

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

КВАЛІФІКАЦІЙНА РОБОТА

Метод розроблення моделі мехатронного робота-гексапода

Назва теми

КвРАКІТР.2024075.01.02.ПЗ

Рівень вищої освіти другий (магістерський)

Галузь знань 17 «Електроніка, автоматизація та електронні комунікації»

Шифр, назва

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

Шифр, назва

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

Назва

Виконав:

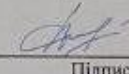
студент II курсу, група АКІТРм-24-1



Підпис

Артем БІГУС
Ім'я, ПРІЗВИЩЕ

Керівник канд. техн. наук, доцент



Підпис

Людмила КОРЕЦЬКА
Ім'я, ПРІЗВИЩЕ

Нормоконтролер



Підпис

Галина РАДЕЛЬЧУК
Ім'я, ПРІЗВИЩЕ

До захисту допускаю:
зав. кафедри АКІТтаР



Підпис

Людмила КОРЕЦЬКА
Ім'я, ПРІЗВИЩЕ

« 18 » грудня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій
Кафедра автоматизації, комп'ютерно-інтегрованих технологій та робототехніки
Рівень вищої освіти другий (магістерський)
Галузь знань 17 – Електроніка, автоматизація та електронні комунікації
Спеціальність 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Освітня програма Автоматизація, комп'ютерно-інтегровані технології та робототехніка

ЗАТВЕРДЖУЮ

Завідувач кафедри АКІТтаР

Людмила КОРЕЦЬКА

01 вересня 2025р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Бігусу Артему Романовичу

Прізвище, ім'я, по батькові студента

1 Тема роботи Метод розроблення моделі мехатронного робота-гексапода

Керівник роботи Корецька Людмила Олександрівна, канд. техн. наук, доцент

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

Затверджено наказом ректора університету від 25.08.2025 р. №65

2 Строк подання студентом роботи на кафедру 08.12.2025р.



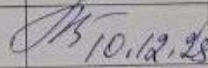
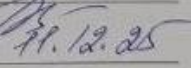
3 Вихідні дані до роботи Аналіз сучасного стану проєктування мехатронних роботів-гексаподів. Розробка алгоритмічного та програмного забезпечення для керування мехатронними роботами-гексаподами.

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Вступ. Проблематика проєктування моделі мехатронного робота-гексапода. Моделювання конструкції мехатронного робота-гексапода. Підбір основних елементів для моделі мехатронного робота-гексапода. Дослідження роботи моделі мехатронного робота-гексапода. Висновки.

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

6 Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Антиплагіат	Федула М.В., доцент кафедри АКІТтаР		
Нормоконтроль	Радельчук Г.І., доцент кафедри АКІТтаР	 10.12.25	 11.12.25

7 Дата видачі завдання 01 вересня 2025р.

КАЛЕНДАРНИЙ ПЛАН

Назва розділу кваліфікаційної роботи	Строк виконання	Примітка
1 Вступ	15.09.2025р.	виконано
2 Проблематика проєктування моделі мехатронного робота-гексапода	30.09.2025р.	виконано
3 Моделювання конструкції мехатронного робота-гексапода	16.10.2025р.	виконано
4 Підбір основних елементів для моделі мехатронного робота-гексапода	01.11.2025р.	виконано
5 Дослідження роботи моделі мехатронного робота-гексапода	12.11.2025р.	виконано
6 Висновки	14.11.2025р.	виконано
7 Оформлення пояснювальної записки до КРМ	01.12.2025р.	виконано
8 Оформлення презентаційних матеріалів	01.12.2025р.	виконано

Студент


Підпис

Артем БІГУС
Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи


Підпис

Людмила КОРЕЦЬКА
Ім'я, ПРІЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Метод розроблення моделі мехатронного робота-гексапода».

Автор роботи: Бігус Артем Романович

Керівник роботи: канд. техн. наук, доцент Корецька Людмила Олександрівна

Пояснювальна записка: 95 с., 61 рис., 9 табл., 2 дод., 70 джерел.

Графічна частина: 10 презентаційних слайдів.

МЕХАТРОННИЙ, РОБОТ-ГЕКСАПОД, БІОНІКА, МОДЕЛЮВАННЯ,
СИСТЕМА КЕРУВАННЯ, MATLAB, ARDUINO.

Об'єкт дослідження: процеси керування моделі мехатронного робота-гексапода.

Мета роботи: розробити модель мехатронного робота-гексапода. Виконано розробку системи керування мехатронним роботом-гексаподом та підбір основних елементів керування. Вся керуюча частина МРГ реалізується шляхом проектування та аналізу сервоприводів Tower Pro MG996r, плати керування Arduino mega2560, ПДК – пульт від PS2, камери Nanying A8 та модуля живлення електроприводної частини LM2596S, а також із використанням програмування Arduino.

Виконано моделювання роботи конструкції мехатронного робота-гексапода в програмному середовищі MATLAB. Розроблено програмне забезпечення для моделі мехатронного робота-гексапода для мікроконтролерів типу Arduino.



Підпис студента

05.12.2025

Дата

ЗМІСТ

	с.
ВСТУП	4
1 ПРОБЛЕМАТИКА ПРОЄКТУВАННЯ МОДЕЛІ МЕХАТРОННОГО РОБОТА-ГЕКСАПОДА.....	8
1.1 Огляд та аналіз існуючих конструкцій мехатронних роботів-гексаподів.....	8
1.2 Розроблення конструкції механічної частини моделі мехатронного робота-гексапода.....	12
1.3 Висновки до першого розділу.....	19
2 МОДЕЛЮВАННЯ КОНСТРУКЦІЇ МЕХАТРОННОГО РОБОТА-ГЕКСАПОДА	20
2.1 Перевірка загальних параметрів і міцності ключових конструкцій мехатронного робота-гексапода	20
2.2 Аналіз ходи мехатронного робота-гексапода	26
2.3 Дослідження варіантів ходи мехатронного робота-гексапода	28
2.4 Висновки до другого розділу.....	41
3 ПІДБІР ОСНОВНИХ ЕЛЕМЕНТІВ ДЛЯ МОДЕЛІ МЕХАТРОННОГО РОБОТА-ГЕКСАПОДА.....	42
3.1 Аналітичне дослідження стійкості ходи мехатронних роботів-гексаподів	42
3.2 Підбір основних елементів для моделі мехатронного робота-гексапода ..	46
3.3 Висновки до третього розділу.....	63
4 ДОСЛІДЖЕННЯ РОБОТИ МОДЕЛІ МЕХАТРОННОГО РОБОТА-ГЕКСАПОДА	64
4.1 Моделювання роботи конструкції мехатронного робота-гексапода.....	64
4.2 Розроблення програмного забезпечення для моделі мехатронного робота-гексапода	73

4.3 Висновки до четвертого розділу.....	76
ВИСНОВКИ.....	77
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	78
Додаток А Програмний код для керування моделлю МРГ.....	86
Додаток Б Стаття у фаховому виданні.....	100

ВСТУП

Біоніка в якості науки була запропонована на першій конференції з біоніки в 1960-х роках минулого сторіччя. Це продукт поєднання наук про життя та інженерних технологій [15-18]. Біонічні роботи (БР) – це досить новий напрямок розвитку в галузі робототехніки. Це такий тип робота, що заснований на біологічних характеристиках конкретної істоти, тобто робот імітує зовнішню структуру істоти, рух і загальну поведінку [10-18].

На сучасному етапі розвитку науки і техніки існує три класифікації БР: так звані роботи-гуманоїди, стандартні БР та, навіть, біологічні роботи [10-18]. Більшість з цих БР використовуються для виявлення та передачі інформації в небезпечних для людей чи просто складних для досягнення областях.

Конструкція мехатронного робота-гексапода (МРГ), що проектується створена шляхом поєднання біологічних механізмів і робототехніки із загальними принципами науки біоніки [10-18]. Проєкт використовує біологічну структуру павукоподібних членистоногих і моделює механізм руху та систему моделювання руху. Тобто це робот, який повинен переміщуватись, визначати орієнтацію власного руху і сприймати зовнішні подразники.

МРГ відрізняється від людини або стандартного колісного робота саме механічною структурою, для того, щоб адекватно працювати в навколишньому середовищі. Конструкція, що проектується, повинна справлятися із суворими природними умовами, вільно переміщатися по нерівних поверхнях і повністю уникати встановлених контрольних точок. МРГ мають надзвичайно високу цінність у багатьох галузях промисловості і володіють досить великою тенденцією розвитку.

В останні роки, завдяки розвитку теорії біоніки та науки і техніки, дослідження і розробки інтелектуальних МРГ стали предметом занепокоєння дослідників. Було проведено безліч досліджень МРГ за кордоном, і були створені різні конструкції МРГ.

Від чисто механічного руху до інтелектуального механічного руху інтелект і всебічні характеристики МРГ швидко розвивалися.

Метою роботи є розв'язання задачі розроблення моделі мехатронного робота-гексапода.

Для досягнення цієї мети необхідно розв'язати наступні завдання:

- розглянути та проаналізувати існуючі конструкції мехатронних роботів-гексаподів, підібрати вихідних даних для проектування;
- перевірити загальні параметри та міцність ключових конструкцій мехатронного робота-гексапода, аналітично дослідити варіанти ходи мехатронного робота-гексапода;
- дослідити стійкості ходи мехатронних роботів-гексаподів, виконати підбір основних елементів для моделі мехатронного робота-гексапода;
- виконати моделювання роботи конструкції мехатронного робота-гексапода в програмному середовищі MATLAB, розробити програмне забезпечення для моделі мехатронного робота-гексапода.

Об'єктом дослідження є процеси керування моделі мехатронного робота-гексапода.

Предметом дослідження є методи та алгоритми керування моделі мехатронного робота-гексапода.

Методи досліджень. Використовувались основні положення теорії моделювання процесів, загальної теорії систем, системного аналізу (ієрархічності, декомпозиції та ін.). Внаслідок проведення моделювання процесу та розроблення методів керування моделі мехатронного робота-гексапода використано методи концептуального моделювання, теоретико-множинні підходи, алгебру систем, принципи формування логічного висновку, евристичні оцінки.

Наукова новизна отриманих результатів: розроблено модель системи автоматичного керування автоматизованою конструкцією мехатронного робота-гексапода.

Практична значимість отриманих результатів: розробленні програмно-технічні засоби для керування автоматизованою конструкцією мехатронного робота-гексапода.

Апробація результатів дослідження: опубліковано повнотекстову статтю у фаховому виданні Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка в технологічних процесах» Випуск №1, 2026 рік.

Кваліфікаційна робота складається із вступу, чотирьох розділів, висновків до кожного розділу, загальних висновків, списку використаних джерел, одного додатку. Загальний обсяг роботи складає 95 сторінок комп'ютерного тексту, у тому числі: 61 рисунок та 9 таблиць, перелік джерел посилянь вміщує 70 найменувань.

У вступі сформульовано мету та завдання досліджень, обґрунтована актуальність поставленої задачі, представлено основні результати та значення.

У першому розділі магістерської кваліфікаційної роботи описано проблематику проєктування моделі мехатронного робота-гексапода, для чого виконано огляд та аналіз існуючих конструкцій мехатронних роботів-гексаподів. Базуючись на проведеному аналітичному огляді було проведено розроблення конструкції механічної частини моделі мехатронного робота-гексапода.

У другому розділі кваліфікаційної роботи виконано моделювання конструкції, проведено перевірку загальних параметрів і міцності ключових конструкцій мехатронного робота-гексапода. Було проведено аналіз виконання руху, виконано дослідження варіантів виконання руху мехатронним роботом-гексаподом.

У третьому розділі кваліфікаційної роботи магістра проведено аналітичне дослідження стійкості ходи мехатронних роботів-гексаподів. Було виконано розроблення системи керування мехатронним роботом-гексаподом та підбір основних елементів керування. Вся керуюча частина МРГ реалізується шляхом проєктування та аналізу сервоприводів Tower Pro MG996r, плати керування Arduino mega2560, ПДК – пульт від PS2, камери Nanying A8 та модуля

живлення електроприводної частини LM2596S, а також із використанням програмування Arduino.

У четвертому розділі кваліфікаційної роботи проведено дослідження роботи розробленої моделі мехатронного робота-гексапода, а саме виконано моделювання роботи конструкції мехатронного робота-гексапода в програмному середовищі MATLAB. Розроблено програмне забезпечення для моделі мехатронного робота-гексапода для мікроконтролерів типу Arduino, в майбутніх дослідженнях необхідно використовувати МК сімейства STM32. Було виконано практичну перевірку на прототипі.

У загальних висновках по кваліфікаційній роботі магістра представлено отримані результати роботи «Метод розроблення моделі мехатронного робота-гексапода».

1 ПРОБЛЕМАТИКА ПРОЄКТУВАННЯ МОДЕЛІ МЕХАТРОННОГО РОБОТА-ГЕКСАПОДА

1.1 Огляд та аналіз існуючих конструкцій мехатронних роботів-гексаподів

У 1970-х роках минулого століття фірма виробник Mosher створила 4-ногий механізм під назвою «Walking Truck» [1]. Загальний вигляд представлено на рисунку 1.1, конструкція оснащена 4-ма ногами і приводиться в загальний рух за допомогою використання гідравліки. Давач положення може реалізувати функцію визначення місцезнаходження відносно людського тіла [1]. Хоча загальний рух 4-ногого робота досить трудомісткий, механізм виконує функції обходу перешкод. Це був досить важливий поворотний момент у розробці сучасних робототехнічних комплексів для автоматизації процесу ходьби, який заклав хорошу основу для подальшого розвитку технології МРГ.

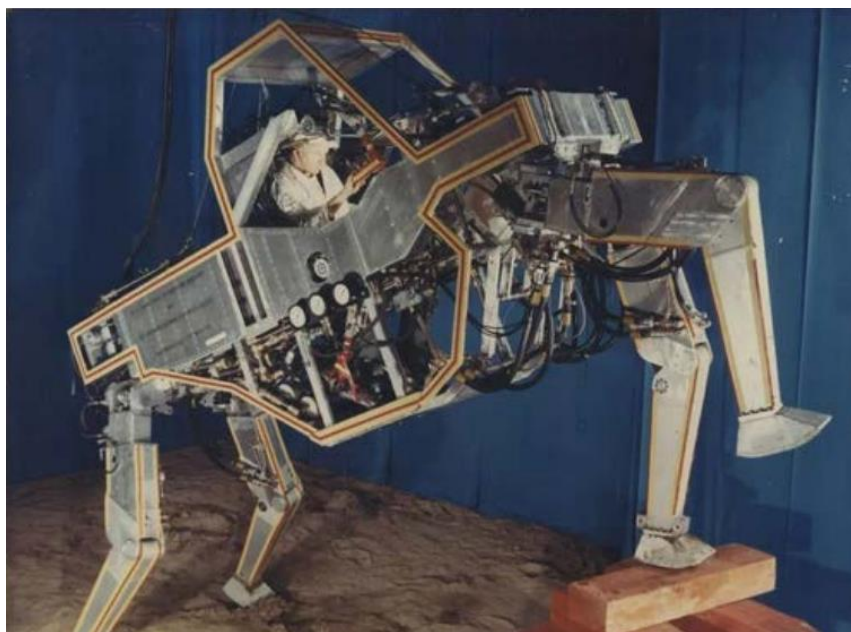


Рисунок 1.1 – Загальний вигляд механізму Walking Truck [1]

Щоб знизити ризик десантних операцій, ВПС США і Масачусетський технологічний інститут (МІТ) створили конструкцію 6-ногого автоматизованого робота Agilx, призначеного для виявлення мін на мілководді

[2]. Кожна нога даного робота володіє двома ступенями свободи обертання, які можуть виконувати функцію перевертання і руху. Давач положення і компас також встановлені на корпусі, який закриває всі схеми машини і ПЛК в одному корпусі.

Автоматичний інтелектуальний робот (AIP) DANTE [3] представлено на рисунку 1.2. Даний AIP використовувався для вивчення будови вулкана Елібус в Антарктиді і надіслав велику кількість цінних зображень та інформації для аналізу даних.



Рисунок 1.2 – Загальний вигляд AIP DANTE [3]

Spider pot – маленький арахноподібний робот, створений Лабораторією реактивного руху НАСА [4] – використовуючи пару антен для виявлення перешкод і спритні задні лапи, може легко виконувати різні спеціальні дії. Дана конструкція також володіє усіма перевагами невеликого розміру і досить швидкого руху. Може уникати перешкод, справлятися з різними досить складними місцевостями і досліджувати важкодоступні місця. Конструкція

робота, схожого на павука, також може бути використана для дослідження невеликих небесних тіл, крім того використовувався в якості обслуговуючого персоналу на Міжнародній космічній станції (МКС) і використовувався для ліквідації аварійних ситуацій, наприклад, виток повітря.

Лабораторія нано-роботів США Carnegie Mellon University/CMU [5] розробила конструкцію міні-робота «Water spider», який не тільки виглядає як справжній павук, але і вільно переміщається по поверхні води. Корпус даної конструкції робота складається з коробчастого карбонізованого волокна довжиною близько 0,0127 м і має в будові 8 сталевих дротяних ніжок, покритих водонепроникним пластиком [6]. Даний робот використовує дроти для підключення п'єзоелектричного приводу до зовнішнього джерела живлення. Коли рівень напруги підвищується, металевий край зминається, створюючи рушійну силу, що дозволяє працювати конструкції робота. «Water spider» [6] має вагу менше одного грама і є відносно недорогим, що дуже підходить для великомасштабного виробництва.

Щоб підвищити безпеку солдатів під час бою, військові США спільно з МІТ розробили мобільного робота Ariel для виявлення мін у 1995 році минулого століття [2]. Дана конструкція робота оснащена хорошою водонепроникністю, також може бути оснащена компасом і давачами положення, а також обладнанням для моніторингу в режимі реального часу. Якщо є несправність, її можна компенсувати використанням автоматичного виявлення перешкод.

Агентство перспективних дослідницьких проєктів в області оборони США і Північно-східний університет в Сполучених Штатах, спільно розробили конструкцію робота, що створює рух на 8-ми ногах, та імітує справжнього лобстера [7]. Дана конструкція може використовуватись для виявлення мін на дні океану. Конструкція оснащена ніжками розміром чотири на вісім дюймів із вісьмома ступенями свободи, які можуть виконувати рухи повзання та плавання. Конструкція голови оснащена парою плоскогубців, мобільний робот гідравлічно керується.

Дослідження і розробки конструкцій мобільних роботів в Китаї почалися пізно в порівнянні із іншими зарубіжними країнами приблизно в кінці 1980-х років минулого століття. Проте завдяки постійним зусиллям і дослідженням технологія даного обладнання в Китаї досягла досить суттєвих успіхів. У 2007 році дослідницька лабораторія Харбінського технологічного інституту (НІТ) розробила конструкцію мобільного робота НІТ-Spider101 на основі палички-комахи [8]. Як представлено на рисунку 1.3, шість ніг мобільного робота Spider101 рівномірно розподілені по боках тіла, кожна окрема нога володіє трьома ступенями свободи, конструкція передачі паралельна чотирилангова передача, сервомеханізм встановлений на тулубі МРГ, що значно покращує загальну стійкість конструкції. Проте головним недоліком даної конструкції робота є те, що він володіє єдиним типом ходу і не може впоратися зі складними умовами навколишнього середовища.

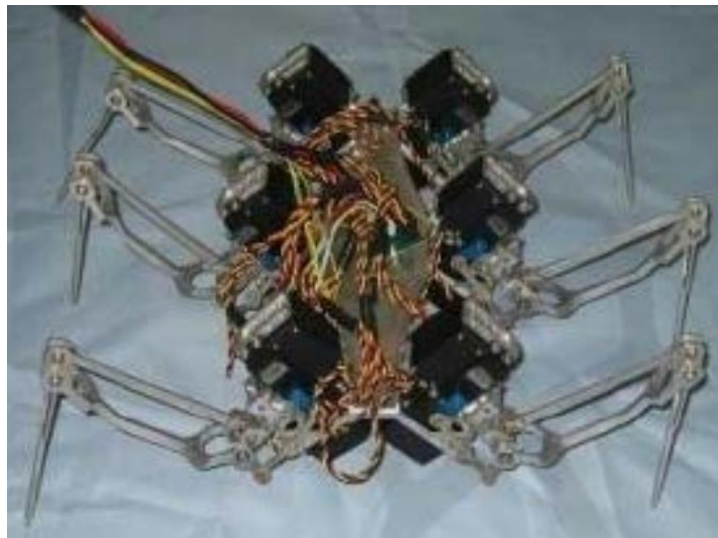


Рисунок 1.3 – Загальний вигляд НІТ-Spider101 [8]

У 2012 році НІТ перепроєктував мобільного 6-ногого робота із центральною симетричною структурою і великим відношенням навантаження до ваги, що приводиться в рух електродвигуном. Як представлено на рисунку 1.4, нова конструкція МРГ володіє високою адаптивністю і великим

співвідношенням навантаження до ваги в складних умовах навколишнього середовища [8]. В даний час відношення навантаження до ваги даної конструкції МРГ може досягати 0,46, а кут підйому не більше 35° [8].



Рисунок 1.4 – Загальний вигляд нової конструкції МРГ від НІТ [8]

У 2018 році було створено конструкцію 4-ногого робота «Jueying», в Університеті Чжецзян та фірмою DEEP Robotics [9]. З основною функцією вправ і стрибків, конструкція МР також може ходити по більш пересіченій місцевості, такій як сніг, схили пагорбів чи, навіть, гравій. Час автономної роботи МР може досягати двох годин.

Наука робототехніка досить швидко розвивається в напрямку штучного інтелекту (ШІ), а технологія біонічних МР розвивається особливо швидко. Найбільша цінність використання робототехніки полягає в можливості виконувати завдання, які більшість людей просто не можуть виконати. Робототехніка висуває більш високі вимоги до матеріалу тіла при роботі в суворих природних умовах.

1.2 Розроблення конструкції механічної частини моделі мехатронного робота-гексапода

1.2.1 Функціональний аналіз

Конструкція МРГ, що проєктується, в основному призначена для небезпечних і складних середовищ. Першочергове завдання проєктування полягає в тому, щоб конструкція МРГ могла виявляти і передавати зображення в тих місцях, де будівлі можуть бути пошкодженими в результаті вибуху чи пожежі, в яких людям рятувальникам буде важко отримати доступ і де вони не змогли б працювати.

Відповідно до роботи, яка повинна бути виконана МРГ, і умов місцевості, в яких конструкція повинна працювати, проводиться всебічний аналіз, і наводяться наступні вимоги до конструкції [10-14]:

- у складному робочому просторі, для того щоб конструкція МРГ могла виконувати поставлене завдання, загальний рух повинен бути дуже гнучким і адаптуватися до складного навколишнього рельєфу, кожна окрема нога МРГ повинна володіти як мінімум трьома ступенями свободи, а весь МРГ також повинен володіти трьома ступенями свободи;

- під час виконання поставлено завдання конструкції МРГ необхідно носити спеціальну камеру, яка вимагає, щоб МРГ володів певною вантажопідйомністю;

- МРГ повинен легко виявляти перешкоди в складних умовах навколишнього простору і повинен мати невеликий розмір з точки зору форми тіла;

- конструкція МРГ повинна бути оснащена спеціальним освітленням для темних місць простору;

- конструкція МРГ повинна мати певний рівень швидкості загального руху за умови забезпечення плавності самого руху.

1.2.2 Характеристика конструкції мехатронного робота-гексапода

Згідно з теорією природної еволюції Чарльза Дарвіна [15], після відбору та еволюції істот із 6-ма ногами, він перетворився на структуру тіла та здатність до руху, здатну адаптуватися до різноманітного природного середовища, і більшість роботів із великою кількістю ніг у сучасному світі використовують фізіологічну структуру істот із багатьма ногами (наприклад, павуків, мурах) в якості еталону для проектування механічної структури частини МРГ. Така будова як у арахноподібних істот із 8-ми ногами володіє досить великою перевагою з точки зору спритності, але не жорсткості конструкції, тому морфологічна будова і характер руху кінцівок арахноподібних істот вивчаються і аналізуються, і на основі цього розробляється механічна структура МРГ.

Загалом відомо понад 40 000 відомих арахноподібних істот у всьому світі, лише в Китаї близько 1000 видів таких істот.

Кожна нога реальної істоти складається із вузлів: базального, поворотного, ноги, коліна, великогомілкового та заднього [15]. Як представлено на рисунку 1.5, а представлено структуру ноги реальної істоти та відповідно до нижченаведених умов представлена спрощена схема структури ноги МРГ (рис. 1.5, б).

