,Ltd., 2018/2/9, v.1.2.5, p.138.
2. Chakraborty, Sudip, & Aithal, P. S., (2022). Open Loop Automated Baby Cradle Using Dobot Magician and C#. International Journal of Applied Engineering and Management Letters (IJAEML),6 (1), 344-349.
3. Pol R., Dhhoka S. Applications of a Robotic Arm (A Dobot). nternational Research Journal of Engineering and Technology (IRJET) Vol. 10, 2023.
4. Raspberry Pi - перший запуск: [Електронний ресурс] – Режим доступу: <https://geekelectronics.org/raspberry-pi/raspberry-pi-pervyj-zapusk.html>

#### References

1. Dobot Magician. User Manual. - Shenzhen Yuejiang Technology Co.,Ltd., 2018/2/9, v.1.2.5, p.138.
2. Chakraborty, Sudip, & Aithal, P. S., (2022). Open Loop Automated Baby Cradle Using Dobot Magician and C#. International Journal of Applied Engineering and Management Letters (IJAEML),6 (1), 344-349.
3. Pol R., Dhhoka S. Applications of a Robotic Arm (A Dobot). nternational Research Journal of Engineering and Technology (IRJET) Vol. 10, 2023.
4. Raspberry Pi - first launch: [Electronic resource] – Access mode: <https://geekelectronics.org/raspberry-pi/raspberry-pi-pervyj-zapusk.html>.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Колос Андрій Богданович

Тема: Удосконалений метод машинного навчання для керування роботом

Спеціальність: 151 «Автоматизація та комп'ютерно-інтегровані технології»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 16 Кількість сторінок записки 75

1. Короткий зміст роботи та прийнятих рішень: Вступ. Огляд літературних джерел та патентних даних. Математична модель процесу машинного навчання для керування роботом. Імітаційна модель машинного навчання для керування роботом. Експериментальні дослідження удосконаленого методу машинного навчання для керування роботом. Висновки.

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

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі були розглянуті наступні питання: будова і принцип роботи робота-маніпулятора із чотирма ступенями свободи Dobot Magician. У другому розділі розроблено математичну модель з використанням модифікованих параметрів Denavit-Hartenberg. У третьому розділі досліджено пропорційно-інтегрально-похідне (PID) керування, обчислене керування крутним моментом (СТС) і декартове відстеження траєкторії з керуванням на основі суглобів. У четвертому розділі проведені експериментальні дослідження робота-маніпулятора із чотирма ступенями свободи Dobot Magician.

4. Позитивні сторони роботи: використання інструменту агента глибокого підкріплення навчання Unity3D може допомогти розробникам досліджень у складних сценаріях навчання.

5. Негативні сторони роботи: у роботі недостатньо уваги приділяється розробці програмного забезпечення Dobot Magician

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

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

8. Інші зауваження: відсутні

9. Оцінка дипломної роботи: добре (4,00/С)

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

Клюва Юрій Павлович, к.т.н., доцент,  
завідувач кафедри кібербезпеки

"19" грудня 2023 р.

 (підпис)

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ  
КАФЕДРИ АВТОМАТИЗАЦІЇ, КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ ТА  
РОБОТОТЕХНІКИ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

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

Назва: Удосконалений метод машинного навчання для керування роботом

Автор: Колос Андрій Богданович

Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології

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

Науковий керівник: Мартинюк Валерій Володимирович, доктор технічних наук, професор

Після аналізу звіту подібності зроблено такий висновок:

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

Підтвердження:

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

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

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

3) виявлені модифікації тексту не впливають на відсоток схожості.

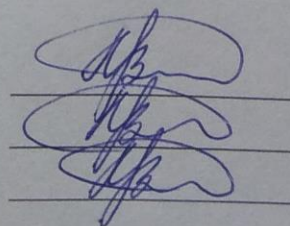
Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає 2,27% і адресується до 124 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата 29.12.2023р.

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи



Валерій МАРТИНЮК

Валерій МАРТИНЮК

Валерій МАРТИНЮК

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

Колос А.Б.

---

ПІБ здобувача вищої освіти

ФІТ, 2 курс, групи АКІТм-22-1

### ЗАЯВА

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

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

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

10.12.2023р.

дата

  
\_\_\_\_\_

підпис

## Anti-Plagiarism v-15.257

**Максимальне співпадіння з одним документом 2.0%**

**Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 9%**

ID: 125138 Назва: МКР Удосконалений метод машинного навчання для керування роботом Додано в БД: 2023-12-29 Автора: Андрій КОЛОС Керівники: Валерій МАРТИНЮК Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	66887	534	1715 (3%)	22 (4%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:  
Кафедра АКІПТК

ID перевірки:  
1016041370

Дата перевірки:  
29.12.2023 15:22:46 EET

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
29.12.2023 15:26:52 EET

ID користувача:  
100005862

Назва документа: Колос\_кваліфікаційна\_робота

Кількість сторінок: 69 Кількість слів: 10570 Кількість символів: 79259 Розмір файлу: 2.50 MB ID файлу: 1015735968

373 слова позначені як "вилучені" та не враховуються у підрахунку слів

## 2.27% Схожість

Найбільша схожість: 1.04% з джерелом з Бібліотеки (ID файлу: 1015735707)

1.33% Джерела з Інтернету

30

Сторінка 71

1.53% Джерела з Бібліотеки

34

Сторінка 71

## 0% Цитат

Вилучення цитат вимкнено

Посилання

1

Сторінка 71

## 0.03% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 0%)

0.03% Вилучення з Інтернету

52

Сторінка 72

Немає вилучених бібліотечних джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

45