

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

КВАЛІФІКАЦІЙНА РОБОТА

магістр

Освітній рівень

Удосконалений метод керування роботом маніпулятором

Назва теми

КВРАКІТ. 2022172.01.02.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_м-22-1



Підпис

Ігор РИЖКОВ

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

АКІТ та Р
В. Мартинюк

«01» 09 2023р.

ЗАВДАННЯ НА ДИПЛОМНУ РОБОТУ

Рижков Ігор Сергійович

1 Тема роботи: Удосконалений метод керування роботом маніпулятором
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 керування роботом
маніпулятором.

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

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

Завдання отримав

Ігор Рижков

Науковий керівник

В. Мартинюк

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Удосконалений метод керування роботом маніпулятором».

Автор роботи: Рижков Ігор Сергійович.

Керівник роботи: Мартинюк Валерій Володимирович.

Пояснювальна записка: 74 с., 56 рис., 0 табл., 1 дод., 40 джерел.

Графічна частина: 16 презентаційних слайдів.

РОБОТ-МАНІПУЛЯТОР, ІМІТАЦІЙНА МОДЕЛЬ, 3-D ВІЗУАЛІЗАЦІЮ РУХУ РОБОТА МАНІПУЛЯТОРА З ТРЬОМА СТУПЕНЯМИ СВОБОДИ.

Мета роботи: розробка удосконаленого методу керування роботом маніпулятором за допомогою ПД контролера та підвищення його продуктивності роботи.


Підпис студента

29.12.2023р.
Дата

ЗМІСТ

ВСТУП.....	3
1 ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ ТА ПАТЕНТНИХ ДАНИХ.....	5
1.1 Особливості роботів маніпуляторів	5
1.2 Особливості математичних моделей роботів маніпуляторів	11
1.3 Висновки до першого розділу.....	16
2 МАТЕМАТИЧНА МОДЕЛЬ ПРОЦЕСУ КЕРУВАННЯ РОБОТОМ МАНІПУЛЯТОРОМ	17
2.1 Математичні моделі складових компонентів робота маніпулятора	17
2.2 Удосконалений метод керування роботом маніпулятором	22
2.3 Висновки до другого розділу.....	32
3 ІМІТАЦІЙНА МОДЕЛЬ ПРОЦЕСУ КЕРУВАННЯ РОБОТОМ МАНІПУЛЯТОРОМ	34
3.1 Імітаційна модель робота маніпулятора в Matlab/Simulink.....	34
3.2 Імітаційна модель процесу керування роботом маніпулятором за допомогою ПІД-регулятора.....	47
3.3 Висновки до третього розділу.....	53
4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ УДОСКОНАЛЕНОГО МЕТОДУ КЕРУВАННЯ РОБОТОМ МАНІПУЛЯТОРОМ.....	54
4.1 Особливості моделювання робота маніпулятора із двома ступенями свободи та прямою кінематикою.....	54
4.2 Імітаційна модель робота маніпулятора з трьома ступенями свободи	64
4.3 Висновки до четвертого розділу.....	69
ВИСНОВКИ.....	70
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	71
ДОДАТОК А Стаття у фаховому журналі	75

ВСТУП

Актуальність теми. Для академічних або промислових роботів-маніпуляторів ключовим завданням є досягнення бажаних положень та орієнтації їх кінцевих механізмів або інструментів для виконання заданого завдання. Для досягнення цієї мети необхідно володіти ґрунтовними знаннями обернених кінематичних задач. Роботи-маніпулятори використовуються в багатьох галузях для виконання різних завдань, таких як переміщення матеріалів, підбір і розміщення, інтерактивна робота, спільна робота, небезпечні польові роботи тощо.

Тому, підвищення ефективності, взаємодія з маніпулятором робота, вирішення прямих та зворотних кінематичних задач з використанням механізму представлення та оптимізації, операції вибору та розміщення за допомогою механізму захоплення, інтерактивної роботи декількох роботів в MATLAB, є дійсно актуальними задачами.

Мета роботи – розробка удосконаленого методу керування роботом маніпулятором за допомогою ПІД контролера 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 керування роботом маніпулятором;
- розробити алгоритм та програмну реалізацію удосконаленого методу керування роботом маніпулятором;
- розробити імітаційну модель удосконаленого методу керування роботом маніпулятором.

Об’єкт дослідження – процес керування роботом маніпулятором та підвищення продуктивності його роботи.

Предмет дослідження – удосконалений метод керування роботом

маніпулятором за допомогою ПДД контролера та підвищення продуктивність його роботи.

Наукова новизна одержаних результатів. У результаті проведеного дослідження:

Розроблено удосконалений метод керування роботом маніпулятором за допомогою ПДД контролера та підвищення продуктивність його роботи.

1 ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ ТА ПАТЕНТНИХ ДАНИХ

1.1 Особливості роботів маніпуляторів

У сучасному світі спостерігається швидка революція у сфері автоматизації. Ми живемо в епоху, коли автоматизація охоплює світ. Автоматизація захоплює світ в наші дні. Прикладом такого розвитку є зростаюче прагнення до впровадження роботів, або роботів-маніпуляторів, попит на яких постійно зростає.

Наприклад, однією з найпомітніших тенденцій останнього часу є зростаюче бажання створювати роботів або роботизовані маніпулятори. Це особливо цікаво, коли ми розглядаємо зростаюче бажання впровадити робототехніку в наше повсякденне життя.

Були повідомлення про те, що, хоча роботи в промисловості, як правило, створюються і використовуються для дуже специфічних цілей, це призвело до відсутності різноманітності, коли мова йде про програмне забезпечення та утиліти, що також може призвести до відсутності різноманітності.

Цей тип застосувань має спільну характеристику: вони вимагають від робота роботи в неструктурованому середовищі, на відміну від структурованої промислової робочої клітини. Через невизначеності в моделюванні системи, якості датчиків і спрацьовування робота, управління рухами робота і планування траєкторії в неструктурованому середовищі представляє значні труднощі.

Більшість застосувань роботів в даний час можна розділити на дві категорії: ті, що мають справу як зі структурованими, так і з неструктурованими середовищами, в той час як решту можна віднести до більш широкої області застосування роботів [10]. Суворе трактування тем, наведених у цьому тексті, які мали слугувати першим вступом до робототехніки, повинно бути включено до будь-якого першого вступу до робототехніки.

Робот допомагає людині, виконуючи додаткові зручні завдання, допомагаючи їй на робочому місці, а також забезпечуючи ізольоване середовище,

яке запобігає будь-якій шкоді, дискомфорту, повторенням і т.д., які зазвичай присутні в такій ситуації. Майже всі завдання, які виконують машини, стають складнішими з розвитком технологій.

Машини стають більш здатними виконувати завдання, які раніше виконували люди, а тепер виконують роботи, незважаючи на небезпеку, яку вони становлять. Роботи майбутнього повинні перевершувати людину за своєю структурою, інтелектом, витонченістю, а також здатністю своєчасно реагувати, виконувати людські завдання людськими способами і мати ефективну і безпечну співпрацю між людьми і роботами.

Вищезазначені причини роблять роботів, що складаються з електромеханічних систем, здатних до високого рівня автономності, надзвичайно складними електромеханічними системами, аналітичний опис яких вимагає розвинених математичних методів.

Розробка таких пристроїв є однією з найцікавіших і найскладніших проблем і завдань в галузі робототехніки. Здатність до перепрограмування є більш важливою, ніж будь-яка інша особливість роботів. Комп'ютерне управління - це те, що дає роботу його корисність і адаптивність, таким чином надаючи роботу його корисність і адаптивність.

Роботи-маніпулятори здатні виконувати найрізноманітніші завдання в найрізноманітніших сферах. Існує низка галузей, де ця технологія використовується, зокрема, в автомобілях, побутових товарах, під час збирання та розбирання, знешкодження радіоактивних полів і вибухових речовин. Вона знайшла застосування в таких галузях, як реконструкція, хірургія та реабілітація, які включають такі види діяльності, як кінематичні, динамічні та контрольні операції.

З єдиною метою створення зв'язку між маніпулятором робота та MATLAB, прямим та зворотнім кінематичним розрахунком, операцією Pick and Place та технікою оптимізації для побудови власного зворотнього кінематичного двигуна.

Було обрано MATLAB, оскільки це програма, з якою багато людей, наприклад, студенти, вже знайомі, і вона має підтримку Microsoft, яка є найпопулярнішою операційною системою, коли мова йде про персональні комп'ютери.

Рекомендується, щоб комунікація між роботами була побудована навколо декількох сервоприводів, щоб рухи робота можна було контролювати на розсуд користувача, в межах безпеки та надійності.

Під час роботи робота маніпулятори рухаються по заданих траєкторіях, які являють собою послідовності точок, що відображають положення кінцевих ефекторів, а також їхню орієнтацію. Залежно від кінематики, траєкторія може бути як у спільному просторі, так і в декартовому просторі, що базується на часі.

Робот, призначений для промислового застосування, може бути явно зрозумілий як механізм з відкритим ланцюгом, що складається з жорсткого тіла і ряду шарнірів, які з'єднують жорсткі тіла разом [11].

Шарніри дозволяють з'єднаним тілам здійснювати певні рухи відносно один одного. Шарнірне з'єднання діє як шарнір між з'єднаними тілами, і обертальний рух між ними обмежується невеликим відносним поворотом навколо осі з'єднання.

Відомо, що кінематичний ланцюг складається з групи жорстких тіл, з'єднаних між собою шарнірами [10]. Він складається з ланок, які є окремими твердими тілами, що утворюють ланцюг. Відкритий, закритий або розгалужений кінематичний ланцюг може бути послідовним, паралельним або їх комбінацією.

Це означає, що кінематичні ланцюги можуть бути випадковими, відкритими або замкнутими. Для успішної роботи необхідно обчислити всі точки в декартових координатах. Обернена кінематика відноситься до процесу перетворення положень траєкторії з декартових координат у спільні координати шляхом перетворення траєкторії з декартових у спільні координати [10].

Слово "робот" походить від "robota", що в рабовласницьких мовах означає підпорядковану працю [11]. Письменник-фантаст Азімов у своєму романі

"Перезавантаження" писав про три фундаментальні закони, які регулюють взаємодію між роботами та людьми.

Ці правила були сформульовані в 1940 році, невдовзі після публікації роману [10]. У середині ХХ століття розпочалася нова ера досліджень штучного інтелекту (ШІ), коли були виявлені перші зв'язки між людським інтелектом і машинами, що спричинило тривалий період плідних досліджень. У цей час були розроблені перші роботи [10].

У міру того, як розвивалися механіка, управління, комп'ютери та електроніка, досягнення в цих галузях ставали все більш і більш важливими. З часом це замкнене коло почало продукувати знання і розуміння, необхідні для виникнення так званої робототехніки - науки і технології роботів.

Поєднання двох технологій призвело до створення перших роботів у 1960-х роках: верстатів з числовим програмним управлінням для виготовлення прецизійних виробів і телеуправління для дистанційного поводження з радіоактивними матеріалами.

У порівнянні з людською рукою, ці руки-ведучі один в один відтворювали механіку людської руки, мали мінімальний контроль і мало усвідомлювали навколишню обстановку [11].

У середині та наприкінці ХХ століття інтегральні схеми, цифрові комп'ютери та мініатюрні деталі призвели до того, що з'явилася можливість проектування та програмування роботів, керованих комп'ютером.

У 1970-х роках промислові роботи - також відомі як автоматизовані промислові роботи - відіграли важливу роль в автоматизації гнучких виробничих систем [10].

Промислові роботи також успішно використовувалися в загальній промисловості на додаток до їх широкого застосування в автомобільній промисловості.

Залежно від кількості ступенів свободи (КСС) або осей, які має робот, та їхніх кінематичних характеристик, роботів можна класифікувати за наступними параметрами.

За ступенем свободи, якою володіє робот-маніпулятор, можна судити про його робочу досконалість. Типові 6-координатні роботи-маніпулятори, як правило, можуть виконувати загальне завдання в 3-вимірному просторі, де об'єкт може перебувати в будь-якій точці простору в будь-який час, незалежно від його положення або орієнтації.

Однак може виникнути потреба у розробці роботів-маніпуляторів або інших маніпуляторів для конкретних застосувань, залежно від таких параметрів, як ступінь свободи (КСС) або кінематичні характеристики. Це правда, що існує безліч критеріїв для класифікації роботів-маніпуляторів, але зазвичай можна вибирати на основі ступеня свободи або кількості осей [10].

На противагу цьому, такі організації, як Американський інститут робототехніки (RIA), Французька асоціація робототехніки (AFR) та Японська асоціація промислових роботів згрупували робототехніку в шість окремих категорій наступним чином.

1. Пристрої для ручного переміщення.
2. Робот з фіксованою послідовністю.
3. Робот зі змінною послідовністю.
4. Робот для відтворення.
5. Робот з числовим програмним управлінням.
6. Розумний робот.

Варто зазначити, що окрім вищезгаданих модулів, промислові роботи-маніпулятори також можна розділити на категорії на основі їх механізмів, ступенів свободи, приведення в дію, робочого простору, управління, руху та застосування [10].

Роботи-маніпулятори зазвичай бувають послідовними або паралельними, залежно від того, чи містять вони відкритий або замкнутий контур. Існує два типи роботів-маніпуляторів, які використовуються в реальних умовах: призматичні (П) і револьверні (Р) з'єднання, причому один з можливих типів з'єднань може бути жорстким і гнучким.

Все просто: осі двох сусідніх шарнірів можуть бути паралельними або ортогональними, залежно від того, як ви вирішите комбінувати ці шарніри та ланки. Завдяки з'єднанню Р і П осі двох сусідніх осей можуть бути паралельними або ортогональними. В ортогональних з'єднаннях одна вісь повертається на 90° відносно іншої, а дві осі перетинаються під кутом 90° відносно їхньої спільної нормалі [10], як зображено на рисунку 1.1.

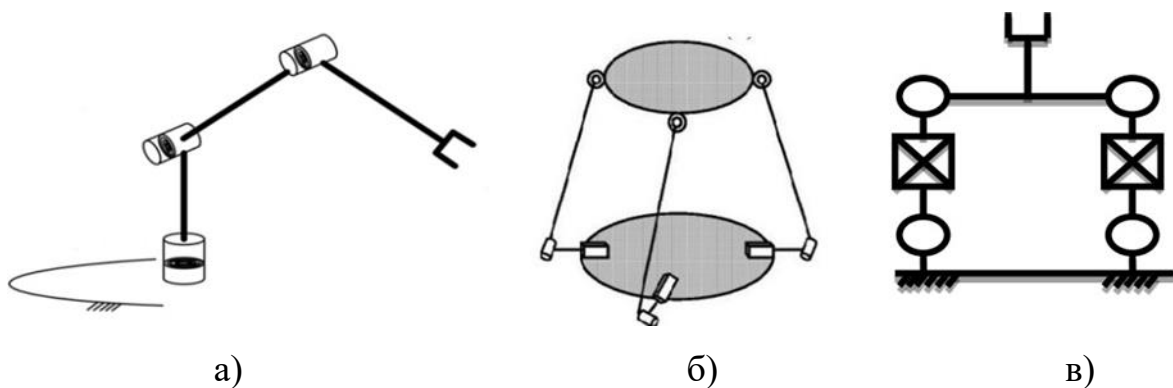


Рисунок 1.1 - (а) Послідовний [11], (б) паралельний [11] та (в) гібридний механізми [11]

Розглядаючи структуру та кінематику маніпуляторів і підкреслюючи важливість конфігурації маніпуляторів, можна зробити висновок, що кінематика завжди вважалася золотою жилою для дизайнерів роботів. Справедливо буде сказати, що з кінця 80-х років було кілька дослідників, які намагалися розробити зворотні кінематичні рішення, використовуючи різні методи і для різних конфігурацій роботів [10].

Роботи-маніпулятори все частіше використовуються в різних промислових умовах для виконання таких операцій, як взяття та розміщення на роботах, тому основним обмеженням було знаходження спільних змінних маніпулятора для досягнення бажаного положення з певними координатами об'єкта.

В даний час зростає потреба у використанні роботів-маніпуляторів в інших сферах, окрім промислового світу. Зараз вони можуть використовуватися в різних сферах, починаючи від клінічної реабілітації, підводного застосування,

складальних завдань, сільського господарства, гірничодобувної промисловості, космосу і т.д., і закінчуючи взаємодією з людиною.

На основі огляду літератури можна зробити висновок, що для досягнення бажаного положення і орієнтації кінцевого ефекту або інструменту також дуже важливо, щоб зберігалися рухомість і точність. Існує потреба в прямій і зворотній кінематиці при плануванні траєкторії, а також моделюванні.

Немає сумнівів, що людська рука є ключовим моментом у мотивації та рушійною силою розвитку роботи-маніпулятора. Для того, щоб з'ясувати оптимальну конструкцію з точки зору проектування, необхідно розрахувати кінематичні зв'язки між кожною спільною змінною, щоб прийти до найкращого можливого проектування.

Конструкція механізму або робота-маніпулятора може мати кілька конфігурацій, структур або функцій, але найбільш важливими для проектування характеристиками механізму або робота-маніпулятора є кінематичний аналіз, аналіз робочого простору, генерація траєкторії і управління.

Значення явних властивостей завжди надається застосуванню робота-маніпулятора щодо застосувань робота-маніпулятора. Коли кількість спільних змінних маніпулятора збільшується, то збільшується також робочий простір і рухомість робота-маніпулятора, що призводить до більш складних математичних формулювань для зворотних кінематичних дозволів і труднощів в управлінні маніпулятором.

1.2 Особливості математичних моделей роботів маніпуляторів

У багатьох сферах життєдіяльності людини, а також у промисловому застосуванні, роботи-маніпулятори зараз використовуються в різних конструкціях і конфігураціях. Важливою метою цього огляду-аналізу є вивчення різних методів і прийомів, які можуть бути використані для вирішення питань кінематики та оптимізації будь-якої конфігурації робота-маніпулятора.

Алгебраїчне розв'язання оберненої кінематики було найпоширенішим методом серед усіх розроблених методологій. Якщо конфігурація робота складна і має більший ступінь свободи, звичайну алгебру важко моделювати і отримувати відповідні рішення.

Формулювання оберненої кінематики ще більше спрощуються за допомогою кількох ефективних методів елімінування. З точки зору геометричної алгебри, коли перші три шарніри будь-якого маніпулятора або механізму не утворюють між собою жодних кутів, обернену кінематичну задачу не можна розв'язати точно [11].

Крім того, якщо матриця Якобіана знаходиться в поганому стані або страждає від сингулярностей, задача стає нестійкою. Таким чином, традиційний метод є надійним, але конфігурації маніпуляторів і КСС завжди представляють проблему математичної складності.

Дослідницька спільнота прийняла ряд інтелектуальних методів для вирішення цих проблем, включаючи штучні нейронні мережі, нечітку логіку, гібридні ШНМ та інші [10]. Використовувати ці методи дешевше і простіше, ніж традиційні алгоритмічні рішення.

Крім цих підходів, оптимізаційні підходи, такі як евристика, метаевристика, чисельні підходи тощо, показали багатообіцяючі результати для розв'язання задач оберненої кінематики роботів-маніпуляторів будь-якої конфігурації [11]. Часто в локальних точках оптимуму спостерігається ефект зациклення, який заважає більшості алгоритмів оптимізації забезпечити глобальний оптимальний розв'язок. Для того, щоб досягти оптимуму, необхідно розробити алгоритм.

Роботів-маніпуляторів можна розглядати як групу жорстких шарнірів, з'єднаних специфічними з'єднаннями. Шарніри можуть бути обертовими, призматичними, гвинтовими, універсальними або циліндричними. Робот-маніпулятор вважається таким, у якого перша ланка закріплена в основі, а остання ланка може переміщатися в робочому просторі.

Обертові шарніри обертаються навколо своїх осей, тоді як призматичні шарніри ковзають уздовж своїх осей без обертання. Існує дві основні категорії роботів-маніпуляторів, які класифікуються за типом кінематичного ланцюга, з якого вони виготовлені.

Необхідно не тільки визначити класифікацію на основі типу роботи, але також важливо зробити її на основі шарнірів і з'єднань, як пояснювалося в попередньому розділі. В ідеалі, з точки зору досліджень, ми прагнемо розробити робота-маніпулятора університетського рівня, найпростішою конфігурацією якого є робот з 4-6 осями обертання.

Існують різні типи роботів, які використовуються в промисловості для виконання різних завдань, і очевидно, що в академічних дослідженнях здебільшого використовуються роботи-маніпулятори і револьверні роботи з 6 ступенями свободи.

Роботи-маніпулятори використовують два різних типи сервомоторів: DYNAMIXEL XM-430-W350T і DYNAMIXEL XL430-W250-T. Обидва вони забезпечують моніторинг температури, а також зворотний зв'язок по положенню, рівні напруги і навантаження, і все це за допомогою зв'язку з DYNAMIXEL.

U2D2 - це комунікаційний протокол, який надає користувачеві доступ до зв'язку з серводвигунами DYNAMIXEL. Сервоприводи з'єднуються через Daisy Chain. Daisy Chain - це зв'язок між декількома елементами за допомогою одного з'єднання.

Що стосується роботів-маніпуляторів, то всі сервоприводи маніпулятора з'єднані за допомогою протокола Daisy Chain, що дозволяє оператору спілкуватися з кожним сервоприводом, незважаючи на те, що він з'єднаний лише з точкою з'єднання в основі маніпулятора робота.

Це дозволяє спростити комунікацію, а також запобігти необхідності використання декількох портів або проводів для скупчення або гальмування руху руки робота [12].

Robot Operating System (ROS) - це фреймворк, створений головним чином для підтримки роботів в Ubuntu Linux. Хоча існують експериментальні версії для

інших систем, таких як Microsoft Windows і macOS, основною метою є створення вільного фреймворку, який підтримує пакети і бібліотеки від декількох різних дистриб'юторів з точки зору знань про роботів.

Завдяки цій філософії фреймворк спирається на досвід різних секторів у створенні операційної системи (ОС), яка містить корисну документацію та код для різних застосувань для різних роботів [12].

Interbotix пропонує пакети та документацію для більшості своїх маніпуляторів для роботів, створені та протестовані в Ubuntu Linux, з метою створення простого для розуміння програмного забезпечення для моделювання.

MATLAB - це мова кодування, а точніше платформа/середовище програмування, що належить компанії MathWorks і призначена для програмування з більшим акцентом на математику, моделювання та загальну маніпуляцію з великими масивами даних.

Основне призначення MATLAB полягає в тому, щоб інтерфейсувати програми на різних мовах на основі матричних даних, тобто використовувати матриці для маніпулювання сирими даними, незважаючи на оригінальну мову збору даних.

Ця матрична спрямованість також є частиною назви MATLAB, яка повністю розшифровується як "матрична лабораторія". Сама мова має схожість з іншими мовами, такими як Python, C++ та простішими версіями C#.

MATLAB також підтримує надбудови, створені як самою компанією, так і MathWorks або іншими третіми сторонами. Ці надбудови можуть додавати багато специфічних функцій і коду, необхідних для більш специфічних цілей, таких як технологія технічного зору, машинне навчання та автомати [12].

Simulink - це надбудова для графічного моделювання, що базується на MATLAB, яка фокусується на блоковому кодуванні, а не на кодуванні за допомогою сценаріїв, хоча допускається використання блоків, визначених користувачем, на основі створених ним сценаріїв. Подібно до базової версії MATLAB, Simulink має можливість додавання великої кількості коду та блоків, створених MathWorks або третіми сторонами [12].

Частково метою в управлінні рукою робота є повне використання кінематики як частини фізики. Кінематика базується на русі фізичних елементів, яким у цьому випадку є рука робота, і на тому, як сила використовується для маніпулювання фізичним елементом [12].

Кінематика може суттєво відрізнитися у вираженні від елемента до елемента, а також між різними ситуаціями. Основна увага приділяється зворотній та прямій кінематиці, яку також називають прямою та зворотною кінематикою.

Пряма/прямолінійна кінематика охоплює більш лінійний погляд на кінематику і також може розглядатися як більш причинно-наслідковий підхід. Пряма кінематика базується на кінематиці, в якій ми можемо задати деякі відомі кути зчленування маніпулятора робота, що дозволить розрахувати положення кінцевого ефектора в результаті руху маніпулятора.

Використовуючи пряму кінематику, ми можемо самі вибирати положення суглобів, а також знаходити/бачити положення кінцевого ефектора в 3D-просторі на основі положень суглобів.

Інверсійна кінематика фокусується на кінцевій цілі руху і, виходячи з цього, розраховує необхідні кути, необхідні для з'єднань. Це може бути, наприклад, за допомогою знання положення захоплювача, руки робота і заявленого положення кінцевого ефектора, і на основі цього відхилення розрахувати результуючі кути/крутний момент, необхідні для переміщення захоплювача до цілі.