До механізму МРГ, що проєктується висуваються наступні вимоги [10-14]:

- МРГ повинен володіти вантажопідйомністю, яка може підтримувати основного тіла МРГ, забезпечуючи стабільність МРГ під час виконання руху;
- кожна нога може виконувати рухи окремо, а всі шість ніг одночасно можуть виконувати складні рухи шляхом об'єднання;
- розмір, конструкція, суглоби та параметри кожної окремої ноги повинні бути однаковими;
- ступінь свободи кожної окремої ноги складає три ступені свободи, які в основному можуть виконувати пряму ходьбу і рульове керування, роблячи конструкцію МРГ більш гнучкою;
- у випадку коли всі вимоги виконані, вага МРГ також повинна бути

зменшена, щоб зменшити складність обробки конструкції.

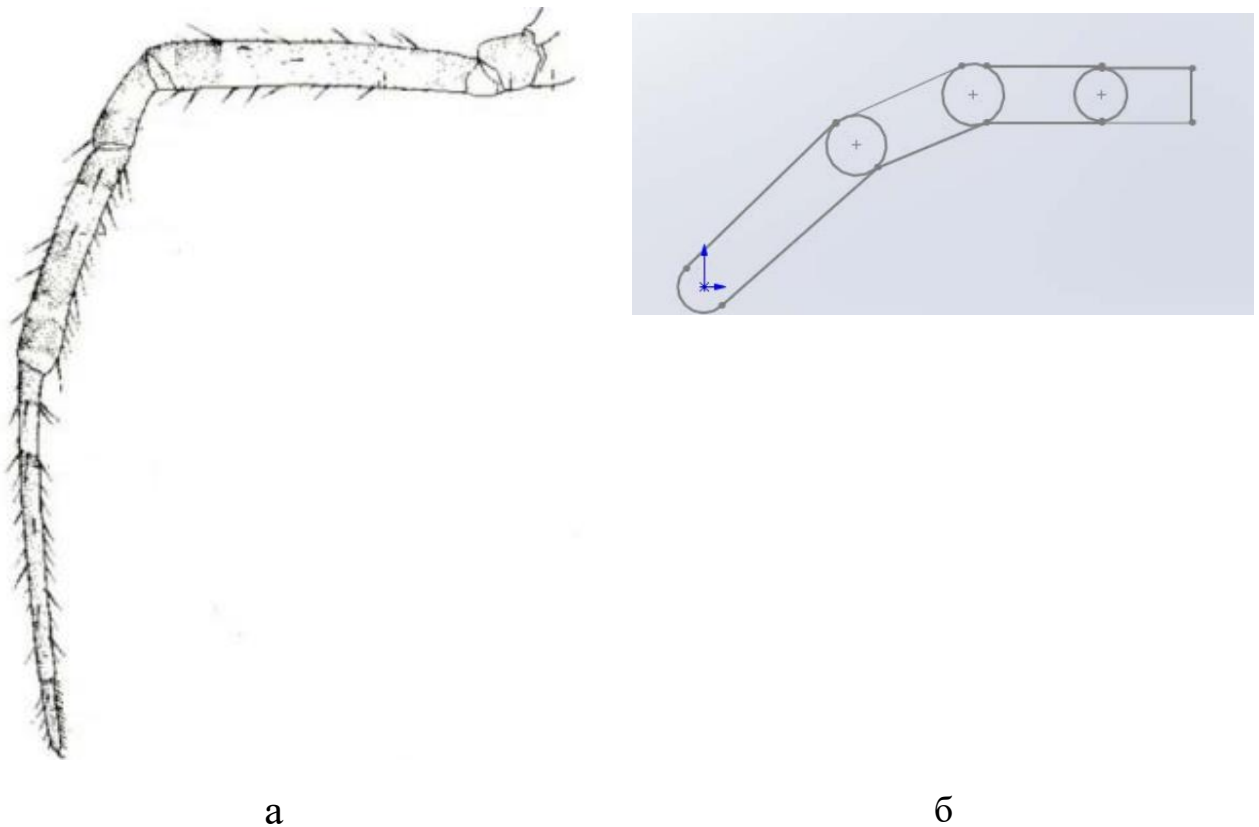


Рисунок 1.5 – Загальний вигляд:

а – структури ноги реальної істоти; б – структури ноги МРГ, що проєктується

Нога МРГ в такому типі конструкції механізму складається як мінімум із трьох суглобів кожен із яких зі ступенем свободи обертання, і кожен суглоб буде приводитись в рух окремим сервоприводом. Керуючи поворотом рульового механізму відповідного суглобу, досягаються функції лінійного переміщення і рульового керування всієї конструкції МРГ.

1.2.3 Загальний вигляд конструкції ноги МРГ

Механічна конструкція ноги МРГ представлена на рисунку 1.6, який розроблений за зразком структури ноги реальної природної істоти вище. Відповідно до 3-х ступенів свободи ніг реальної істоти необхідно спроектувати

базовий вузол, щоб з'єднати корпус тіла МРГ і поворотний вузол; 2 – нога, з'єднувальний поворотний вузол і колінний вузол; 3 – великогомілковий вузол, з'єднувальний колінний вузол і задній клубовий; 4 – задня частина, що з'єднує великогомілкову і клубову частини: кожна частина приводиться в рух окремим сервоприводом, що імітує суглоби реальної істоти, з'єднує основний корпус МРГ і виконує необхідні види рухів на ногах.

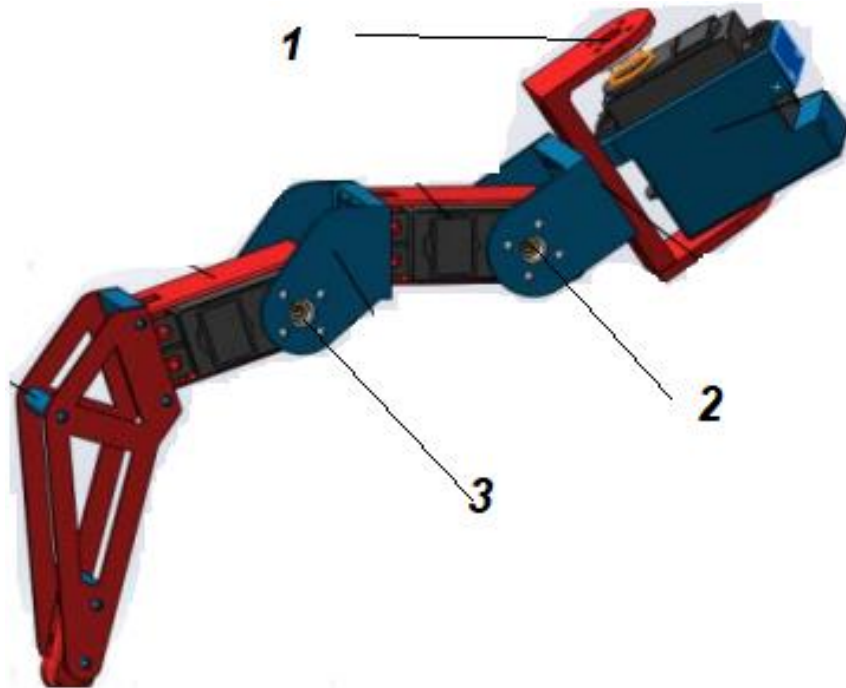


Рисунок 1.6 – Загальний вигляд ноги МРГ – опора мехатронної конструкції

Перший механізм (рис. 1.6) керування еквівалентний базовому вузлу ніжки природної істоти, який відіграє роль у з'єднанні корпусу і поворотного вузла; Другий механізм керування еквівалентний поворотному з'єднанню, яке відіграє роль з'єднання ноги і базового з'єднання, яке може піднімати і опускати механічну ногу МРГ; перший та другий механізми керування з'єднані між собою двома U-подібними кронштейнами; другий та третій механізми керування з'єднані шатуном, який відповідає нозі реальної істоти, посилена частина шатуна має форму Н, що може ефективно захищати від зовнішнього впливу весь суглоб, уникнути деформації ніжки і вплинути на нормальну роботу МРГ; третій

механізм керування еквівалентний колінному суглобу, який відіграє роль з'єднання ножного та великогомілкового суглобів і може виконувати удари ногами, імітуючи рух реальної істоти. З'єднувач, закріплений на третьому механізмі керування до підошовної конструкції, еквівалентний великогомілковій кістці і задній частині ноги реальної істоти, і на з'єднувачі знову встановлені арматурні елементи, що ускладнює деформацію конструкції ноги МРГ під навантаженням. Структура підошви еквівалентна вузлу, який приблизно розглядається як точковий контакт із поверхнею, щоб запобігти ковзанню МРГ під час виконання руху (див. рис. 1.6).

1.2.4 Проєктування конструкції МРГ

Загальний вигляд МРГ представлено на рисунку 1.7, який включає в себе форму самої конструкції і структуру ніг. Опора складаються з 6 окремих ніг, що імітують ноги реальної істоти, а корпус також виконано відповідно до форми тіла істоти: довгий еліпс і кругла голова, що відповідає принципу структури і науки функціональної біоніки [16-18].

Розподіл ніг в таких істот по тулубу буває двох видів – або рівномірний або осесиметричний.

Вигляд рівномірного розподілу представлено на рисунку 1.8, навколо корпусу розташований розподіл ніг під певним кутом. Перевага такого виду розподілу полягає в тому, що відстань між ногами велика, і при русі ноги не будуть заважати одна одній.

Недоліком є те, що параметри руху кожної ноги не збігаються в процесі руху, необхідно окремо регулювати і контролювати параметри кожної окремої ноги, що збільшує складність керування МРГ [10-14].

Осесиметричний розподіл представлено на рисунку 1.8, на шести симетричних вершинах тіла. Перевага полягає в тому, що такий вид розподілу дозволяє забезпечити стійкість тіла МРГ, крім того, аналіз ходи МРГ дозволяє в

основному узгодити параметри руху 6 ніг, що робить керування всім МРГ відносно простим. Головним недоліком є те, що простір руху 6 окремих ніг обмежений, і вони легко заважають одна одній при виконанні руху.

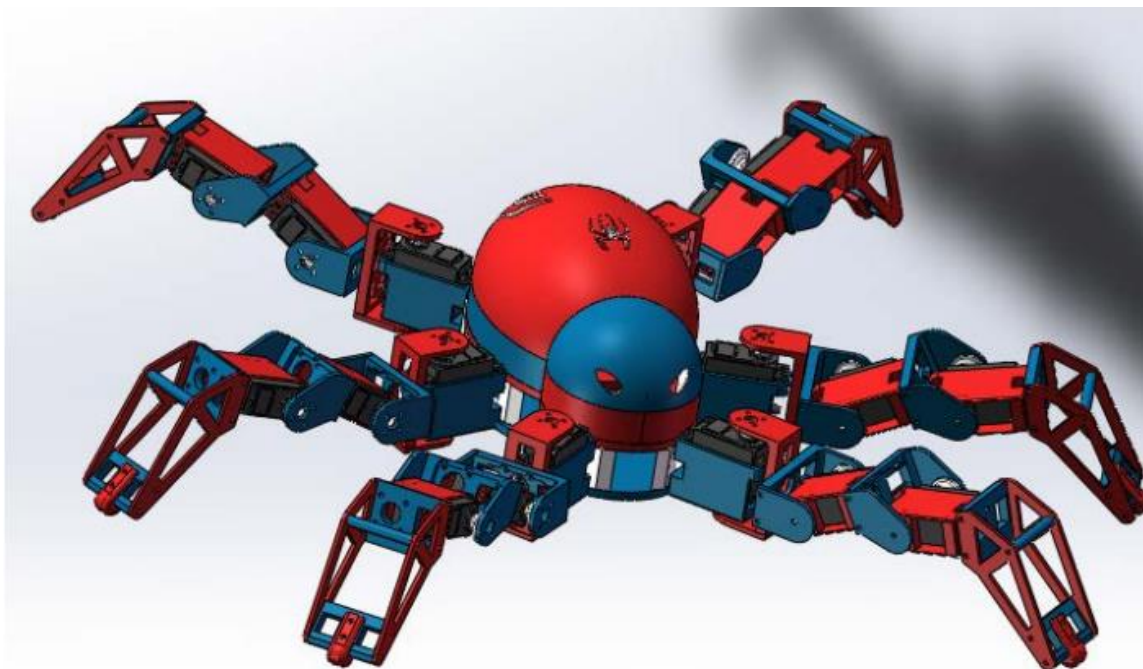


Рисунок 1.7 – Загальна структура МРГ з рівномірним розподілом ніг

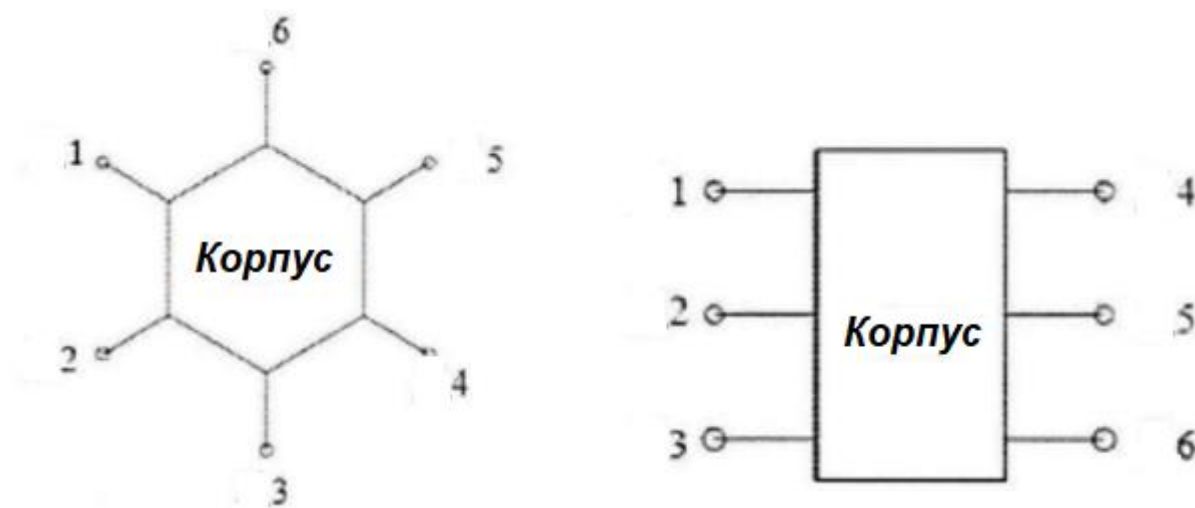


Рисунок 1.8 – Рівномірний та осесиметричний розподіл ніг МРГ

Відповідно до аналізу і порівняння вищезазначених двох видів руху, наша конструкція буде використовувати осесиметричний метод в якості основи,

покращує форму корпусу і використаємо форму еліпса, щоб зменшити взаємні перешкоди між окремими ногами, як представлено на рисунку 1.9.

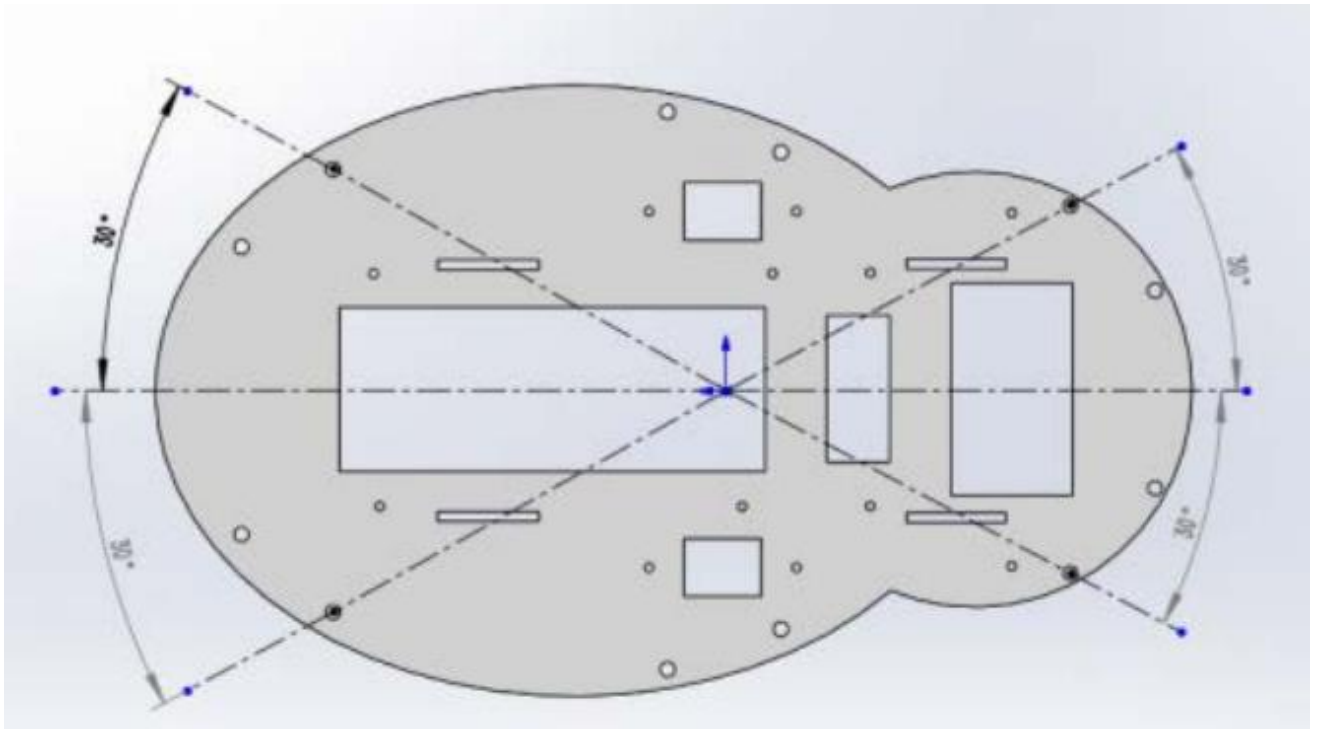


Рисунок 1.9 – Загальний вигляд основи для корпусу МРТ

1.3 Висновки до першого розділу

Описано проблематику проектування моделі мехатронного робота-гексапода, для чого виконано огляд та аналіз існуючих конструкцій мехатронних роботів-гексаподів. Базуючись на проведену аналітичному огляді було проведено розроблення конструкції механічної частини моделі мехатронного робота-гексапода.

2 МОДЕЛЮВАННЯ КОНСТРУКЦІЇ МЕХАТРОННОГО РОБОТА-ГЕКСАПОДА

2.1 Перевірка загальних параметрів і міцності ключових конструкцій мехатронного робота-гексапода

Загальний розмір МРГ складає 25 см у довжину, 20 см у ширину та 15 см у висоту. Максимальне подовження кожної ноги не буде перевищувати 27 см, як представлено на рисунку 1.7.

Реальні загальні параметри МРГ представлені в табличному вигляді.

Таблиця 2.1 – Загальні параметри конструкції МРГ

Назва деталі	Розмір деталі, см	Вага деталі, г
Корпус	20x12	500
Базовий вузол	7	85
Вузол стегна	7	85
Гомілка	5,5	90
Загальне	20x20x15	2000

Робоче середовище даного проєкту МРГ являє собою більш складну місцевість, тому необхідно перевірити міцність ключових конструкцій. Далі виконаємо статичний аналіз крутного моменту приводу з'єднання МРГ, а потім виконаємо перевірку міцності ключової конструкції з використанням програмного середовища ANSYS [19-25].

Виберемо середню частину ноги і корпус МРГ для проведення аналізу, як представлено на рисунку 2.1, виконаємо аналіз сили і моментів в трьох точках А, В і С [26-30].

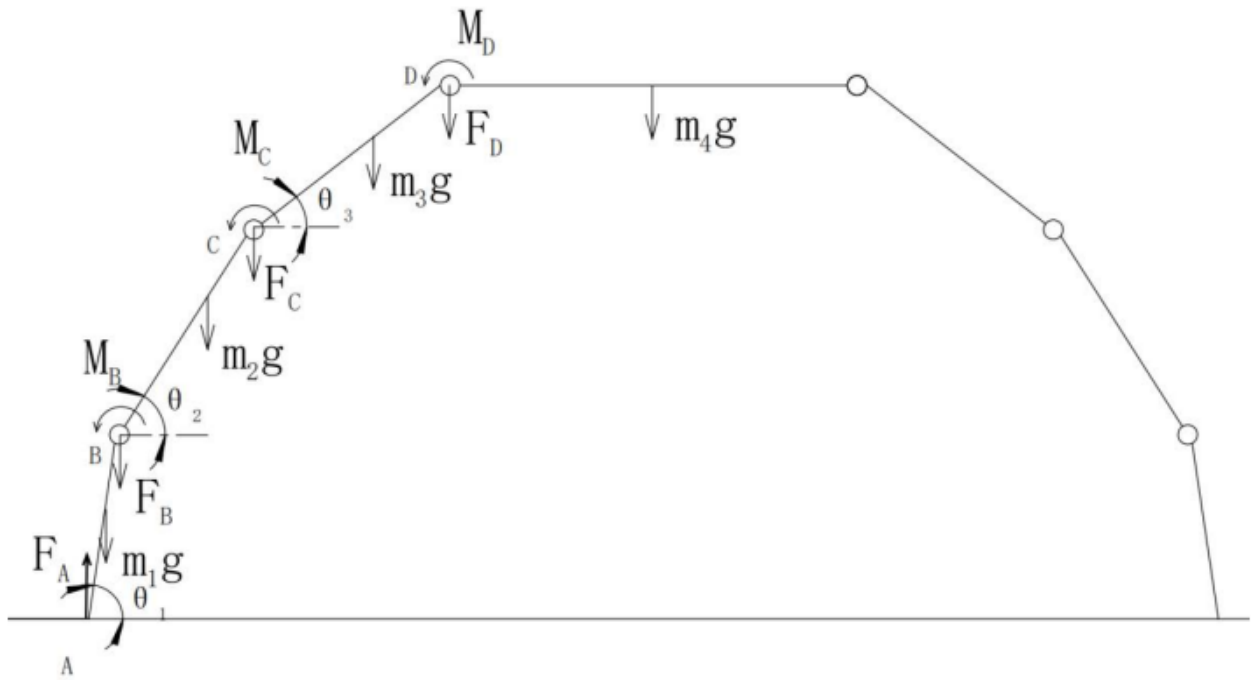


Рисунок 2.1 – Фізична модель механічного аналізу ноги та корпусу МРГ

Згідно рисунку 2.1 для точки А запишемо рівняння [26-30]:

$$F_A = m_1 \cdot g + F_B; \quad (2.1)$$

$$\frac{1}{2} M_1 \cdot G \cdot \cos 1 + F_B \cdot L_1 \cdot \cos 1 = M_B. \quad (2.2)$$

Аналогічно для точки В [26-30]:

$$F_B = m_2 \cdot g + F_C; \quad (2.3)$$

$$\frac{1}{2} M_2 \cdot G \cdot \cos 2 + F_C \cdot L_2 \cdot \cos 2 = M_C. \quad (2.4)$$

Тоді для точки С [26-30]:

$$F_C = m_3 \cdot g + F_D; \quad (2.5)$$

$$\frac{1}{2} M_3 \cdot G \cdot \cos 3 + F_D \cdot L_3 \cdot \cos 3 = M_D; \quad (2.6)$$

$$F_D = \frac{1}{2} m_4 \cdot g. \quad (2.7)$$

Виконавши підстановку рівнянь отримаємо [26-30]:

$$F_A = m_1 \cdot g + m_2 \cdot g + m_3 \cdot g + \frac{1}{2} m_4 \cdot g; \quad (2.8)$$

$$F_B = m_2 \cdot g + m_3 \cdot g + \frac{1}{2} m_4 \cdot g; \quad (2.9)$$

$$M_B = \frac{1}{2} m_1 \cdot g \cdot \cos 1 + m_2 \cdot g \cdot L_1 \cdot \cos 1 + m_3 \cdot g \cdot L_1 \cdot \cos 1 + \frac{1}{2} m_4 \cdot g \cdot L_1 \cdot \cos 1; \quad (2.10)$$

$$F_C = m_3 \cdot g + \frac{1}{2} m_4 \cdot g; \quad (2.11)$$

$$M_C = \frac{1}{2} m_2 \cdot g \cdot \cos 2 + m_3 \cdot g \cdot L_2 \cdot \cos 2 + \frac{1}{2} m_4 \cdot g \cdot L_2 \cdot \cos 2; \quad (2.12)$$

$$M_D = \frac{1}{2} m_3 \cdot g \cdot \cos 3 + m_4 \cdot g \cdot L_3 \cdot \cos 3, \quad (2.13)$$

де M_B, M_C, M_D – це приводний крутний момент, необхідний для кожного з'єднання ноги МРГ.

Підставивши табличні значення характеристик МРГ у формули отримаємо одноногу статичну модель МРГ представлену на рисунку 2.2. Використавши програмне середовище ANSYS [19-25] для імпорту деталей і виконання аналізу міцності, отримаємо діаграму деформації, представлену на рисунку 2.3.

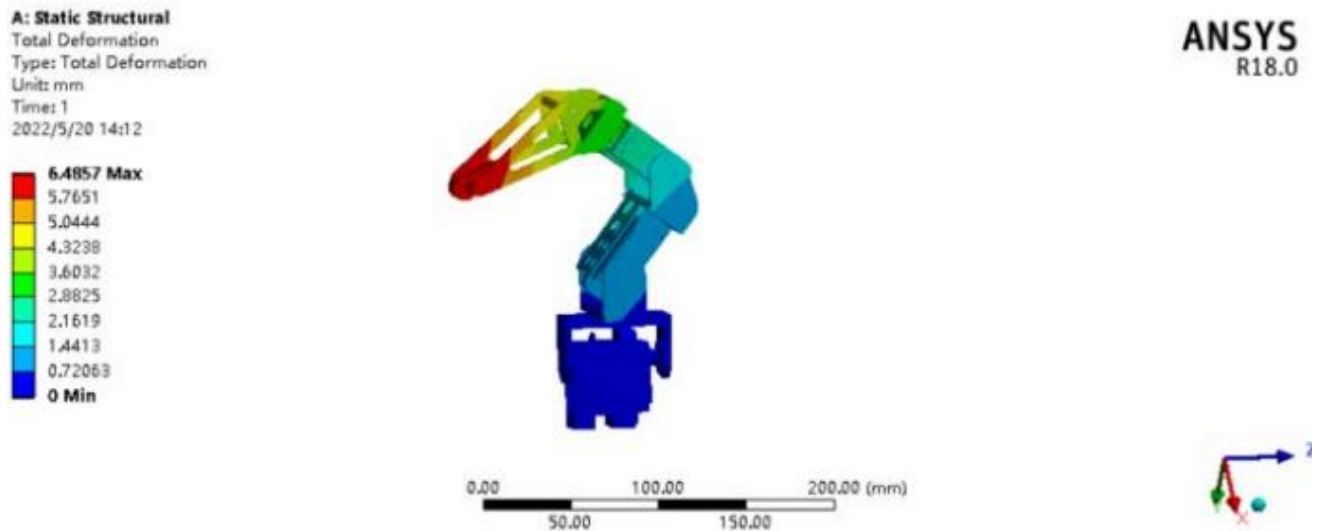


Рисунок 2.2 – Загальний вигляд статичної моделі МРГ

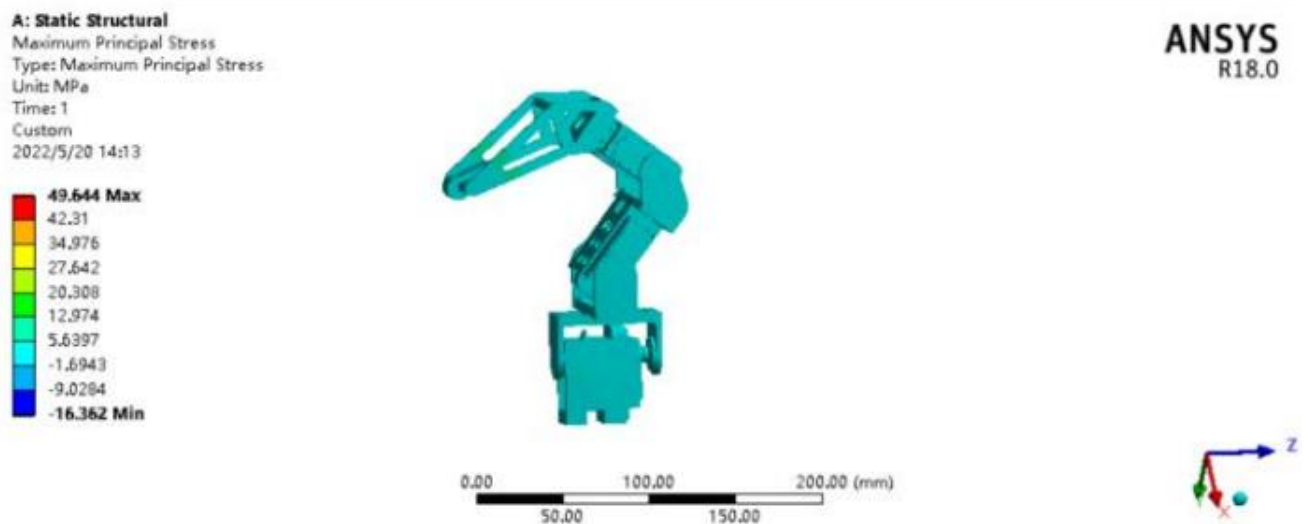


Рисунок 2.3 – Діаграма деформації моделі МРГ

Відомо, що нога МРГ буде виготовлятися методом 3D-друку із матеріалу PLA [31], міцність на розтягнення складає 40-60 МПа. Аналіз методом кінцевих

елементів конструкції однієї ноги МРГ показав, що максимальне напруження складає не більше 49,644 МПа, так що проектні параметри кожної частини задовольняють встановленим вимогам і мають достатній залишок, тому даний проєкт відповідає встановленим вимогам до проєктування МРГ.

Процес виготовлення

Виготовлення МРГ буде виконуватись в наступному порядку:

– моделювання МРГ – для створення моделі буде використано програмний комплекс SolidWorks [32-35], що належить компанії Dassault [36]. У порівнянні з іншим програмним забезпеченням, наприклад, UG компанії Siemens [37], дизайн програмного комплексу SolidWorks більш зручний та простий. Система також має можливості моделювання і розрахунку методом кінцевих елементів, тому програмний комплекс SolidWorks і було обрано для роботи;

– наступний крок вибір і обробка матеріалів, для цього знадобляться 3D-друк, верстати з ЧПК, лазерне різання та інше. Якщо в якості матеріалу обрати з метал, тоді необхідно спеціалізоване обладнання для різання і лазерного різання та застосування верстатів з ЧПК. Проте, ціна такого матеріалу відносно висока, а вага конструкцію буде більшою.

Тому в даній конструкції буде використано спеціальний нейлоновий матеріал для 3D-друку і лазерне гравіювання акрилових пластин, верхня і нижня пластини (основний ствол) МРГ будуть виготовлені за допомогою використання лазерного гравіювання акрилових пластин, а інші механічні деталі – із нейлонового 3D-друку. Матеріал більш легкий і дешевий. На рисунку 2.4 представлено схему процесу 3D-друку корпусу МРГ, а на рисунку 2.5 представлено процес лазерної різки нижньої пластини корпусу МРГ.

Виконано статичний аналіз, та перевірено міцність ключових конструкцій за допомогою програмного середовища ANSYS [19-25] для відповідності встановленим проєктним вимогам, щоб отримати остаточний план проєкту.



Рисунок 2.4 – Загальний вигляд процесу 3D-друку корпусу МРГ



Рисунок 2.5 – Загальний вигляд процесу лазерної різки нижньої пластини корпусу МРГ

2.2 Аналіз ходи мехатронного робота-гексапода

У світі майже всі реальні істоти із багатьма ногами рухаються за допомогою координації між власними ногами такий тип руху носить назву – хода. Теоретично існують різні типи ходи, але не кожен тип ходи дозволяє МРГ ефективно і стабільно рухатися, тому необхідно проаналізувати ходу МРГ і визначити відповідну схему ходи. Загально відомо, що кожна хода володіє власним темпом тому необхідно виконати аналіз типу ходи і визначити схеми для МРГ шляхом порівняння реальних істот і багатьма ногами наприклад, павуків та мурах.

Відомо, що мобільні роботи колісного типу можуть рухатися лише на відносно рівній поверхні, мобільні роботи із рухом за допомогою ніг можуть виконувати підйом ніг та, відповідно, рухатися по нерівній поверхні. Для подолання перешкоди можливо виконати відповідний рух за допомогою підйому ноги, щоб подолати перешкоду. Володіючи високою адаптивністю до зміни рельєфу, оскільки даний проєкт призначений для виявлення в недоступних і непрацездатних середовищах, наприклад, якщо будівлі були пошкоджені в результаті пожежі, рух за допомогою використання ніг є найкращим варіантом для проєкту [10-14].

Однак було виявлено, що проблема регулярності ходи МРГ, що рухається за допомогою використання ніг, знаходиться на досліджуваній поверхні, і немає точного та стабільного рішення для роботи МРГ з достатньо пересіченими поверхнями.

Ґрунтуючись на спостереженнях реальних істот із багатьма ногами і огляді літературних джерел, виконаємо аналіз кілька типів ходи, які зазвичай використовуються мобільними МРГ, і виконує експерименти з множинними рухами ходи [38].

Виконаємо необхідний аналіз робочого циклу.

Початковий робочий цикл відноситься до частки періоду часу вмикання

відносно загального періоду часу в імпульсному циклі [38]. У нашому випадку робочий цикл виражається як відношення кількості ніг МРГ, піднятих в циклі виконання руху, до кількості всіх ніг всього МРГ. Формула виглядає наступним чином: кількість тільки піднятих ніг відноситься до кількості всіх ніг МРГ.

Згідно з формулою, можливо визначити, що бажаний діапазон робочого циклу знаходиться в проміжку $(0, 6/6)$, і різні значення позначатимуть використання різного типу ходи.

При діапазоні робочого циклу $(0, 3/6)$, шість загальних ніг МРГ еквівалентні тому, що більше трьох ніг одночасно не мають контакту із поверхнею, при гойданні менше як три ноги не стикаються з поверхнею, а також можливо, що жодна нога МРГ не стикається з поверхнею, як у вільному стані, в цьому стані МРГ рухається надзвичайно швидко, але не має загальної стійкості та сейсмостійкості.

При діапазоні робочого циклу $3/6$, три ноги з загальних шести ніг МРГ в кожен момент не стикаються із поверхнею, при хитанні три ноги стикаються із поверхнею, даний стан називається рухом у триногій ході, в цьому стані МРГ відносно стабільний і швидкий. Тринога хода досить часто використовується в шестиногих МРГ.

При діапазоні робочого циклу в проміжку $(3/6, 4/6)$ три з загальних шести ніг МРГ не стикаються із поверхнею, при гойданні три ноги стикаються із поверхнею, але в цей час одна з ніг збирається стикнутися із поверхнею в стадії гойдання. У цьому стані МРГ не підходить для ні для аналізу ні для контролю.

Коли діапазон робочого циклу дорівнює $4/6$, дві з загальних шести ніг МРГ в кожен момент не стикаються із поверхнею, а при гойданні чотири ноги стикаються із поверхнею. Даний стан носить назву рух чотириновою ходою, МРГ більш стабільний, ніж МРГ з триновою ходою, але зі рух виконується із зменшеною швидкістю.

При діапазоні робочого циклу в проміжку $(4/6, 5/6)$ дві з загальних шести ніг МРГ не стикаються із поверхнею, при гойданні чотири ноги стикаються із

поверхнею, але в цей час одна з ніг збирається стикнутися із поверхнею в стадії гойдання ноги, в даному стані МРГ не підходить ні для аналізу ні для контролю.

При діапазоні робочого циклу $5/6$, одна з загальних шести ніг МРГ в кожен момент циклу не стикається із поверхнею, при гойданні п'ять ніг стикаються із поверхнею, даний стан носить назву рух п'ятиногої ходи, в даному стані МРГ більш стабільний, ніж МРГ із триногою і чотириногою ходою, але значно знижує швидкість руху МРГ в цілому.

При діапазоні робочого циклу в проміжку $(5/6,1)$ одна з шести ніг МРГ не стикається із поверхнею, при гойданні п'ять ніг стикаються із поверхнею, але при цьому одна з них збирається стикнутися із поверхнею в стадії гойдання, в цьому стані ефективність руху МРГ дуже низька, практично немає можливості виконувати рух.

Коли діапазон робочого циклу рівний одиниці, всі шість ніг МРГ стикаються з поверхнею в кожен момент циклу, і немає ніг, що гойдаються. Такий стан носить назву стаціонарний стан.

З вищевказаного видно, що чим ближче значення діапазону до одиниці, тим більше опорних ніг МРГ до поверхні, тим вища стійкість, проте тим нижча швидкість.

Відповідно до виконаного аналізу переваг і недоліків вказаних восьми типів ходи, з них обрано для подальшого дослідження три типи ходи із робочими циклами $3/6$, $4/6$, $5/6$ [10-14, 38].

2.3 Дослідження варіантів ходи мехатронного робота-гексапода

Після описаного вище аналізу робочого циклу для дослідження було обрано три варіанти ходи з робочими циклами $3/6$, $4/6$ і $5/6$ [10-14, 38].

Хода МРГ при робочому циклі, що складає $3/6$.

Коли робочий цикл дорівнює $3/6$, МРГ має як мінімум три ноги, які не стикаються з поверхнею в кожен момент, а при виконанні руху три ноги

стикаються з поверхнею. Як вже вказувалось даний стан носить назву рух триноги ходи. Тринога хода володіє певною стійкістю і не низьким рівнем швидкості руху, вона також є найбільш популярною ходою в світі МРГ [38].

На рисунку 2.6 представлена фазова діаграма положення ніг при виконанні триногої ходи МРГ. Білий колір на рисунку позначає фазу виконання руху, чорний позначає фазу контакту ноги з поверхнею. Тринога хода володіє стабільністю, тому що МРГ має три ноги, що контактують із поверхнею в кожен момент руху, утворюючи стійку трикутну опорну конструкцію з корпусом МРГ. Тим часом, три ноги, які в повітрі, стають ногами, які стикаються з поверхнею після підняття, виконання руху і приземлення, попередні три ноги, що стикаються з поверхнею, піднімаються в повітря ноги, утворюючи іншу трикутну опорну конструкцію і так далі. Така хода дозволяє рівномірно розділити шість ніг на дві групи, рух кожної окремої ноги є одиничним циклом, керування також відносно просте, тому воно достатньо широко використовується.



Рисунок 2.6 - Фазова діаграма положення ніг триногої ходи МРГ

Характеристики виконання прямого руху вперед, назад і повороту є основними вимогами для конструкції МРГ, тому виконаємо аналіз прямого руху вперед, назад і виконання повороту при використанні триногої ходи.

На рисунку 2.7 представлено схематичне зображення виконання триногої

ходи МРГ вперед та назад. Форма, утворена перетином двох еліпсів, представляє основний корпус МРГ, відрізок прямої лінії представляє ногу, чорне коло на кінці лінії представляє собою підошву стопи, що стикається із землею, а коло білого кольору на кінці лінії представляє підошву ноги, що піднімається:

– перший етап (див. рис. 2.7) – початковий стан МРГ, в цей момент часу ноги 1, 2, 3, 4, 5 і 6 знаходяться в контакті з поверхнею;

– другий етап – перша стадія руху МРГ, в цей момент часу ноги 2, 4, 6 – контакт із поверхнею, ноги 1, 3, 5 – підйом і поворот ніг вперед на півкроку, потім приземлення, корпус МРГ рухається вперед, ноги 2, 4, 6 – контакт з поверхнею;

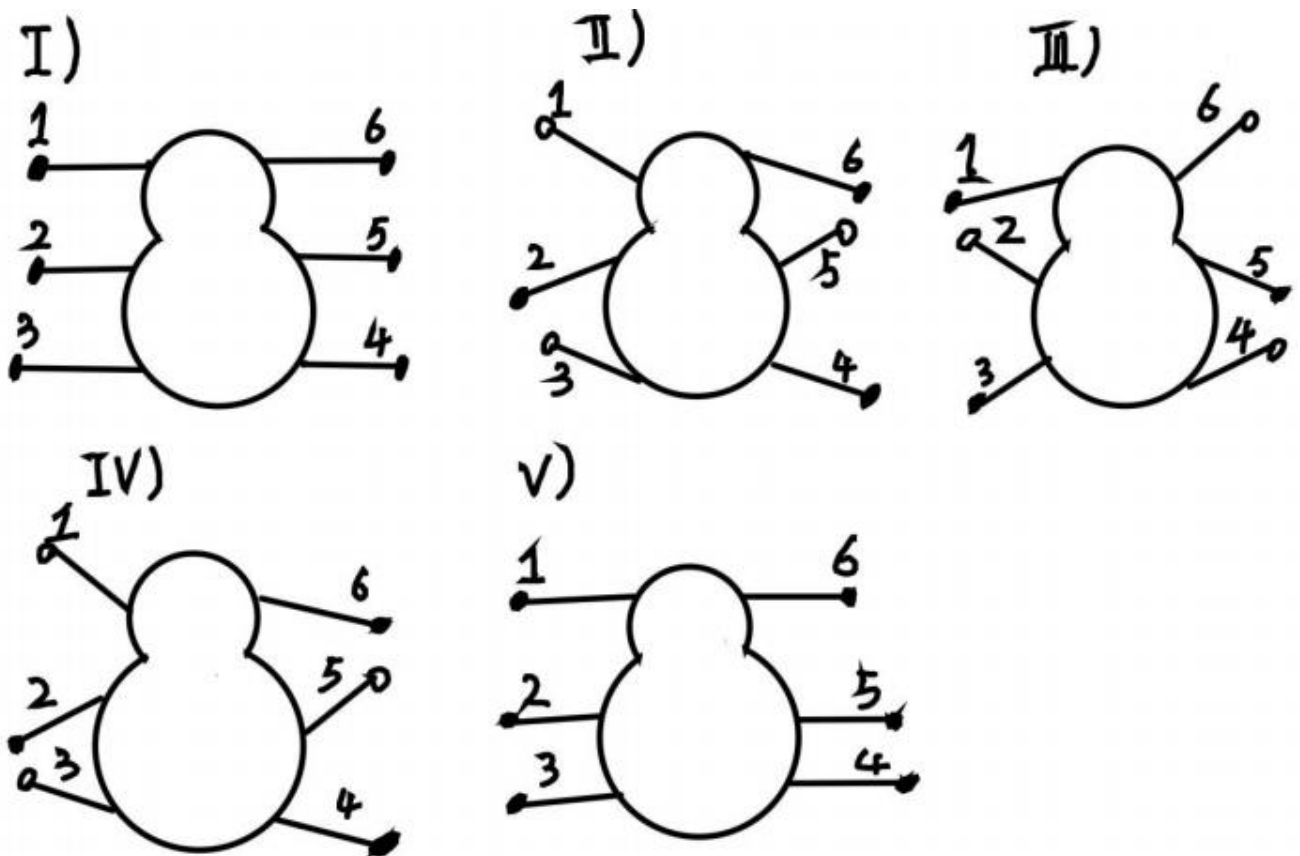


Рисунок 2.7 – Схематичне зображення виконання МРГ триногої ходи вперед

– третій етап – представляє другу стадію руху МРГ, в цей момент часу ноги 1, 3, 5 отримують контакт із поверхнею, ноги 2, 4, 6 починають підйом і

поворот ноги вперед на крок, потім приземлення, весь корпус рухається вперед, ноги 1, 3, 5 мають контакт з поверхнею;

– четвертий етап – являє собою третю стадію руху МРГ, в цей момент часу ноги 2, 4, 6 мають контакт із поверхнею, ноги 1, 3, 5 виконують рух підйому і повороту ноги вперед на півкроку, потім приземлення, весь корпус рухається вперед, ноги 2, 4, 6 знаходяться в контакті з поверхнею і збираються виконувати рух;

– п'ятий етап – показує, що МРГ зупиняється, повертається в початковий стан і завершує цикл руху вперед.

Принцип руху назад аналогічний принципу руху МРГ прямо вперед [38].

На рисунку 2.8 представлено схематичне зображення ходи МРГ вправо (виконання повороту за годинниковою стрілкою):

– перший етап (рис. 2.8) – вихідний стан МРГ, при якому ноги 1 - 6 стикаються із поверхнею;

– другий етап – перша стадія повороту МРГ вправо (виконання повороту за годинниковою стрілкою), в цей час ноги 2, 4, 6 мають контакт із поверхнею, ноги 1, 3, 5 виконують рух і поворот ноги за годинниковою стрілкою під тим самим кутом, потім приземлення, весь корпус повертається на заданий кут, ноги 2, 4, 6 мають контакті із поверхнею;

– третій етап – друга стадія повороту МРГ вправо (виконання повороту за годинниковою стрілкою), при цьому ноги 1, 3, 5 – контакт із поверхнею, ноги 2, 4, 6 – підйом і поворот за годинниковою стрілкою під тим же кутом, потім приземлення, весь корпус повертається під заданим кутом, ноги 1, 3, 5 – контакт із поверхнею;

– четвертий етап – остання стадія повороту МРГ вправо (виконання повороту за годинниковою стрілкою), зупинка повороту, повернення у вихідний стан і завершення циклу руху повороту із фіксованою точкою.

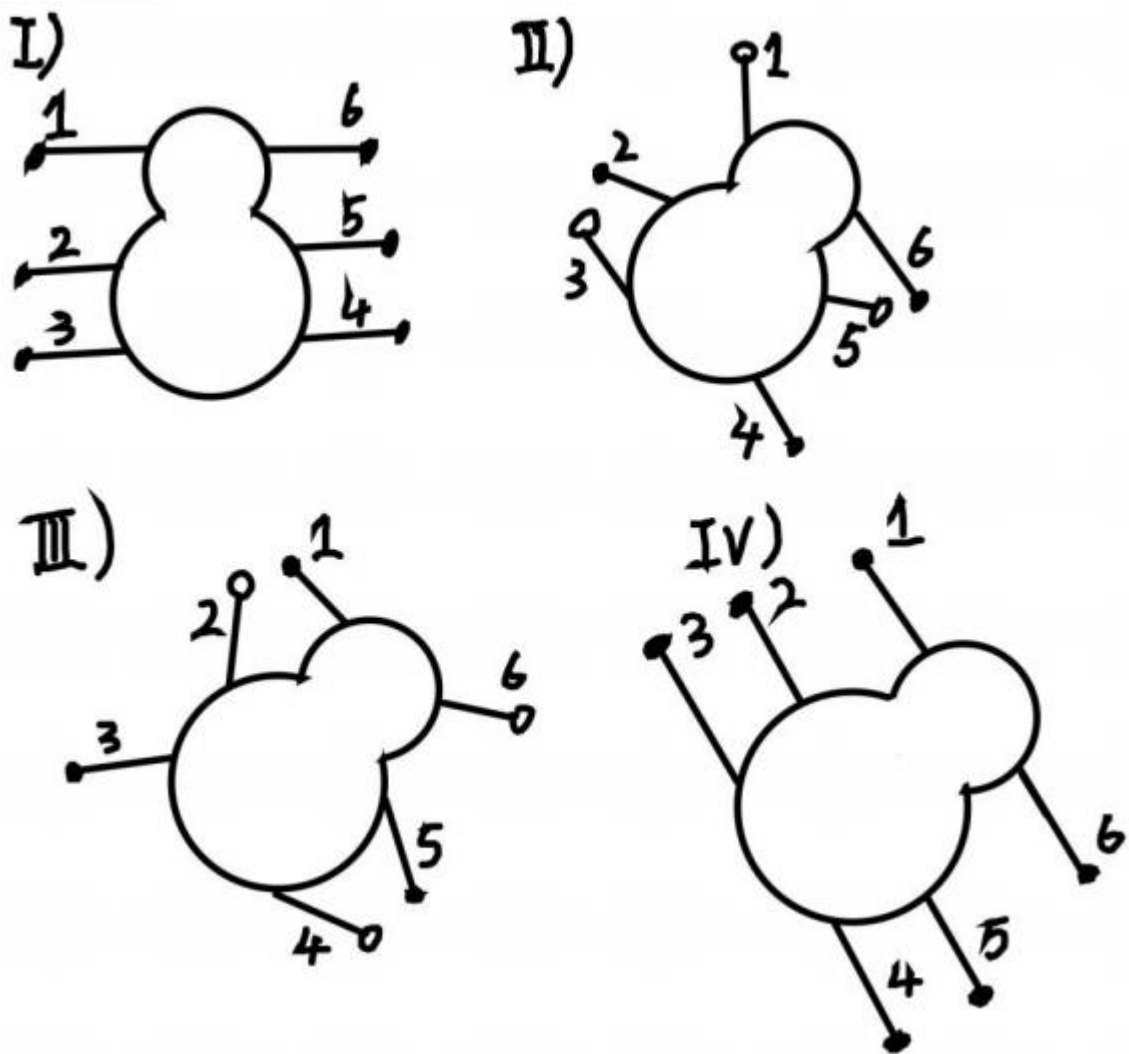


Рисунок 2.8 – Схематичне зображення виконання повороту з фіксованою точкою в триногій ході МРГ

План виконання триногої ходи повороту вліво МРГ (виконання повороту проти годинникової стрілки) аналогічний із планом повороту вправо МРГ (за годинниковою стрілкою).