Для того, щоб розв'язати обернену кінематичну задачу, ми припускаємо, що положення кінцевого ефектора руки робота відоме. Кінцевим ефектором маніпулятора, як ми вже згадували раніше, є центр лівої та правої ланок захоплювача. Для роботи інверсної кінематики необхідно спочатку визначити положення кінцевих ефektorів у просторі. Генерація траєкторії - це процес визначення цього положення.

1.3 Висновки до першого розділу

1. Під час роботи робота маніпулятори рухаються по заданих траєкторіях, які являють собою послідовності точок, що відображають положення кінцевих ефекторів, а також їхню орієнтацію. Залежно від кінематики, траєкторія може бути як у спільному просторі, так і в декартовому просторі, що базується на часі.

2. За ступенем свободи, якою володіє робот-маніпулятор, можна судити про його робочу досконалість. Типові 6-координатні роботи-маніпулятори, як правило, можуть виконувати загальне завдання в 3-вимірному просторі, де об'єкт може перебувати в будь-якій точці простору в будь-який час, незалежно від його положення або орієнтації.

3. Роботи-маніпулятори зазвичай бувають послідовними або паралельними, залежно від того, чи містять вони відкритий або замкнутий контур. Існує два типи роботів-маніпуляторів, які використовуються в реальних умовах: призматичні (П) і револьверні (Р) з'єднання, причому один з можливих типів з'єднань може бути жорстким і гнучким.

4. На основі огляду літератури можна зробити висновок, що для досягнення бажаного положення і орієнтації кінцевого ефекту або інструменту також дуже важливо, щоб зберігалися рухомість і точність. Існує потреба в прямій і зворотній кінематиці при плануванні траєкторії, а також моделюванні.

5. Роботів-маніпуляторів можна розглядати як групу жорстких шарнірів, з'єднаних специфічними з'єднаннями. Шарніри можуть бути обертовими, призматичними, гвинтовими, універсальними або циліндричними. Робот-маніпулятор вважається таким, у якого перша ланка закріплена в основі, а остання ланка може переміщатися в робочому просторі.

6. Обертові шарніри обертаються навколо своїх осей, тоді як призматичні шарніри ковзають уздовж своїх осей без обертання. Існує дві основні категорії роботів-маніпуляторів, які класифікуються за типом кінематичного ланцюга, з якого вони виготовлені.

2 МАТЕМАТИЧНА МОДЕЛЬ ПРОЦЕСУ КЕРУВАННЯ РОБОТОМ МАНІПУЛЯТОРОМ

2.1 Математичні моделі складових компонентів роботи маніпулятора

Перетворення (поворот або трансляція) системи координат також називається однорідним перетворенням. Для того, щоб повернути систему координат на кут q_1 , ми можемо використати матрицю повороту у двовимірній площині у відповідності до виразу (2.1).

$$R_{(q_1)} = \begin{bmatrix} \cos_{q_1} & -\sin_{q_1} & 0 \\ \sin_{q_1} & \cos_{q_1} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.1)$$

Для того, щоб перевести координатну рамку на l_x і l_y довжини/відстані, ми можемо використовувати матрицю перекладу у двовимірній площині у відповідності до виразу (2.2).

$$T(l_x l_y) = \begin{bmatrix} 1 & 0 & l_x \\ 0 & 1 & l_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.2)$$

Щоб знайти положення кінцевого ефектора (враховуючи кути суглоба) за допомогою прямої кінематики, слід виконати наступні перетворення в їхньому порядку.

1. Повернути опорну систему координат x-y на кут q_1 проти годинникової стрілки у відповідності до рисунку 2.1 (b).
2. Перемістити у напрямку x на довжину l_1 у відповідності до рисунку 2.1 (c).
3. Поворот на кут з'єднання q_2 проти годинникової стрілки у відповідності до рисунку 2.1 (d).

4. Перемістити у напрямку x на довжину l_2 у відповідності до рисунку 2.1 (e).

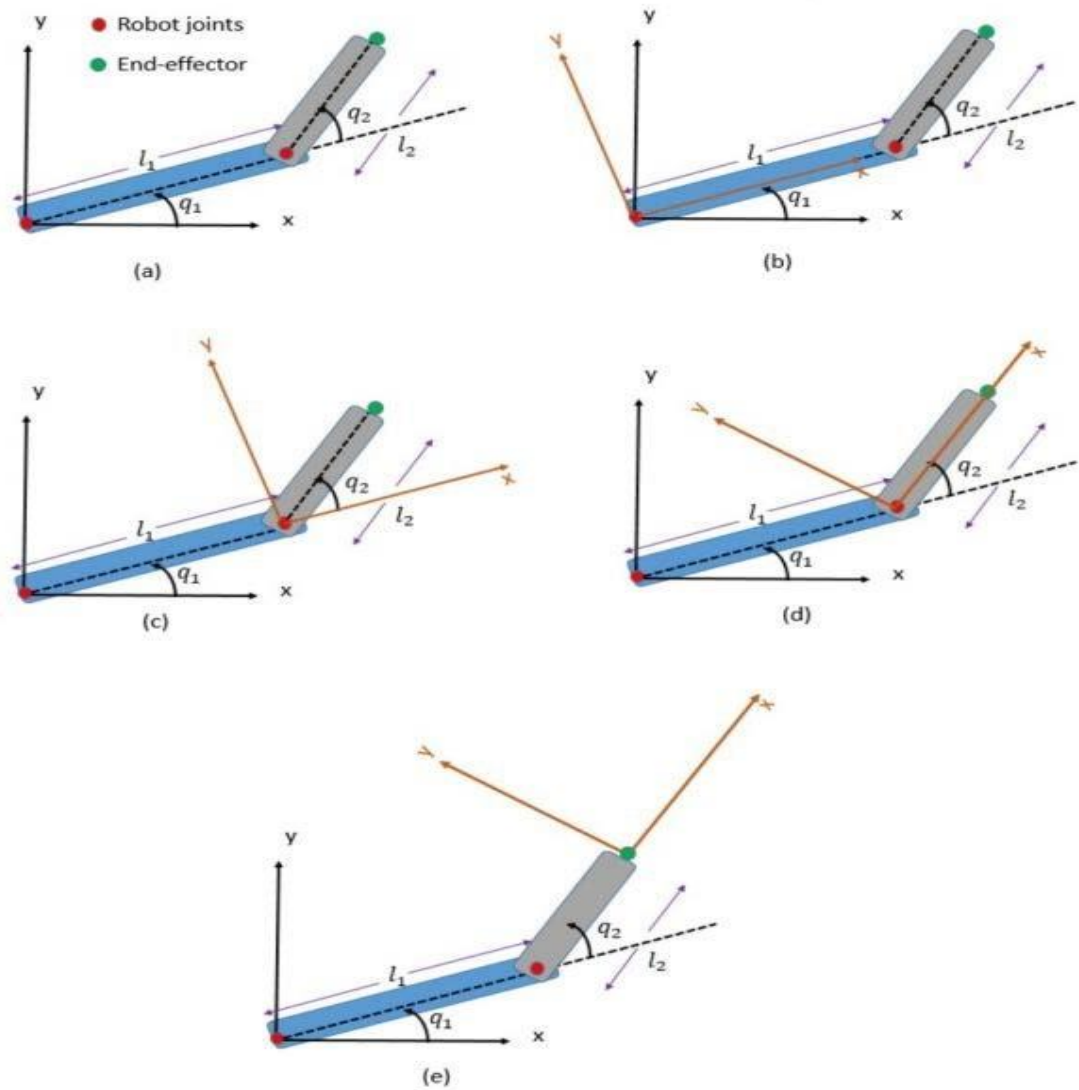


Рисунок 2.1 – Перетворенням координат для робота-маніпулятора [2]



Рисунок 2.2 – Зображення робота-маніпулятора [3]

Тоді однорідна матриця перетворення задається добутком матриць обертання та трансляції у відповідності до виразу (2.3).

$$E = R_{q_1} \cdot T_{xl_1} \cdot R_{q_2} \cdot T_{xl_2} =$$

$$= \begin{bmatrix} \cos(q_1+q_2) & -\sin(q_1+q_2) & l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) \\ \sin(q_1+q_2) & \cos(q_1+q_2) & l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.3)$$

З матриці однорідних перетворень ми можемо знайти положення кінцевого ефектора (x_e, y_e) у відповідності до виразів (2.4) і (2.5).

$$x_e = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2); \quad (2.4)$$

$$y_e = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2). \quad (2.5)$$

Наведені вище рівняння обчислюють результуючі координати x і y на основі заданої довжини кожного плеча шарніра, l_1 і l_2 , і кута кожного шарніра, q_1 і q_2 . Це пряма кінематика. Відомі кути і довжини дають координату.

Для оберненої кінематики проблема полягає в тому, щоб знайти кути зчленування q_1 і q_2 , які б орієнтували руку робота таким чином, щоб кінцевий ефект досягав заданого положення (x_e, y_e) .

$$q_1 = \tan^{-1} \left(\frac{y_e}{x_e} \right) - \tan^{-1} \left(\frac{l_2 \sin(q_2)}{l_1^2 + l_1 l_2 \cos(q_2) + l_2 \cos(q_2)} \right); \quad (2.6)$$

$$q_2 = \cos^{-1} \left(\frac{x_e^2 + y_e^2 - l_1^2 - l_2^2}{2l_1 l_2} \right). \quad (2.7)$$

Наведені вище рівняння для q_1 і q_2 ґрунтуються на знанні довжин шарнірів, l_1 і l_2 , як і раніше, але замість кутів тепер потрібні координати x і y . Це зворотна кінематика. Оскільки координати задані, для знаходження відповідних кутів потрібні більш складні рівняння.

На додаток до того, що рівняння для q_1 є складнішим, воно вимагає також результуючого кута для q_2 , таким чином показуючи залежність між кутами при використанні оберненої кінематики. Векторна схема маніпулятора робота з двома ступенями свободи показана на рисунку 2.1, а більш реалістичне зображення робота-маніпулятора показано на рисунку 2.2.

Обернені кінематичні рішення можуть бути визначені за допомогою різних алгоритмів оптимізації. Для застосування алгоритмів оптимізації достатньо визначити цільову функцію для відповідного маніпулятора.

Для кожної окремої спільної змінної за допомогою цільової функції, яка може бути визначена вектором конфігурації маніпулятора, також визначається можливий розв'язок для кожної окремої спільної змінної.

Пряме кінематичне рівняння і пов'язані з ним константи або параметри - це все, що потрібно для цього методу. Використовуючи цю методологію, можна вирішити багато завдань, пов'язаних з роботами-маніпуляторами, таких як проектування, кінематичний аналіз і синтез кінематичних структур.

Крім того, функція оптимізації 'fmincon' та алгоритм 'sqp' у MATLAB можуть бути використані для оптимізації задач роботів-маніпуляторів з більшою кількістю ступенів свободи та більш складних задач.

Нелінійність повинна оброблятися алгоритмами оптимізації. Обернену кінематику можна розв'язати за допомогою будь-якого алгоритму оптимізації, який здатен розв'язувати різні мультимодальні функції.

Оптимізаційний підхід може бути застосований для подолання обмежень звичайних інструментів та інтелектуальних методів [10]. Вважається, що оптимізаційні методи є більш стабільними і часто призводять до збіжних глобальних розв'язків.

Поточне положення маніпулятора описується виразом (2.8).

$$P_{current} = [x_e, y_e, z_e]. \quad (2.8)$$

Бажане положення кінцевого ефектора можна позначити виразом (2.9).

$$P_{desired} = [x_{e_hat}, y_{e_hat}, z_{e_hat}]. \quad (2.9)$$

Однорідна евклідова відстань між поточним та бажаним положеннями кінцевих ефекторів оцінюється за кількістю ітерацій.

$$e_{min} = p_{desired(i)} - P_{current(i)}. \quad (2.10)$$

Поточне положення x_e, y_e, z_e можна визначити за допомогою виразів.

$$\begin{aligned} x_e = & (\cos(q1) \cdot \cos(q2))/20 + (3 \cdot \cos(q4) \cdot (\cos(q1) \cdot \cos(q2) \cdot \cos(q3) - \\ & - \cos(q1) \cdot \sin(q2) \cdot \sin(q3)))/20 - (3 \cdot \sin(q4) \cdot (\cos(q1) \cdot \cos(q2) \cdot \sin(q3) + \\ & + \cos(q1) \cdot \cos(q3) \cdot \sin(q2)))/20 + (3 \cdot \cos(q1) \cdot \cos(q2) \cdot \cos(q3))/20 - \\ & - (3 \cdot \cos(q1) \cdot \sin(q2) \cdot \sin(q3))/20. \end{aligned} \quad (2.11)$$

$$\begin{aligned} y_e = & (\cos(q2) \cdot \sin(q1))/20 - (3 \cdot \cos(q4) \cdot (\sin(q1) \cdot \sin(q2) \cdot \sin(q3) - \\ & - \cos(q2) \cdot \cos(q3) \cdot \sin(q1)))/20 - (3 \cdot \sin(q4) \cdot (\cos(q2) \cdot \sin(q1) \cdot \sin(q3) + \\ & + \cos(q3) \cdot \sin(q1) \cdot \sin(q2)))/20 - (3 \cdot \sin(q1) \cdot \sin(q2) \cdot \sin(q3))/20 + \\ & + (3 \cdot \cos(q2) \cdot \cos(q3) \cdot \sin(q1))/20. \end{aligned} \quad (2.12)$$

$$\begin{aligned} z_e = & \sin(q2)/20 + (3 \cdot \cos(q2) \cdot \sin(q3))/20 + \\ & + (3 \cdot \cos(q3) \cdot \sin(q2))/20 + (3 \cdot \cos(q4) \cdot (\cos(q2) \cdot \sin(q3) + \\ & + \cos(q3) \cdot \sin(q2)))/20 + (3 \cdot \sin(q4) \cdot (\cos(q2) \cdot \cos(q3) - \\ & - \sin(q2) \cdot \sin(q3)))/20 + 1/4. \end{aligned} \quad (2.13)$$

За умови дотримання спільних обмежень в радіусі у відповідності до виразів.

$$-1,57 \leq q_1 \leq 1,57; \quad (2.14)$$

$$-1,57 \leq q_2 \leq 1,57; \quad (2.15)$$

$$-1,57 \leq q_3 \leq 1,57; \quad (2.16)$$

$$-1,57 \leq q_4 \leq 1,57. \quad (2.17)$$

2.2 Удосконалений метод керування роботом маніпулятором

Перед підключенням і запуском будь-якого коду дуже корисно підключитися до маніпулятора робота за допомогою програми DYNAMIXEL Wizard, яка надається компанією DYNAMIXEL. З безлічі різних функцій програми найважливішою функцією є перегляд всієї інформації про кожен регістр/сервопривід, тобто поточне положення, температуру сервоприводу, значення PID і т.д.

Більше того, керування сервоприводами також можна здійснювати з цієї програми. За допомогою цього майстра також можна встановити межі безпеки для різних сервоприводів. Найбільш корисною частиною цієї програми є початкове сканування на наявність різних швидкостей передачі даних і DXL_ID, який є ідентифікатором для різних сервоприводів. Майстер DYNAMIXEL показано на Рисунку 2.2 [12].

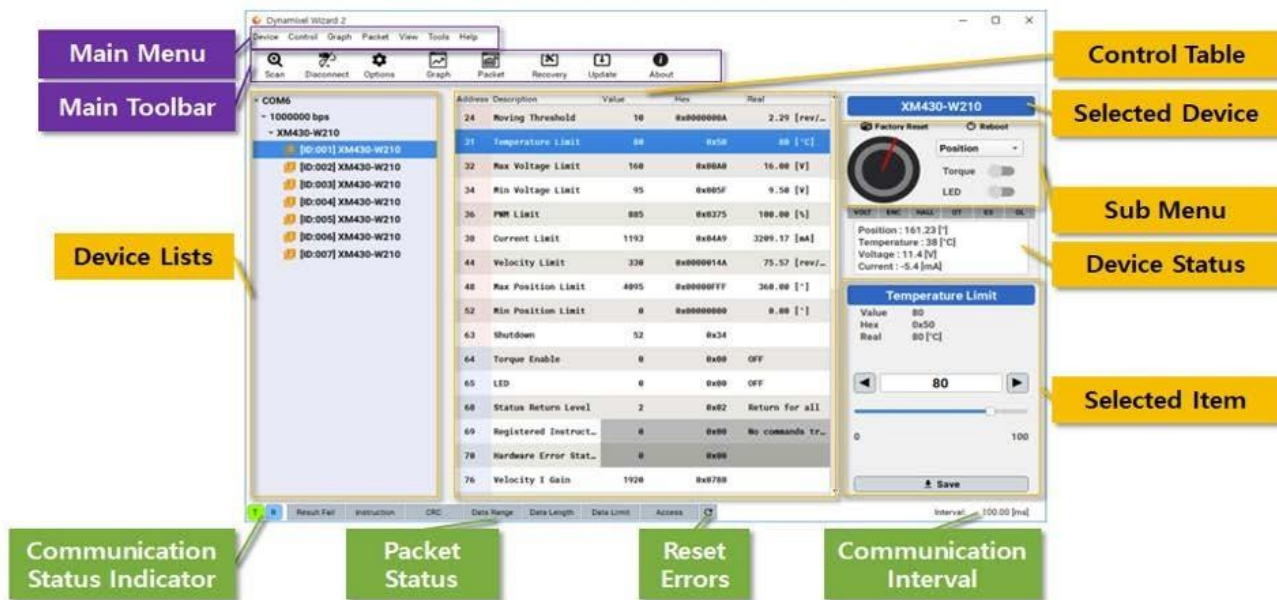


Рисунок 2.2 - Майстер DYNAMIXEL

Маніпулятор робота використовує серводвигуни Dynamixel у кожному суглобі. У цій руці маніпулятора використовуються два типи серводвигунів

(XL430 і XM430). Серводвигуни Dynamixel мають різні режими роботи, які представлені в таблиці 2.1.

Таблиця 2.1 - Різні режими роботи сервоприводу

Mode	Description
(0) Current Control	This mode can also be referred and torque mode. You need to have own controller.(Built-in controller is disabled in this mode)
(1) Velocity Control	To control velocity of wheel
(3) Position Control	Controlled by internal PID. Only need to update goal position.
(5) Extended-Position Control	Same and Position Control but <u>upto</u> 512 full turn.
(5) Current base position control	Controls both position and current (torque)
(16) PWM control	Directly controls PWM for output

Інший спосіб підключення до маніпулятора робота - це DYNAMIXEL SDK. Це програмна бібліотека, в основному створена на мовах C і C++, яка дозволяє користувачеві спілкуватися з різними блоками DYNAMIXEL, такими як сервоприводи. Ця бібліотека є невід'ємною для взаємодії з будь-яким іншим маніпулятором робота на базі ROS. Ця бібліотека доступна на декількох різних мовах програмування, таких як C, C#, C++, LabVIEW, JAVA, MATLAB та ін. [12].

Бібліотека MATLAB використовувалася як інтерфейс для спілкування серводвигунами робота. Бібліотека включає декілька методів, які спрощують процес комунікації. Деякі з найбільш актуальних методів включають файл "read_write.m", який використовується для простого спілкування з рукою, "bulk_read_write.m", який використовується для декількох повідомлень, і "rebooting.m", який допомагає перезавантажити робота, не маючи можливості зробити щось неправильно [12].

Файл "read_write.m" має бути першою програмою MATLAB, яку слід змінити, оскільки інші програми будуються на методі обміну даними. Програма використовує цикл while, в якому виконується обмін даними, і в кожному циклі програма чекає на наступне введення даних користувачем.

Спочатку ця програма надсилає команду першому сервоприводу (DXL_ID_1) для повороту на 360°, який є базовим сервоприводом. Пізніше програма модифікується, щоб дати команду всім 5 сервоприводам в одному циклі while. Для звернення до окремих сервоприводів використовуються різні DXL_ID від 1 до 5 [12].

Файл починається з різних адрес, які є важливими для комунікації. Це можуть бути, наприклад, адреси крутного моменту, швидкості, положення тощо. Фрагмент коду "read_write.m" з адресами.

% Адреса таблиці керування

```
ADDR_PRO_TORQUE_ENABLE      = 64; % У моделі DYNAMIXEL інша адреса таблиці
                               керування
ADDR_PRO_GOAL_POSITION      = 116;
ADDR_PRO_PRESENT_POSITION   = 132;
ADDR_PRO_GOAL_VELOCITY      = 104;
```

Необхідно перевірити АДРЕСУ регістрів, яка може не збігатися з роботою робота. Швидкість передачі даних за замовчуванням зазвичай становить 1 000 000, але за потреби її можна змінити.

Зв'язок з маніпулятором здійснюється через порт комп'ютера, номер якого можна знайти в диспетчері пристроїв у Windows. Після з'єднання комп'ютера з маніпулятором цей порт з'являється в програмі як порт зв'язку.

Після з'єднання наступна мета програми - записати інструкції за відповідними адресами на руку робота. Після успішного з'єднання наступною частиною коду буде ввімкнення крутного моменту сервоприводу. Код, який вмикає крутний момент, має вигляд.

Якщо з'єднання не вдалося, програма завершується [12]. Після ввімкнення крутного моменту програма запускає цикл, який безперервно зчитує та очікує на введення користувачем. Приклад коду наразі зв'язується з сервоприводом номер 1, який є сервоприводом на основі обертання.

Розширення на всі п'ять сервоприводів здійснюється шляхом розширення програми для включення перевірок інших входів клавіатури, пов'язаних з іншими сервоприводами, що розрізняються за відповідними номерами DXL_ID.

```

% Увімкнути Dynamixel Torque DXL_ID_2
writelByteTxRx(port_num, PROTOCOL_VERSION, DXL_ID_2,
ADDR_PRO_TORQUE_ENABLE, TORQUE_ENABLE);
dxl_comm_result = getLastTxRxResult(port_num, PROTOCOL_VERSION);
dxl_error = getLastRxPacketError(port_num, PROTOCOL_VERSION);
if dxl_comm_result ~= COMM_SUCCESS
    fprintf('%s\n', getTxRxResult(PROTOCOL_VERSION, dxl_comm_result));
elseif dxl_error ~= 0
    fprintf('%s\n', getRxPacketError(PROTOCOL_VERSION, dxl_error));
else
    fprintf('Dynamixel було успішно підключено \n');

```

Робот-маніпулятор має 2 сервоприводи XM430-W350-T та 4 сервоприводи XL430-W250-T. Обидва типи сервоприводів працюють з 32-бітним контролером ARM Cortex-M3.

Вони також оснащені 12-бітним безконтактним енкодером, що дозволяє здійснювати рух на 360° без використання кабелів. Він підтримує швидкість передачі даних від 9600 до 4,5 Мбіт/с, використовується швидкість 1 Мбіт/с. Кожен сервопривід має роздільну здатність 4096 імпульсів/об, що забезпечує високу точність і має PID-регулювання [12].

Кожен сервопривід має чотири різні режими: Керування швидкістю, керування положенням, розширене керування положенням і ШІМ-керування. ШІМ-керування працює зі спектром напруги, а розширене керування положенням дозволяє здійснювати кілька поворотів замість обмеження на 360° у режимі керування положенням. У цій дипломній роботі використовується керування положенням, оскільки воно вважається найбільш відповідним для позиціонування.

Фізичне з'єднання між комп'ютером і роботом-маніпулятором - це напівдуплексний асинхронний послідовний зв'язок TTL. Напівдуплекс означає, що обидва пристрої можуть читати і писати, але не одночасно. Пакети послідовного зв'язку складаються з 8 біт, 1 стартового біта і без перевірки на парність.

Сервоприводи DYNAMIXEL підтримують два різних протоколи: DYNAMIXEL Protocol 1.0 і DYNAMIXEL Protocol 2.0, де протокол 2.0 є

рекомендованим. Ці протоколи використовують пакети інструкцій для виконання завдань маніпулятором робота.

Пакети побудовані відповідно до Рисунку 2.3. "ID" - це ідентифікатор сервоприводу, "Length" - довжина пакета, яка повинна гарантувати отримання всього пакета, "Inst" - дана інструкція для руки робота, а "Param" - додаткові параметри для виконання інструкції, які не є обов'язковими для всіх інструкцій [12].

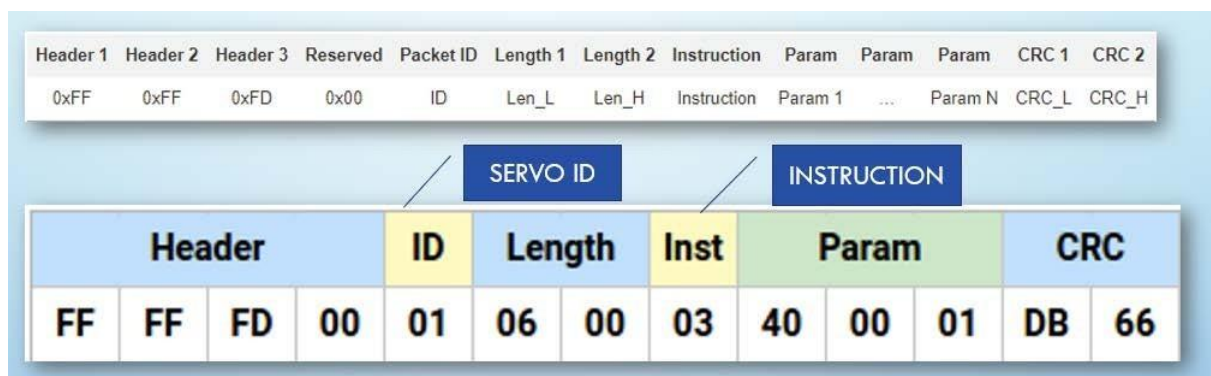


Рисунок 2.3 - Накопичення комунікаційних пакетів

Окрім надсилання пакетів, можна також зчитувати статус надісланого пакета, щоб перевірити наявність помилок зв'язку. Приклад такого зчитування пакета зображено на рисунку 2.4.

Header 1	Header 2	Header 3	Reserved	Packet ID	Length 1	Length 2	Instruction	ERR	PARAM	PARAM	PARAM	CRC 1	CRC 2
0xFF	0xFF	0xFD	0x00	ID	Len_L	Len_H	Instruction	Error	Param 1	...	Param N	CRC_L	CRC_H

Error Number	Error	Description
0x01	Result Fail	Failed to process the sent Instruction Packet
0x02	Instruction Error	Undefined Instruction has been used Action has been used without Reg Write
0x03	CRC Error	CRC of the sent Packet does not match
0x04	Data Range Error	Data to be written in the corresponding Address is outside the range of the minimum/maximum value
0x05	Data Length Error	Attempt to write Data that is shorter than the data length of the corresponding Address (ex: when you attempt to only use 2 bytes of a item that has been defined as 4 bytes)
0x06	Data Limit Error	Data to be written in the corresponding Address is outside of the Limit value
0x07	Access Error	Attempt to write a value in an Address that is Read Only or has not been defined Attempt to read a value in an Address that is Write Only or has not been defined Attempt to write a value in the ROM domain while in a state of Torque Enable(ROM Lock)

Рисунок 2.4 – Пакет стану

Існує декілька різних інструкцій, а також різні інструкції, що залежать від таблиці керування, на якій працюють сервоприводи. Як вже згадувалося раніше, сервоприводи отримують інструкції у вигляді пакету. Приклад такої таблиці інструкцій наведено на рисунку 2.5, а схема передачі пакетів інструкцій між програмним забезпеченням і маніпулятором робота показана на рисунку 2.6.

Value	Instructions	Description
0x01	Ping	Instruction that checks whether the Packet has arrived to a device with the same ID as Packet ID
0x02	Read	Instruction to read data from the Device
0x03	Write	Instruction to write data on the Device
0x04	Reg Write	Instruction that registers the Instruction Packet to a standby status; Packet is later executed through the Action command
0x05	Action	Instruction that executes the Packet that was registered beforehand using Reg Write
0x06	Factory Reset	Instruction that resets the Control Table to its initial factory default settings
0x08	Reboot	Instruction to reboot the Device
0x10	Clear	Instruction to reset certain information
0x20	Control Table Backup	Instruction to store current Control Table status data to a Backup area or to restore EEPROM data.
0x55	Status(Return)	Return packet for the Instruction Packet
0x82	Sync Read	For multiple devices, Instruction to read data from the same Address with the same length at once
0x83	Sync Write	For multiple devices, Instruction to write data on the same Address with the same length at once
0x8A	Fast Sync Read	For multiple devices, Instruction to read data from the same Address with the same length at once
0x92	Bulk Read	For multiple devices, Instruction to read data from different Addresses with different lengths at once
0x93	Bulk Write	For multiple devices, Instruction to write data on different Addresses with different lengths at once
0x9A	Fast Bulk Read	For multiple devices, Instruction to read data from different Addresses with different lengths at once

Рисунок 2.6 - Таблиця інструкцій

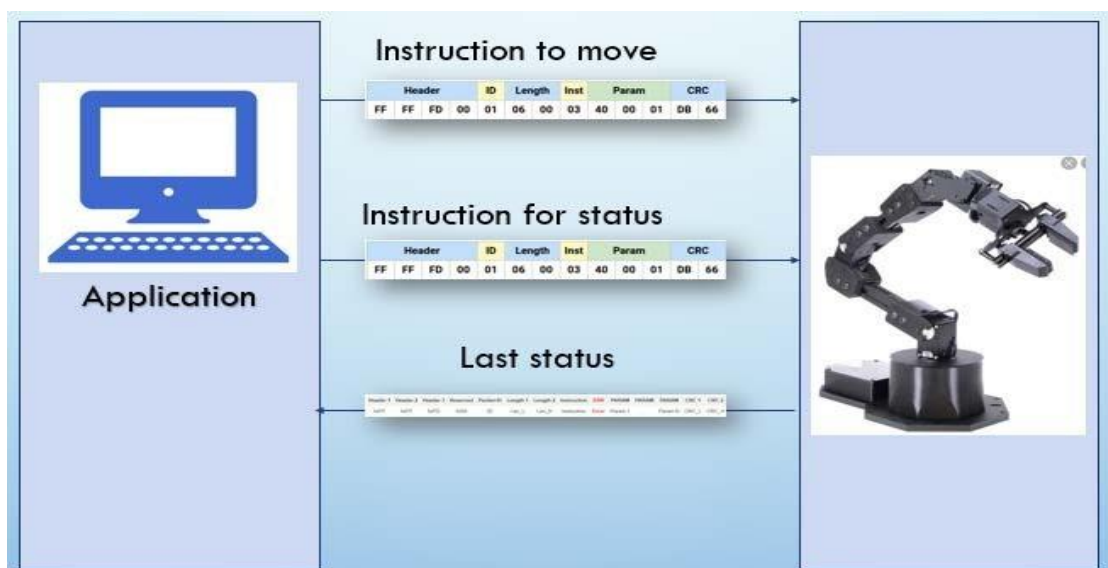


Рисунок 2.6 - Схема передачі пакетів інструкцій між програмним забезпеченням і маніпулятором робота

В рамках декількох налаштувань і значень регістрів, доступних в сервоприводах, також можна встановити обмеження або отримати значення регістрів з сервоприводів.

Ці значення розділені на дві різні таблиці: таблиця області EEPROM і таблиця області RAM. Дані оперативної пам'яті - це дані, які видаляються при вимиканні, тоді як дані EEPROM - це дані, які зберігаються разом з сервоприводами.

Дані ОЗП можуть, наприклад, містити інформацію про поточну швидкість, положення, якщо ввімкнено крутний момент, і траєкторію, тоді як дані EEPROM - це дані про номер моделі, ідентифікатор номер, швидкість передачі та обмеження. Зображення початку кожної таблиці даних зображено на рисунку 2.7 та рисунку 2.8.

Address	Size(Byte)	Data Name	Access	Initial Value	Range	Unit
0	2	Model Number	R	1,020	-	-
2	4	Model Information	R	-	-	-
6	1	Firmware Version	R	-	-	-
7	1	ID	RW	1	0 ~ 252	-
8	1	Baud Rate	RW	1	0 ~ 7	-
9	1	Return Delay Time	RW	250	0 ~ 254	2 [µsec]
10	1	Drive Mode	RW	0	0 ~ 5	-
11	1	Operating Mode	RW	3	0 ~ 16	-
12	1	Secondary(Shadow) ID	RW	255	0 ~ 252	-
13	1	Protocol Type	RW	2	1 ~ 2	-
20	4	Homing Offset	RW	0	-1,044,479 ~ 1,044,479	1 [pulse]

Рисунок 2.7 - Початок таблиці даних області EEPROM

Програмні компоненти, які складають програмний стек, можуть включати віртуалізацію або абстрактні апаратні ресурси, а також інші компоненти, необхідні для запуску програми. Стекування - це процес, який об'єднує окремі компоненти для підтримки роботи програми.

Address	Size(Byte)	Data Name	Access	Initial Value	Range	Unit
64	1	Torque Enable	RW	0	0 ~ 1	-
65	1	LED	RW	0	0 ~ 1	-
68	1	Status Return Level	RW	2	0 ~ 2	-
69	1	Registered Instruction	R	0	0 ~ 1	-
70	1	Hardware Error Status	R	0	-	-
76	2	Velocity I Gain	RW	1,920	0 ~ 16,383	-
78	2	Velocity P Gain	RW	100	0 ~ 16,383	-
80	2	Position D Gain	RW	0	0 ~ 16,383	-
82	2	Position I Gain	RW	0	0 ~ 16,383	-
84	2	Position P Gain	RW	800	0 ~ 16,383	-
88	2	Feedforward 2nd Gain	RW	0	0 ~ 16,383	-
90	2	Feedforward 1st Gain	RW	0	0 ~ 16,383	-
98	1	Bus Watchdog	RW	0	1 ~ 127	20 [msec]
100	2	Goal PWM	RW	-	-PWM Limit(36) ~ PWM Limit(36)	0.113 [%]

Рисунок 2.8 - Початок таблиці даних області оперативної пам'яті

В ієрархічній структурі кожен компонент складається з архітектурного рівня, протоколів, операційної системи, середовища виконання, баз даних і викликів функцій. Компоненти вищого рівня зазвичай виконують конкретні завдання і надають послуги кінцевим користувачам - це додаток MATLAB, в той час як компоненти нижчого рівня зазвичай взаємодіють з апаратним забезпеченням, яким є комунікаційна плата маніпулятора робота.

За допомогою складних інструкцій, що проходять через стек, компоненти взаємодіють безпосередньо з додатками. Бібліотека "MATLAB C" та програмний протокол U2D2 використовуються сервоприводом Dynamixel на найнижчому рівні, як показано на рисунку 2.9.

Налаштування, калібрування, тестування серводвигуна виконується за допомогою майстра Dynamixel, показаного на рисунку 2.10, перед тим, як підключатись до MATLAB.

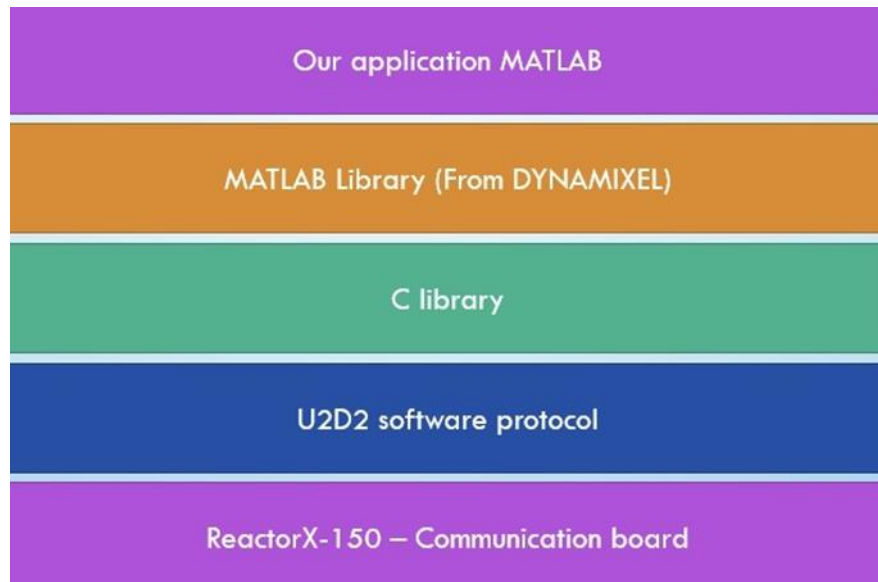


Рисунок 2.9 - Потік програмного стеку

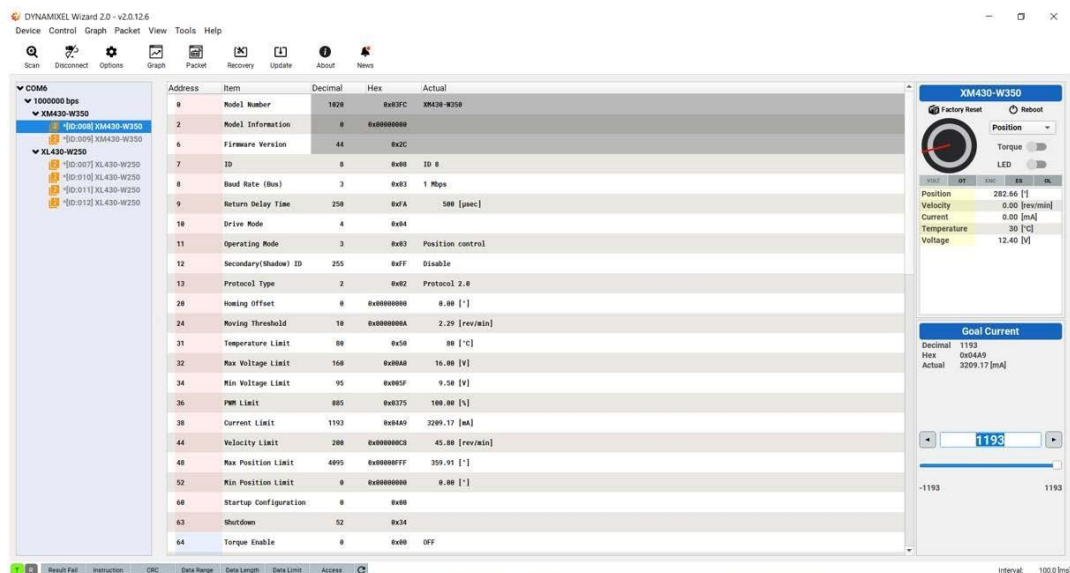


Рисунок 2.10 - Налаштування, калібрування, тестування серводвигуна

Ідентифікатор сервоприводу Dynamixel, режим керування, порт для зв'язку, профільне прискорення та швидкість, адреса цілі та поточної позиції, перезавантаження/скидання сервоприводу двигуна в разі переважання можна контролювати і змінювати за допомогою цього додатку.

Під час інтерфейсу через MATLAB, перше, що потрібно зробити, це запустити скрипт ініціалізації обладнання та перевірити, чи всі сервоприводи взаємодіють з ним.

На рисунку 2.11 показано результат зв'язку з серводвигуном у командному вікні MATLAB.

The screenshot shows the MATLAB R2020a academic use interface. The Editor window displays a script named 'Hardware_initialization_file.m' with the following code:

```

1 % clear;
2 % clear all;
3
4 lib_name = '';
5
6 if strcmp(computer, 'PCWIN')
7     lib_name = 'dxl_x86_c';
8 elseif strcmp(computer, 'PCWIN64')
9     lib_name = 'dxl_x64_c';
10 elseif strcmp(computer, 'GLNX86')
11     lib_name = 'libdxl_x86_c';
12 elseif strcmp(computer, 'GLNX64')
13     lib_name = 'libdxl_x64_c';
14 elseif strcmp(computer, 'MACI64')
15     lib_name = 'libdxl_mac_c';
16 end
17

```

The Command Window shows the following output:

```

>> Hardware_initialization_file
Succeeded to open the port!
Succeeded to change the baudrate!
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
Dynamixel has been successfully connected
fx >>

```

The Workspace window shows the following variables:

Name	Value
A1	dxl sym
A2	dxl sym
A3	dxl sym
A4	dxl sym
ADDR_PROG_GOAL_POSITION	116
ADDR_PROG_GOAL_VELOCITY	104
ADDR_PROG_PRESENT_POSIT...	132
ADDR_PROG_TORQUE_ENAB...	64
B1	0.0500
BAUDRATE	1000000

Рисунок 2.11 - Результат зв'язку з серводвигуном у командному вікні MATLAB

Якщо у зв'язку з будь-яким серводвигуном виникають проблеми, то у командному вікні з'являється повідомлення про помилку "There is no status packet" ("Немає пакету стану"). У цьому випадку користувач повинен перевірити у майстрі Dynamixel, чи невірно вказано ідентифікатор сервоприводу у скрипті MATLAB, чи не відключився серводвигун.

Після успішного підключення серводвигуна Dynamixel через програмний протокол U2D2, користувач може запустити потрібний сценарій MATLAB або модель Simulink, де адреса запису були записані всередині функції MATLAB. На рисунку 2.12 показано функцію MATLAB Hardware у Simulink.

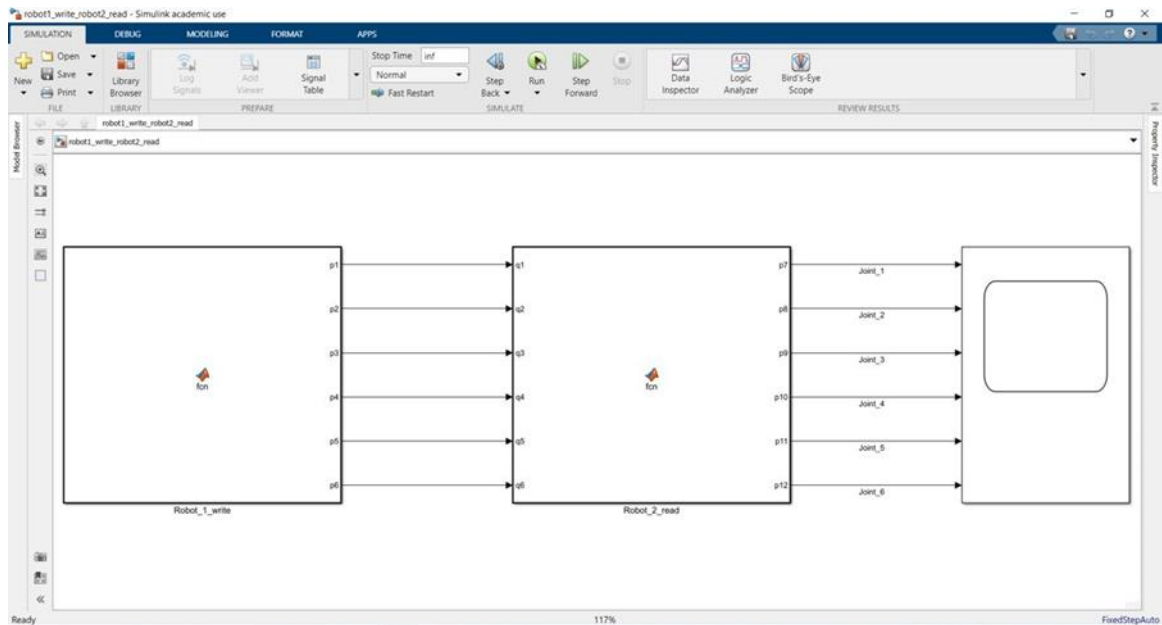


Рисунок 2.12 - Функція MATLAB Hardware у Simulink

2.3 Висновки до другого розділу

1. Перетворення (поворот або трансляція) системи координат також називається однорідним перетворенням. Для того, щоб повернути систему координат на кут q_1 , ми можемо використати матрицю повороту у двовимірній площині.

2. Для оберненої кінематики проблема полягає в тому, щоб знайти кути зчленування q_1 і q_2 , які б орієнтували руку робота таким чином, щоб кінцевий ефект досягав заданого положення (x_e, y_e) .

3. Перед підключенням і запуском будь-якого коду дуже корисно підключитися до маніпулятора робота за допомогою програми DYNAMIXEL Wizard, яка надається компанією DYNAMIXEL. З безлічі різних функцій програми найважливішою функцією є перегляд всієї інформації про кожен регістр/сервопривід, тобто поточне положення, температуру сервоприводу, значення PID і т.д.

4. Зв'язок з маніпулятором здійснюється через порт комп'ютера, номер якого можна знайти в диспетчері пристроїв у Windows. Після з'єднання комп'ютера з маніпулятором цей порт з'являється в програмі як порт зв'язку.

5. Після з'єднання наступна мета програми - записати інструкції за відповідними адресами на руку робота. Після успішного з'єднання наступною частиною коду буде ввімкнення крутного моменту сервоприводу. Код, який вмикає крутний момент.

6. В ієрархічній структурі кожен компонент складається з архітектурного рівня, протоколів, операційної системи, середовища виконання, баз даних і викликів функцій. Компоненти вищого рівня зазвичай виконують конкретні завдання і надають послуги кінцевим користувачам - це додаток MATLAB, в той час як компоненти нижчого рівня зазвичай взаємодіють з апаратним забезпеченням, яким є комунікаційна плата маніпулятора робота.

7. За допомогою складних інструкцій, що проходять через стек, компоненти взаємодіють безпосередньо з додатками. Бібліотека "MATLAB C" та програмний протокол U2D2 використовуються сервоприводом Dynamixel на найнижчому рівні

3 ІМІТАЦІЙНА МОДЕЛЬ МЕТОДУ КЕРУВАННЯ РОБОТОМ МАНІПУЛЯТОРОМ

3.1 Особливості імітаційної моделі методу керування роботом маніпулятором

Щоб успішно інтегрувати MATLAB з роботом-маніпулятором, спочатку необхідно створити функціонуючий симулятор. Цей симулятор повинен відтворювати функції фізичної руки, а також працювати безпечно, оскільки можна уникнути фізичних пошкоджень.

Interbotix пропонує багато симуляцій та пакетів для ROS2, і оскільки всі симуляції зберігаються у форматі хасго, основною метою є перетворення цих файлів та симуляцій у формат, придатний для MATLAB.

Після встановлення та запуску ROS2 на платформі Linux хасго-файли були конвертовані у формат Unified Robot Description Format (URDF). Ці файли були отримані з репозиторію Interbotix GitHub [4].

Таке перетворення підтримується в ROS2 і просто вказує на конкретний маніпулятор робота, на якому базуються файли. Існують симулятори для всіх вищезгаданих маніпуляторів і в залежності від маніпулятора ім'я файлу має бути змінено [12].

Формати файлів URDF і хасго засновані на стандартному форматі XML, що робить конвертацію можливою. Файл URDF можна імпортувати і використовувати в MATLAB з Simulink для моделювання робота.

Для моделювання маніпулятора робота в середовищі MATLAB необхідні моделі URDF і дерева жорсткого тіла. Деревоподібна модель - це один із способів представлення механічних властивостей маніпулятора робота в MATLAB, а також для візуалізації та керування маніпулятором робота в Simulink.

Жорстка ланка в цьому випадку жорстко прикріплена до батьківського тіла. Як впливає з назви, обертове з'єднання дозволяє тілу/ланці обертатися навколо однієї зі своїх осей (осі x , y або z) по відношенню до батьківської.

Слід зазначити, що при використанні призматичного з'єднання є тільки можливість трансляції (і немає обертання). Згідно з віссю руху, тіло рухається лінійно відносно своєї батьківської ланки.

Для опису робота необхідно вказати декілька деталей, таких як тип шарнірів, довжину ланок, масу ланки, а також інерційні властивості ланки/тіла і, в ідеалі, початкове положення (координату) тіла [12].

Існує файл під назвою URDF (Universal Robot Description Format - універсальний формат опису роботів), який містить набір інформації, що описує фізичні властивості робота. До файлу URDF також може бути включена 3D-модель робота, яка дозволяє описати робота.

Цей тип тривимірної моделі робота може бути створений за допомогою програм автоматизованого проектування (CAD), таких як AutoCAD, Solidworks, TinkerCAD тощо. Для 3D-рендерингу URDF підтримує експорт .stl- файлів. Імпорт URDF-моделі в Simulink.

Симуляція була створена шляхом запуску новоствореного URDF-файлу, а точніше, за допомогою команди MATLAB "smimport" у командному вікні. Таким чином, файл геометрії повинен бути створений у тій самій папці, що й файл URDF. Після запуску симуляції файл геометрії ReactorX 150 відкривається і виконує налаштування для чіткої візуалізації, яка потім готує симуляцію [12].

Першим кроком після імпорту команди є підключення кожного з'єднання до відповідних файлів геометрії та налаштувань. Блок-схема цього файлу показана на рисунку 3.1. Блок-схема в основному базується на блоках "зв'язок", білих квадратах і блоках "обертальне з'єднання", які є частиною системи моделювання.

Крім того, п'яти основним з'єднанням присвоєно відповідні ідентифікаційні номери від 1 до 5, де номер 5 не використовується, оскільки він не є релевантним для цієї роботи, наприклад, цей сервопривід використовується для затягування гвинта.

Файл також змінює налаштування для суглобів, коли файли пов'язані, і змінює одиницю виміру, яка змінюється з метрів на міліметри, щоб отримати чітке уявлення про всю руку робота. Як приклад, візуальний блок для верхньої частини руки показано на рисунку 3.2, а також зміна одиниць виміру для обертового суглоба. Ця зміна виконується для всіх суглобів.