Хода МРГ при робочому циклі, що складають 4/6 [38].

Коли робочий цикл складає 4/6, шість ніг МРГ діляться на дві ноги, які не стикаються з поверхнею в кожен момент, а при виконанні руху чотири ноги стикаються з поверхнею. Такий стан називається рухом ходи чотириногих істот.

З точки зору стабільності виконання руху, чотиринога хода відносно більш стабільна, ніж тринога хода, але з точки зору рівня швидкості чотиринога хода

поступається триногій ході і має відносно нижчий рівень загальної швидкості.

Чотиринога хода рахується більш стабільною, тому що МРГ має як мінімум чотири ноги, що контактують із поверхнею в кожен момент руху, і разом з корпусом МРГ утворює стійку чотирикутну опорну конструкцію. У той же час дві ноги після підйому, виконання руху і приземлення стають ногами, що стикаються із поверхнею, а дві з чотирьох інших ніг, що стикаються із поверхнею, піднімаються, щоб стати ногами які виконують рух, утворюючи іншу чотирикутну опорну конструкцію і так далі. Така чотири хода може виконуватись шістьма ногами з різними типами комбінацій, що складніше контролювати, ніж виконання триногій ходи (рис. 2.9).



Рисунок 2.9 – Фазова діаграма положення ніг чотириногій ходи МРГ

Проаналізуємо виконання прямої ходи вперед, назад і поворот чотириногій ходи МРГ.

На рисунку 2.10 представлено схематичне зображення чотириногій ходи МРГ вперед і назад. Форма, утворена перетином двох еліпсів, представляє основний корпус МРГ, відрізок прямої лінії представляє ногу, коло чорного кольору на кінці лінії представляє підошву стопи, що стикається із поверхнею, а коло білого кольору на кінці лінії представляє підошву ноги, що виконує рух:

– перший етап (див. рис. 2.10) – вихідний стан МРГ, при цьому ноги 1 - 6 стикаються із поверхнею;

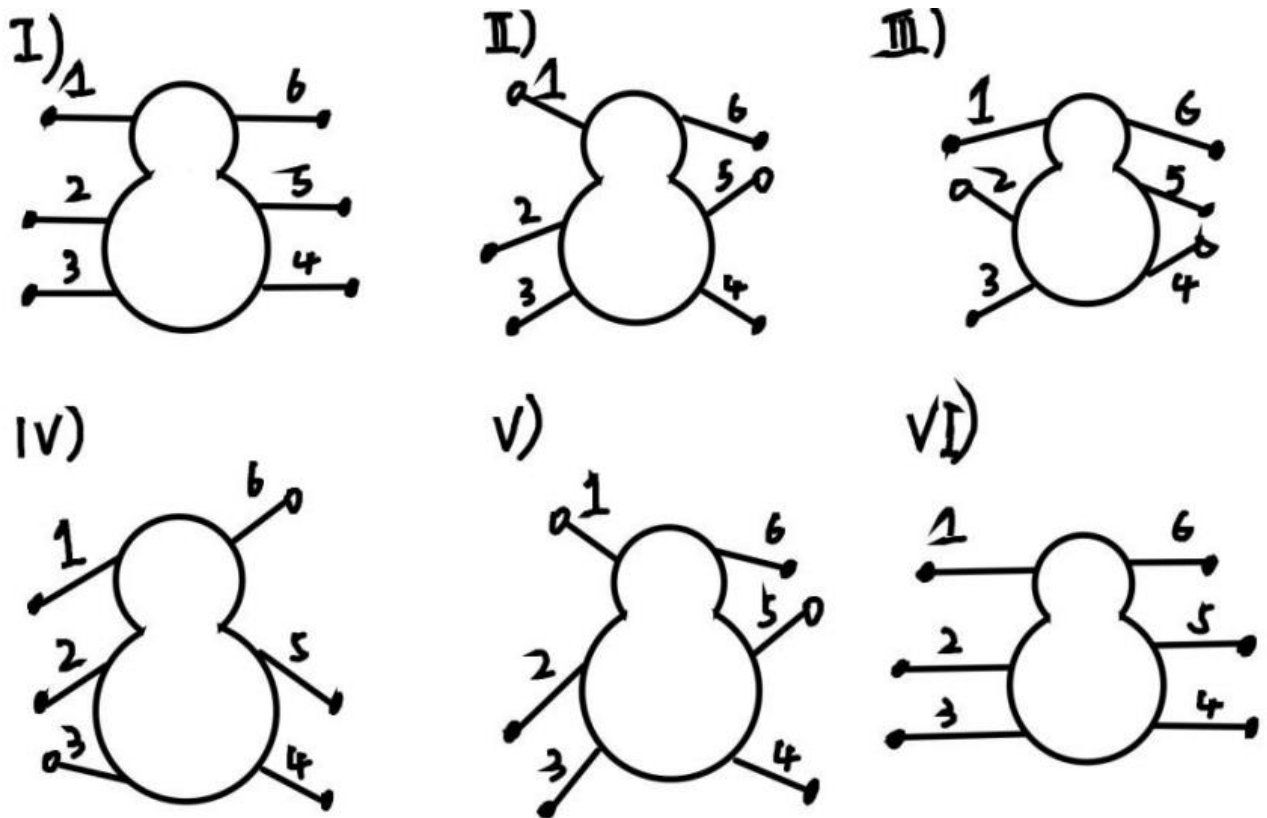


Рисунок 2.10 – Схематичне зображення лінійного руху чотиринової ходи МРГ

– другий етап – перша стадія просування МРГ, в цей час ноги 2, 3, 4, 6 мають контакт із поверхнею, ноги 1, 5 виконують рух і поворот вперед на півкроку, потім приземлення, весь корпус МРГ рухається вперед, ноги 2, 3, 4, 6 контактують з поверхнею, ноги 2, 4 будуть виконувати рух;

– третій етап – представляє другу стадію виконання руху МРГ, в цей час ноги 1, 3, 5, 6 мають контакт із поверхнею, ноги 2, 4 виконують рух і поворот вперед на крок, потім приземлення, весь корпус МРГ рухається вперед, ноги 1, 3, 5, 6 контактують з поверхнею, ноги 3, 5 будуть виконувати рух;

– четвертий етап – представляє третю стадію виконання руху МРГ, в цей час ноги 1, 2, 4, 6 мають контакт із поверхнею, ноги 3, 5 виконують рух і поворот вперед на крок, потім приземлення, весь корпус МРГ рухається вперед, ноги 1, 2, 4, 6 контактують з поверхнею, ноги 1, 5 будуть виконувати рух;

– п'ятий етап – четверта стадія виконання руху МРГ, при цьому ноги 2, 3, 4, 6 – контакт із поверхнею, ноги 1, 5 – виконання підйому і повороту вперед на

півкроку, потім приземлення, весь корпус МРГ рухається вперед, ноги 2, 3, 4, 6 – контакт з поверхнею;

– шостий етап – остання стадія виконання руху МРГ, зупинка руху, повернення у вихідний стан і завершення циклу виконання руху вперед МРГ.

Етапи виконання руху назад, є аналогічними як і в разі виконання руху вперед [38].

На рисунку 2.11 представлено схематичне зображення виконання правого повороту (за годинниковою стрілкою) чотиринової ходи МРГ:

– перший етап (рис. 2.11) – вихідний стан МРГ, при якому ноги 1 - 6 стикаються із поверхнею;

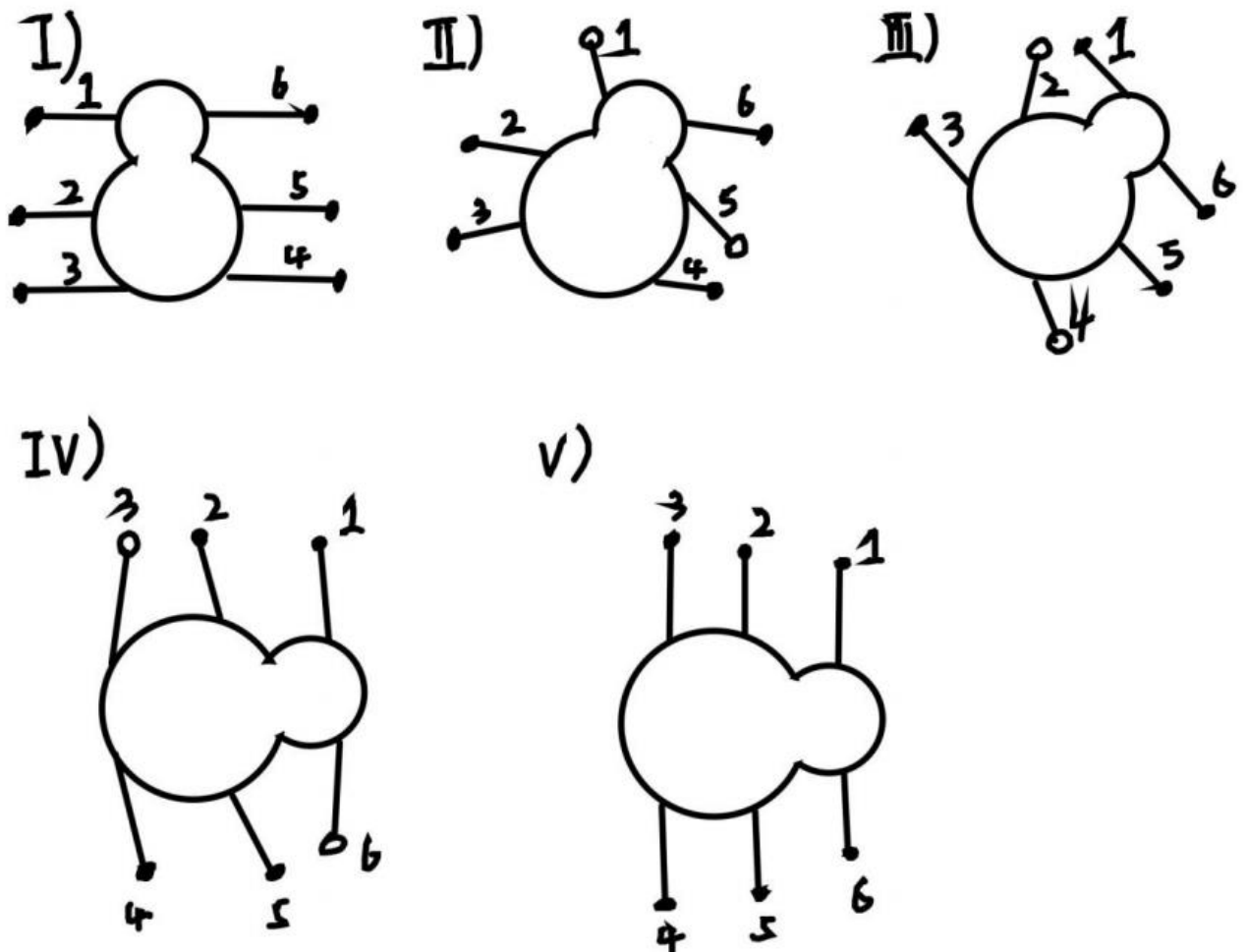


Рисунок 2.11 – Схематичне зображення виконання правого повороту (за годинниковою стрілкою) чотиринової ходи МРГ

– другий етап – перша стадія виконання повороту МРГ вправо (за годинниковою стрілкою), в цей час ноги 2, 3, 4, 6 мають контакт із поверхнею, ноги 1, 5 виконують рух і поворот за годинниковою стрілкою під тим самим кутом, потім приземляються, весь корпус повертається під певним кутом, ноги 2, 3, 4, 6 контактують із поверхнею, ноги 2, 4 будуть виконувати рух;

– третій етап – друга стадія виконання повороту МРГ вправо (за годинниковою стрілкою), при цьому ноги 1, 3, 5, 6 мають контакт із поверхнею, ноги 2, 4 виконують підйом і поворот за годинниковою стрілкою під тим самим кутом, потім приземляються, весь корпус МРГ повертається під певним кутом, ноги 1, 3, 5, 6 контактують із поверхнею, ноги 3, 6 будуть виконувати рух;

– четвертий етап представляє собою третю стадію виконання повороту МРГ вправо (за годинниковою стрілкою), при цьому ноги 1, 2, 4, 5 мають контакт із поверхнею, ноги 3, 6 виконують підйом і поворот за годинниковою стрілкою під тим самим кутом, потім приземляються, весь корпус МРГ повертається під певним кутом, ноги 1, 2, 4, 5 контактують із поверхнею;

– п'ятий етап – остання стадія виконання повороту МРГ вправо (за годинниковою стрілкою), зупинка повороту, повернення в початковий стан і завершення циклу руху повороту із фіксованою точкою.

План виконання чотириноного повороту вліво (проти годинникової стрілки) аналогічний з планом виконання повороту вправо (за годинниковою стрілкою) [10-14, 38].

Виконання руху при робочому циклі, що складає $5/6$ [38].

Коли робочий цикл дорівнює $5/6$, одна з шести ніг МРГ не має контакту із поверхнею в кожен момент виконання руху. При виконанні руху п'ять ніг стикаються з поверхнею – цей стан носить назву п'ятиного хода.

П'ятиного хода, також носить назву хвильова хода, є найбільш стабільною і має найнижчий рівень швидкості із всіх розглянутих варіантів руху.

На рисунку 2.12 представлено фазову діаграму п'ятиногої ходи МРГ. МРГ має п'ять ніг, що контактують із поверхнею в кожен момент виконання руху, і

разом з корпусом МРГ утворює стійку п'ятикутну опорну конструкцію. У той же час, нога, що виконує рух – піднімається, рухається і приземляється, перетворюється на ногу, що стикається із поверхнею, і одна з п'яти інших ніг, що стикаються із поверхнею, піднімається ввєрх, утворюючи іншу п'ятикутну опорну конструкцію і так далі.



Рисунок 2.12 – Фазова діаграма положення ніг п'ятиногої ходи МРГ

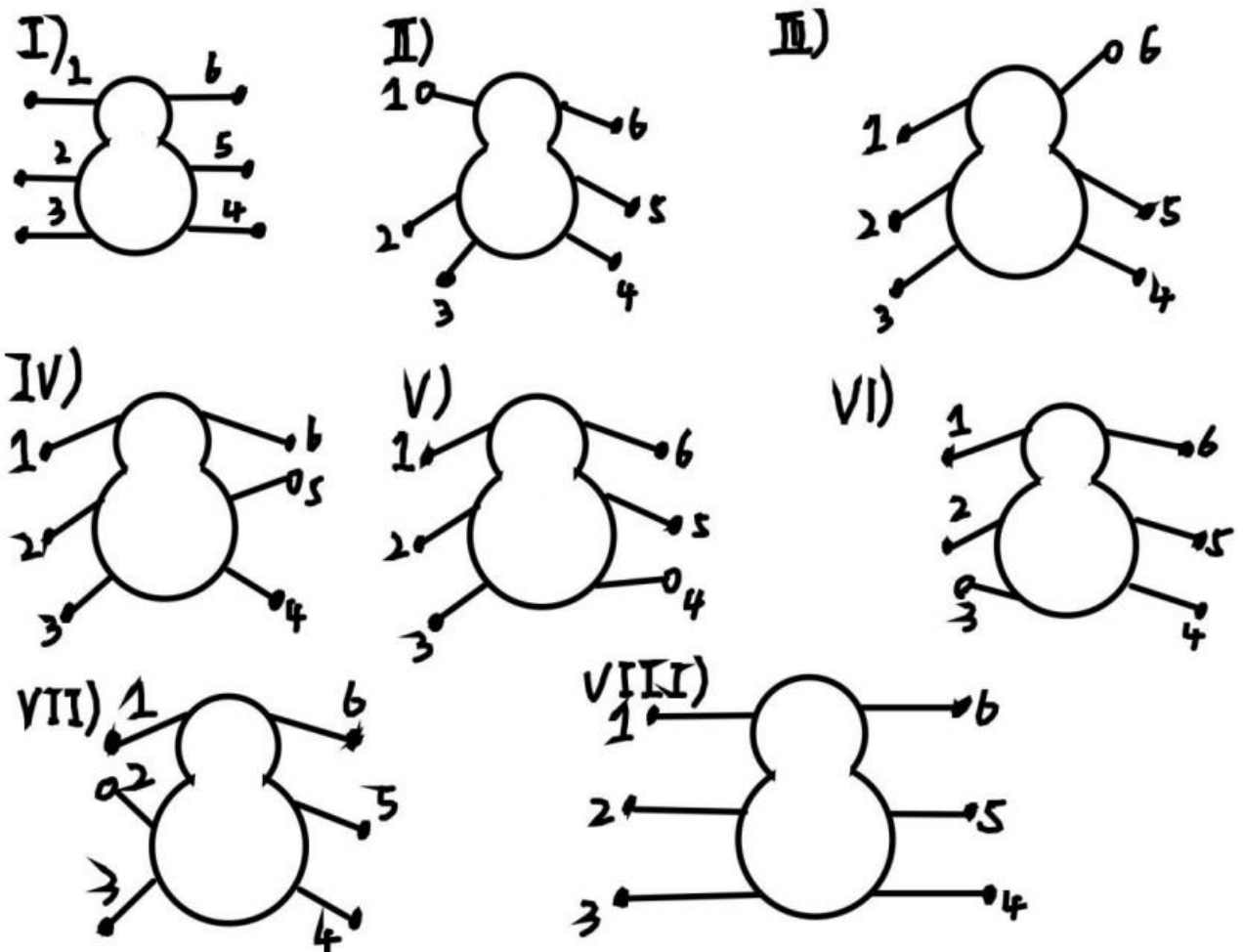


Рисунок 2.13 – Схематичне зображення лінійного руху п'ятиногої ходи МРГ

Проведемо аналіз виконання прямих рухів вперед та назад і виконання поворотів при використанні п'ятиногої ходи МРГ.

На рисунку 2.13 представлено схематичне зображення п'ятиногої ходи МРГ вперед/назад. Форма, утворена перетином двох еліпсів, представляє основний корпус МРГ, відрізок прямої лінії представляє ногу, коло чорного кольору на кінці лінії представляє підошву стопи, що стикається із поверхнею, а коло білого кольору на кінці лінії представляє підошву ноги, що виконує рух.

– перший етап (рис. 2.13) – вихідний стан МРГ, при цьому ноги 1 - 6 стикаються із поверхнею;

– другий етап – перша стадія виконання руху МРГ, в цей час ноги 2 - 6 мають контакт із поверхнею, нога 1 виконує підйом і поворот вперед на півкроку, потім приземлення, весь корпус МРГ рухається вперед, ноги 2 - 6 контактують з поверхнею, нога 6 буде виконувати рух;

– третій етап – представляє собою другу стадію виконання руху МРГ, в цей час ноги 1 - 5 мають контакт із поверхнею, нога 6 виконує підйом і поворот вперед на крок, потім приземлення, весь корпус МРГ рухається вперед, ноги 1 - 5 контактують із поверхнею, нога 5 буде рухатись;

– четвертий етап – представляє собою третю стадію виконання руху МРГ, в цей час ноги 1, 2, 3, 4, 6 мають контакт із поверхнею, нога 5 виконує підйом і поворот вперед на крок, потім приземлення, весь корпус МРГ рухається вперед, ноги 1, 2, 3, 4, 6 контактують із поверхнею, нога 4 буде виконувати рух;

– п'ятий етап – четверта стадія виконання руху МРГ, в цей час ноги 1, 2, 3, 5, 6 – контакт із поверхнею, нога 4 – підйом і поворот вперед на крок, потім приземлення, весь корпус МРГ рухається вперед, ноги 1, 2, 3, 5, 6 контактують із поверхнею, нога 3 буде виконувати рух;

– шостий етап – п'ята стадія руху МРГ, в цей час ноги 1, 2, 4, 5, 6 мають контакт із поверхнею, нога 3 виконує підйом і поворот вперед на крок, потім приземлення, весь корпус рухається вперед, ноги 1, 2, 4, 5, 6 контактують із поверхнею, нога 2 буде виконувати рух;

– сьомий етап – представляє собою шосту стадію виконання руху МРГ, при цьому ноги 1, 3, 4, 5, 6 мають контакт із поверхнею, нога 2 виконує рух і поворот вперед на півкроку, потім приземлення, весь корпус МРГ рухається вперед, ноги 1, 3, 4, 5, 6 контактують із поверхнею.

– восьмий етап – остання стадія виконання руху МРГ, зупинка руху, повернення до вихідного стану і завершення циклу руху вперед.

Етапи на які ділиться рух назад, є аналогічними руху МРГ вперед [38].

На рисунку 2.14 схематично зображено виконання повороту вправо (за годинниковою стрілкою) при використанні п'ятинової ходи МРГ.

– перший етап (див. рис. 2.14) – вихідний стан МРГ, при цьому ноги 1 - 6 мають контакт із поверхнею;

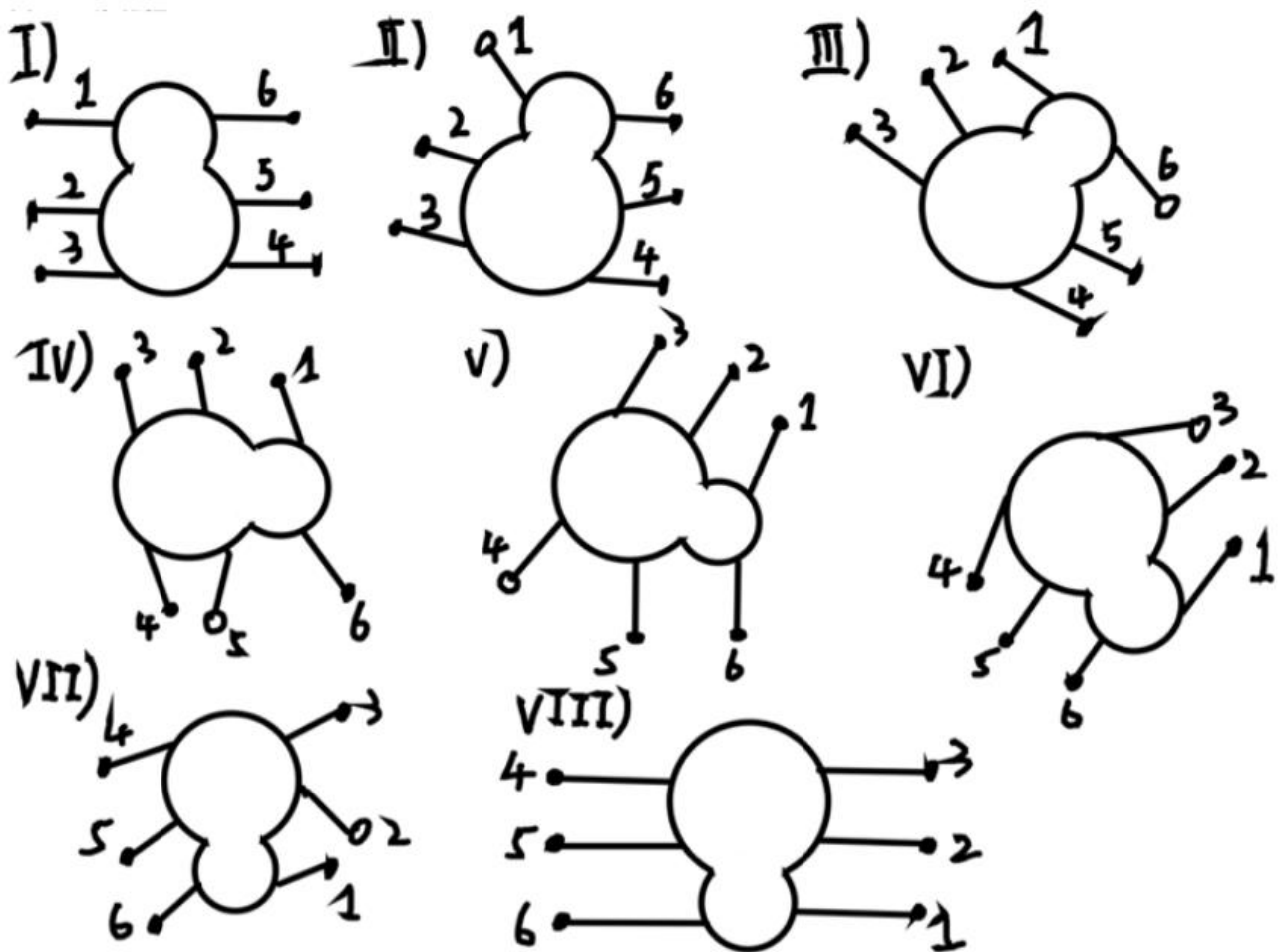


Рисунок 2.14 – Схематичне зображення виконання повороту вправо при використанні п'ятинової ходи МРГ

– другий етап – перша стадія виконання повороту МРГ вправо (за годинниковою стрілкою), при цьому ноги 2 - 6 мають контакт із поверхнею, нога 1 виконує рух і поворот за годинниковою стрілкою під тим же кутом, потім приземляються, весь корпус МРГ повертається під певним кутом, ноги 2 - 6 контактують із поверхнею, нога 6 буде виконувати рух;

– третій етап – друга стадія виконання повороту МРГ вправо (за годинниковою стрілкою), при цьому ноги 1 - 5 мають контакт із поверхнею, нога 6 виконує рух і поворот за годинниковою стрілкою під тим же кутом, потім приземляються, весь корпус повертається під певним кутом, ноги 1 - 5 контактують із поверхнею, нога 5 буде виконувати рух;

– четвертий етап – представляє собою третю стадію виконання повороту МРГ вправо (за годинниковою стрілкою), при цьому ноги 1, 2, 3, 4, 6 мають контакт із поверхнею, нога 5 виконує рух і поворот за годинниковою стрілкою під тим же кутом, потім приземляється, весь корпус повертається під певним кутом, ноги 1, 2, 3, 4, 6 контактують із поверхнею, нога 4 буде виконувати рух;

– п'ятий етап – четверта стадія виконання повороту МРГ вправо (за годинниковою стрілкою), при цьому ноги 1 - 3, 5, 6 – контакт із поверхнею, нога 4 – підйом і поворот за годинниковою стрілкою під тим самим кутом, потім приземлення, весь корпус МРГ повертається під певним кутом, ноги 1 - 3, 5, 6 контактують із поверхнею, нога 3 буде виконувати рух;

– шостий етап – п'ята стадія виконання повороту МРГ вправо (за годинниковою стрілкою), при цьому ноги 1, 2, 4 - 6 мають контакт із поверхнею, нога 3 виконує рух і поворот за годинниковою стрілкою під тим же кутом, потім приземляються, весь корпус повертається під певним кутом, ноги 1, 2, 4 - 6 контактують із поверхнею, нога 2 буде виконувати рух;

– сьомий етап – п'ята стадія виконання повороту МРГ вправо (за годинниковою стрілкою), при цьому ноги 1, 3, 4 - 6 позначає контакт із поверхнею, нога 2 виконує рух і поворот за годинниковою стрілкою під тим же кутом, потім приземляються, весь корпус МРГ повертається під певним кутом,

ноги 1, 3, 4 - 6 контактують із поверхнею;

– восьмий етап – остання стадія виконання повороту МРГ вправо (за годинниковою стрілкою), зупинка повороту, повернення в початковий стан і завершення одного циклу руху повороту із фіксованою точкою.