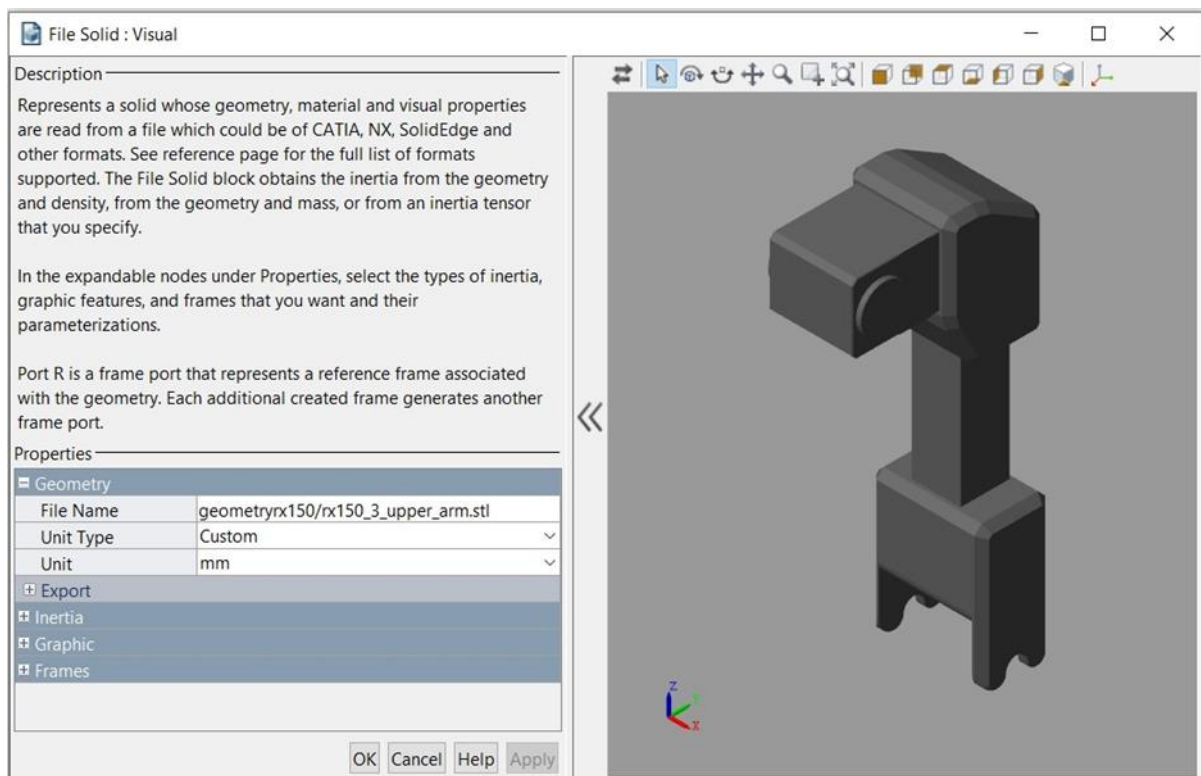


Рисунок 3.2 - Візуальний блок для плеча

Наступним кроком є зміна дії всіх п'яти рухомих суглобів, а саме: основи, талії, плечового, ліктювого та променезап'ясткового суглобів. Дії цих суглобів

встановлюються на "Автоматично обчислюється" для крутного моменту і "Забезпечується введенням" для руху.

Таке налаштування приводів призводить до зворотної кінематики, оскільки вхідним сигналом є результуючий рух, а необхідний крутний момент обчислюється. Входи для руху задаються за допомогою декількох повзунків.

Приклад візуального блоку, в якому цей параметр змінено, показано на рисунку 3.3.

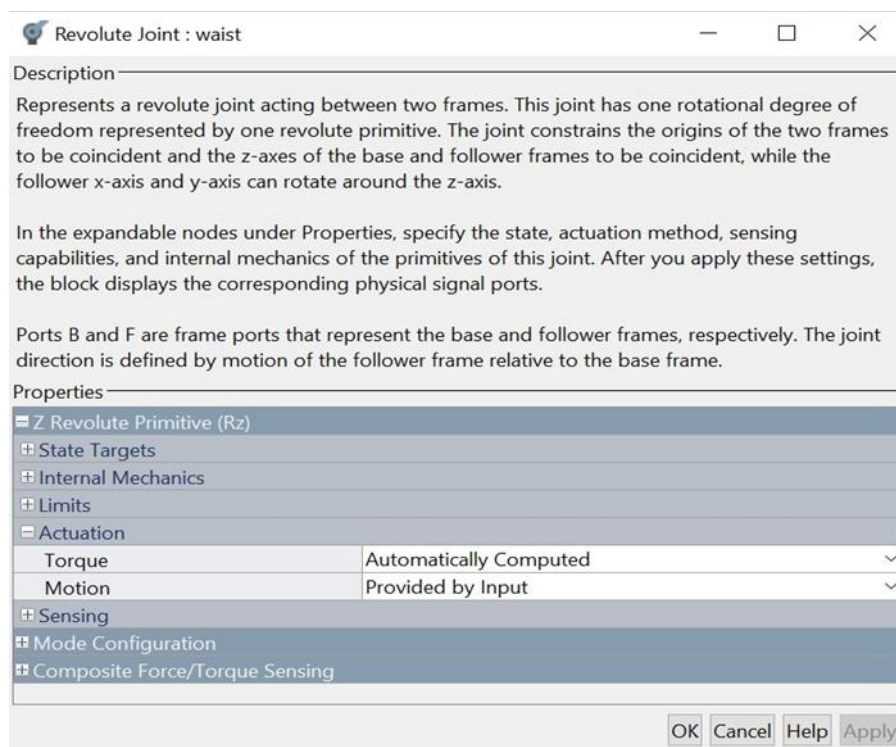


Рисунок 3.3 - Візуальний блок для вибору активації суглоба

Третім, і останнім, кроком у цьому файлі налаштувань є з'єднання вищезгаданих входів повзунка з блоком симуляції. Це основний блок для симуляції. Для підключення цих повзунків передбачено п'ять входів. Ці входи базуються на виборі входів з попереднього кроку.

Кожен вхід кута зчленування з'єднаний з повзунком на основі градусів, і сигнал в градусах перетворюється в радіани перед відправкою на симулятор або Simulink-модель. На додаток до цього входу використовується функція бібліотеки MATLAB для перетворення сигналу MATLAB Simulink у фізичний сигнал.

Цей перетворювач називається "Simulink PS", як показано на рисунку 3.1. Наразі система працює як симулятор, але сама програма створюється як фізичний рух, тому конвертер потрібен незалежно від цього. Блок-схема симулятора з входами показана на рисунку 3.4, а симуляція руки робота - на рисунку 3.5.

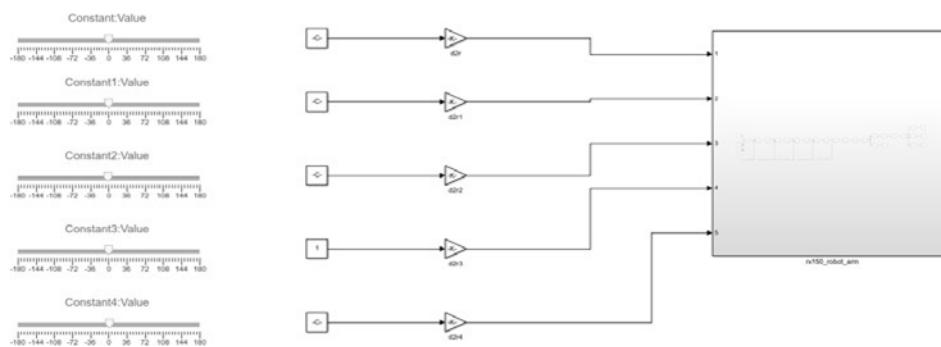


Рисунок 3.4 - Блок-схема симулятора з входами

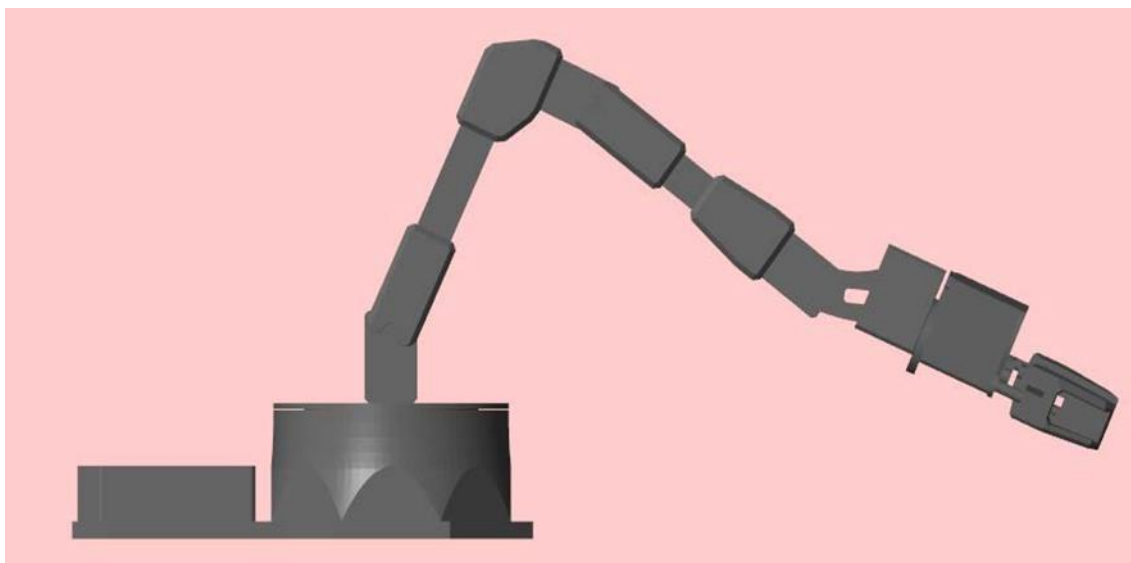


Рисунок 3.5 – Візуалізація робота-маніпулятора

Реалізація симулятора була необхідна не тільки для тестування руху та комунікації, але й для тестування інших функцій, таких як відстеження траєкторії та зворотна кінематика. Симулятор пропонує графічну систему, в якій можна тестувати комунікацію та рух руки робота.

На рисунку 3.6, показано нульове положення (ZDP) для симуляції руки. Для суглобів 2, 3, 4 і 5 ця точка дорівнює 0° , тоді як для суглоба 1, де відбувається обертання, вона дорівнює 180° .

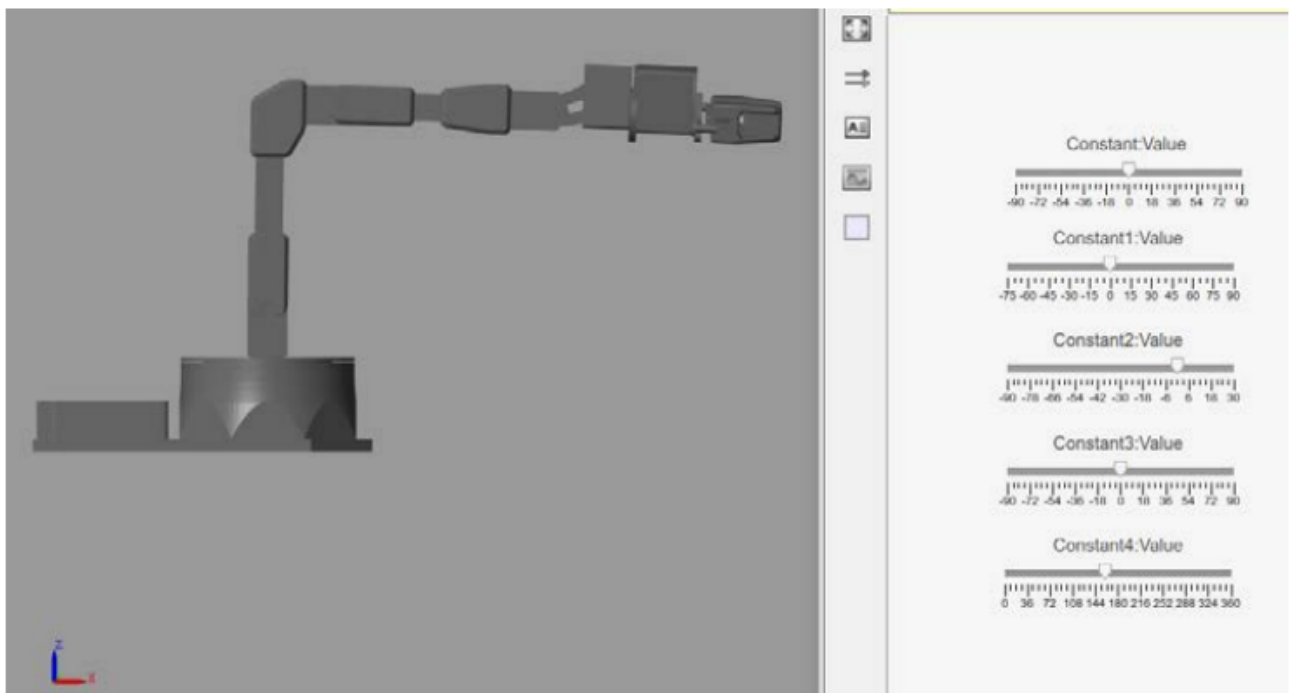


Рисунок 3.6 - Нульове положення (ZDP) для симуляції руки

Ця точка ZDP використовується як початкова точка, тобто всі рухи базуються на її різниці з ZDP, яка потім використовується для розрахунку поточного положення. Оскільки задекларована ZDP відрізняється від значень 0° деяких з'єднань, для відстеження задекларованої ZDP були використані рівняння.

Обмеження повороту на 180° базується на фізичній конструкції роботизованої руки, яка враховується в симуляції. Обмеження діапазонів руху суглобів 2 і 3 також ґрунтується на фізичних обмеженнях.

Рух симулятора може базуватися як на позиції, використовуючи зворотну кінематику, так і на основі повзунка, надсилаючи вхідні дані кожному суглобу. Як приклад руху на основі повзуна, симуляційній руці було надано кілька кутів для встановлення, і один із цих вхідних даних наведено на рисунку 3.6, що є нульовим положенням руки робота, що означає, що вхідні дані повзунка є нульовими для всіх суглобів.

На рисунку 3.6 показано як положення плеча моделювання, так і вхідні кути повзунків. Симуляція базується на великому пальці правила робототехніки, яке є від'ємним градусом кута для руху за годинниковою стрілкою та додатним градусом кута проти годинникової стрілки. Це налаштування наведено в оригінальному файлі URDF.

На рисунку 3.7, 2-й повзунок подано під кутом 90° , що призводить до руху проти годинникової стрілки на 90° у суглобі 2. У суглобах 3 і 4 забезпечується від'ємний ступінь руху, що призводить до руху за годинниковою стрілкою. Графіки для кожного зі з'єднаних значень у випадковому положенні наведено на рисунку 3.8.

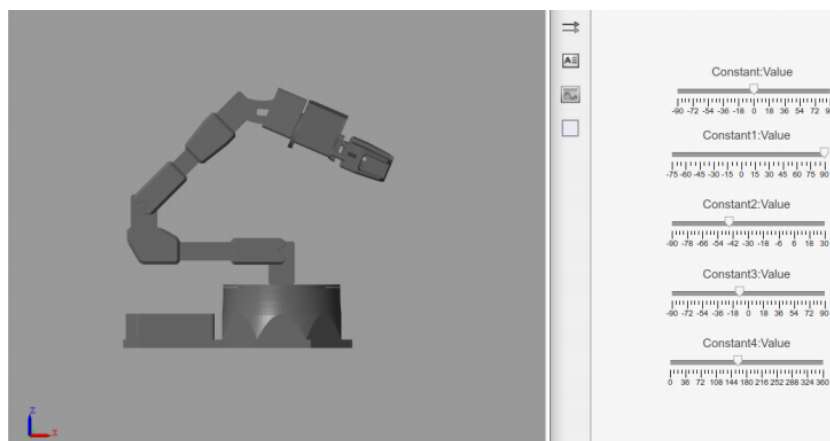


Рисунок 3.7 - Випадкова позиція робота-маніпулятора

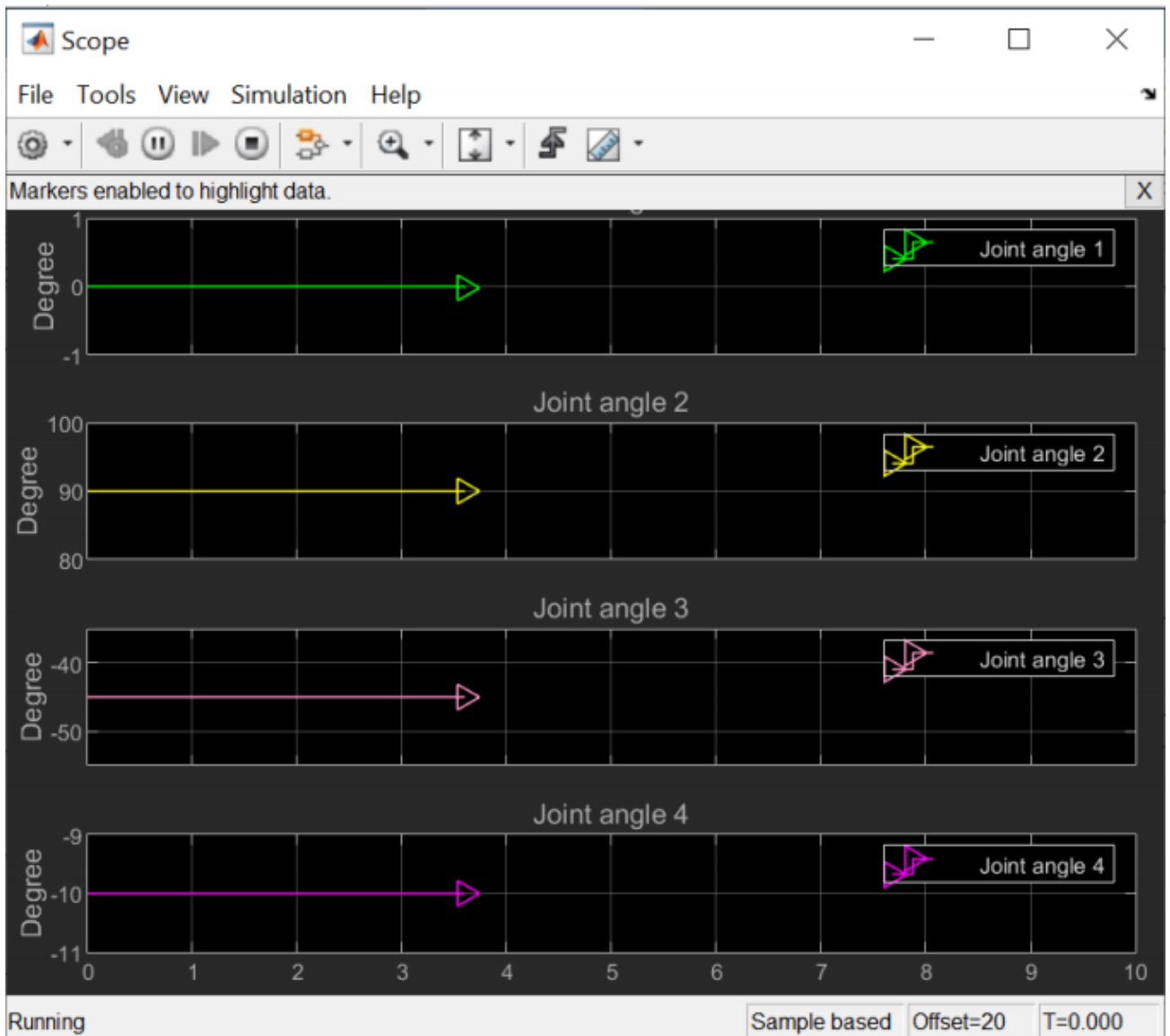


Рисунок 3.8 - Графіки для кожного зі з'єднаних значень у випадковому положенні

Приклад використання руху на основі позиції в моделюванні наведено на рисунку 3.9. За допомогою зворотної кінематики, вибране положення використовується для обчислення необхідних кутів з'єднання. У рамках цих обчислень траєкторію можна побудувати разом із засобом перегляду моделювання. Цей графік зображено на рисунку 3.9.

Рух документується червоними точками, а координати траєкторії відображаються синіми колами. Тому також можна побачити найефективнішу лінію руху, спостерігаючи за червоною лінією між двома координатами.

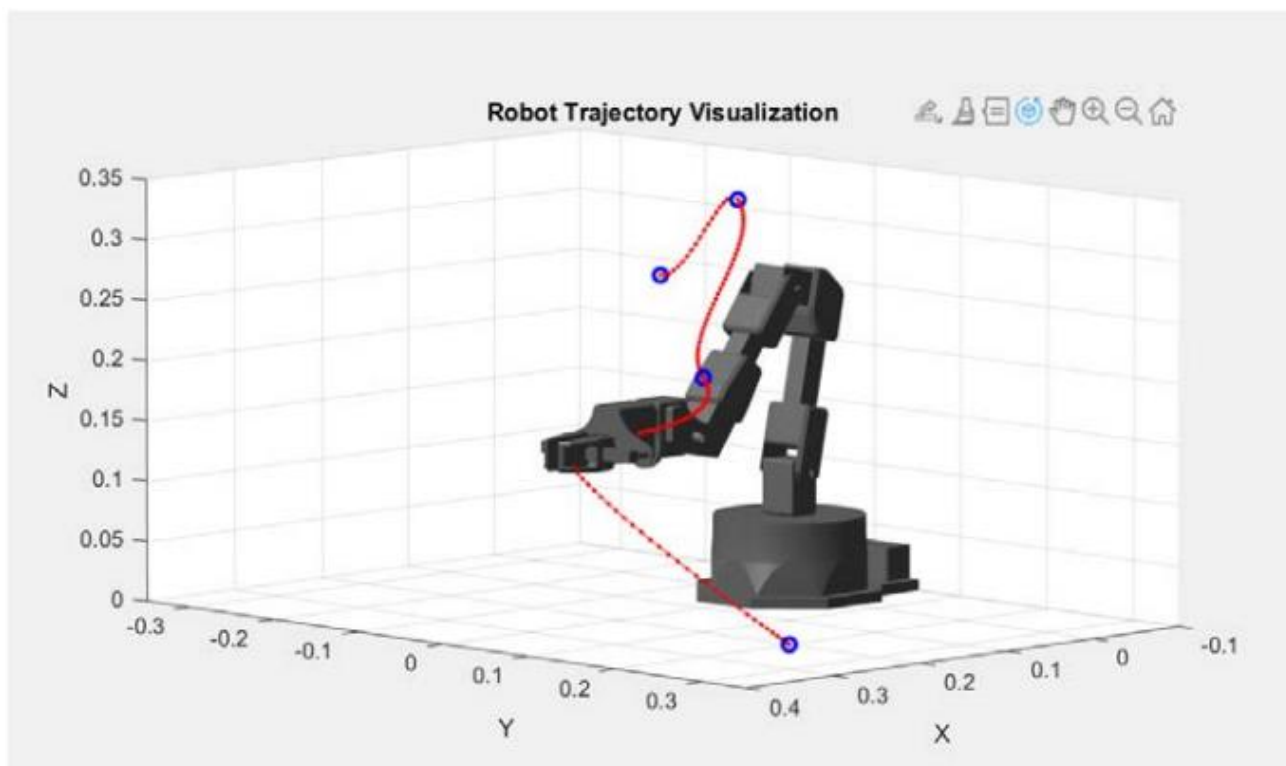


Рисунок 3.9 - Приклад використання руху на основі позиції в моделюванні

У результаті цього ми визначили траєкторію (сукупність позицій кінцевого ефектора в 3D-просторі), а також швидкість маршрутних точок. Руку робота потрібно змусити слідувати цій траєкторії за допомогою зворотної кінематики, яка у своїй найпростішій формі запитує, який кут з'єднання має бути застосований, щоб досягти положення кінцевого ефектора з огляду на положення кінцевого ефектора. Зворотню кінематичну задачу для руки робота буде сформульовано та розв'язано за допомогою Simulink, щоб визначити та розв'язати проблему.

Перші три «постійних блоку» зліва на рисунку 3.10 представляють три вхідні дані: точку шляху, швидкість і час. Точка маршруту містить координати траєкторії для руки робота, швидкість – це швидкість руху від однієї координати до наступної координати траєкторії, а час – це часовий проміжок усього руху.

Ці вхідні дані надсилаються до блоку «Поліноміальна траєкторія», який дає матричний вихід заданих вхідних даних, який потім перетворюється перед

використанням як основа для необхідних кінематичних обчислень, які в даному випадку є зворотною кінематикою.