Виконання повороту вліво (проти годинникової стрілки) аналогічне із виконанням повороту вправо (за годинниковою стрілкою) [10-14, 38].

2.4 Висновки до другого розділу

Виконано моделювання конструкції, проведено перевірку загальних параметрів і міцності ключових конструкцій мехатронного робота-гексапода. Було проведено аналіз виконання руху, виконано дослідження варіантів виконання руху мехатронним роботом-гексаподом.

3 ПІДБІР ОСНОВНИХ ЕЛЕМЕНТІВ ДЛЯ МОДЕЛІ МЕХАТРОННОГО РОБОТА-ГЕКСАПОДА

3.1 Аналітичне дослідження стійкості ходи мехатронних роботів-гексаподів

Рельєф руху, з яким стикається МРГ у даній конструкції, є складним, тому він повинен володіти певною перевагою в стійкості. Для цього необхідно виконати дослідження стійкості трьох видів ходи, який поєднується із заданим рівнем швидкості та іншими аспектами, щоб визначити рішення для даної конструкції та перевірити його на практиці. Існує два методи дослідження стійкості МРГ: динамічний та статичний. Серед них методи статичного дослідження в основному включають методи – межі статичної стійкості, запасу статичної стійкості, запасу енергетичної стійкості тощо.

Методи динамічного дослідження в основному включають методи – запасу динамічної стійкості, критерію стійкості кута сили і точки нульового моменту (ZMP) [39-45].

Дана конструкція МРГ використовує метод статичної стійкої межі в методі статичного дослідження. Граничний метод статичної стійкості використовується для визначення стійкості МРГ. У русі, коли центр ваги всього тіла знаходиться в замкнутій області, утвореній точками стопи ноги, що стикається із поверхнею, МРГ буде вважатись стійким, в іншому випадку МРГ буде перекидатися і ковзати.

На рисунках 3.1 – 3.6 представлено аналітичне дослідження граничного методу статичної стійкості трьох видів виконання ходи, обраних при аналізі робочого циклу вище.

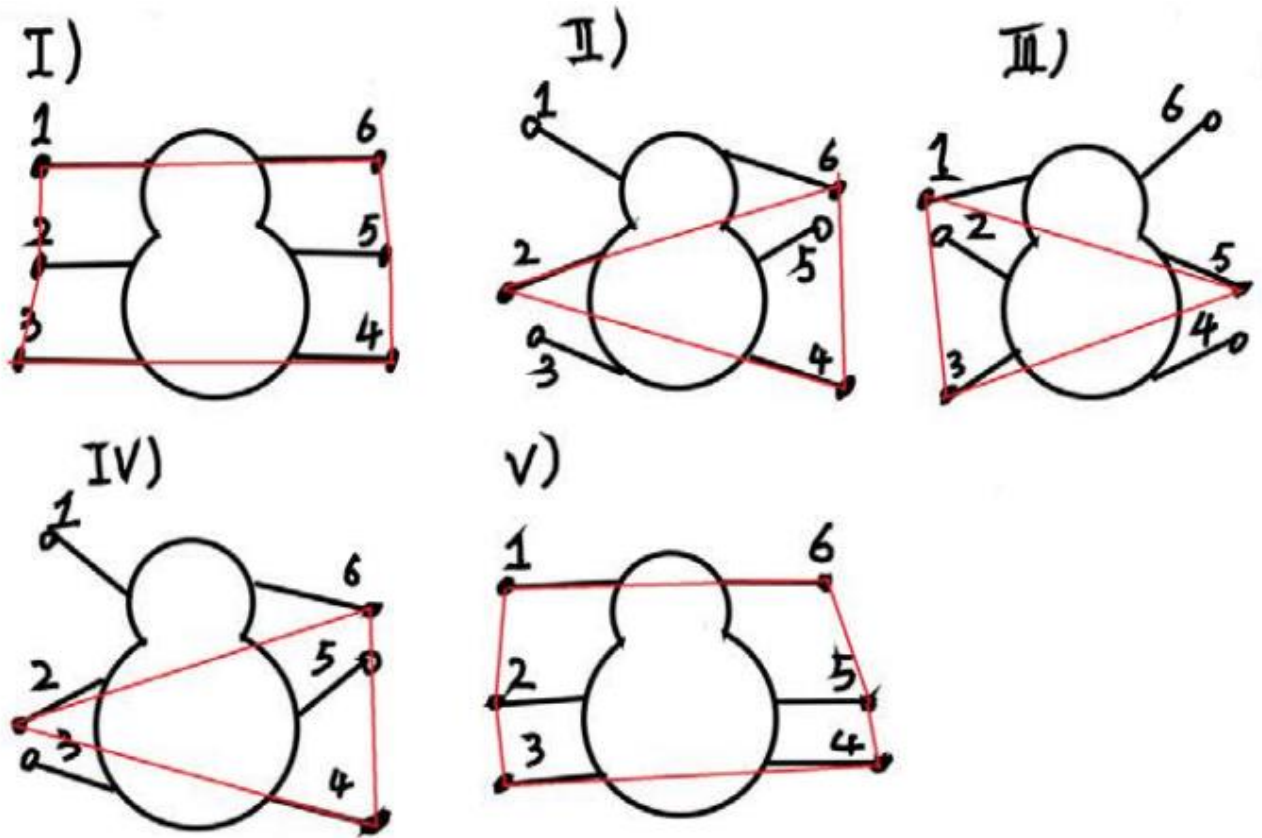


Рисунок 3.1 – Виконання руху по прямій лінії, тринога хода

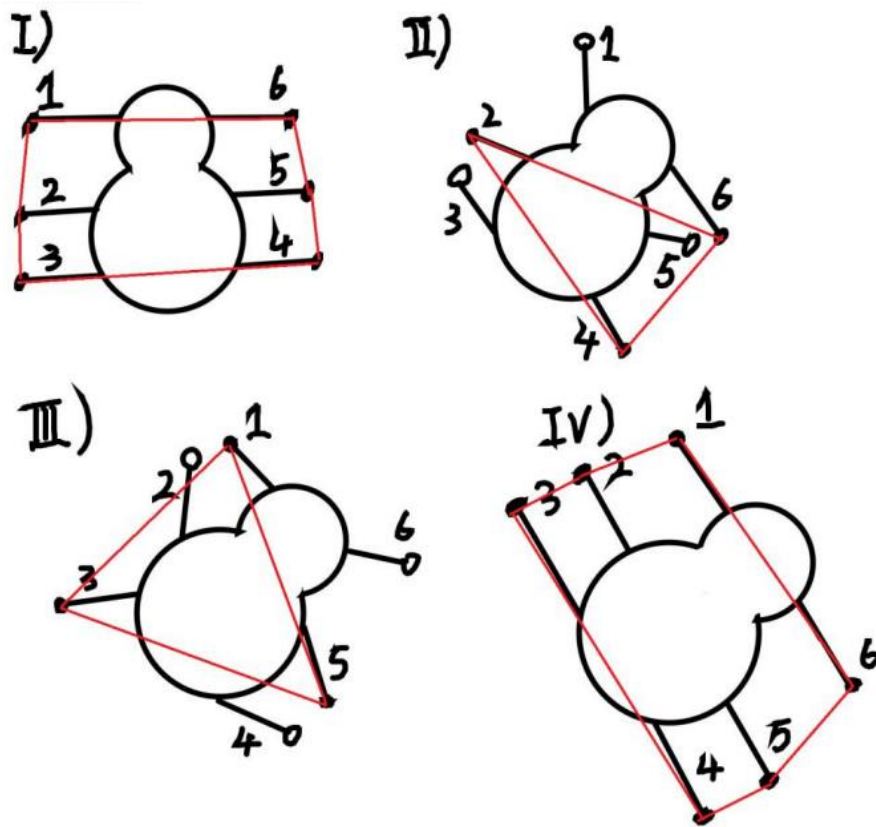


Рисунок 3.2 – Виконання повороту із фіксованою точкою, тринога хода

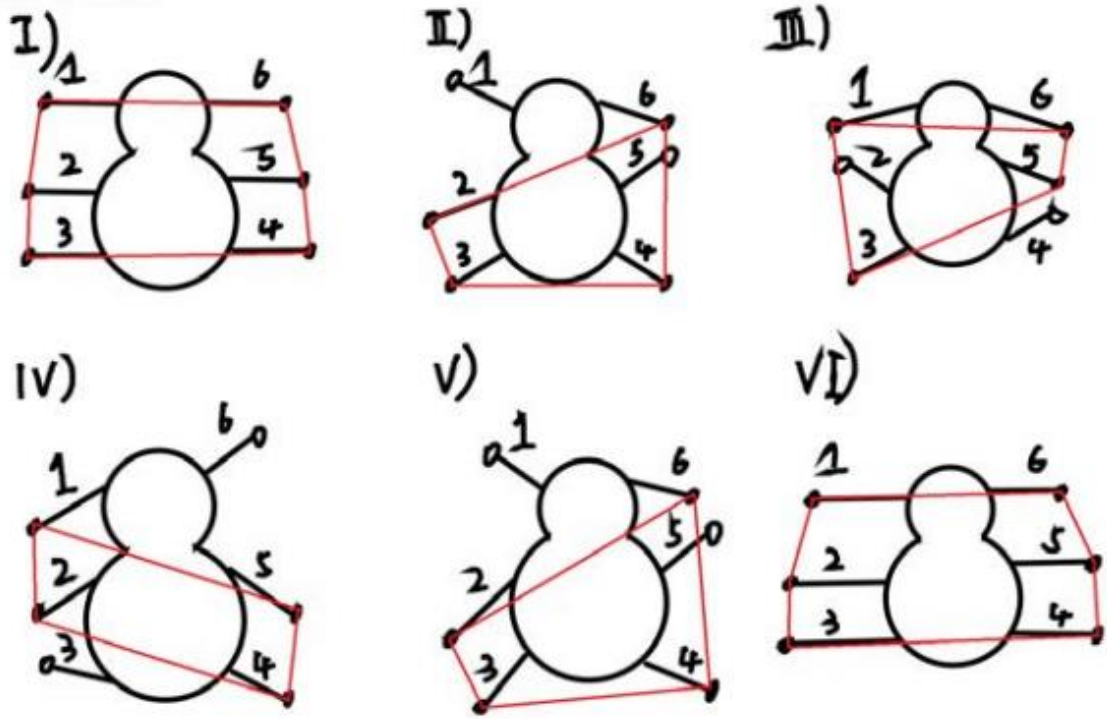


Рисунок 3.3 – Виконання руху по прямій лінії, чотиринога хода

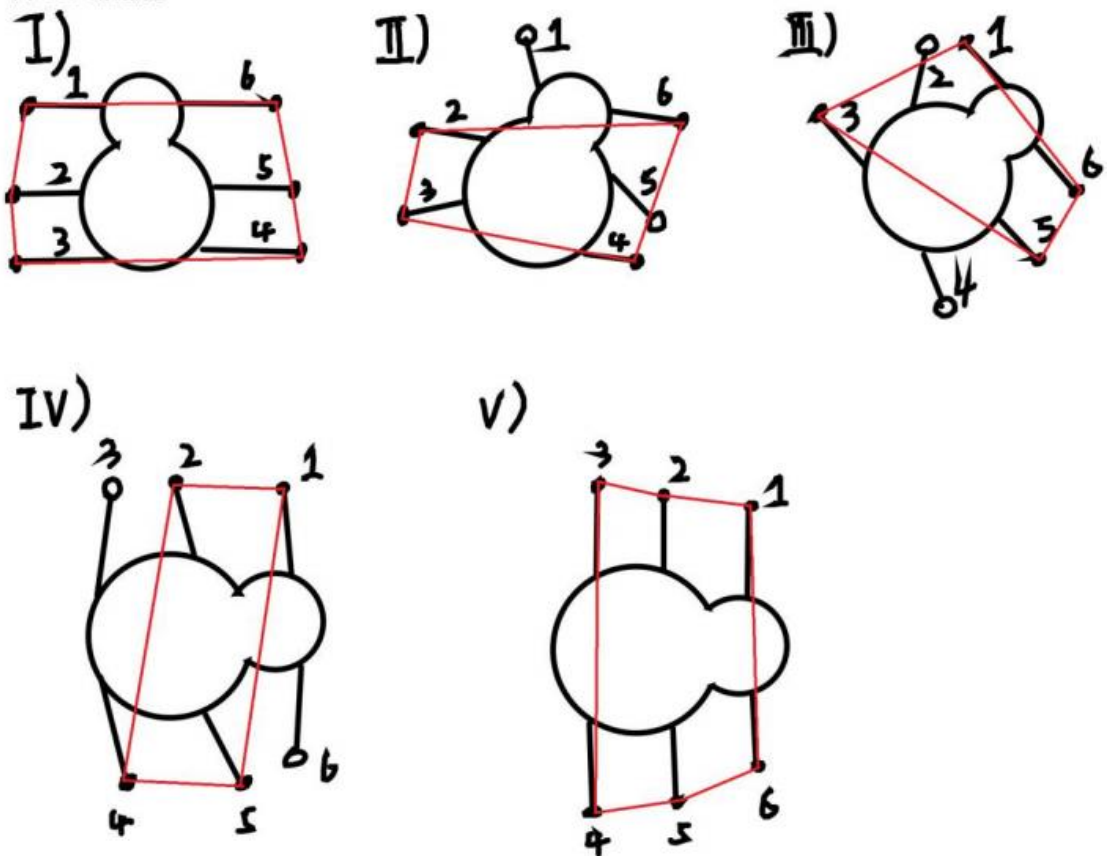


Рисунок 3.4 – Виконання повороту із фіксованою точкою, чотиринога хода

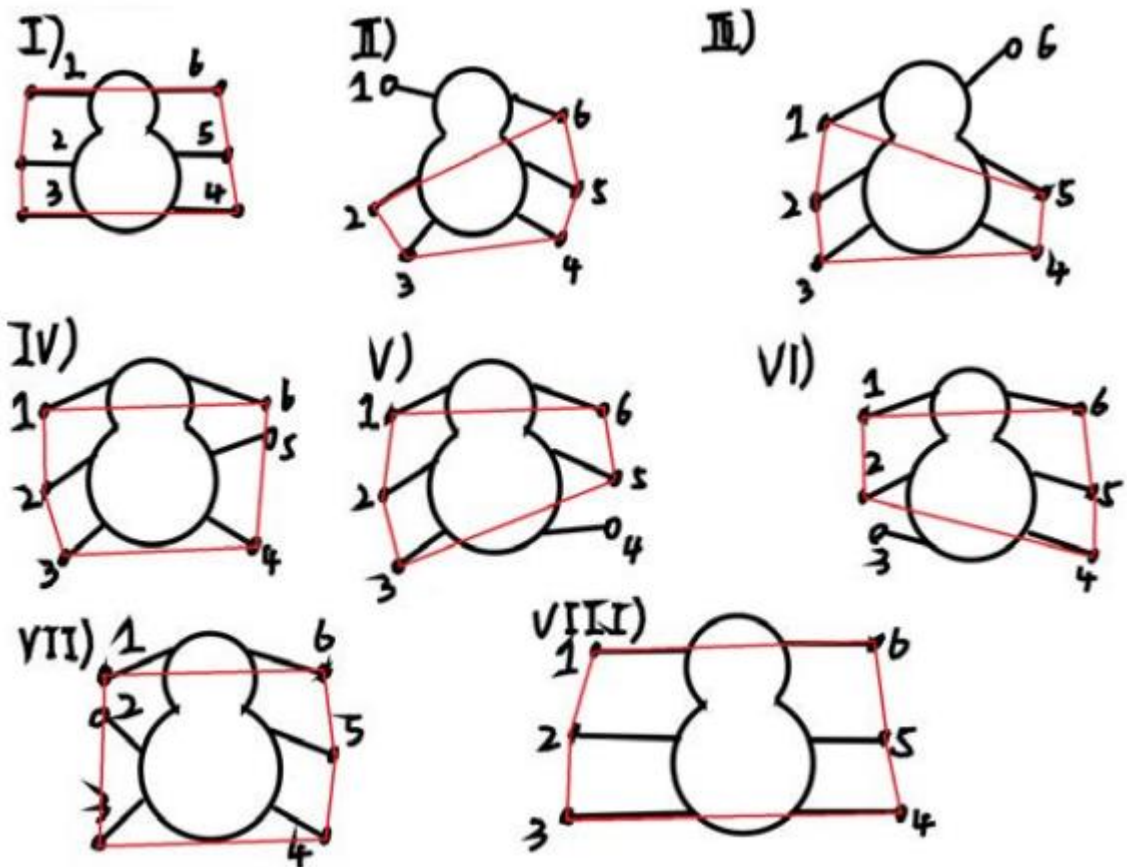


Рисунок 3.5 – Виконання руху по прямій лінії, п'ятиного хода

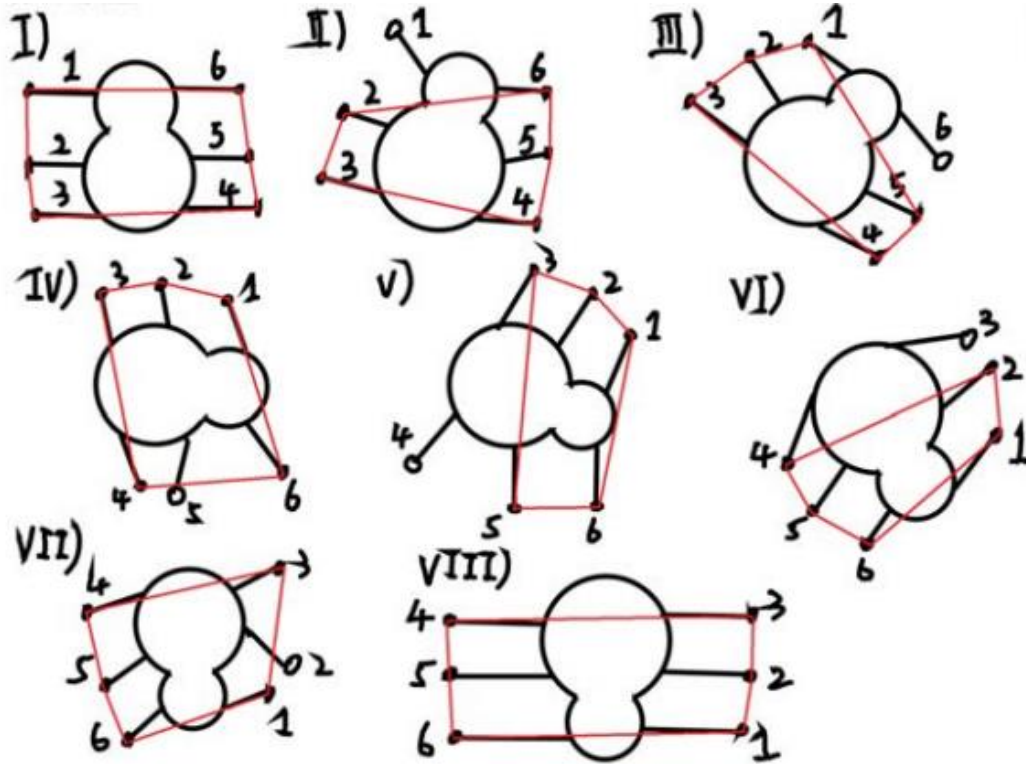


Рисунок 3.6 – Виконання повороту із фіксованою точкою, п'ятиного хода

Аналітичне дослідження показує, що центр ваги ходи трьох, чотирьох і п'яти ніг знаходиться в багатокутнику, утвореному підошвами ніг, що стикаються із поверхнею, і всі три види ходи володіють певним рівнем стійкості. Порядок розміру – три ноги < чотири ноги < п'ять ніг, але з точки зору рівнів швидкості три ноги краще, ніж чотири ноги та п'ять ніг. Оскільки конструкція вимагає не лише стабільності, але і певного рівня швидкості руху, в системі також переважає тринога хода. Інші види ходи будуть використовуватись в якості запасного виду ходи.

Використовуючи аналіз робочого циклу наведений в розділі 2, аналіз діаграм планування ходи для трьох, чотирьох і п'ятиногих МРГ, а також аналіз статичної стійкості під час руху, були отримані переваги і недоліки трьох видів ходи і виконано аналітичне дослідження.

Тринога хода була обрана в якості основної ходи руху, а чотиринога і п'ятинога хода були обрані в якості альтернативної ходи для виконання руху МРГ [39-45].

3.2 Підбір основних елементів для моделі мехатронного робота-гексапода

МРГ потребує вісімнадцять суглобів (механізм керування) для одночасної роботи для виконання поступального руху. У процесі руху кожен суглоб повинен керуватися впорядкованим чином, що є необхідною умовою для того, щоб МРГ міг плавно виконувати поступальний рух.

Отже, в даній моделі МРГ також необхідно розробити систему керування, яка відповідає біонічному роботу. На додаток до того, що МРГ повинен плавно виконувати рух, система керування (СК) також повинна мати систему камери.

СК МРГ умовно поділена на три частини-модулі: руху, камери і живлення. Плата керування і електропривід утворюють модуль руху для виконання руху МРГ, камера і модуль бездротової передачі 4G утворюють модуль камери, який може контролювати і передавати зображення в режимі

реального часу під час виконання руху МРГ, а модуль живлення забезпечує живлення всього МРГ [10-14, 38].