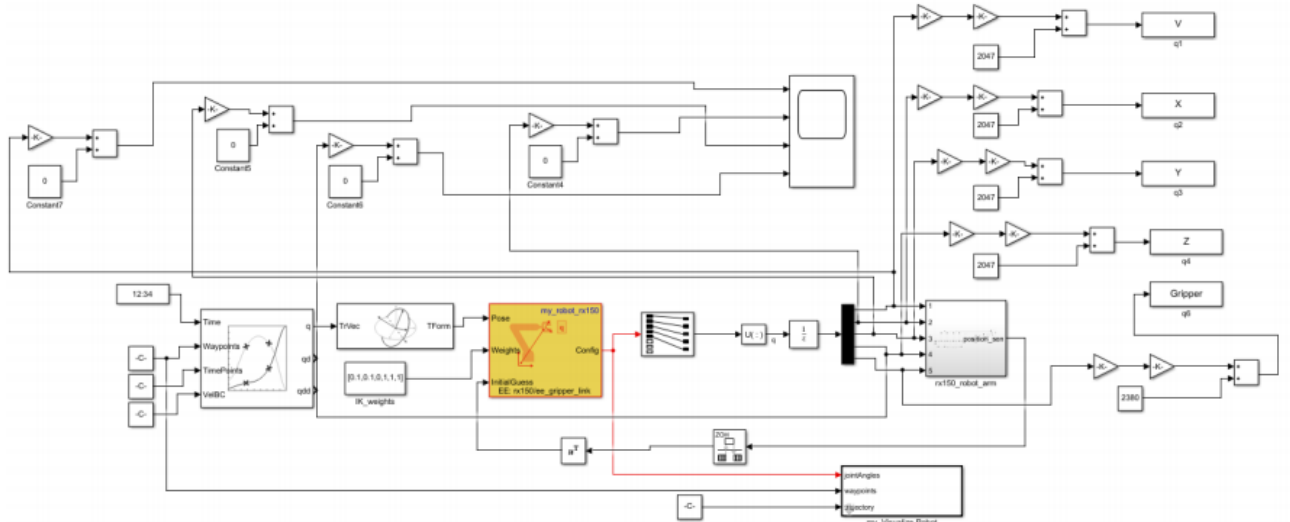


Рисунок 3.10 - Блок-схема зворотного кінематичного моделювання руки робота

Розраховані кути з'єднання потім передаються до функціонального блоку симуляції робота на основі зворотного зв'язку початкового положення, отриманого від моделі робота, а також зберігаються в робочій області MATLAB як ім'я змінної V як $q1$, X як $q2$, Y як $q3$ і Z як $q4$, як показано на рисунку 3.12, щоб пізніше використовувати для перевірки прямої та зворотної кінематичної задачі з реальним апаратним блоком.

Деякі незначні дії виконуються, щоб уникнути алгебраїчного циклу та усунути затримку, перш ніж остаточні значення надсилаються до симулятора, який потім виконує рухи. Після руху в симуляторі поточне положення повертається для обчислення помилки/відхилення, тим самим створюючи цикл зворотного зв'язку для врахування будь-яких помилок.

Останнім блоком є візуалізація в Simulink, тобто функція, яка робить рухи, а також симуляцію графічно видимими. Цей процес показано на блок-схемі, показаній на рисунку 3.10.

Зворотна кінематика також дає змогу відстежувати рух манипулятора моделювання. Файл «visualizationRobot.m» також потрібен для візуалізації руху руки робота, яка слідує за траєкторією. Відповідно до наведених шляхових точок і формулювання кубічної поліноміальної траєкторії зворотне кінематичне рішення було відображено на тривимірному графіку, як на рисунку 3.11.

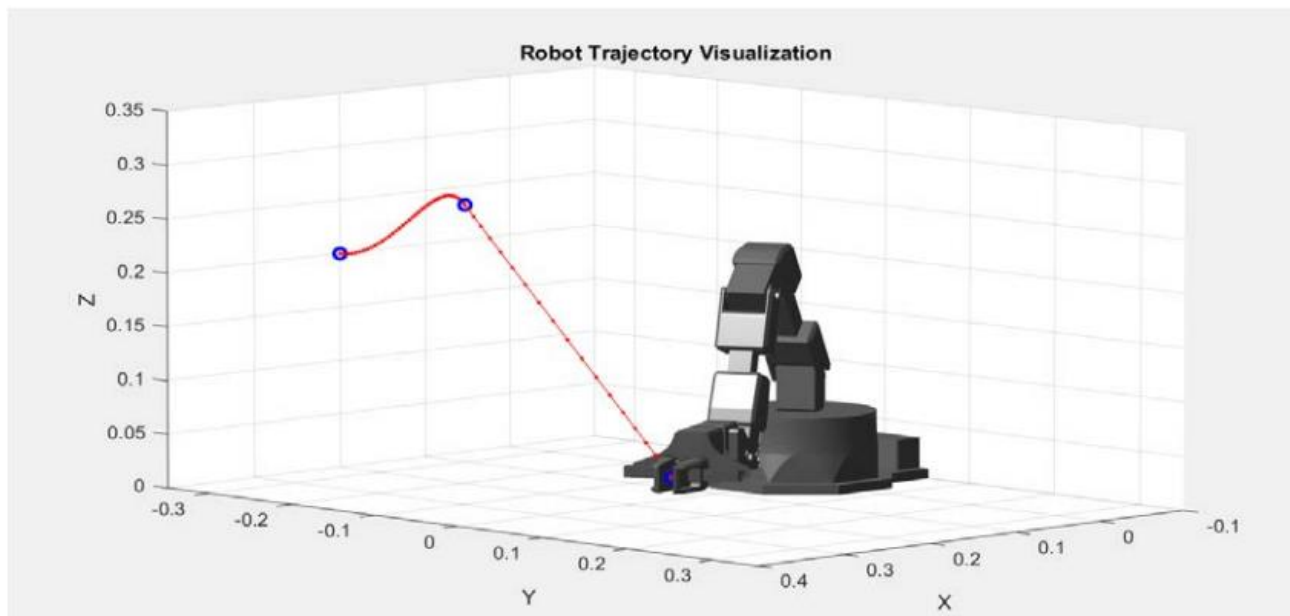


Рисунок 3.11 - Результат моделювання руки робота

Інверсний кінематичний вихід для визначеної траєкторії, що зберігається в робочому просторі MATLAB за допомогою бібліотечного блоку, показано на рисунку 3.12. Метою використання цього бібліотечного блоку є використання спільних кутів під час реального апаратного інтерфейсу.

Крім того, ці кути з'єднань можна використовувати в прямому кінематичному рівнянні як дані часового ряду для переміщення робота по бажаній траєкторії.

У різних областях останніх технологій і тенденцій традиційний підхід до вирішення кінематики є ключовим, цей розділ поширюється від комп'ютерної

графіки (наприклад, аналіз анімаційних персонажів) до подальшого розширення просторових маніпуляцій і моделювання.

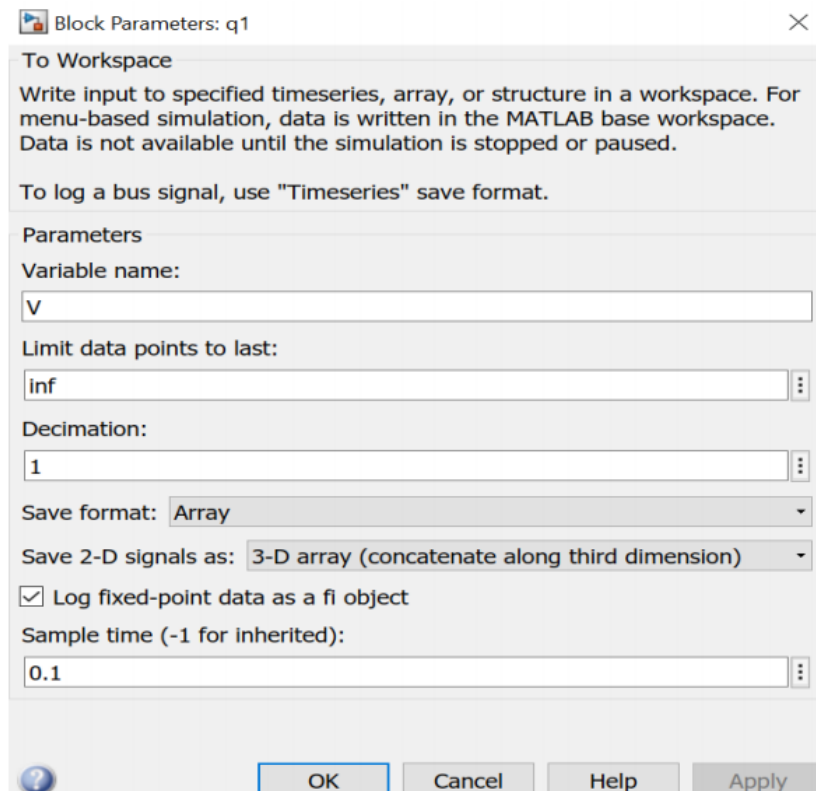


Рисунок 3.12 - Інверсний кінематичний вихід для визначеної траєкторії, що зберігається в робочому просторі MATLAB

Як наслідок, усі ці додатки фундаментально базуються на оцінці як позиції, так і орієнтації декартових координат кінцевих ефекторів роботів-маніпуляторів, а також спільних змінних робота. Метод однорідної матриці перетворення можна використовувати для оцінки положення та орієнтації кінцевих ефекторів та їхніх спільних змінних. Існує два основних типи методів для опису кінематичних зв'язків між шарнірами і для опису руху ланок.

Можна зрозуміти кінематику за допомогою ланцюга або ланок, з'єднаних з шарнірами для створення відносного руху, без вивчення крутних моментів, сил або механізмів, за допомогою яких відбувається рух.

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

Кінематичну систему робота можна розуміти як рух ланки робота відносно однієї фіксованої системи координат або базової системи координат з часом, як це сприймається [10]. Також можна вивчати вищі похідні кінематики ланки робота, такі як швидкість, ривок, прискорення тощо.

Зазвичай кінематичний ланцюг складається з пари ланок, можливо, з'єднаних обертальними або призматичними з'єднаннями з обертовими чи трансляційними ступенями свободи.

Як описано в літературі, існує багато підходів до математичного представлення кінематичних ланцюгів. Існують значні відмінності в тому, як до цих підходів приєднуються координатні системи. З цієї причини, зазвичай використовуються параметри Денавіта-Гартенберга.

Для розміщення систем координат для зв'язків і спільних змінних більш доцільними є однорідні матриці перетворення [11]. У методі є чотири параметри, які називаються параметрами ДН кінематичного ланцюга.

Геометрія ланки та відносне зміщення суглоба визначаються цими скалярами. У результаті цього методу представлення кінематичний опис спрощується, а математичні/арифметичні операції зменшуються. Використовуючи рисунок 3.13 як приклад, можна визначити положення та орієнтацію осі суглоба відносно базисних координат X , Y , Z з мінімальним відключенням наших параметрів.

У результаті величина довжини (a), яка розташована від початку рамки до точки іншого з'єднання на відстані зміщення (d). θ представляє кут, утворений OA , який паралельний осі x , і OP .

Альфа - це кут, утворений PQ, як показано на діаграмі, цей кут являє собою поворот навколо осі z, який вимірюється через вісь x. Вісь суглоба може повертатися на певний кут, якщо вісь паралельна осі z і вимірюється в напрямку z.

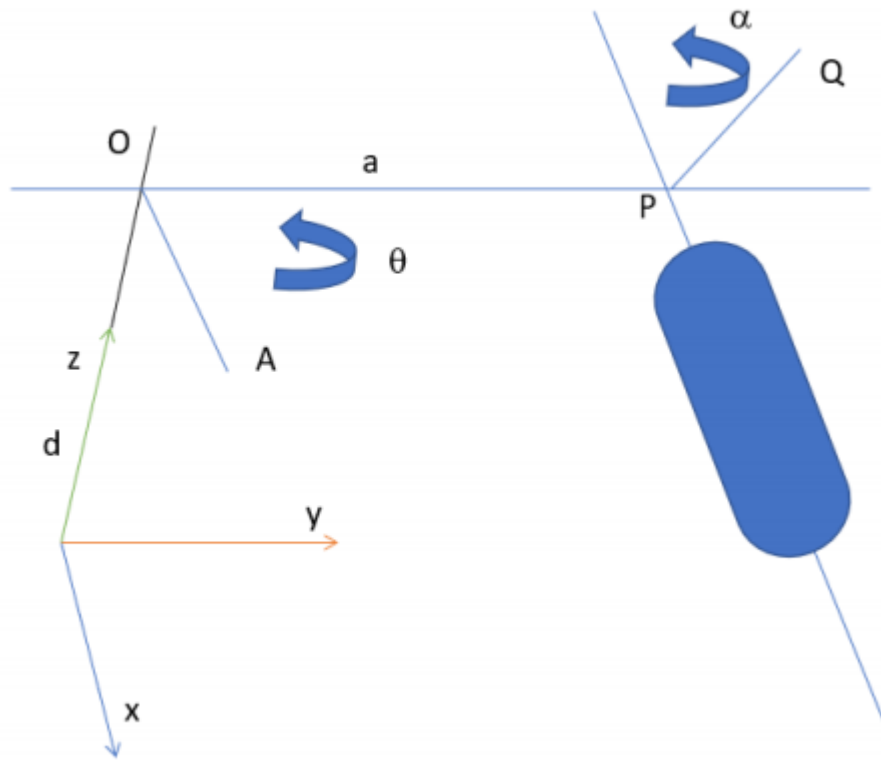


Рисунок 3.13 - Положення і напрямок циліндричного шарніра в системі координат

Чотири скаляри a , α , θ і d представляють параметри Денавіта Гартенберга, які використовуються для представлення положення осі з'єднань у декартових системах координат. У наступному розділі є більш детальне обговорення цих чотирьох параметрів.

Представлення Денавіта-Хартенберга (DH) є кращим вибором, коли ви маєте справу зі складними роботизованими системами, включаючи манипулятори робота зі зміщенням. У конфігураціях рук промислових роботів подання DH широко використовується і розглядається як універсальний метод ідентифікації маніпуляторів роботизованих рук.

Для конфігурацій руки робота зі зміщенням представлення ДН є дуже корисним. Використовуючи представлення ДН, можна розрахувати кінематику складних робототехнічних систем лише з чотирма основними параметрами.

У цьому списку є два параметри переміщення та два параметри обертання. Відповідно до домовленості про представлення ДН обертання проти годинникової стрілки (CCW) є позитивним, а обертання за годинниковою стрілкою (CW) є негативним.

На рисунку 3.14 показано блок симуляції, що розв'язує зворотну кінематику за допомогою функції зворотної кінематичної бібліотеки, щоб змусити робота слідувати траєкторії.

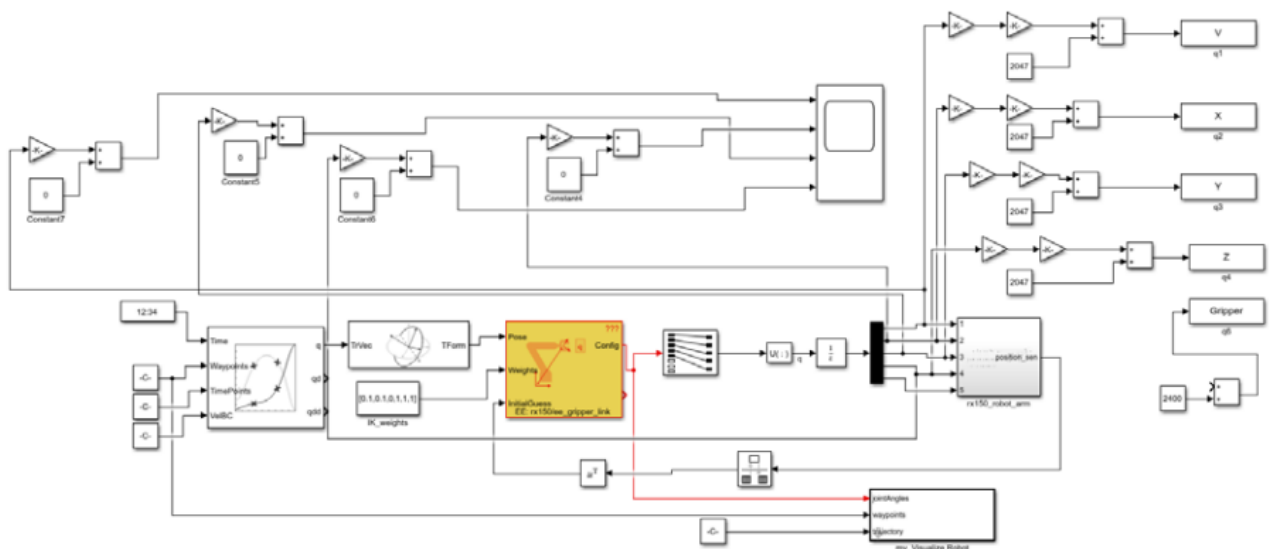


Рисунок 3.14 - Блок інверсного кінематичного моделювання

Мета цього моделювання полягає в тому, щоб реалізувати його на реальному апаратному блоці та порівняти рух руки робота з зворотним кінематичним результатом за допомогою техніки оптимізації.

На рисунку 3.15 показана рука робота, яка відстежує траєкторію в 3D площині. Зворотне кінематичне рішення, що зберігається в блоках, які

називаються «До робочої області» за назвами «V», «X», «Y», «Z» і «Gripper» як «q1», «q2», «q3» і «q4». спільні кути.

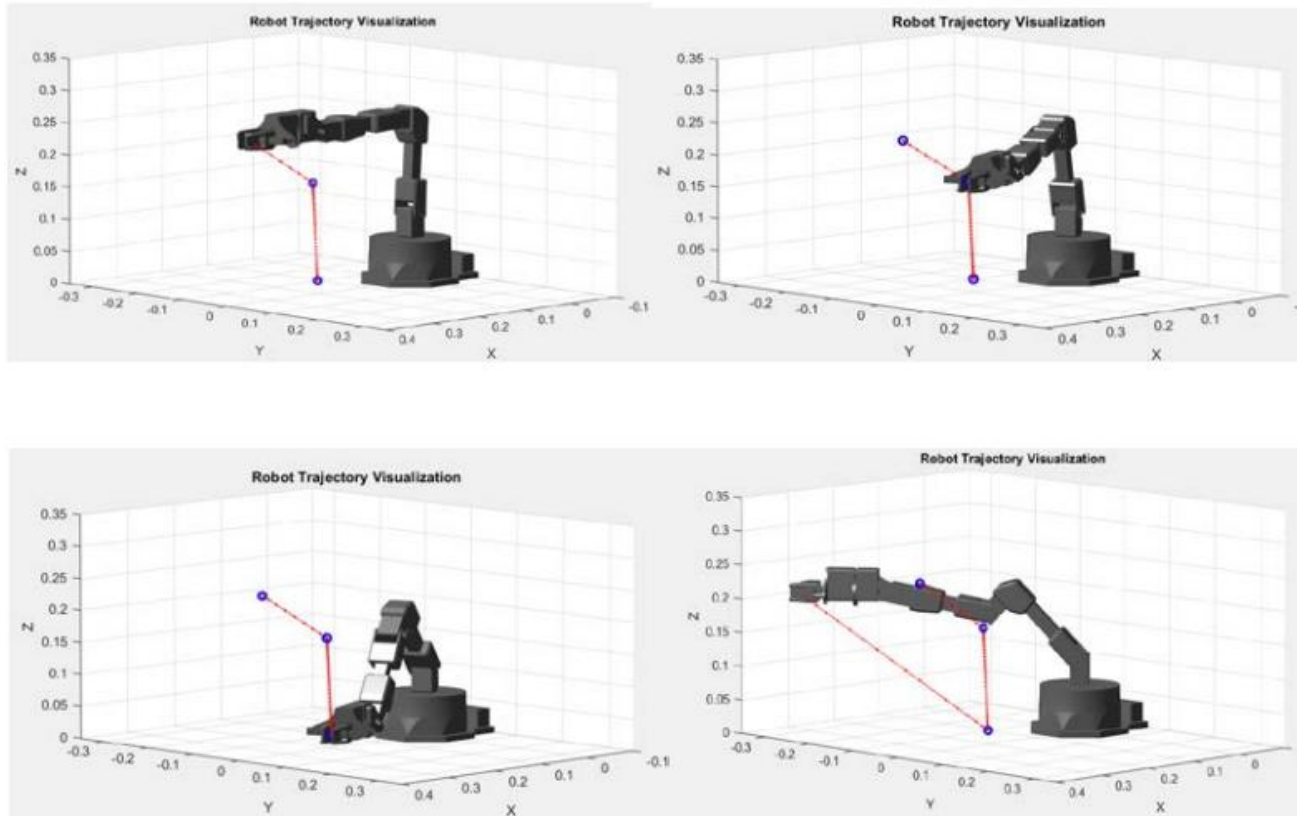


Рисунок 3.15 - Рука робота, яка відстежує траєкторію в 3D площині

Значення кутів з'єднань вибирається зі змінної робочої області для реалізації в реальному апаратному блоці. Реалізація стикових кутів у реальному апаратному забезпеченні виконана у двох варіантах. Один - із Simulink, як показано на рисунку 3.16, і шляхом написання кодів у сценарії редактора MATLAB.

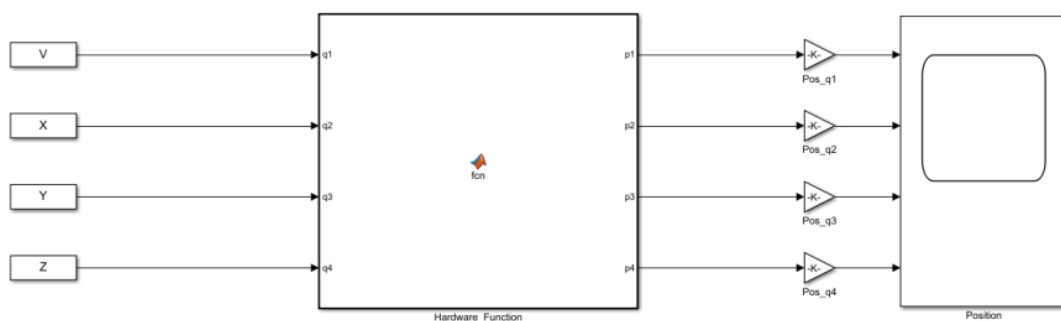


Рисунок 3.16 - Інверсний кінематичний результат, реалізований на руці робота

Під час розробки інтерфейсу було створено інше рішення, засноване на введенні з клавіатури. Це рішення ґрунтувалося на прив'язуванні клавіш до кількох рухів руки робота, як-от обертання за годинниковою стрілкою та проти годинникової стрілки та переміщення кожного кута в обидва боки, за допомогою різних клавіш на клавіатурі, що призвело до більш зосередженої взаємодії оператора, а не повзунків у код.

Це рішення було змінено, щоб спростити введення від оператора, як з точки зору кількості вводів, так і простоти повзунка з одним іменем, а не двох вибраних кнопок для кожного суглоба [12].

Проблема зі ривками під час слідування траєкторії за допомогою реального апаратного блоку, яку було подолано шляхом зміни адреси 108 «Профіль прискорення (700 мс)» та адреси 112 «Профіль швидкості (700 мс)» у майстрі Dynamixel. Це показано на рисунку 3.17 зі значенням за замовчуванням «Профіль прискорення» та «профіль швидкості» та на рисунку 3.18 зі значенням зміни, де профіль визначає швидкість на запрограмованому шляху. Значення за замовчуванням (0 мс).

DYNAMIXEL Wizard 2.0 - v2.0.12.6

Device Control Graph Packet View Tools Help

Scan Disconnect Options Graph Packet Recovery Update About News

Address	Item	Decimal	Hex	Actual
98	Bus Watchdog	0	0x00	Disable
100	Goal PWM	885	0x0375	100.00 [%]
102	Goal Current	1193	0x04A9	3209.17 [mA]
104	Goal Velocity	10	0x0000000A	2.29 [rev/min]
108	Profile Acceleration	0	0x00000000	0 [ms]
112	Profile Velocity	0	0x00000000	0 [ms]

Рисунок 3.17 - Стандартне прискорення та швидкість сервоприводу майстра Dynamixel

1. Зазначені допустимі швидкості.
2. Прискорення.
3. Зберігається встановлений ривок.

Необхідно звернути увагу на те, що кожного разу після вимкнення адреси живлення 108 і 112 міститимуть значення за замовчуванням, тому користувачеві слід записати значення знову з майстра Dynamixel або зсередини апаратного коду читання та запису з MATLAB.

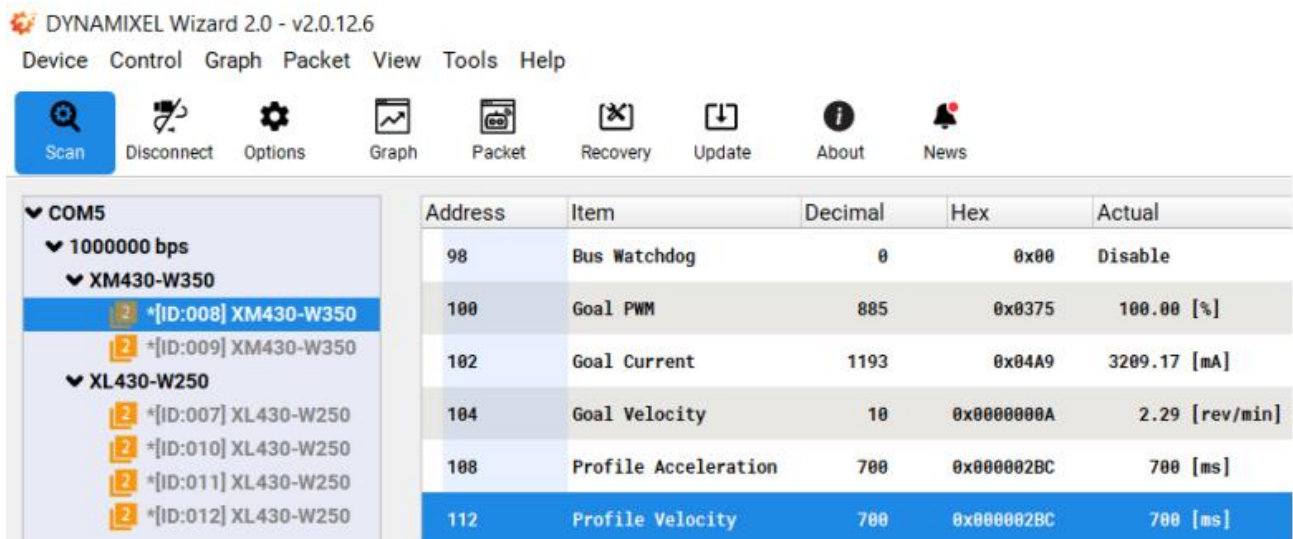


Рисунок 3.18 - Майстер Dynamixel змінив прискорення та швидкість сервоприводу

3.3 Висновки до третього розділу

1. Щоб успішно інтегрувати MATLAB з роботом-маніпулятором 150, спочатку необхідно створити функціонуючий симулятор. Цей симулятор повинен відтворювати функції фізичної руки, а також працювати безпечно, оскільки можна уникнути фізичних пошкоджень.

2. Interbotix пропонує багато симуляцій та пакетів для ROS2, і оскільки всі симуляції зберігаються у форматі хасго, основною метою є перетворення цих файлів та симуляцій у формат, придатний для MATLAB.

3. Після встановлення та запуску ROS2 на платформі Linux хасго-файли були конвертовані у формат Unified Robot Description Format (URDF). Ці файли були отримані з репозиторію Interbotix GitHub.

4. Таке перетворення підтримується в ROS2 і просто вказує на конкретний маніпулятор робота, на якому базуються файли. Існують симулятори для всіх вищезгаданих маніпуляторів і в залежності від маніпулятора ім'я файлу має бути змінено.

5. Формати файлів URDF і хасго засновані на стандартному форматі XML, що робить конвертацію можливою. Файл URDF можна імпортувати і використовувати в MATLAB з Simulink для моделювання робота.

6. Для моделювання маніпулятора робота в середовищі MATLAB необхідні моделі URDF і дерева жорсткого тіла. Деревоподібна модель - це один із способів представлення механічних властивостей маніпулятора робота в MATLAB, а також для візуалізації та керування маніпулятором робота в Simulink.

7. Мета цього моделювання полягає в тому, щоб реалізувати його на реальному апаратному блоці та порівняти рух руки робота з зворотним кінематичним результатом за допомогою техніки оптимізації.

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ УДОСКОНАЛЕНОГО МЕТОДУ КЕРУВАННЯ РОБОТОМ МАНІПУЛЯТОРОМ

4.1 Особливості моделювання робота маніпулятора із двома ступенями свободи та прямою кінематикою

На рисунку 4.1 зображено модель робота маніпулятора із прямою кінематикою. До складу моделі робота маніпулятора із прямою кінематикою входять два джерела сигналів q_1 та q_2 , підсистема робот (Robot) та підсистема пряма кінематика (Forward Kinematics).

Для побудови графіків сигналів q_1 та q_2 , сигналів q_{1m} та q_{2m} та сигналу із виходу підсистеми прямої кінематики використовується триканальний осцилограф.

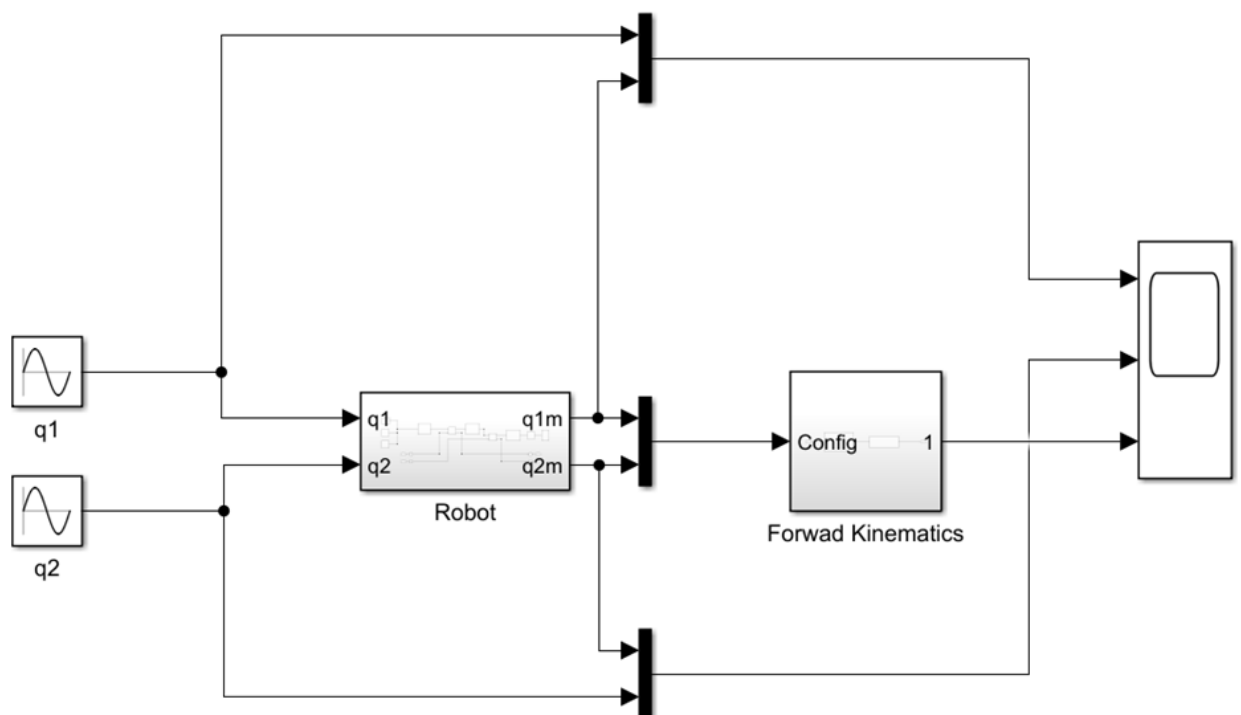


Рисунок 4.1 - Модель робота маніпулятора із прямою кінематикою

Підсистема робот (Robot) зображена на рисунку 4.2 і моделює робот маніпулятор із двома ступенями свободи. До складу підсистеми робот (Robot)

входять два поворотні з'єднання робота маніпулятора Joint_1 та Joint_2, а також одне нерухоме з'єднання, які з'єднуються за допомогою блоків Base, Link1, Link2 та EE.

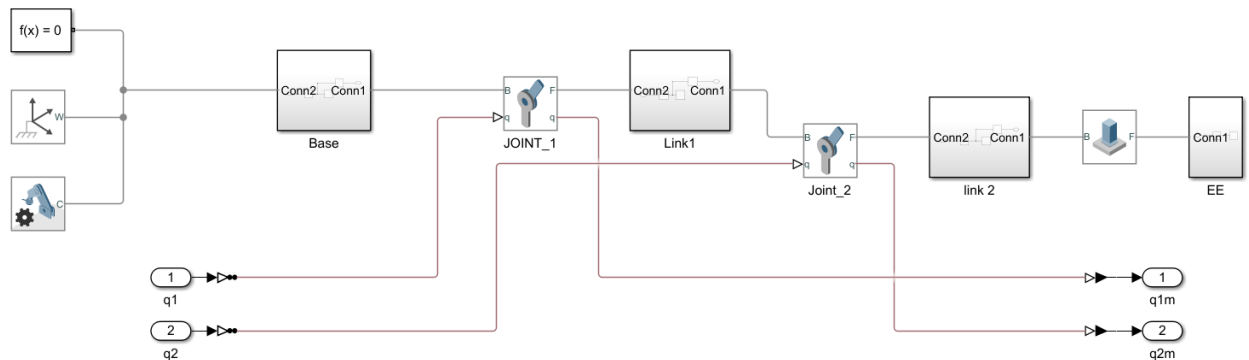


Рисунок 4.2 – Підсистема робот

Вхідні параметри підсистеми робот задаються блоком інерційна система відліку (Inertial reference fram), який зображено на рисунку 4.3.

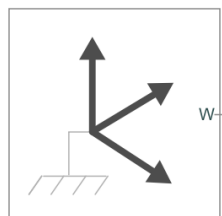


Рисунок 4.3 – Блок інерційна система відліку

Цей блок визначає глобальну систему відліку в моделі. Ця система відліку є інерційною і знаходиться в стані абсолютного спокою. Жорстке з'єднання блоку із нерухомою основою робить цю систему відліку інерційною. Осі системи відліку ортогональні і розташовані за правилом правої руки. Інші вхідні параметри підсистеми робот задаються блоком конфігурація механізму (Mechanism Configuration), який задає параметри гравітації та моделювання для всього механізму і зображений на рисунку 4.4.



Рисунок 4.4 – Блок конфігурація механізму

Блок конфігурації механізму визначає параметри сили тяжіння та моделювання механізму, до якого під'єднується блок. Параметри моделювання містять значення збурення для обчислення чисельних часткових похідних під час лінеаризації та номер ітерації для спільного переходу режиму.

Мережа Simscape™ Multibody™ може мати не більше одного екземпляра блоку конфігурації механізму. Якщо модель не має блоку конфігурації механізму, рівномірна сила тяжіння, яка застосовується до механізму, дорівнює нулю.

Наступним блоком є підсистема база (Base), яка зображена на рисунку 4.5.

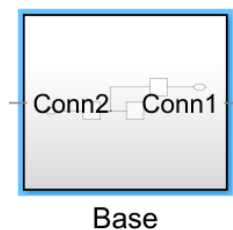


Рисунок 4.5 – Підсистема база (Base)

Підсистема база (Base) складається з наступних блоків, які зображені на рисунку 4.6.

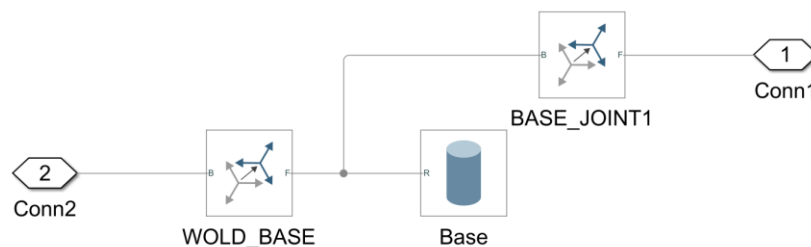


Рисунок 4.5 – Підсистема база (Base)

Першим блоком є блок WOLD_BASE (Rigid Transform), який зображено на рисунку 4.6.

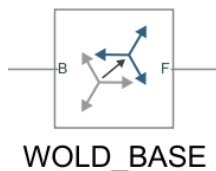


Рисунок 4.6 - Блок WOLD_BASE (Rigid Transform)

Блок Rigid Transform визначає та підтримує фіксоване просторове співвідношення між двома кадрами під час симуляції. Просторове відношення може включати переміщення та обертання. Блок має різні методи для вказівки положення та орієнтації слідкуючої системи координат відносно базової системи координат.

Наступним блоком є блок Base (Циліндричне тіло), який моделює суцільний циліндричний елемент з геометрією, інерцією та кольором, і зображений на рисунку 4.7.



Рисунок 4.7 - Блок Base (Циліндричне тіло)

Циліндричний суцільний блок - це циліндрична форма з центром геометрії, що збігається з координатами системи відліку, а вісь симетрії збігається з віссю z системи відліку.

Циліндричний суцільний блок додає до прикріпленої системи координат твердий елемент із геометрією, інерцією та кольором. Твердий елемент може бути простим твердим тілом або частиною складеного твердого тіла - групи жорстко зв'язаних твердих тіл, часто розділених у просторі жорсткими перетвореннями.

Параметри геометрії включають форму та розмір. Можна вибрати зі списку попередньо встановлених фігур або імпортувати спеціальну фігуру із зовнішнього файлу у форматі STL або STEP. За замовчуванням для всіх форм, окрім STL, блок автоматично обчислює масові властивості твердого тіла на основі вказаної геометрії та маси або масової щільності. Ви можете змінити це налаштування в параметрі Inertia > Type block.

Система відліку кодує положення та орієнтацію твердого тіла. У конфігурації за замовчуванням блок забезпечує лише опорний кадр. Інтерфейс створення фреймів надає засоби для визначення додаткових фреймів на основі твердотільних геометричних елементів.

Також до складу підсистеми база (Base) входить блок BASE_JOINT1, який аналогічний блоку WOLD_BASE (Rigid Transform) і зображений на рисунку 4.8

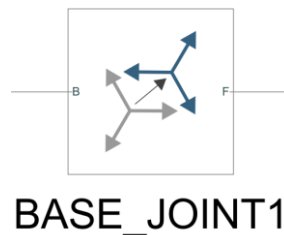


Рисунок 4.8 - Блок BASE_JOINT1 (Rigid Transform)

Поворотне з'єднання робота маніпулятора зображене на рисунку 4.9 і має вигляд шарніра, який розміщений між двома плечами.



Рисунок 4.9 – Поворотне з'єднання робота маніпулятора

Цей шарнір має один обертовий ступінь свободи, представлений одним поворотним елементом. З'єднання обмежує збіг початків двох рамок і збігання осей z базової та слідкуючої рам, у той час як осі x і y слідкуючі можуть обертатися навколо осі z .

Порти B і F є портами кадрів, які представляють базовий і наступний кадри відповідно. Напрямок з'єднання визначається рухом ведучої рами відносно базової рами.

Наступною є підсистема $Link1$, яка зображена на рисунку 4.10.

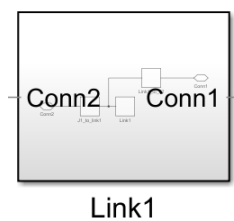


Рисунок 4.10 - Підсистема $Link1$

Підсистема $Link1$ складається з наступних блоків, які зображені на рисунку 4.11.

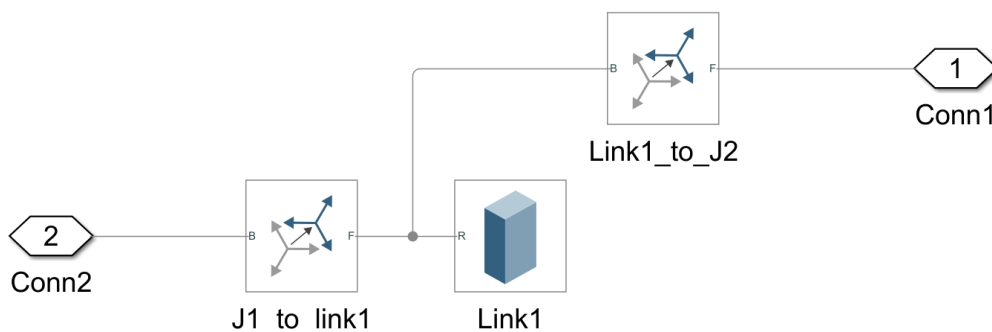


Рисунок 4.11 – Структурна схема підсистеми $Link1$

По аналогії із підсистемою база (Base) підсистеми $Link1$ також складається з двох блоків $J1_to_link1$ та $Link1_to_J2$ (Rigid Transform), які визначають та підтримують фіксоване просторове співвідношення між двома кадрами під час симуляції.

Підсистема Link1 відрізняється від підсистеми база (Base) блоком Link1, який моделює прямокутний паралелепіпед і зображений на рисунку 4.12.



Рисунок 4.12 – Блок прямокутний паралелепіпед Link1

Блок прямокутний паралелепіпед - це призматична форма з центром геометрії, що збігається з координатами системи відліку, і призматичними поверхнями, нормальними до осей x , y і z системи відліку.

Блок прямокутний паралелепіпед додає до прикріпленої рами твердий елемент із геометрією, інерцією та кольором. Цегляний твердий елемент може бути простим твердим тілом або частиною складеного твердого тіла - групи жорстко з'єднаних твердих тіл, часто розділених у просторі через жорсткі перетворення.

Нерухоме з'єднання зображено на рисунку 4.13 і моделює зварний шов між двома рамами.

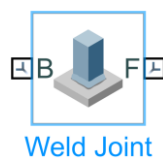


Рисунок 4.13 – Нерухоме з'єднання робота маніпулятора

Цей суглоб має нульовий ступінь свободи. Ведуча та базова рами завжди збігаються. Порти B і F є портами кадрів, які представляють базовий і наступний кадри відповідно.

Підсистема прямої кінематики зображена на рисунку 4.14 і складається із двох: блоку прямої кінематики (Forward Dynamics) та блоку перетворення координат (Coordinate Transformation Conversion).

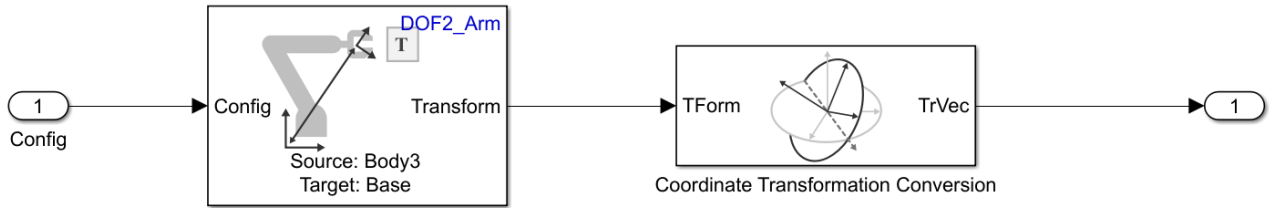


Рисунок 4.14 – Підсистема пряма кінематика

Блок прямої кінематики (Forward Dynamics) зображений на рисунку 4.15 і визначає спільні прискорення з урахуванням крутних моментів і станів суглобів.

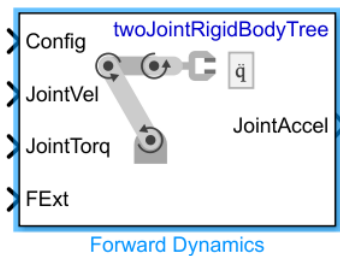


Рисунок 4.15 – Блок прямої кінематики (Forward Dynamics)

Блок прямої кінематики (Forward Dynamics) обчислює прискорення суглобів для моделі робота з урахуванням стану робота, який складається з крутних моментів суглобів, станів суглобів і зовнішніх сил. Щоб отримати прискорення суглобів, необхідно вказати конфігурацію робота (положення суглобів), швидкість суглобів, прикладені крутні моменти та зовнішні сили.

Блок перетворення координат (Coordinate Transformation Conversion) зображений на рисунку 4.16 і здійснює задане перетворення координат.

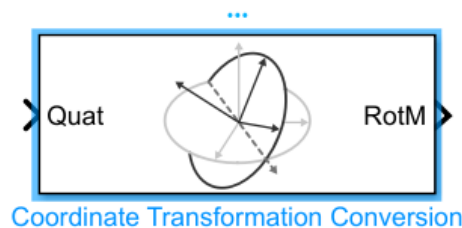


Рисунок 4.16 – Блок перетворення координат (Coordinate Transformation Conversion)

Блок Coordinate Transformation Conversion здійснює перетворення координат із вхідного подання на вказане вихідне подання. Вхідні та вихідні представлення використовують такі форми:

Кут осі (AxAng) – $[x \ y \ z \ \theta]$.

Кути Ейлера (Eul) – $[z \ y \ x]$, $[z \ y \ z]$ або $[x \ y \ z]$.

Однорідне перетворення (TForm) – матриця 4 на 4.

Кватерніон (Quat) – $[w \ x \ y \ z]$.

Матриця обертання (RotM) – матриця 3 на 3.

Вектор перекладу (TrVec) – $[x \ y \ z]$.

Усі вектори мають бути векторами-стовпцями.

Для представлень, які містять лише інформацію про положення або орієнтацію (наприклад, TrVec або Eul), можна вказати два входи або виходи для обробки всієї інформації про перетворення.

Коли вибирається однорідне перетворення як вхід або вихід, у масці блоку вибирається необов'язковий параметр «Показати порт вводу/виводу TrVec», щоб перемикає кілька портів.

В результаті проведення математичного моделювання отримали 3-D візуалізацію руху робота маніпулятора з двома ступенями свободи в середовищі Simulink, яка зображена на рисунку 4.17.

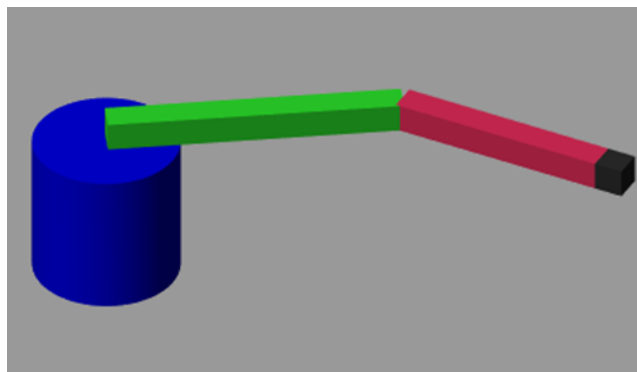


Рисунок 4.17 – 3-D візуалізація руху робота маніпулятора з двома ступенями свободи

За допомогою осцилографа можна побудувати графіки фізичних сигналів, які приймають профілі руху суглоба. Сигнал забезпечує обертання ведучої рами відносно базової рами навколо сполучної осі. На рисунку 4.18 верхній графік відображає сигнал збудження першого суглоба, середній графік відображає сигнал збудження другого суглоба, а нижні графіки відображають вектор переміщення.

Вектор переміщення заданий як 3-елементний вектор-стовпець $[x \ y \ z]$, який відповідає переміщенню по осях x , y та z відповідно. Вектор переміщення використовується для виведення інформації про координати x , y та z кінцевого суглоба робота маніпулятора.

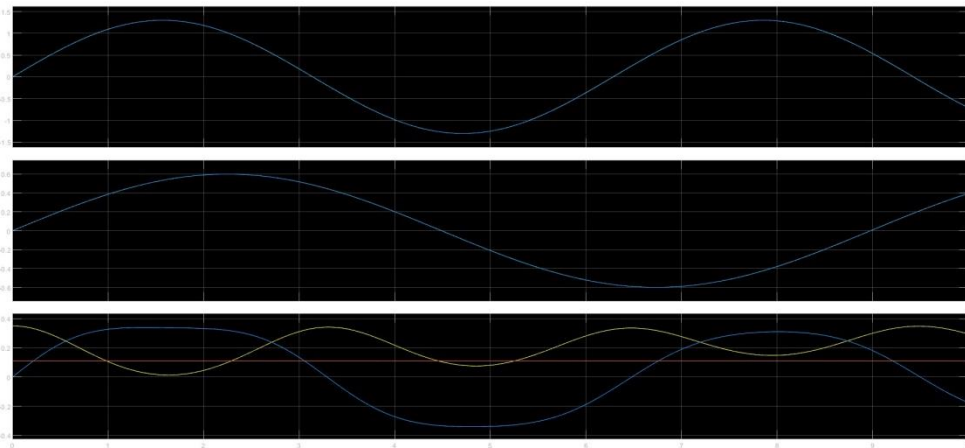


Рисунок 4.18 - Результати імітаційного моделювання робота маніпулятора з двома ступенями свободи

Жовтим кольором зображено графік зміни координати x , синім кольором зображено графік зміни координати y , а оранжевим кольором зображено графік зміни координати z .

Аналізуючи нижні графіки, які відображають вектор переміщення кінцевого суглоба робота маніпулятора можна зробити висновок, що координата z не змінюється у часі. Це пояснюється тим, що використовується робот маніпулятор з двома ступенями свободи, який може рухатися лише по двох координатах x та y в одній площині.

По координаті z робот маніпулятор не рухається. Для того, щоб робот маніпулятор рухався по осі z він повинен мати три суглоби і відповідно три ступеня свободи.