3.2.1 Конструювання модуля руху МРГ

Функція руху є основною функцією МРГ, вона пов'язана з тим, чи може БР рухатися ефективно і плавно, а також з успіхом або невдачею всієї конструкції. Модуль функції руху МРГ включатиме в свою конструкцію компоненти електроприводу і панель керування.

Конструкція електроприводної частини – ключем до функції МРГ є здатність реалізовувати рух. Розумний вибір електроприводу може значно поліпшити робочі характеристики і зменшити загальні труднощі при проектуванні конструкції. Оскільки розмір МРГ відносно великий, сервопривід використовується в якості рушійної частини.

У зв'язку з тим, що матеріал ноги МРГ, що приводиться в даній конструкції, є 3D-друкованим, матеріал відносно легкий, і в відношенні руху існують певні вимоги до кута і рівня швидкості руху, остаточно обрано сервопривід з великим крутним моментом типу Tower Pro MG996r [46, 47] в якості приводу з'єднань МРГ, загальний вигляд представлено на рисунку 3.7. Даний метод керування сервоприводом Tower Pro MG996r [46, 47] достатньо простий, високо точний, мало коштує і є невеликого розміру, з крутним моментом до 13 кг, електродвигуном з порожнистою чашкою і подвійним кульковим підшипником. Кут повороту всього рульового механізму складає повних 180° .

Рульовий механізм на 180° може контролювати рівні швидкості і може відповідати руху всього МРГ з точки зору рівнів швидкості. Сигнал керування сервоприводом Tower Pro MG996r [46, 47] є сигналом ШІМ. Положення сервоприводу змінюється за допомогою зміни робочого циклу. Зміни в циклі ШІМ представлені на рисунку 3.8, а параметри даного сервоприводу Tower Pro

MG996r [46, 47] наведені в таблиці 3.1.



Рисунок 3.7 – Сервопривід Tower Pro MG996r [46, 47]

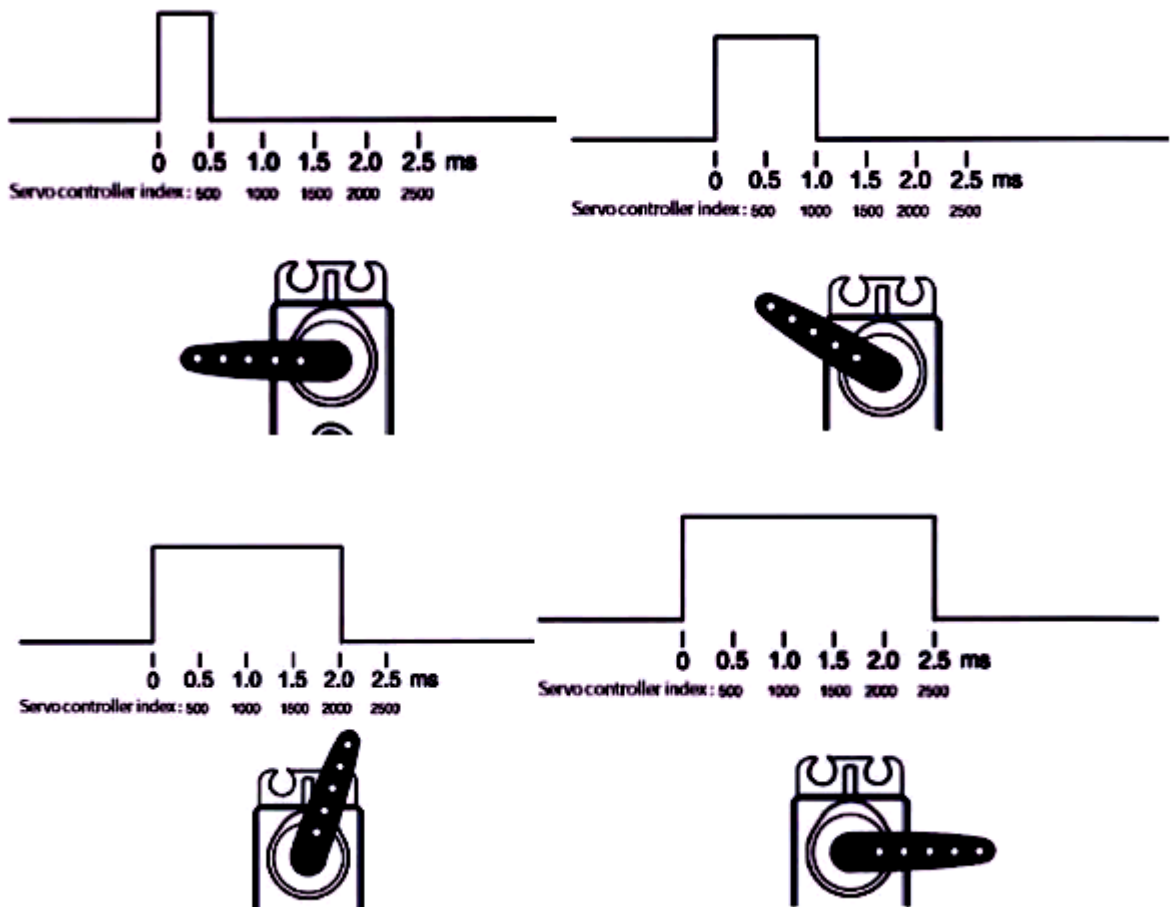


Рисунок 3.8 – Імпульсна діаграма сервоприводу Tower Pro MG996r [46, 47]

Таблиця 3.1 – Параметри сервоприводу Tower Pro MG996r [46, 47]

Назва параметру	Значення параметру
Робоча сила струму, мА	100
Робочий рівень напруги, В	від 3 до 7,2
Крутний момент, кгс	13
Діапазон робочих температур, °С	Від мінус 30 до +60
Закінчення табл. 3.1	
Максимальний кут повороту, °	360
Тип	MG996r
Габаритні розміри, мм	40,7 x 19,7 x 42,9
Вага, г	55

3.2.2 Панель керування

У конструкції МРГ передбачено 18 сервоприводів Tower Pro MG996r [46, 47] для керування рухом, кожен сервопривід повинен бути підключений до головної плати керування і керується за принципом керування ШІМ. Керування цими 18 сервоприводами Tower Pro MG996r [46, 47] безпосередньо впливає на рух всього МРГ, тому в даній конструкції буде використовуватись МК Arduino mega2560 [48-50] для керування сервоприводом Tower Pro MG996r [46, 47].

Arduino mega2560 [49] показаний на рисунку 3.9, параметри представлені в табличному вигляді (див. табл. 3.2), в Arduino mega2560 [48-50] використовується базова друкована плата з інтерфейсом USB, найбільша особливість в тому, що Arduino mega2560 [48-50] оснащено до 54 цифрових входів і виходів, що особливо підходить для проектування, що вимагає великої кількості інтерфейсів вводу/виводу. Ядром процесора Mega 2560 є ATmega 2560 [51], який також має 54 цифрових порти вводу/виводу, 18 з яких можна використовувати в якості ШІМ-виходу, 18 аналогових входів, 4 інтерфейси UART, кварцовий генератор 16МГц, USB-порт, рознім живлення,

ICSP і кнопку скидання. Щоб задовольнити керування 18 сервоприводами Tower Pro MG996r [46, 47] в проєктованій конструкції, немає необхідності додавати додаткову плату керування сервоприводом, що економить накладні витрати на проєктування.



Рисунок 3.9 – Загальний вигляд плати керування Arduino mega2560 [49]

Таблиця 3.2 – Параметри Arduino mega2560 [48-50]

Назва параметру	Значення параметру
Кількість цифрових портів вводу/виводу, шт	54
Рівень робочої напруги, В	5
Порт постійної сили струму (вводу/виводу), мА	40
Порт постійного рівня напруги 3,3 В, мА	50
Частота, МГц	16
Тип мікроконтролеру	ATmega2560
Габаритні розміри, мм	101,52 x 53,3
Вага, г	37

Конструкція пульта дистанційного керування МРГ

У даній конструкції до керування додано можливість використання пульта дистанційного керування (ПДК), який порушує традиційне керування комп'ютером і мобільним телефоном. В якості ПДК буде використано пульт від PS2 [52]. На рисунку 3.10 представлено загальний вигляд ПДК. У СК встановлений приймальний модуль ПДК на МК Arduino mega2560 [48-50], ПДК передає сигнал ШІМ і передає його приймачу, який, у свою чергу, передає сигнал на МК Arduino mega2560 [48-50], який передає сигнал сервоприводам для керування рухом. Реальні параметри представлені в табличному вигляді (див. табл. 3.3).



Рисунок 3.10 – Загальний вигляд ПДК [52]

Таблиця 3.3 – Реальні параметри ПДК [52]

Назва параметру	Значення параметру
Рівень робочої напруги, В	від 3, 3до 5
Частота, Гц	2,4
Робоча відстань, м	2

На рисунку 3.11 представлено загальний вигляд виводів модуля прийому сигналу дистанційного керування, опис функції виводів представлено в табличному вигляді (див. табл. 3.4).

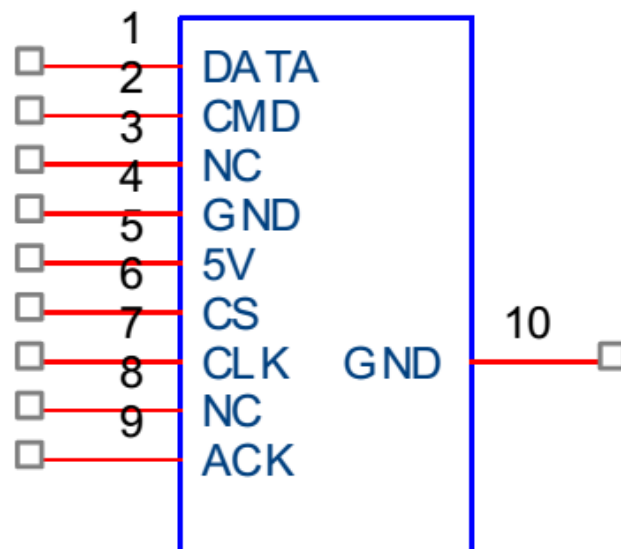


Рисунок 3.11 – Модуль прийому сигналів ПДК [52]

Таблиця 3.4 – Функція виводів приймального модуля ПДК [52] (рис. 3.6)

Номер порту	Призначення порту	Ім'я порту
1	Сигнал рухається від маркера до хосту, 8-бітні послідовні дані сигналу передаються синхронно по подаючому фронту тактового сигналу. Зчитування сигналу завершується при зміні тактового сигналу від високого до низького рівня.	DATA

Кінець таблиці 3.4 – Функція виводів приймального модуля ПДК [52]
(рис. 3.6)

Номер порту	Призначення порту	Ім'я порту
2	Сигнал рухається від хоста до дескриптора. Даний сигнал протилежний до сигналу DATA. Сигнал являє собою 8-бітні послідовні дані, які синхронно передаються по передньому фронту тактового сигналу.	CMD
3	Не використовується, пустий порт.	NC
4	Джерело живлення.	GND
5	Порт підключення до робочого рівня напруги.	5V
6	Забезпечення тригерного сигналу ПДК, який знаходиться на низькому рівні під час зв'язку.	CS
7	Тактовий сигнал, що відправляється хостом для синхронізації отриманих даних.	CLK
8	Не використовується, пустий порт.	NC
9	Відповідь від ПДК до хосту. Сигнал стає нижчим в останньому циклі кожної передачі даних 8BITS, і CS-прямий залишається низьким. Якщо сигнал CS не змінюється на низькому рівні, хост PS спробує інший ПДК протягом приблизно 60 мікросекунд.	ACK
10	Джерело живлення.	GND

Вхідна частина контролера обробки для реалізації ПДК в програмному середовищі Arduino IDE [53, 54] представлена на рисунку 3.12.

```

void process_gamepad()
{
  if(ps2x.ButtonPressed(PSB_PAD_DOWN)) //stop & select gait 0
  {
    mode = 0;
    gait = 0;
    reset_position = true;
  }
  if(ps2x.ButtonPressed(PSB_PAD_LEFT)) //stop & select gait 1
  {
    mode = 0;
    gait = 1;
    reset_position = true;
  }
  if(ps2x.ButtonPressed(PSB_PAD_UP)) //stop & select gait 2
  {
    mode = 0;
    gait = 2;
    reset_position = true;
  }
  if(ps2x.ButtonPressed(PSB_PAD_RIGHT)) //stop & select gait 3
  {
    mode = 0;
    gait = 3;
    reset_position = true;
  }

  if(ps2x.ButtonPressed(PSB_TRIANGLE)) //select walk mode
  {
    mode = 1;
    reset_position = true;
  }
  if(ps2x.Button(PSB_TRIANGLE)) //vibrate controller if walk button held
    gamepad_vibrate = 64;
  else
    gamepad_vibrate = 0;
  if(ps2x.ButtonPressed(PSB_SQUARE)) //control x-y-z with joysticks mode
  {
    mode = 2;
    reset_position = true;
  }
  if(ps2x.ButtonPressed(PSB_CIRCLE)) //control y-p-r with joysticks mode
  {
    mode = 3;
    reset_position = true;
  }
  if(ps2x.ButtonPressed(PSB_CROSS)) //one leg lift mode
  ,

```

Рисунок 3.12 – Частина програми керування через ПДК

3.2.3 Реалізація функції руху

Відповідно до МК Arduino mega2560 [48-50] даний проєкт МРГ використовує Arduino IDE [53, 54] в якості програмного забезпечення для програмування і розроблення коду для здійснення передачі інформації між МК

Arduino mega2560 [48-50] і сервоприводом Tower Pro MG996r [46] і ПДК для керування рухом МРГ [52].

Електрична схема МРГ представлена на рисунку 3.13.

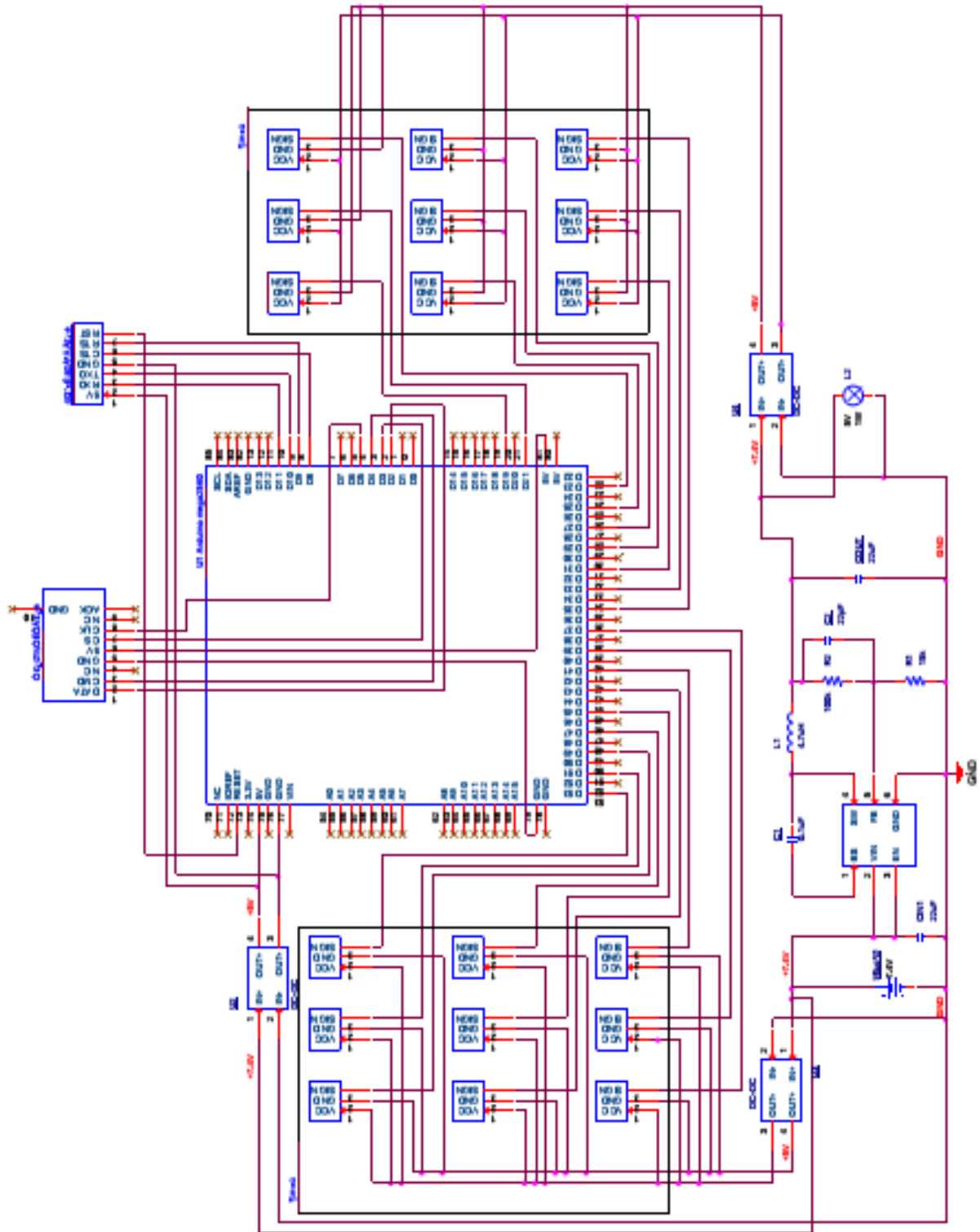


Рисунок 3.13 – Електрична схема МРГ

На рисунку 3.14 представлено схему портів МК Arduino mega2560 [48-50], призначення портів представлені в табличному вигляді (див. табл. 3.5).

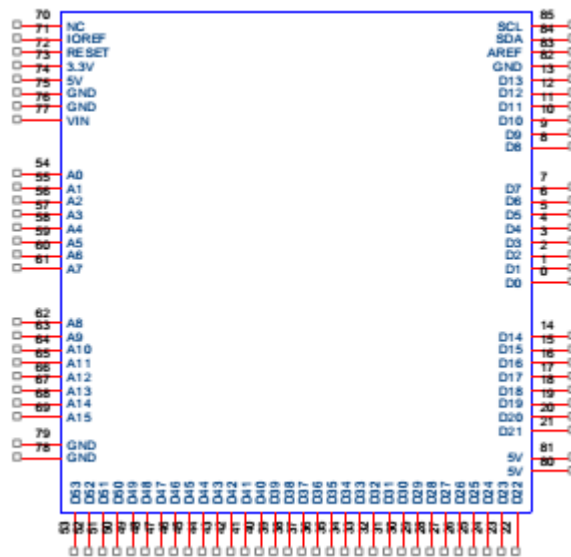


Рисунок 3.14 – Електрична схема портів МК Arduino mega2560 [48-50]

Таблиця 3.5 – Призначення портів МК Arduino mega2560 [48-50]

Номер порту	Призначення порту	Ім'я порту
2	Для отримання доступу до плати дистанційного керування DATA, потік сигналу від ПДК до хосту, 8-бітні послідовні дані сигналу передаються синхронно по спадаючому фронту тактового сигналу. Зчитування сигналу завершується при переході тактових імпульсів від високого до низького рівня.	D2
3	Для подачі сигналу, який знаходиться на низькому рівні під час зв'язку до плати дистанційного керування CS.	D3
4	При доступі до плати дистанційного керування CMD потік сигналу від хоста до ПДК. Сигнал протилежний DATA, представляє собою 8-бітні послідовні дані, які синхронно передаються по спадаючому фронту тактового сигналу.	D4

Кінець таблиці 3.5 – Призначення портів МК Arduino mega2560 [48-50]

Номер порту	Призначення порту	Ім'я порту
5	Для підключення до тактового сигналу ПДК, який надсилається хостом для синхронізації даних.	D5
19-53	Виходи сигналу сервоприводу Tower Pro MG996r	D19-D53
74	Підключення до вихідної клеми джерела живлення модуля зниження рівня напруги	5V
75-79	Джерело живлення.	GND
81	Підключення до вихідної клеми джерела живлення модуля зниження рівня напруги	5V
10	Джерело живлення.	GND

Очевидно, що порт МК Arduino mega2560 [48-50] D2 з'єднується з портом DATA приймального модуля ПДК, порт МК Arduino mega2560 [48-50] D3 з'єднується з портом CS приймального модуля ПДК, порт МК Arduino mega2560 [48-50] D4 з'єднується з портом CMD приймального модуля ПДК, порт МК Arduino mega2560 [48-50] D5 з'єднується з портом CLK приймального модуля ПДК, порт МК Arduino mega2560 [48-50] D19-D53 з'єднується з портом SIGN для прийому команд і керування сервоприводом Tower Pro MG996r [46] для виконання руху.

Буде використовуватись програма Arduino IDE [53, 54], в якій прописані три типи ходи: 3/6, 4/6 і 5/6 для перевірки плану ходи розглянутому раніше. В основному буде використовуватись саме програма ходи в 3/6, на рисунку 3.15 нижче показані деякі основні програми.

Конструкція функціонального модуля камери.

У функціональному дизайні встановлено, що МРГ повинні виявляти, фотографувати складні середовища, а потім передавати зображення людині-оператору, тому необхідно розробити модуль для реалізації функцій зйомки та

передачі отриманих зображень.

```

//*****
// Main Program
//*****
void loop()
{
  //exit if no controller found or GuitarHero controller
  if((gamepad_error == 1) || (gamepad_type == 2))
  {
    Serial.println("Invalid Controller!");
    return;
  }

  //set up frame time
  currentTime = millis();
  if((currentTime - previousTime) > FRAME_TIME_MS)
  {
    previousTime = currentTime;

    //read controller and process inputs
    ps2x.read_gamepad(false, gamepad_vibrate);
    process_gamepad();

    //reset legs to home position when commanded
    if(reset_position == true)
    {
      for(leg_num=0; leg_num<6; leg_num++)
      {
        current_X[leg_num] = HOME_X[leg_num];
        current_Y[leg_num] = HOME_Y[leg_num];
        current_Z[leg_num] = HOME_Z[leg_num];
      }

      reset_position = false;
    }

    //position legs using IK calculations - unless set all to 90 degrees mode
    if(mode < 99)
    {
      for(leg_num=0; leg_num<6; leg_num++)
        leg_IK(leg_num, current_X[leg_num]+offset_X[leg_num], current_Y[leg_num]+offset_Y[leg_num], current_Z[leg_num]+offset_Z[leg_num])
    }

    //reset leg lift first pass flags if needed
    if(mode != 4)
    {
      leg1_IK_control = true;
      leg6_IK_control = true;
    }

    //battery_monitor(); //battery monitor and output to LEDs
    print_debug(); //print debug data

    //process modes (mode 0 is default 'home idle' do-nothing mode)
    if(mode == 1) //walking mode
    {
      if(gait == 0) tripod_gait(); //walk using gait 0
      if(gait == 1) wave_gait(); //walk using gait 1
      if(gait == 2) ripple_gait(); //walk using gait 2
      if(gait == 3) tetrapod_gait(); //walk using gait 3
    }
    if(mode == 2) translate_control(); //joystick control x-y-z mode
    if(mode == 3) rotate_control(); //joystick control y-p-r mode
    if(mode == 4) one_leg_lift(); //one leg lift mode
    if(mode == 99) set_all_90(); //set all servos to 90 degrees mode
  }
}

```

Рисунок 3.15 – Частина основної програми керування ходи в 3/6 [53, 54]

Буде використовуватись камера типу Nanying A8 [55], дана камера представляє собою панорамну камеру 360°, без підключення і проводів, може передавати створені зображення по мережі 4G на мобільний телефон, здійснювати моніторинг в реальному режимі часу, володіє функцією індукції тіла, час очікування супер великий. Це хороший вибір для використання на МРГ для пошуку і роботи. Реальні параметри камери наведені в табличному вигляді.

Таблиця 3.6 – Параметри камери Nanying A8 [55]

Назва параметру	Значення параметру
Робочий рівень напруги, В	5
Робоча сила струму, А	1
Величина кута огляду лінзи, град	180
Максимальна якість відеозйомки, пікс	1080
Габаритні розміри, мм	12x12



Рисунок 3.16 – Загальний вигляд модуля передачі даних 4G

3.2.4 Блок передачі отриманих даних

Коли МРГ переміщається з поля зору людини-оператора, оператор не зможе керувати ним відповідно до змiну поверхні місцевості в реальному режимі

часу. Тому на МРГ встановлено модуль передачі даних в 4G, загальний вигляд якого представлено на рисунку 3.16, який з'єднується з камерою і передає зображення камери Nanuīng A8 [55] на мобільний телефон людини-оператора за допомогою мережевого сигналу 4G, щоб МРГ міг переміщатися під керуванням людини-оператора без прямої видимості.