4.2 Імітаційна модель робота маніпулятора з трьома ступенями свободи

На рисунку 4.19 зображена імітаційна модель робота маніпулятора з трьома ступенями свободи.

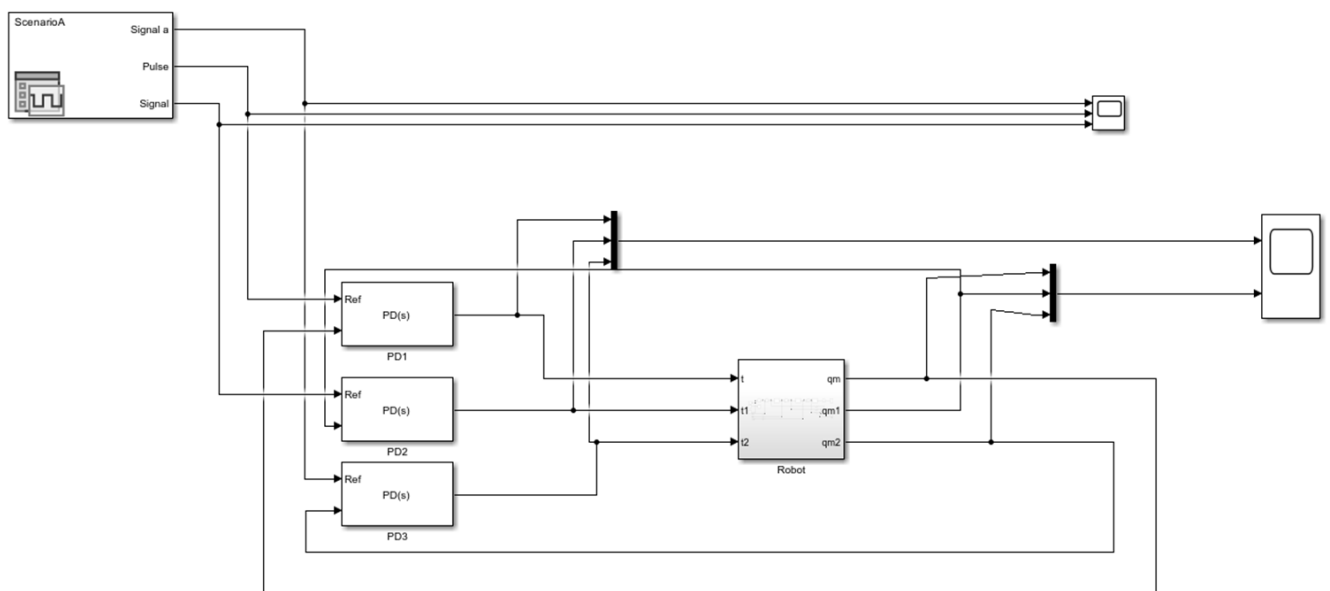


Рисунок 4.19 - Імітаційна модель робота маніпулятора з трьома ступенями свободи

До складу імітаційної моделі робота маніпулятора з трьома ступенями свободи входить блок редактор сигналів, який зображений на рисунку 4.20.

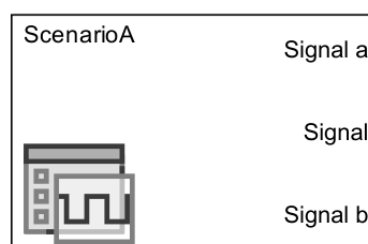


Рисунок 4.20 - Блок редактор сигналів

За допомогою блока редактор сигналів здійснюється відображення, створення, редагування та перемикання взаємозамінних сценаріїв. Редактор сигналів використовується для створення та редагування вхідних сигналів, які можна організувати для кількох симуляцій.

Дані сигналів збережено у MAT-файлі для моделювання. Редактор сигналів створює дані сигналу у форматі часових рядів. Графіки вхідних сигналів для кожної осі робота відображаються осцилографом і зображені на рисунку 4.21.

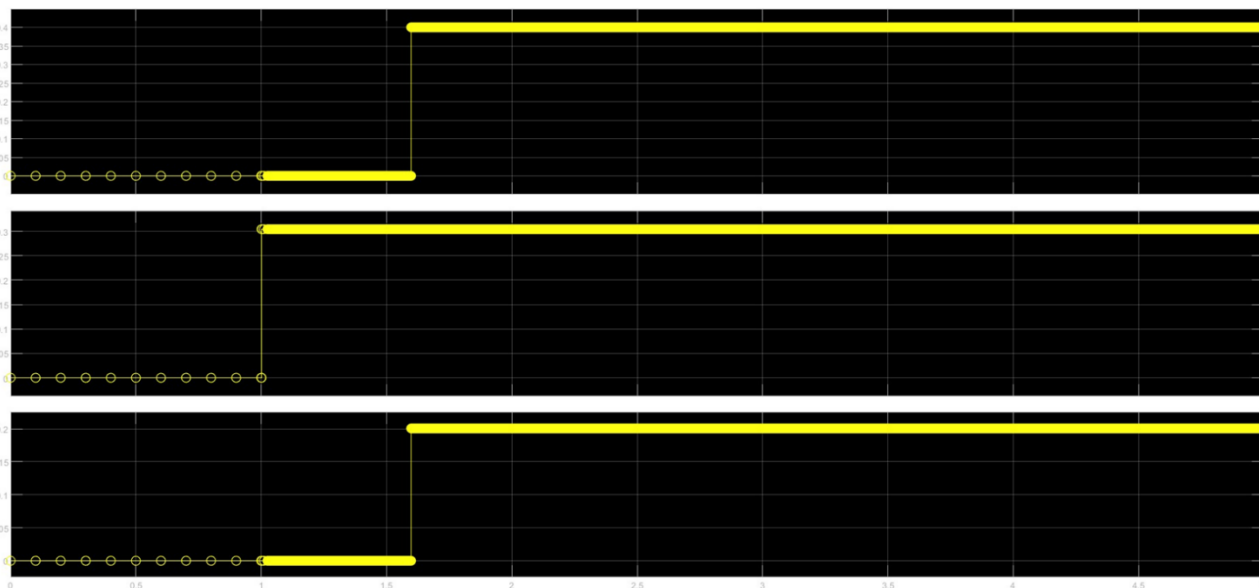


Рисунок 4.21 - Графіки вхідних сигналів для кожної осі робота

Далі вхідні сигнали для кожної осі робота поступають на три ПД-регулятора для кожної осі робота маніпулятора, які зображені на рисунку 4.22.

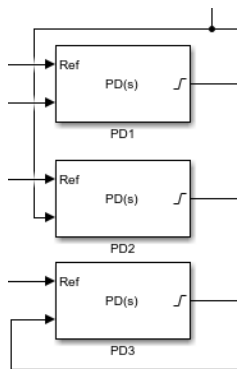


Рисунок 4.22 - Три ПД-регулятора для кожної осі робота маніпулятора

Імітаційна модель підсистеми ПД-регулятор для осі робота маніпулятора зображена на рисунку 4.23.

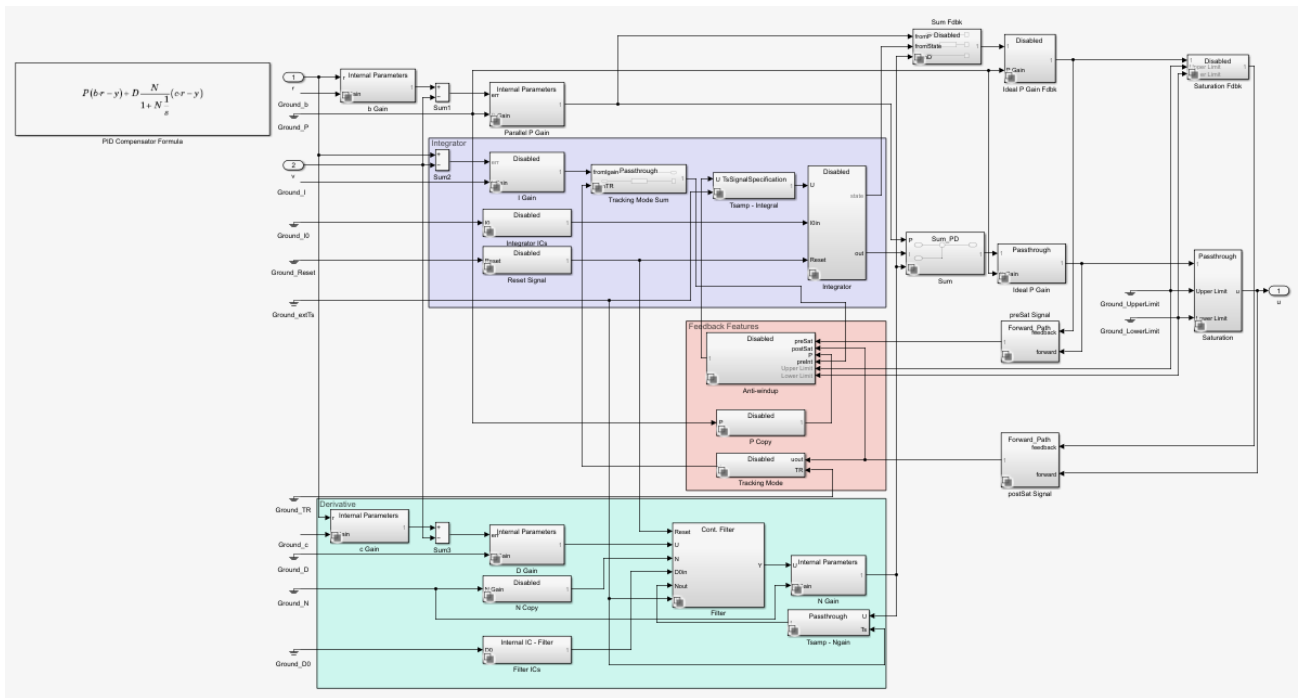


Рисунок 4.23 - Імітаційна модель підсистеми ПД-регулятор для осі робота маніпулятора

Вихідний сигнал блоку ПД-регулятор є зваженою сумою вхідного сигналу, який помножується на коефіцієнт пропорційності, а також похідної вхідного сигналу. Ваговими коефіцієнтами є пропорційний та похідний параметри підсилення. Полос першого порядку фільтрує похідну дію.

Сигнали з трьох ПД-регуляторів для кожної осі робота маніпулятора поступають на підсистему Robot, яка зображена на рисунку 4.24.

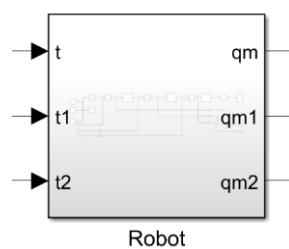


Рисунок 4.24 - Підсистему Robot

Підсистема робот (Robot) зображена на рисунку 4.25 і моделює робот маніпулятор із трьома ступенями свободи. До складу підсистеми робот (Robot) входять три поворотні блоки з'єднання робота маніпулятора, блоки з'єднання між ними, а також два нерухомих з'єднання.

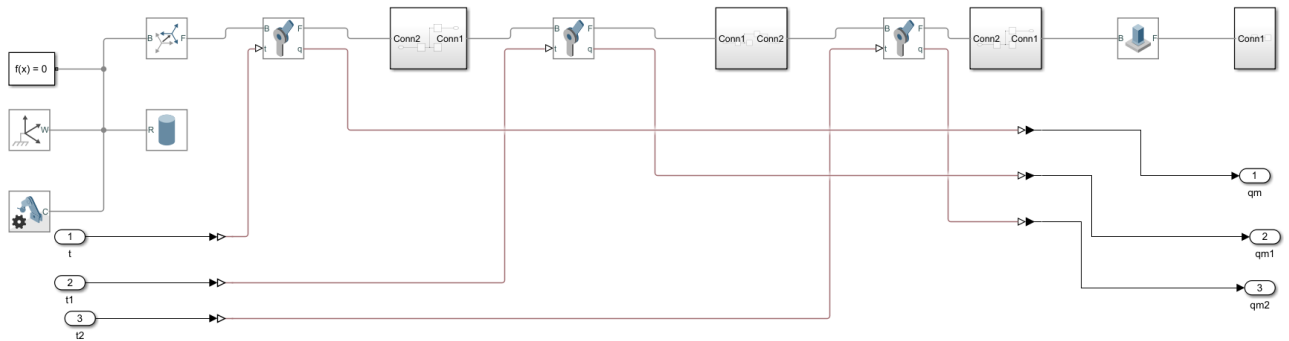


Рисунок 4.25 - Імітаційна модель підсистеми робот маніпулятор з трьома ступенями свободи

Три поворотні блоки з'єднання робота маніпулятора, блоки з'єднання між ними, а також два нерухомих з'єднання є аналогічними блокам, які використовувалися для імітаційного моделювання робота маніпулятора з двома ступенями свободи і були описані в підрозділі 4.1.

Вихідні сигнали підсистеми робот (Robot) поступають на осцилограф для відображення результатів імітаційного моделювання удосконаленого методу керування роботом і зображені на рисунку 4.26.

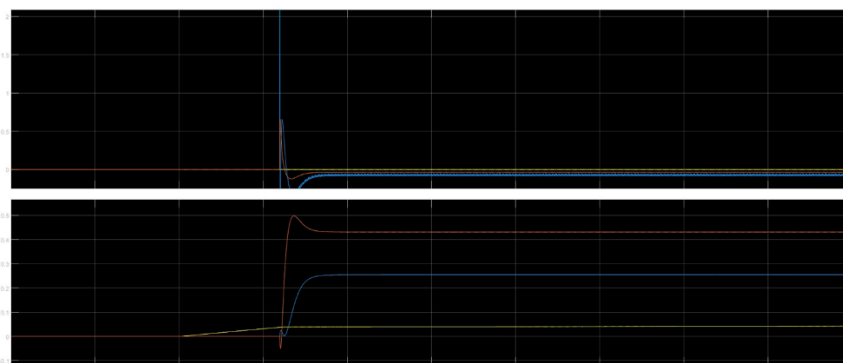


Рисунок 4.26 - Результати імітаційного моделювання удосконаленого методу керування роботом

Аналізуючи нижні графіки, які відображають вектор переміщення кінцевого суглоба робота маніпулятора можна зробити висновок, що координата z змінюється у часі. Це пояснюється тим, що використовується робот маніпулятор з трьома ступенями свободи, який може рухатися по трьох координатах x , y та z .

В результаті проведення математичного моделювання отримали 3-D візуалізацію руху робота маніпулятора з трьома ступенями свободи в середовищі Simulink, яка зображена на рисунку 4.26.

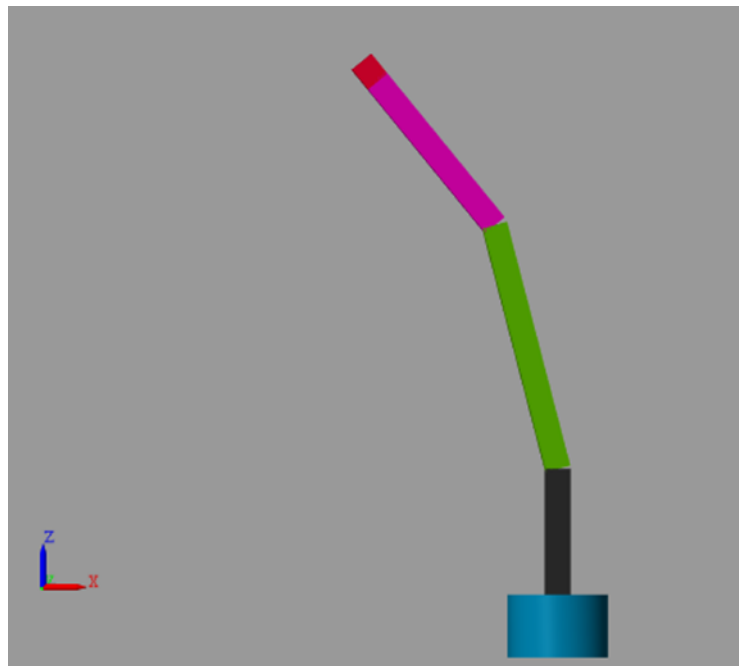


Рисунок 4.26 - 3-D візуалізація руху робота маніпулятора з трьома ступенями свободи

3-D візуалізація руху робота маніпулятора з трьома ступенями свободи дозволяє проводити дослідження руху суглобів робота маніпулятора, визначати їх координати, а також налаштовувати ПД-регулятори для кожного суглоба.

Це дозволяє проводити динамічне моделювання руху робота маніпулятора для досягнення максимально можливої швидкості руху суглобів і максимально можливої точності позиціонування робочого органу робота маніпулятора.

4.3 Висновки до четвертого розділу

1. В процесі експериментального дослідження удосконаленого методу керування робота маніпулятора розроблено і досліджено модель робота маніпулятора із прямою кінематикою.

2. Розроблено і досліджено підсистему робот з двома ступенями свободи на основі стандартних блоків Simulink.

3. Досліджено підсистема прямої кінематика для робота маніпулятора з двома ступенями свободи на основі стандартних блоків Simulink.

4. Отримані результати імітаційного моделювання робота маніпулятора з двома ступенями свободи.

5. Розроблено і досліджено імітаційну модель робота маніпулятора з трьома ступенями свободи у середовищі Simulink.

6. Розроблено і досліджено імітаційну модель підсистеми робота з трьома ступенями свободи та отримано результати імітаційного моделювання

ВИСНОВКИ

1. Конструкція механізму або робота-маніпулятора може мати кілька конфігурацій, структур або функцій, але найбільш важливими для проектування характеристиками механізму або робота-маніпулятора є кінематичний аналіз, аналіз робочого простору, генерація траєкторії і управління.

2. Значення явних властивостей завжди надається застосуванню робота-маніпулятора щодо застосувань робота-маніпулятора. Коли кількість спільних змінних маніпулятора збільшується, то збільшується також робочий простір і рухомість робота-маніпулятора, що призводить до більш складних математичних формулювань для зворотних кінематичних дозволів і труднощів в управлінні маніпулятором.

3. Під час розробки інтерфейсу було створено інше рішення, засноване на введенні з клавіатури.

Це рішення ґрунтувалося на прив'язуванні клавіш до кількох рухів руки робота, як-от обертання за годинниковою стрілкою та проти годинникової стрілки та переміщення кожного кута в обидва боки, за допомогою різних клавіш на клавіатурі, що призвело до більш зосередженої взаємодії оператора, а не повзунків у код. Це рішення було змінено, щоб спростити введення від оператора, як з точки зору кількості вводів, так і простоти повзунка з одним іменем, показано в розділі 4, а не двох вибраних кнопок для кожного суглоба.

4. Проблема зі ривками під час слідування траєкторії за допомогою реального апаратного блоку, яку було подолано шляхом додавання трьох ПД-регуляторів для трьох осей робота маніпулятора.

5. За допомогою звичайних і реактивних підходів, вирішення прямих і зворотних кінематичних проблем вибраних маніпуляторів і їх моделювання. Зворотне кінематичне рішення для вибраного еталонного маніпулятора було визначено за допомогою аналізу. Проте були прийняті алгоритми оптимізації та проведено порівняння. Для вирішення зворотної та прямої задач кінематики для обраних маніпуляторів використовуються програми MATLAB.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The lifecycle of our electronic gadgets and the True Cost to Earth. Author: Philippe Sibaud. Retrieved 2013,
2. ROS Programming: Building Powerful Robots: Design, build and simulate complex robots. Authors: Anil Mahtani, Luis Sanchez, Enrique Fernandez, Aaron Martinez, Lentin Joseph. Retrieved 2018,
3. A Systematic Approach to Learning Robot Programming with ROS. Author: Wyatt S. Newman. Retrieved 2018
4. SIMATIC Ident for MV420/MV440 Code Reader Systems. Operating systems. Retrieved 2012.
5. KUKA youBot User Manual. Retrieved 2012, from KUKA.com.
6. A novel kinematics for a 5 DOF manipulator based on KUKA youBot. Authors: Allan D. Zhang, Xiao Xiao, Yangmin Li. Retrieved 2015,
7. Robotics toolbox for Matlab. Author: Peter Corke. Retrieved 2015,
8. Parámetros Denavit-Hartenberg. Cinemática Directa de un Robot Manipulador. Authors: Alberto Herreros, Juan Carlos Fraile Marinero. Retrieved 2017.
9. Digital Image Processing in Matlab. Authors: Rafael C. González, Richard E. Woods, Steve L. Eddins. Retrieved 2011.
10. E. Kolberg, and N Orlev, "Robotics Learning as a Tool for Integrating Science-Technology curriculum in K-12 Schools," 31st Annual Frontiers in Education Conference. Impact on Engineering & Science Education. Conference Proceedings, Reno, NV, USA, 2001.
11. D.P. Miller and C. Stein, "So That's What Pi is For" and Other Educational Epiphanies from Hands-on Robotics, in Robots for kids: Exploring new technologies for learning experiences, A. Druin, A. & J. Hendler (Eds.) San Francisco, CA: Morgan Kaufmann 2000.
12. K. Wedeward, and S. Bruder, "Incorporating Robotics into Secondary Education," Robotics Manufacturing Automation and Control. Vol.14. Proceeding of the

5th Biannual World Automation Congress (WAC 2002) ISORA 2002, ISIAC 2002 and ISOMA, Orlando, FL, USA.2002.

13. K. T. Sutherland, "Undergraduate robotics on a shoestring," IEEE Intelligent Systems, Volume: 15,. Issue: 6, pp. 28-31, 2000.

14. R. Manseur, "A Software Package For Computer-Aided Robotics Education", pp.1409-1412, 26th Annual Frontiers in Education - Vol 3, 1996.

15. Johan J.Craig, Introduction to Robotics Mechanics and Control, 3rd Edition, pp 109-114, Prentice Hall, 2005.

16. Leniel Braz de Oliveira Macaferi, "Construction and Simulation of a Robot Arm with Opengl", May 16, 2007.

17. Baki koyuncu, and Mehmet Güzel, "Software Development For the Kinematic Analysis Of A AL5B Robot Arm", pwaset volume 24 october 2007 issn 1307-6884.

18. Muhammad Ikhwan Jambak, Habibollah Haron, Dewi Nasien, "Development of Robot Simulation Software for Five Joints Mitsubishi RV-2AJ Robot using MATLAB/Simulink and V-Realm Builder", Fifth International Conference on Computer Graphics, Imaging and Visualization, 2008.

19. Arya Wirabhuanal, Habibollah bin Haron "Industrial Robot Simulation Software Development Using Virtual Reality Modeling Approach (VRML) and MATLAB- Simulink Toolbox", University Teknologi Malaysia, 2004.

20. Ildiko Paşc, Radu şarcă, Florin PopenĢiu-Vlădicescu, The VRML Model And Vr Simluation For A Scara Robot, Annals Of The Oradea University, Fascicle Of Management And Technological Engineering, Volume Vi (Xvi), 2007.

21. Surender Kumar, Mukherjee S. K., Robotic Engineering, Smt. Sumitra Handa, First Edition, 2001.

22. Harish K 1, Megha D 2, Shuklambari M 3, Amit K 4, Chaitanya K Jambotkar 5, Pick and Place Robotic Arm Using Arduino, International Paper of Science, Engineering and Technology Research (IJSETR), December 2017, Volume 6, Issue 12.

23. War War Naing, Kyi Zar Aung, Aung Thike, Position Control of 3-DOF Articulated Robot Arm using PID Controller, International Paper of Science and Engineering Applications, 2018, Volume 7–Issue 09,254-259.

24. Mr. C. Chandra Mouli¹, Ms. P. Jyothi¹, Prof. K. Nagabhushan Raju², Prof. C. Nagaraja², Design and Implementation of Robot Arms Control Using LabVIEW and ARM Controller, IOSR Paper of Electrical and Electronics Engineering (IOSRJEEE), (Jul – Aug) 2013, Volume 6, Issue 5.

25. Myat Noe Wai, Lu Maw, Thwe Thwe Win, Kinematics Modeling and Simulation of the 4-DOF Robotic Manipulator, 1st University Research Conference on Science & Engineering, 23rd January,2020,Vol.1, 50- 54.

26. Sanjay L., Shweta P, Position Control of Pick and Place Robotic Arm, International Conference on Engineering Innovation and Technology, Nagpur, 1st July 2012.

27. Jolly Shah, S.S. Rattan, B.C.Nakra, End-Effector Position Analysis Using Forward Kinematics for 5DOF PravaK Robot Arm, International Paper of Robotics and Automation (IJRA), September 2013, Vol. 2, No. 3.

28. Binbin L., Yimin S., Gang D., Tao S., Yang Q, Dimensional Synthesis of a Planar Parallel Manipulator for Pick-and-Place Operations Based on Rigid-Body Dynamics, Intelligent Robotics and Applications/Lecture Notes in Computer Science, Springer, 2012, Vol. 7506, pp. 261-270.

29. Ahmed Zakari Alassar, Modeling and Control of 5DOF Robot Arm Using Supervisory Control, Degree of Master of Science in Electrical Engineering, The Islamic University of Gaza, Electrical Engineering Department, March 2010.