3.2.5 Проєктування модуля електроживлення для МРГ

Модель МРГ, вимагає бездротового керування і роботи на відстані, тому на корпусі МРГ повинен бути встановлено змінне джерело живлення, щоб МРГ міг працювати автономно після відключення від дротового джерела живлення.

Вибір акумуляторної батареї.

В даний час більшість інтелектуальних роботів використовують літєві батареї для забезпечення енергії для роботи автономно. Тому в даній моделі МРГ буде використовуватись Li-батарея 2S із реакцією Гріньяра [56]. Дана модель акумуляторів має високий рівень робочої напруги, високу щільність енергії, малу габаритність, високу витривалість, може задовольняти автономну роботу МРГ. Робоча сила струму обраного до монтажу сервоприводу Tower Pro MG996r [46] із максимальним рівнем енергоспоживання складає не більше 0,1А, тому необхідна потужність для вісімнадцяти сервоприводів Tower Pro MG996r [46] на годину буде рівним:

$$W = I \cdot t; \quad (3.1)$$

$$W = 18 \cdot 0,1 \cdot 1 = 1800mAh.$$

Не виключаючи додаткову витрату, було вирішено використовувати Li-батарею Гріньяра 2S [56] на 4000 мАгод, загальний вигляд якої представлено на рисунку 3.17, реальні параметри представлені в табличному вигляді.



Рисунок 3.17 – Загальний вигляд Li-батареї GOLDBAT LiPo 2S [57]

Таблиця 3.7 – Реальні параметри Li-батареї GOLDBAT LiPo 2S [57]

Назва параметру	Значення параметру
Номінальний рівень напруги, В	7,4
Ємність акумулятору, мАг	4000
Коефіцієнт заряджання	10С
Коефіцієнт розряджання	30С
Габаритні розміри, мм	4x41x132
Вага, г	288

Модуль регулювання рівня напруги.

У зв'язку з тим, що робочий рівень напруги сервоприводу Tower Pro MG996r [46] складає від 3 до 7,2 В, а робочий напруги напруги акумуляторної батареї складає 7,4 В [57], для забезпечення безпечної експлуатації сервоприводу Tower Pro MG996r [46] потрібен модуль регулювання рівня напруги для зниження рівня вихідної напруги, будемо використовувати модуль регулювання рівня напруги – понижуючого і стабілізуючого джерела живлення постійної сили струму типу DC-DC – LM2596S [58, 59], загальний вигляд якого представлено на рисунку 3.18.



Рисунок 3.18 – Загальний вигляд понижуючого і стабілізуючого джерела живлення постійної сили струму типу DC-DC – LM2596S [58, 59]

LM2596S [58, 59] оснащено потужною, високоефективною понижуючою і низьким рівнем пульсації 5А, із встановленим радіатором і індикатором рівня напруги, який може контролювати потужність підключеного джерела живлення, щоб уникнути недостатньої потужності і хаосу керування, що призводить до пошкодження МРГ. Таким чином, функції джерела живлення LM2596S [58, 59] можуть добре вирішити дану проблему. Реальні параметри представлені в табличному вигляді.

Таблиця 3.8 – Реальні параметри модуля регулювання рівня напруги [59]

Назва параметру	Значення параметру
Діапазон вхідного рівня напруги, В	Від 2,5 до 35
Діапазон вихідного рівня напруги, В	Від 1,25 до 35
Вихідна сила струму, А	3
Робоча частота, Гц	150
Коефіцієнт перемикання, %	92
Діапазон робочих температур, °С	Від мінус 45 до +85
Габаритні розміри, мм	66x36

3.3 Висновки до третього розділу

Проведено аналітичне дослідження стійкості ходи мехатронних роботів-гексаподів. Було виконано розроблення системи керування мехатронним роботом-гексаподом та підбір основних елементів керування. Вся керуюча частина МРГ реалізується шляхом проєктування та аналізу сервоприводів Tower Pro MG996r, плати керування Arduino mega2560, ПДК – пульт від PS2, камери Nanjing A8 та модуля живлення електроприводної частини LM2596S, а також із використанням програмування Arduino.

4 ДОСЛІДЖЕННЯ РОБОТИ МОДЕЛІ МЕХАТРОННОГО РОБОТА-ГЕКСАПОДА

4.1 Моделювання роботи конструкції мехатронного робота-гексапода

В даний час МРГ виконується з використанням стандартних методів виготовлення та програмування.

Використовуються наступні засоби:

- корпус МРГ буде виготовлено зі склеєних пластин з оргскла;
- в корпус МРГ встановлюється спеціальна електронна плата розширення можливостей – Robot Hat [60];
- програма робота «захита» в МК Arduino mega2560 [50, 51] і виконана на стандартній мові програмування;
- також в корпус МРГ поміщається відсік для акумуляторних Li-батареї 2S;
- рушієм ніг МРГ є спеціальний сервопривід Tower Pro MG996r [46], необхідно вісімнадцять сервоприводів;
- в якості датчиків використовуються: відеокамера Nanying A8 [55], та інші датчі.

Креслення для вирізання механічних елементів МРГ були створені в САД системі 3D моделювання SolidWorks 2025 [32-35]. До таких елементів відносяться:

- платформа для закріплення сервоприводів Tower Pro MG996r [46] (рис. 4.1);
- «тазова частина» МРГ (рис. 4.2), створена у вигляді збірки з самим сервоприводом Tower Pro MG996r [46];
- частина ноги МРГ (рис. 4.3) (за кількістю ніг МРГ їх буде 6 штук), накреслена як єдине ціле;

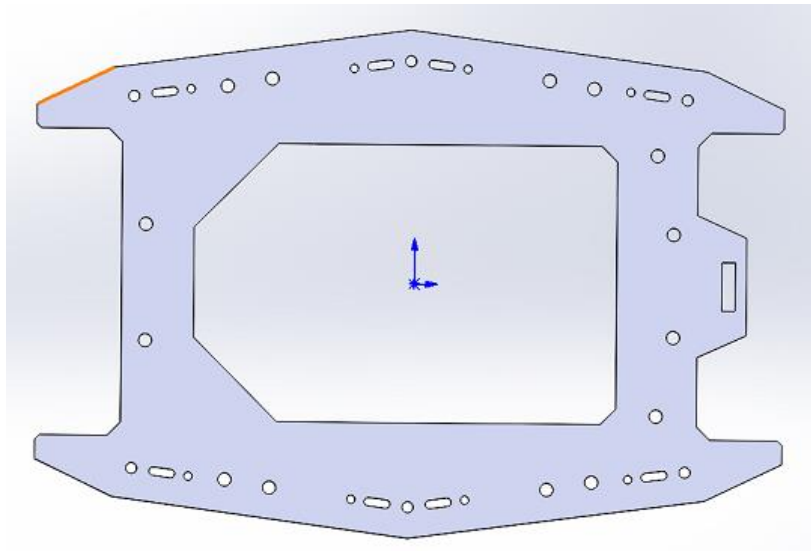


Рисунок 4.1 – Загальний вигляд платформи для закріплення сервоприводів Tower Pro MG996r [46]

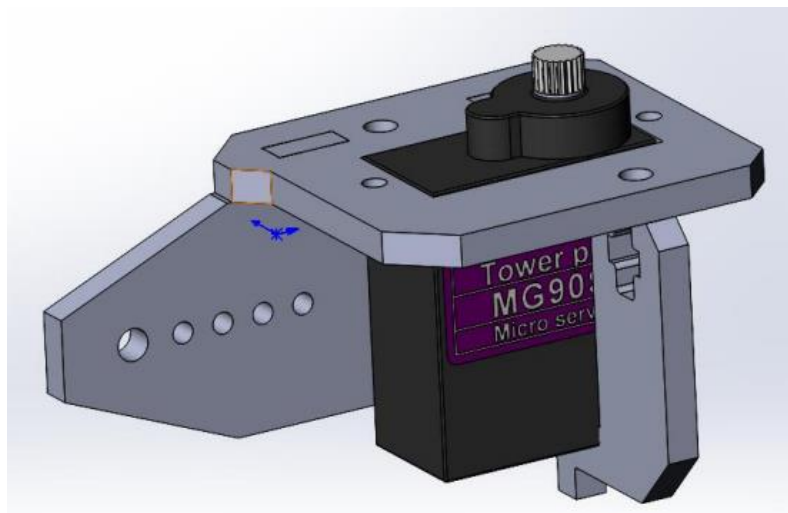


Рисунок 4.2 – Загальний вигляд «тазової частини» МРГ в збірці із сервоприводом Tower Pro MG996r [46]

З'єднавши всі спроектовані механічні частини МРГ в одне ціле (див. рис. 4.4), можна отримати спрощену модель МРГ, рух якої вже можна досліджувати із використанням спеціальних віртуальних засобів (нами було використано програмне середовище MATLAB [61-63]).

Для дослідження руху МРГ спочатку необхідно виконати експортування деталей створених в програмному середовищі SolidWorks 2025 [32-36] для

креслення 3D деталей і збірок у відповідний блок програмного середовища MATLAB (Simmechanics [64]) у форматі STEP. Зібрана нога МРГ, наприклад, у даному форматі виглядає наступним чином (див. рис. 4.5):

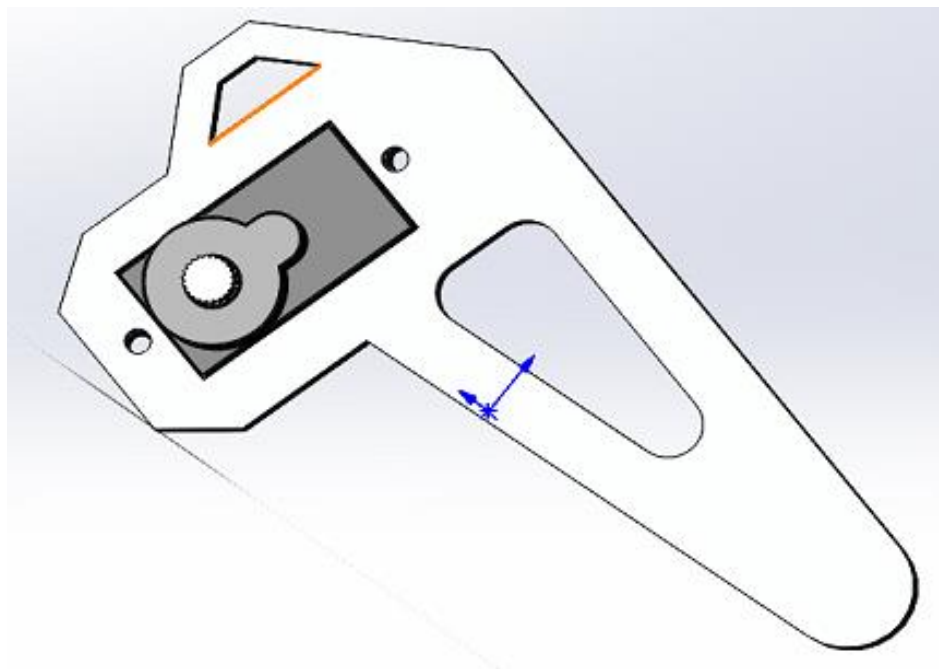


Рисунок 4.3 – Загальний вигляд частини ноги МРГ

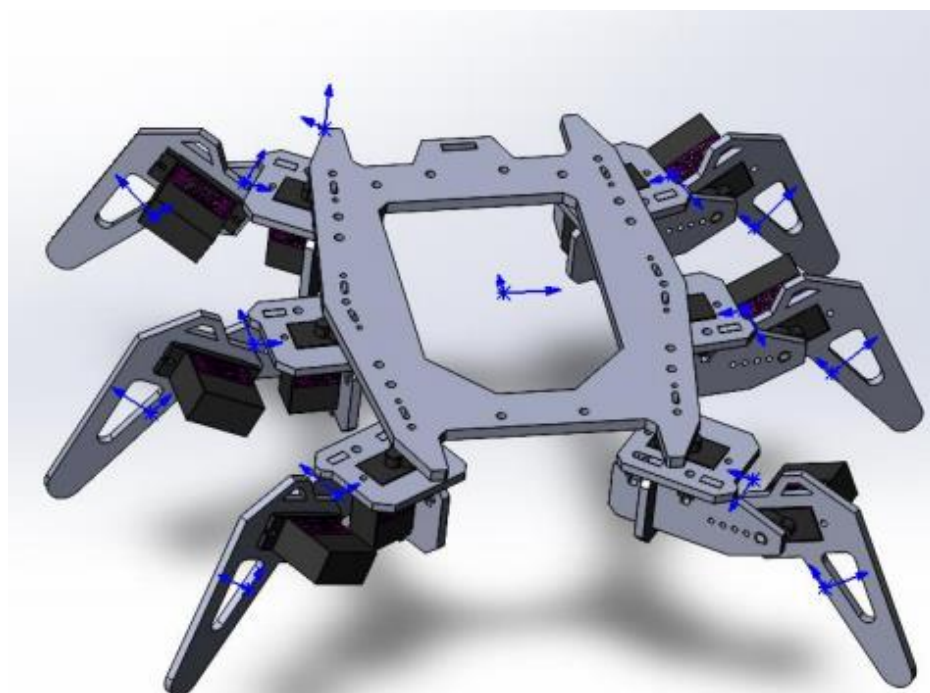


Рисунок 4.4 – Загальний вигляд збірки механічних частин МРГ

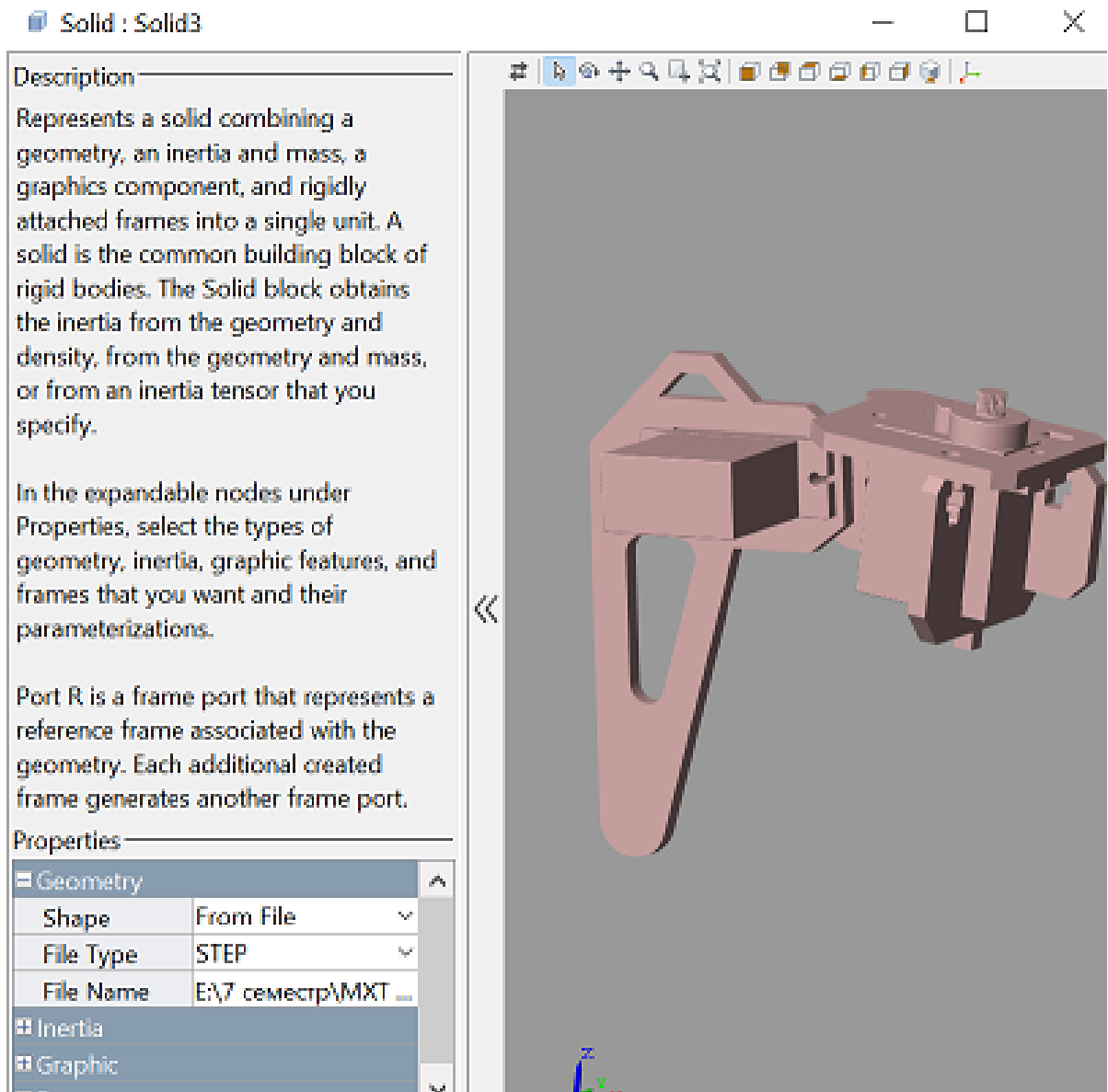


Рисунок 4.5 – Загальний вигляд зібраної ноги МРГ, яка імпортована в програмне середовище MATLAB у форматі STEP [61-64]

Кожен блок програмного середовища MATLAB [61-63], крім візуалізації елемента МРГ, повинен містити опис можливостей переміщення конструкції. Розглянемо створені елементи моделі МРГ в програмному середовищі MATLAB (рис. 4.6) більш детально.

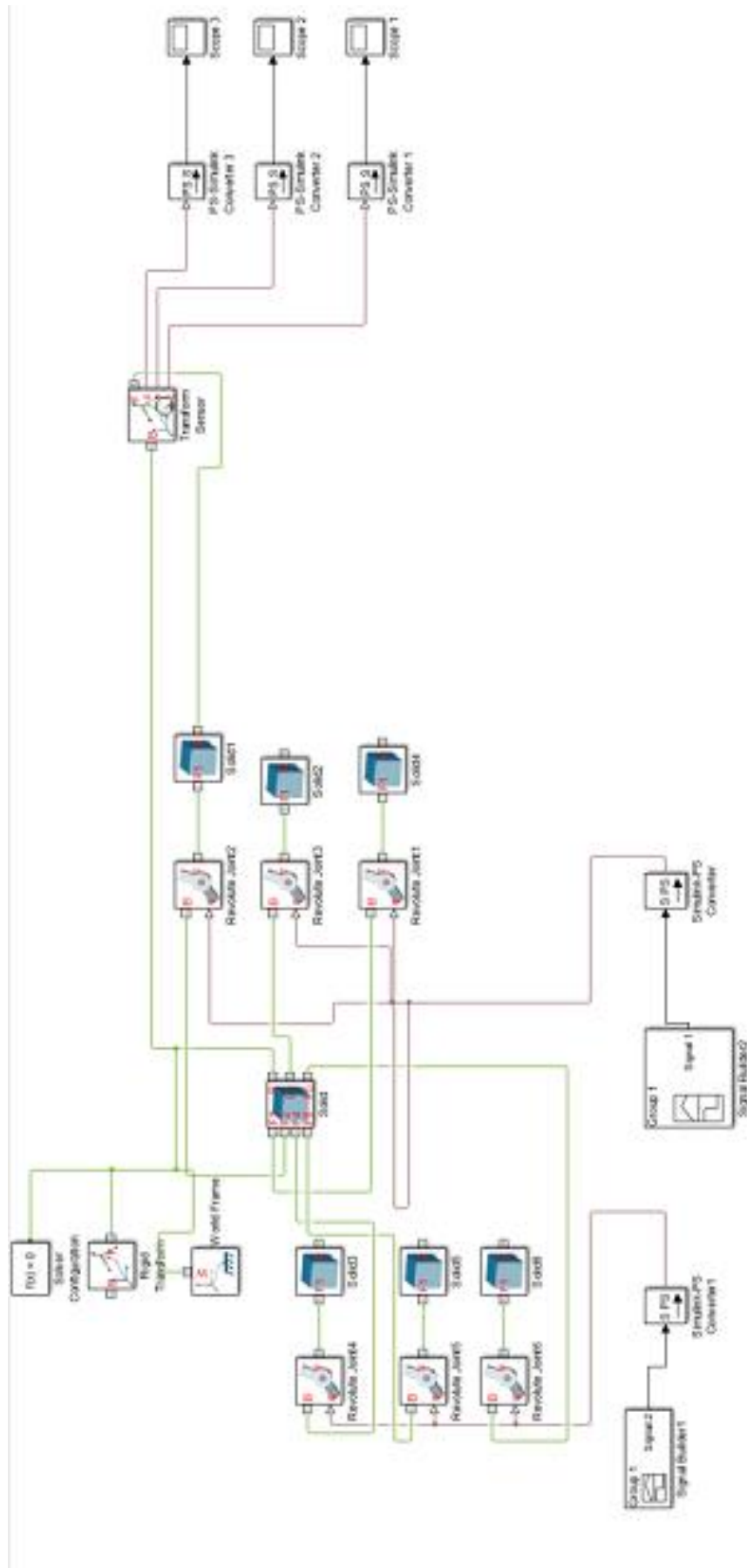


Рисунок 4.6 – Блок-схема МРГ в програмному середовищі MATLAB [61-63]

Спеціальній блок Solver Configuration [61-65] (див. рис. 4.7) описує основні налаштування задавача руху МРГ. Без даного спеціального блоку запуск моделі МРГ неможливий (оскільки жодна задача виконання руху вирішуватися не буде).

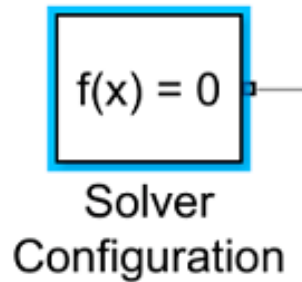


Рисунок 4.7 – Загальний вигляд спеціального блоку Solver Configuration схеми МРГ в програмному середовищі MATLAB [61-65]

Спеціалізований блок World Frame [61-65] (див. рис. 4.8) задає властивості глобальної системи координат (навіть при додаванні декількох подібних блоків в проєкт програмного середовища MATLAB [62] глобальна система координат, відносно якої виконується обрахунок переміщення елементів МРГ, не буде змінюватись).

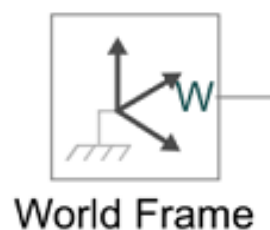


Рисунок 4.8 – Загальний вигляд спеціалізованого блоку World frame схеми МРГ в програмному середовищі MATLAB [61-65]

Спеціалізований блок Solid [61-65] (рис. 4.9) дозволяє створити геометрію самостійно – в даній моделі МРГ він імпортує геометрію з власноруч створених файлів програмного середовища SolidWorks [32-35].

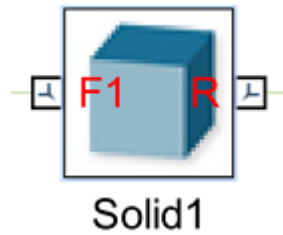


Рисунок 4.9 – Загальний вигляд спеціалізованого блоку Solid схеми МРГ в програмному середовищі MATLAB [61-65]

Спеціалізований блок Revolute Joint [61-65] (рис. 4.10) задає можливий обертовий рух ніг проєктованого за допомогою використання моделі МРГ.

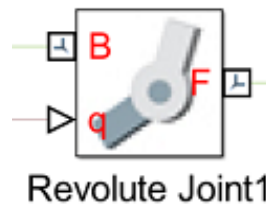


Рисунок 4.10 – Загальний вигляд спеціалізованого блоку Revolute Joint схеми МРГ в програмному середовищі MATLAB [61-65]

Спеціалізований блок Transform Sensor [61-65] (див. рис. 4.11) дозволяє створювати необхідну затримку для виконання програми, що реалізується загальною схемою.

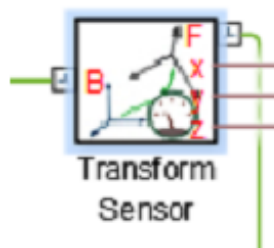


Рисунок 4.11 – Загальний вигляд спеціалізованого блоку Transform Sensor схеми МРГ в програмному середовищі MATLAB [61-65]

Спеціалізований блок Rigid Transform [61-65] (рис. 4.12) дозволяє

виконувати перетворення встановленої системи координат. Спочатку виконується поворот осей, а після цього виконується переміщення вихідної точки встановленої системи координат.