30. Ayokunle A. Awelewa, Kenechukwu C. Mbanisi, Samuel O. Majekodunmi, Ishioma A. Odigwe, Ayoade F. Agbetuyi, Isaac A. Samuel, Development of a Prototype Robot Manipulator for Industrial Pick-and-Place Operations, International Paper of Mechanical & Mechatronics Engineering, November 2013, IJMME-IJENS Vol:13 No:05.

31. YuYang Fu, Research on Structural Design and Motion Trajectory Planning of Industrial Robot. Master Thesis of Wuyi University. 2013.

32. ZiXing Cai, Robotics. Beijing: Tsinghua University Press, 2009: 264.
33. LinShi Gao,XiaoLin Liu And YuanXian Ou,Robotic arm smoothing programming algorithm based on quintic polynomials. Manufacturing Automation, 2013, (11): 16-17, 24.
34. Saeed B.Niku. Introduction to Robotics [M]. BEIJING: Publishing House of Electronics Industry, 2013.
35. Bin Xie, ZiXing Cai,Simulation experiment teaching of Robotics Based on MATLAB Robotics Toolbox. Computer Education, 2010.
36. B.O. Omijeh, R. Uhumwangho, M. Ehikhamenle, “Design Analysis of a Remote Controlled Pick and Place Robotic Vehicle”, International Journal of Engineering Research and Development, Volume 10, PP.57-68, May 2014.
37. Elias Eliot, B.B.V.L. Deepak, D.R. Parhi, and J. Srinivas, “Design & Kinematic Analysis of an Articulated Robotic Manipulator”, Department of Industrial Design, National Institute of Technology-Rourkela.
38. Sanjay Lakshminarayan, Shweta Patil, “Position Control of Pick and Place Robotic Arm”, Department of Electrical Engineering MS Ramaiah Technology, Bangalore, India.
39. Mohamed Naufal Bin Omar, “Pick and Place Robotic Arm Controlled By Computer”, Faculty of Manufacturing Engineering, April 2007.
40. Ankit Gupta, Mridul Gupta, NeelakshiBajpai, Pooja Gupta, Prashant Singh, “Efficient Design and Implementation of 4-Degree of Freedom Robotic Arm”, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, June 2013.

Додаток А

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

МАРТИНЮК Валерій
Хмельницький національний університет
ORCID ID: 0000-0001-5758-4244
e-mail: martynyuk.valeriy@gmail.com
РИЖКОВ Ігор
Хмельницький національний університет
e-mail: igor19997982@gmail.com

УДОСКОНАЛЕНИЙ МЕТОД КЕРУВАННЯ РОБОТОМ МАНІПУЛЯТОРОМ

У статті розроблено імітаційну модель робота маніпулятора з трьома ступенями свободи. Отримані графіки, які відображають вектор переміщення кінцевого суглоба робота маніпулятора можна зробити висновок, що координата z змінюється у часі. Це пояснюється тим, що використовується робот маніпулятор з трьома ступенями свободи, який може рухатися по трьох координатах x , y та z .

Проведено 3-D візуалізацію руху робота маніпулятора з трьома ступенями свободи дозволяє проводити дослідження руху суглобів робота маніпулятора, визначити їх координати, а також налаштовувати ПД-регулятори для кожного суглоба. Це дозволяє проводити динамічне моделювання руху робота маніпулятора для досягнення максимально можливої швидкості руху суглобів і максимально можливої точності позиціонування робочого органу робота маніпулятора.

Ключові слова: робот-маніпулятор, імітаційна модель, 3-D візуалізацію руху робота маніпулятора з трьома ступенями свободи.

MARTYNYUK Valeriy, RYZHKOV Ihor
Khmelnitsky national university, Ukraine

AN IMPROVED METHOD OF CONTROLLING A ROBOT WITH A MANIPULATOR

The article developed a simulation model of a robot manipulator with three degrees of freedom. From the obtained graphs, which display the vector of movement of the end joint of the manipulator robot, it can be concluded that the z coordinate changes over time. This is explained by the fact that a robot manipulator with three degrees of freedom is used, which can move along three coordinates x , y and z .

A 3-D visualization of the movement of the robot manipulator with three degrees of freedom has been carried out, which allows you to study the movement of the joints of the manipulator robot, determine their coordinates, as well as adjust the PD controllers for each joint. This allows dynamic modeling of the movement of the manipulator robot to achieve the maximum possible speed of movement of the joints and the maximum possible accuracy of positioning of the working organ of the manipulator robot.

Keywords: robot manipulator, simulation model, 3-D visualization of the motion of a robot manipulator with three degrees of freedom.

Постановка проблеми

Для академічних або промислових роботів-маніпуляторів ключовим завданням є досягнення бажаних положень та орієнтації їх кінцевих механізмів або інструментів для виконання заданого завдання. Для досягнення цієї мети необхідно володіти ґрунтовними знаннями обернених кінематичних задач. Роботи-маніпулятори використовуються в багатьох галузях для виконання різних завдань, таких як переміщення матеріалів, підбір і розміщення, інтерактивна робота, спільна робота, небезпечні польові роботи тощо.

Тому, підвищення ефективності, взаємодія з маніпулятором робота, вирішення прямих та зворотних кінематичних задач з використанням механізму представлення та оптимізації, операції вибору та розміщення за допомогою механізму захоплення, інтерактивної роботи декількох роботів в MATLAB, є дійсно актуальними задачами.

Роботів-маніпуляторів можна розглядати як групу жорстких шарнірів, з'єднаних специфічними з'єднаннями. Шарніри можуть бути обертовими, призматичними, гвинтовими, універсальними або циліндричними. Робот-маніпулятор вважається таким, у якого перша ланка закріплена в основі, а остання ланка може переміщатися в робочому просторі.

Щоб успішно інтегрувати MATLAB з роботом-маніпулятором, спочатку необхідно створити функціонуючий симулятор. Цей симулятор повинен відтворювати функції фізичної руки, а також працювати безпечно, оскільки можна уникнути фізичних пошкоджень.

Для моделювання маніпулятора робота в середовищі MATLAB необхідні моделі і дерева жорсткого тіла. Деревоподібна модель - це один із способів представлення механічних властивостей маніпулятора робота в MATLAB, а також для візуалізації та керування маніпулятором робота в Simulink. Це перший крок до використання інструментарію робототехніки в MATLAB, а також до візуалізації та керування маніпуляторами робота в Simulink.

Аналіз останніх джерел

Роботизована рука - це робот-маніпулятор, який зазвичай програмується з функціями, аналогічними людській руці [1]. Найважливішими застосуваннями промислових роботів є обробка матеріалів, зварювання, складання, дозування та обробка, де роботизована рука-маніпулятор має безперервно виконувати операції підбору та розміщення. Одним із таких промислових стандартних роботів є звичайна серійна рука, яка складається з основи, ланки або ряду ланок, з'єднаних у з'єднаннях, а також кінцевого ефектора - робочого елемента робота маніпулятора [2].

В процесі аналізу кінематики робота маніпулятора використовується підхід Денавіта-Хартенберга (ДХ), який базується на приєднанні системи координат до кожного з'єднання та визначенні чотирьох параметрів, відомих як ДХ параметри для кожної ланки [3]. Кожний суглоб робота маніпулятора має один ступінь свободи, який може бути або поступальний або обертальний. Для маніпулятора з n шарнірів, пронумерованих від 1 до n , існує $n - 1$ ланок, пронумерованих від 0 до n . Ланка 0 є основою маніпулятора, як правило, фіксованою, а ланка n несе кінцевий ефектор - робочий елемента робота маніпулятора [4].

Виклад основного матеріалу

Розроблено імітаційну модель робота маніпулятора з трьома ступенями свободи, яка зображена на рис. 1.

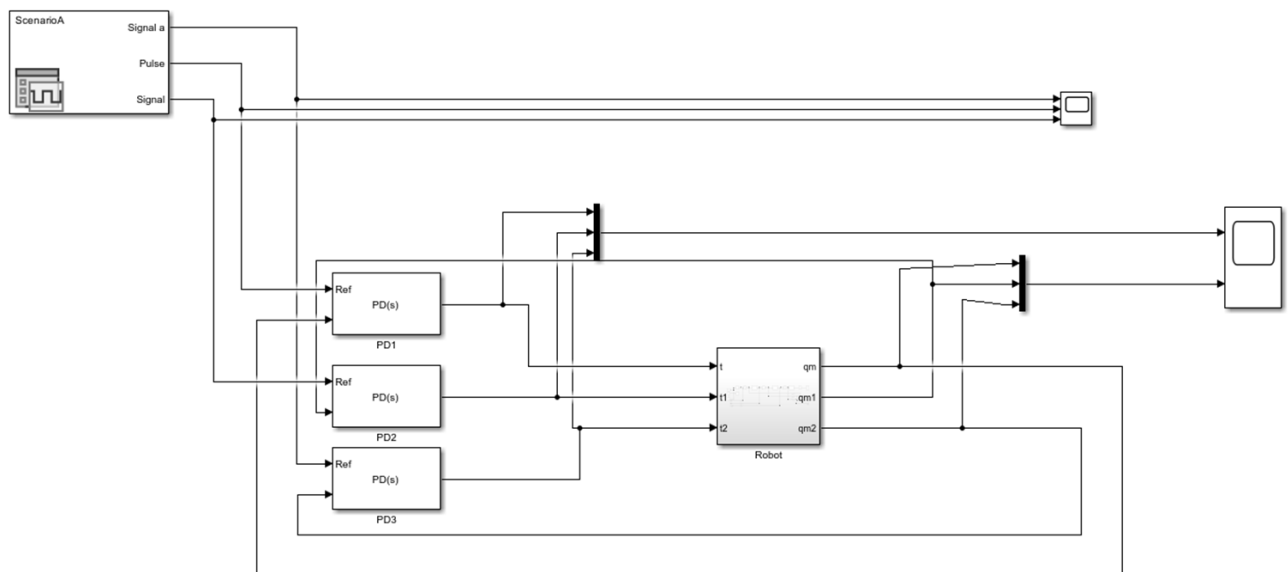


Рис. 1. Імітаційна модель робота маніпулятора з трьома ступенями свободи

До складу імітаційної моделі робота маніпулятора з трьома ступенями свободи входить блок редактор сигналів, за допомогою якого здійснюється відображення, створення, редагування та перемикання взаємозамінних сценаріїв моделювання робота маніпулятора.

Дані сигналів збережено у MAT-файлі для моделювання. Редактор сигналів створює дані сигналу у форматі часових рядів. Графіки вхідних сигналів для кожної осі робота відображаються осцилографом і зображені на рис. 2.

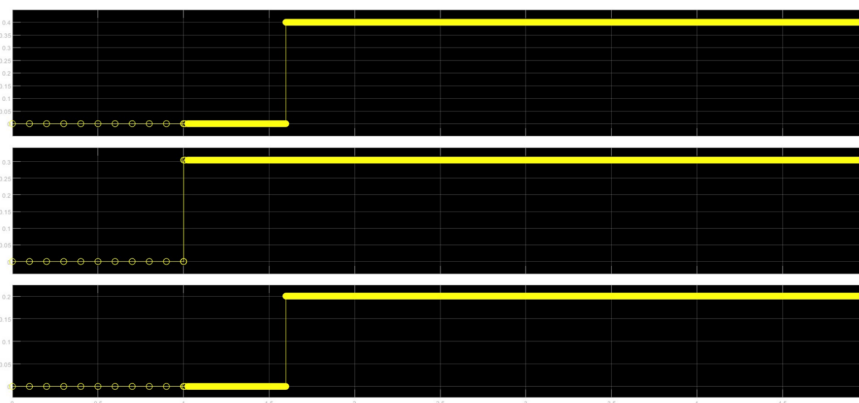


Рис. 2. Графіки вхідних сигналів для кожної осі робота

Далі вхідні сигнали для кожної осі робота поступають на три ПД-регулятора для кожної осі робота маніпулятора. Вихідний сигнал блоку ПД-регулятор є зваженою сумою вхідного сигналу, який помножується на коефіцієнт пропорційності, а також похідної вхідного сигналу. Ваговими коефіцієнтами є пропорційний та похідний параметри підсилення. Поліос першого порядку фільтрує похідну дію. Сигнали з трьох ПД-регуляторів для кожної осі робота маніпулятора поступають на підсистему Robot, яка зображена на рис. 3.

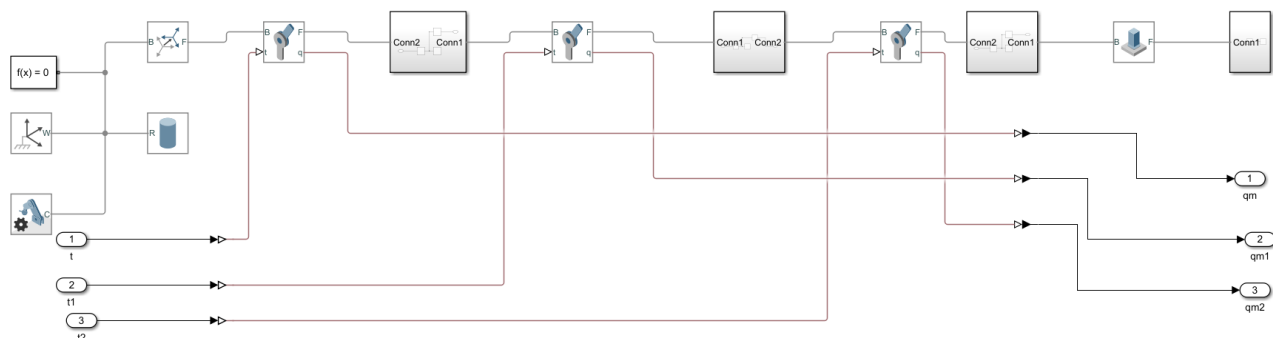


Рис. 3. Імітаційна модель підсистеми робот маніпулятор з трьома ступенями свободи

До складу підсистеми робот (Robot) входять три поворотні блоки з'єднання робота маніпулятора, блоки з'єднання між ними, а також два нерухомих з'єднання. Вхідні параметри підсистеми робот задаються блоком інерційна система відліку (Inertial reference fram). Цей блок визначає глобальну систему відліку в моделі.

Ця система відліку є інерційною і знаходиться в стані абсолютного спокою. Жорстке з'єднання блоку із нерухомою основою робить цю систему відліку інерційною. Осі системи відліку ортогональні і розташовані за правилом правої руки. Інші вхідні параметри підсистеми робот задаються блоком конфігурація механізму (Mechanism Configuration), який задає параметри гравітації та моделювання для всього механізму.

Блок конфігурації механізму визначає параметри сили тяжіння та моделювання механізму, до якого під'єднується блок. Параметри моделювання містять значення збурення для обчислення чисельних часткових похідних під час лінеаризації та номер ітерації для спільного переходу режиму.

Блок Rigid Transform визначає та підтримує фіксоване просторове співвідношення між двома кадрами під час симуляції. Просторове відношення може включати переміщення та обертання. Блок має різні методи для вказівки положення та орієнтації слідкуючої системи координат відносно базової системи координат.

Наступним блоком є блок Base (Циліндричне тіло), який моделює суцільний циліндричний елемент з геометрією, інерцією та кольором. Циліндричний суцільний блок - це циліндрична форма з центром геометрії, що збігається з координатами системи відліку, а вісь симетрії збігається з віссю z системи відліку.

Циліндричний суцільний блок додає до прикріпленої системи координат твердий елемент із геометрією, інерцією та кольором. Твердий елемент може бути простим твердим тілом або частиною складеного твердого тіла - групи жорстко зв'язаних твердих тіл, часто розділених у просторі жорсткими перетвореннями.

Параметри геометрії включають форму та розмір. Можна вибрати зі списку попередньо встановлених фігур або імпортувати спеціальну фігуру із зовнішнього файлу у форматі STL або STEP. За замовчуванням для всіх форм, окрім STL, блок автоматично обчислює масові властивості твердого тіла на основі вказаної геометрії та маси або масової щільності.

Система відліку кодує положення та орієнтацію твердого тіла. У конфігурації за замовчуванням блок забезпечує лише опорний кадр. Інтерфейс створення фреймів надає засоби для визначення додаткових фреймів на основі твердотільних геометричних елементів.

Поворотне з'єднання робота маніпулятора зображене на рис. 4 і має вигляд шарніра, який розміщений між двома плечами.



Revolute Joint

Рис. 4. Поворотне з'єднання робота маніпулятора

Цей шарнір має один обертовий ступінь свободи, представлений одним поворотним елементом. З'єднання обмежує збіг початків двох рамок і збігання осей z базової та слідкуючої рами, у той час як осі x і y слідкуючої можуть обертатися навколо осі z. Порти B і F є портами кадрів, які представляють базовий і наступний кадри відповідно. Напрямок з'єднання визначається рухом ведучої рами відносно базової рами.

Для моделювання кінцевого ефектора - робочого елемента робота маніпулятора, використовується Блок прямокутний паралелепіпед - це призматична форма з центром геометрії, що збігається з координатами системи відліку, і призматичними поверхнями, нормальними до осей x, y і z системи відліку.

Блок прямокутний паралелепіпед додає до прикріпленої рами твердий елемент із геометрією, інерцією та кольором. Цегляний твердий елемент може бути простим твердим тілом або частиною складеного твердого тіла - групи жорстко з'єднаних твердих тіл, часто розділених у просторі через жорсткі перетворення. Цей блок має нульовий ступінь свободи. Ведуча та базова рами завжди збігаються. Порти В і F є портами кадрів, які представляють базовий і наступний кадри відповідно.

Вихідні сигнали підсистеми робот (Robot) поступають на осцилограф для відображення результатів імітаційного моделювання удосконаленого методу керування роботом і зображені рис. 5.

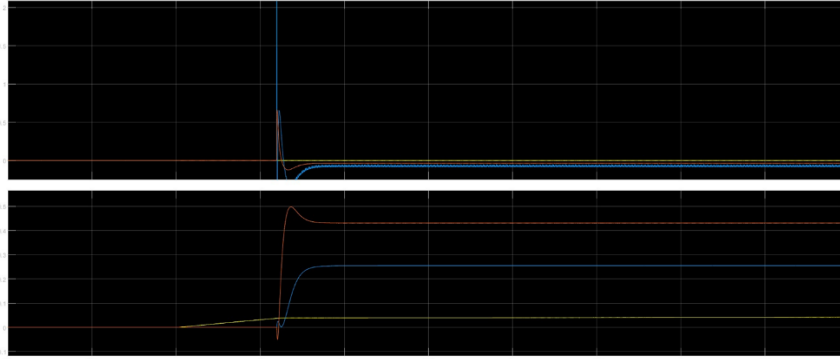


Рис. 5. Результати імітаційного моделювання удосконаленого методу керування роботом

В результаті проведення математичного моделювання отримали 3-D візуалізацію руху робота маніпулятора з трьома ступенями свободи в середовищі Simulink, яка зображена на рисунку рис. 6.

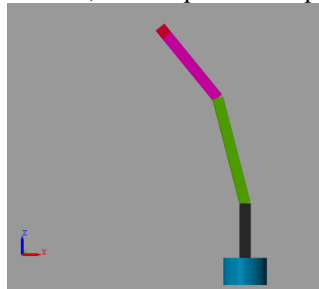


Рис. 6. 3-D візуалізація руху робота маніпулятора з трьома ступенями свободи

Висновки

Аналізуючи графіки, які відображають вектор переміщення кінцевого суглоба робота маніпулятора можна зробити висновок, що координата z змінюється у часі. Це пояснюється тим, що використовується робот маніпулятор з трьома ступенями свободи, який може рухатися по трьох координатах x , y та z .

3-D візуалізація руху робота маніпулятора з трьома ступенями свободи дозволяє проводити дослідження руху суглобів робота маніпулятора, визначати їх координати, а також налаштовувати ПД-регулятори для кожного суглоба.

Це дозволяє проводити динамічне моделювання руху робота маніпулятора для досягнення максимально можливої швидкості руху суглобів і максимально можливої точності позиціонування робочого органу робота маніпулятора.

Література:

1. War War Naing, Kyi Zar Aung, Aung Thike Position Control of 3-DOF Articulated Robot Arm using PID Controller, International Paper of Science and Engineering Applications, 2018, Volume 7 –Issue 09, 254-259.
2. C. Chandra Mouli1, P. Jyothi1, K. Nagabhushan Raju, C. Nagaraja Design and Implementation of Robot Arms Control Using LabVIEW and ARM Controller, IOSR Paper of Electrical and Electronics Engineering (IOSRJEEE), (Jul – Aug) 2013, Volume 6, Issue 5.
3. Myat Noe Wai, Lu Maw, Thwe Thwe Win Kinematics Modeling and Simulation of the 4-DOF Robotic Manipulator, 1st University Research Conference on Science & Engineering, 23rd January, 2020, Vol. 1, 50-54.
4. P.I. Corke Robotics Toolbox for MATLAB. Second edition, Springer, 2013, 208 p.

References

1. 1. War War Naing, Kyi Zar Aung, Aung Thike, Position Control of 3-DOF Articulated Robot Arm using PID Controller, International Paper of Science and Engineering Applications, 2018, Volume 7 –Issue 09, 254-259.
2. C. Chandra Mouli1, P. Jyothi1, K. Nagabhushan Raju, C. Nagaraja, Design and Implementation of Robot Arms Control Using LabVIEW and ARM Controller, IOSR Paper of Electrical and Electronics Engineering (IOSRJEEE), (Jul – Aug) 2013, Volume 6, Issue 5.
3. Myat Noe Wai, Lu Maw, Thwe Thwe Win, Kinematics Modeling and Simulation of the 4-DOF Robotic Manipulator, 1st University Research Conference on Science & Engineering, 23rd January, 2020, Vol. 1, 50-54.
4. P.I. Corke Robotics Toolbox for MATLAB. Second edition, Springer, 2013, 208 p.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Рижков Ігор Сергійович

Тема: Удосконалений метод керування роботом маніпулятором

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

Обсяг кваліфікаційної роботи:

Кількість листів креслень 16 Кількість сторінок записки 74

1. Короткий зміст роботи та прийнятих рішень: Вступ. Огляд літературних джерел та патентних даних. Математична модель процесу керування роботом маніпулятором. Імітаційна модель методу керування роботом маніпулятором. Експериментальні дослідження удосконаленого методу керування роботом маніпулятором. 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. У другому розділі розроблено математичну модель роботи маніпулятором. Розроблено удосконалений метод керування роботом маніпулятором. У третьому розділі розроблено імітаційну модель процесу керування роботом маніпулятором допомогою ПД-регулятора. У четвертому розділі проведені експериментальні дослідження удосконаленого методу керування роботом маніпулятором.
4. Позитивні сторони роботи: додавання трьох ПД-регуляторів для трьох осей робота маніпулятора.

5. Негативні сторони роботи: у роботі недостатньо уваги приділяється аналізу похибок позиціонування робота маніпулятора..

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: задовільно (3,50/D)

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

Клюва Юрій Павлович К.Т.К, доцент, зав. кафедри мінералогії

"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 керування роботом маніпулятором

Автор: Рижков Ігор Сергійович

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.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

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

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

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

3) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає 5,18% і адресується до 184 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.

Рижков І.С.

ПІБ здобувача вищої освіти

ФІТ, 2 курс, групи АКІТм-22-1

ЗАЯВА

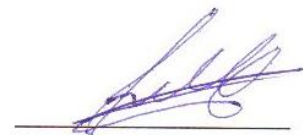
З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, 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: 125116 Назва: МКР Удосконалений метод керування роботом маніпулятором Додано в БД: 2023-12-29 Автора: Ігор РИЖКОВ Керівники: Валерій МАРТИНЮК Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	64811	483	1374 (2%)	19 (4%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:
Кафедра АКІПТК

Дата перевірки:
29.12.2023 12:27:38 EET

Дата звіту:
29.12.2023 12:40:48 EET

ID перевірки:
1016041174

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005862

Назва документа: Рижков_кваліфікаційна_робота

Кількість сторінок: 66 Кількість слів: 9787 Кількість символів: 74384 Розмір файлу: 4.91 MB ID файлу: 1015735707

339 слів позначені як "вилучені" та не враховуються у підрахунку слів

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

5.18% Схожість

Найбільша схожість: 2.76% з Інтернет-джерелом (<https://web01.usn.no/~roshans/cfr/downloads/Control-for-Robotics-Le..>)

4.3% Джерела з Інтернету

179

Сторінка 68

0.98% Джерела з Бібліотеки

5

Сторінка 69

0% Цитат

Вилучення цитат вимкнено

Посилання

1

Сторінка 69

0% Вилучень

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

0% Вилучення з Інтернету

6

Сторінка 70

0% Вилученого тексту з Бібліотеки

3

Сторінка 70

Модифікації

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

Підозріле форматування

16
сторінок