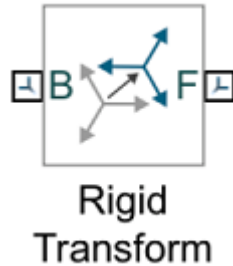


Рисунок 4.12 – Загальний вигляд спеціалізованого блоку Rigid Transform схеми МРГ в програмному середовищі MATLAB [61-65]

Спеціалізований блок PS-Simulink Converter [61-65] (рис. 4.13) – блок для перетворення фізичного сигналу (сигналу з будь-якого датчика, що входить до схеми) у вихідний сигнал SIMULINK [61-65].

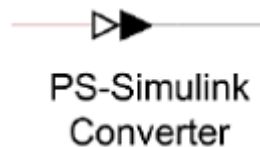


Рисунок 4.13 – Загальний вигляд спеціалізованого блоку PS-Simulink Converter схеми МРГ в програмному середовищі MATLAB [61-65]

Блок Scope [61-65] (осцилограф) (рис. 4.14) може виконувати відображення сигнал, що генерується під час виконання симуляції програмою.



Рисунок 4.14 – Загальний вигляд спеціалізованого блоку Scope схеми МРГ в програмному середовищі MATLAB [61-65]

Спеціалізований блок Signal Builder [61-65] (рис. 4.15) використовується для створення сигналу, який подається на кінематичний ланцюг між корпусом і ногами МРГ.

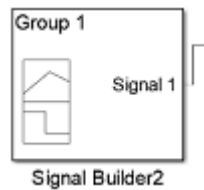


Рисунок 4.15 – Загальний вигляд спеціалізованого блоку Signal Builder схеми МРГ в програмному середовищі MATLAB [61-65]

Описаний набір блоків в програмному середовищі MATLAB [61-65] дозволяє створювати/знімати сигнали як з лівого, так і з правого боку моделі МРГ.

Для розробки програми відпрацювання сигналів спочатку в програмному середовищі MATLAB [61-65] було проведено розрахунок прикладу. Для цього були задані траєкторії руху МРГ в просторі, приклади яких наведено на рисунку 4.16.

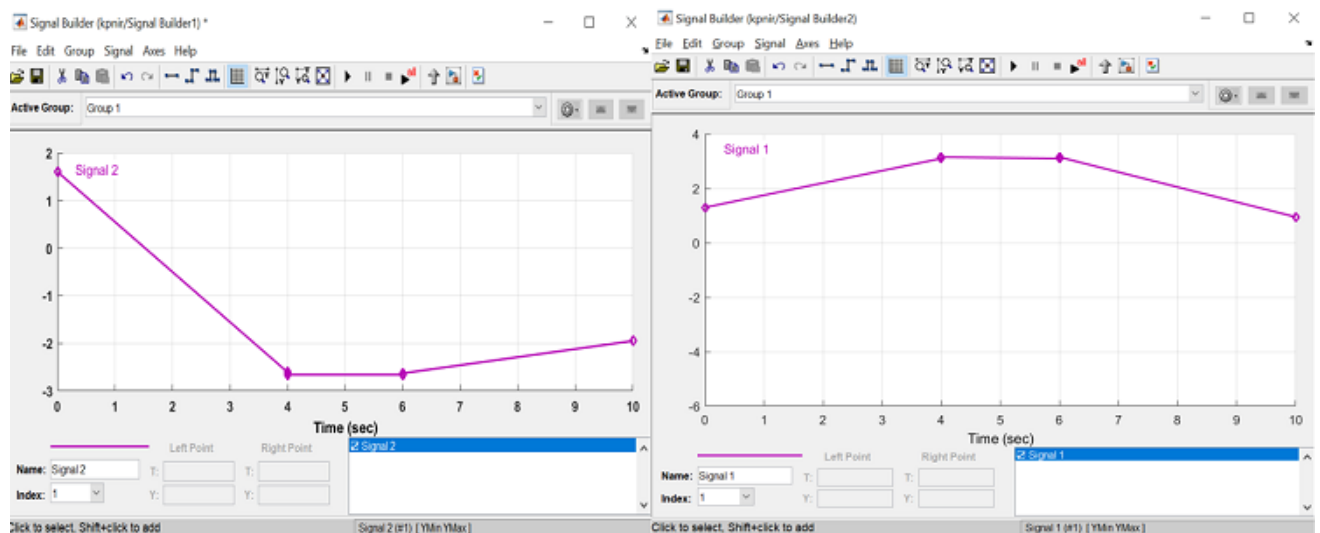


Рисунок 4.16 – Експериментальні траєкторії руху МРГ в просторі

В результаті численних віртуальних експериментів вдалося досягти

впевненого відпрацювання заданих траєкторій на моделі МРГ (див. рис. 4.17), а також стійкого його положення за умов, коли це можливо кінематично [38, 65].

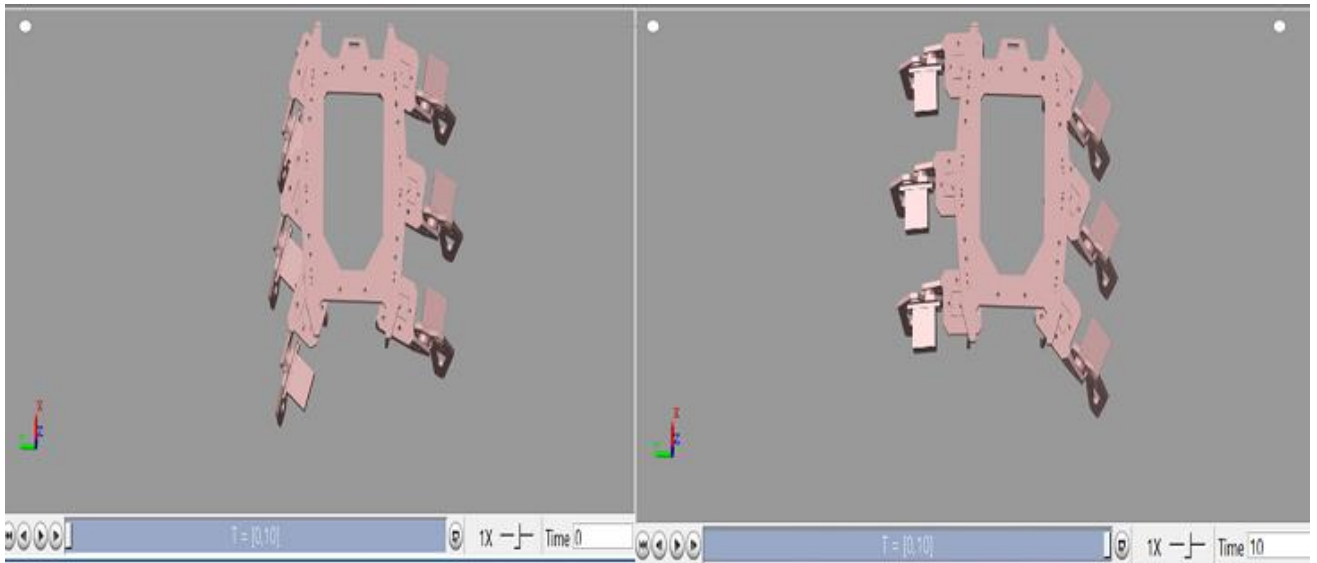


Рисунок 4.17 – Вигляд віртуального руху МРГ в різних точках траєкторій під час виконання дослідження

Було обрано як віртуальні, так і матеріальні засоби виготовлення та випробування МРГ. Відпрацьовано алгоритм, що дозволяє МРГ впевнено рухатися по будь-якій заданій траєкторії (поки лише віртуально в програмному середовищі).

4.2 Розроблення програмного забезпечення для моделі мехатронного робота-гексапода

З точки зору керування в цьому проєкті обрано керування МРГ розроблене на МК типу Arduino, тому, що програмування на МК Arduino [53, 54] відносно просте як в навчанні так і у використанні. Розроблений програмний код наведено у додатку А.

Проте, в майбутніх дослідженнях варто використовувати МК сімейства STM32 [66-69]. Такі МК STM32 [66-69] більш широко використовується і розширює можливості програмування. У цій конструкції МРГ відсутня реалізація функції обходу перешкод, тому в майбутніх дослідженнях слід додати конструкцію датчиків, щоб МРГ могли уникати виникаючі перешкоди.

Завдяки теоретичному дизайну виконаному в попередніх розділах кваліфікаційної роботи та із використанням розробленого програмного коду модель МРГ була остаточно виготовлена для перевірки можливостей роботи, загальний вигляд представлено на рисунку 4.18.

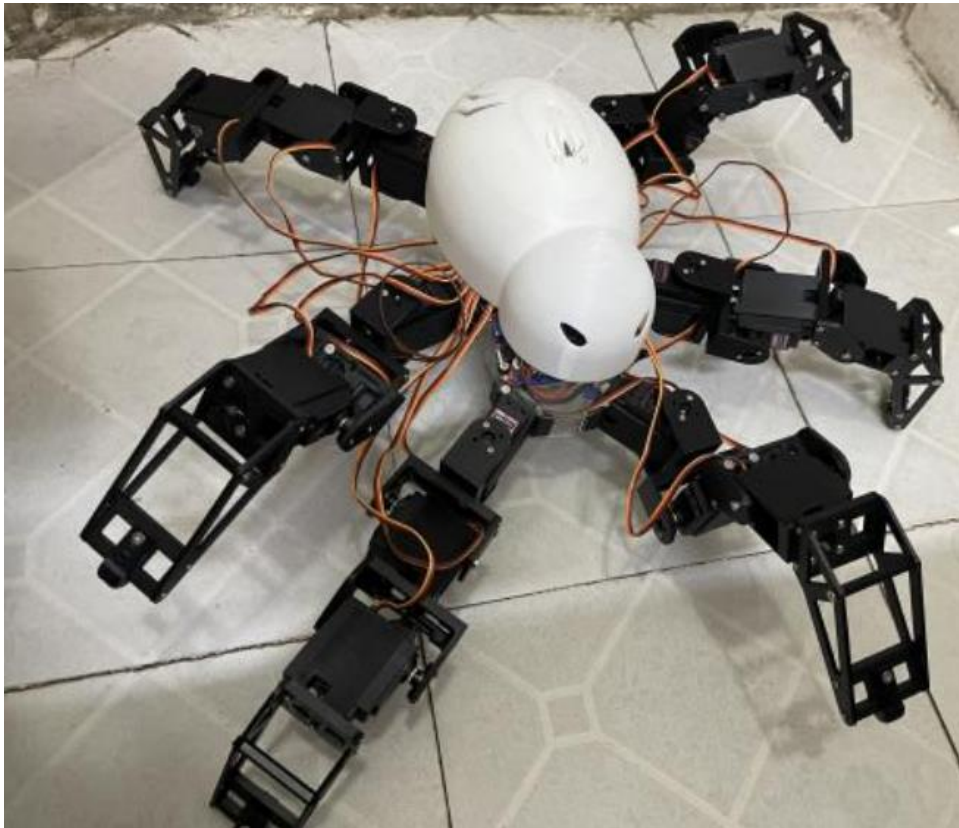


Рисунок 4.18 – Загальний вигляд моделі МРГ

Для виконання експериментального дослідження перевірки було використано два набори режимів прямого руху та повороту з фіксованою точкою. Рисунок 4.19 – виконання руху вперед, а рисунок 4.20 – виконання повороту вправо із фіксованою точкою.

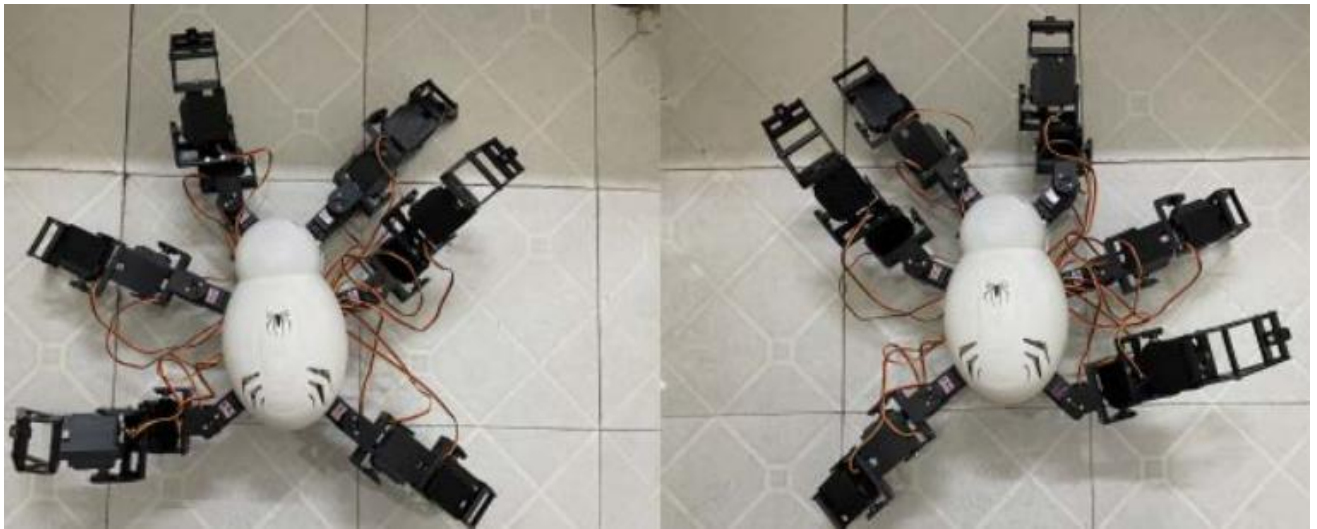


Рисунок 4.19 – Виконання руху вперед моделлю МРГ

З рисунків 4.19 видно, що під час виконання вправ використовується хода 3/6, три ноги піднімаються і приземляються, а потім рухаються вперед, а потім інші три ноги виконують рух.



Рисунок 4.20 – Виконання повороту вправо (за годинниковою стрілкою) з фіксованою точкою моделлю МРГ

Як видно з рисунку 4.20, коли корпус повертається, три ноги обертаються в одному напрямку, щоб піднятися і приземлитися, змушуючи МРГ обертатися всім корпусом.

4.3 Висновки до четвертого розділу

Проведено дослідження роботи розробленої моделі мехатронного робота-гексапода, а саме виконано моделювання роботи конструкції мехатронного робота-гексапода в програмному середовищі MATLAB. Розроблено програмне забезпечення для моделі мехатронного робота-гексапода для мікроконтролерів типу Arduino, в майбутніх дослідженнях необхідно використовувати МК сімейства STM32. Було виконано практичну перевірку на прототипі.

ВИСНОВКИ

В першому розділі кваліфікаційної роботи було описано проблематику проектування моделі мехатронного робота-гексапода, для чого виконано огляд та аналіз існуючих конструкцій мехатронних роботів-гексаподів. Базуючись на проведеному аналітичному огляді було проведено розроблення конструкції механічної частини моделі мехатронного робота-гексапода.

В другому розділі кваліфікаційної роботи магістра виконано моделювання конструкції, проведено перевірку загальних параметрів і міцності ключових конструкцій мехатронного робота-гексапода. Було проведено аналіз виконання руху, виконано дослідження варіантів виконання руху мехатронним роботом-гексаподом.

В третьому розділі проведено аналітичне дослідження стійкості ходи мехатронних роботів-гексаподів. Було виконано розроблення системи керування мехатронним роботом-гексаподом та підбір основних елементів керування. Вся керуюча частина МРГ реалізується шляхом проектування та аналізу сервоприводів Tower Pro MG996r, плати керування Arduino mega2560, ПДК - пульт від PS2, камери Nanying A8 та модуля живлення електроприводної частини LM2596S, а також із використанням програмування Arduino.

В четвертому розділі кваліфікаційної роботи проведено дослідження роботи розробленої моделі мехатронного робота-гексапода, а саме проведено моделювання роботи конструкції мехатронного робота-гексапода в програмному середовищі MATLAB. Розроблено програмне забезпечення для моделі мехатронного робота-гексапода для мікроконтролерів типу Arduino, в майбутніх дослідженнях необхідно використовувати МК сімейства STM32. Було виконано практичну перевірку на прототипі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Quinn RD, Nelson GM, Bachmann RJ. Parallel Complementary Strategies for Implementing Biological Principles into Mobile Robots [J]. The International Journal of Robotics Research. 2013, 22(3). 169-186р.
2. Робо-зірка, черепахо-бот і техно-акула: вчені створюють роботів, які вміють плавати, для дослідження океанів. URL: <https://focus.ua/uk/technologies/736115-chas-mozhe-vzagali-ne-isnuvati-shcho-kazhut-fiziki> (дата звернення: 15.09.2025).
3. M.C. Garcia-Lopez, E. Gorrostieta-Hurtado, Kinematic analysis for trajectory generation in one leg of a hexapod robot, Procedia Technology [J], 2012, 03. 342-350р.
4. NASA продемонструвало можливості робота-павука. URL: https://www.ukrinform.ua/rubric-other_news/1609760-nasa_prodeonstruvalo_moglivosti_robota_pavuka_video_1902746.html (дата звернення: 15.09.2025).
5. Research That Moves the World Forward. URL: <https://www.cmu.edu/> (дата звернення: 15.09.2025).
6. A New Way to Build Soft Robots That Can Walk on Water. URL: <https://www.techbriefs.com/component/content/article/54183-a-new-way-to-build-soft-robots-that-can-walk-on-water> (дата звернення: 15.09.2025).
7. Мехатронні системи в різних сферах виробничої діяльності. Лекція №7. Промислові та мобільні роботи. URL: <https://msn.khmnu.edu.ua/mod/resource/view.php?id=361813> (дата звернення: 15.09.2025).
8. Харбінський технологічний інститут, Міжнародний навчальний центр Вейхай. URL: <https://www.ncuk.ac.uk/uk/where-can-i-study/harbin-institute-of-technology-weihai-international-study-centre/> (дата звернення: 15.09.2025).

9. DEEP Robotics. URL: <https://www.deeprobotics.cn/en> (дата звернення: 15.09.2025).

10. Трет'як А. В., Кльон А. М. Основи робототехніки: навчальний посібник для студентів спеціальностей 133 «Галузеве машинобудування», 141 «Електроенергетика, електротехніка та електромеханіка», 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка». Полтава : Видавництво Національного університету «Полтавська політехніка імені Юрія Кондратюка», 2024. 135 с.

11. Струтинський В. Б., Гуржій А. М. Наземні роботизовані комплекси: монографія. Житомир : ПП «Рута», 2023. 524 с.

12. Цвіркун Л. І., Грулер Г. Робототехніка та мехатроніка: навч. посіб. / під заг. ред. Л. І. Цвіркуна ; М-во освіти і науки України, Нац. гірн. ун-т. 3-тє вид., переробл. і доповн. Дніпро : НГУ, 2017. 224 с. ISBN 978–966–350–645–6.

13. Солона О.В., Купчук І.М., Замрій М.А. Інтелектуальні мехатронні системи керування. Методичні вказівки до виконання практичних та самостійних робіт для студентів денної та заочної форми навчання другого (магістерського) освітнього рівня галузі знань 14 Електрична інженерія спеціальності 141 Електроенергетика, електротехніка та електромеханіка. Вінниця: ВНАУ, 2023. 110 с.

14. Поліщук М. М., Ткач М. М. Робототехнічні системи та комплекси: мобільні роботи довільної орієнтації: підручник для студентів спеціальності «Інформаційні системи та технології». Електронні текстові дані (1 файл: 14,7 Мбайт). Київ : КПІ ім. Ігоря Сікорського, 2023. 301 с. ISBN 978-966-990-076-0.

15. Основи еволюційної теорії: Навчальний посібник з дисципліни «Біологія розвитку та основи еволюційної теорії» для студентів спеціальності 162 – Біотехнології та біоінженерія спеціалізації «Промислова біотехнологія» / Уклад.: О.Ю. Галкін, Л.О. Тітова. Київ : КПІ імені Ігоря Сікорського, 2018. 121 с.

16. Коломієць В.О., Левадний О.М., Морозюк Ю.В. Методичні рекомендації до організації самостійної роботи з навчальної дисципліни

«Біонічна практика» (для здобувачів першого (бакалаврського) рівня вищої освіти денної форми навчання зі спеціальності 022 – Дизайн). Харків : ХНУМГ ім. О. М. Бекетова, 2021. 20 с.

17. Мигаль С. П., Дида І. А., Казанцева Т. Є. Біоніка в дизайні просторово-предметного середовища. Львів : Львівська політехніка, 2022. 228 с. ISBN: 978-966-941-696-4

18. Потапенко М.В. Основи ергономіки, біоніка: методичні рекомендації до практичних занять для здобувачів ступеня вищої освіти бакалавра спеціальності «Дизайн» освітньо-професійної програми «Графічний дизайн» Запоріжжя : ЗНУ, 2018. 56 с.

19. Гришанова І.А., Згуровська Л.П., Киричук Ю.В.. Розв'язок задач проектування приладів та систем з використанням ANSYS і MATHCAD : підручник. Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2022. 180 с.

20. Прибитько І.О., Руденко М.М., Ганєєв Т.Р., Нагорна І.В. Програмний комплекс ANSYS. Методичні вказівки до самостійної роботи для здобувачів другого (магістерського) рівня вищої освіти за освітньою програмою «Технології та устаткування зварювання» (освітній ступінь магістр). Чернігів : НУ «Чернігівська політехніка», 2023. 56 с.

21. Васильків В.В., Данильченко Л.М., Радик Д.Л., Дивдик О.В. Методичний практикум на тему: “Інженерний аналіз в Ansys Workbench” з дисципліни: “Комп’ютерне моделювання процесів обробки матеріалів“ для практичних занять і самостійної роботи здобувачів освітнього рівня доктор філософії за спеціальністю 131 “Прикладна механіка” та блоку вибіркових дисциплін інших спеціальностей. Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2021. 58 с.

22. Загорулько А.В. Програмний комплекс ANSYS в інженерних задачах: навч. посіб. Суми: Вид-во СумДУ, 2008. 201 с.

23. ANSYS Help. Documentation, tutorials, and videos for Ansys products. Mechanical APDL User’s Guide Documentation. ANSYS, Inc., 2023.

24. Thompson M., Thompson J. ANSYS Mechanical APDL for Finite Element Analysis. 1st Edition. Butterworth-Heinemann, 2017. 466 p.
25. Ansys Student - Free Software Download. ANSYS, Inc., 2023. URL: <https://www.ansys.com/academic/students> (дата звернення: 08.10.2025).
26. Заховайко О.П. Теорія механізмів і машин. Курс лекцій для студентів спеціальності «Динаміка і міцність машин». Київ : НТУУ "КПІ", 2010. 243 с.
27. Мороз В.І., Братченко О.В., Павшенко А.В. Теорія механізмів і машин: дослідження та проектування механізмів типових технічних засобів залізничного транспорту : Навч. посібник. Харків : УкрДАЗТ, 2013. 156 с. ISBN 978-966-2033-96-0
28. Бабенко Д.В., Доценко Н.А., Горбенко О.А. Теорія механізмів і машин: практикум для навчання в умовах інформаційно-освітнього середовища : навчальний посібник. Миколаїв : МНАУ, 2019. 168 с. ISBN 978–617–7149–39–1
29. Кінденко М.І. Теорія механізмів і машин: навчальний посібник. для студентів технічних спеціальностей всіх форм навчання. Краматорськ : ДДМА, 2018. 82 с.
30. Кіницький Я.Т., Харжевський В.О., Марченко М.В. Теорія механізмів і машин в системі Mathcad: Навчальний посібник. Хмельницький : РВЦ ХНУ, 2014. 324 с.
31. PLA Filament. Пластик для 3d принтера та 3d ручки. URL: <https://monofilament.com.ua/ua/products/standartnye-materialy/pla/?srsltid=AfmBOooCyzDkbM61qAkihGLQeARqAOQKkpesCFkyzztZ3eE06tcUSOt> (дата звернення: 08.10.2025).
32. Козяр М. М., Фещук Ю.В., Парфенюк О.В. Комп'ютерна графіка. SolidWorks : Навчальний посібник. Херсон : ОЛДІ-ПЛЮС, 2018. 252 с.
33. Ворощук В.Я., Вітенько Т.М. Solidworks у завданнях 3D моделювання та інжинірингу технічних систем. Навч. посібник. Тернопіль : ФОП Паляниця В.А., 2021. 164 с.

34. V. Kharzhevskiy, G. Kalda. Methods of design and calculation of machines and mechanisms. Wydawnictwo: POLITECHNIKA RZESZOWSKA, 2025. 305 p. ISBN: 978-83-7934-759-9.
35. Лістовщик Л.К. Основи геометричного моделювання в програмі solidworks .Частина 1. Навчальний посібник. Київ : НТУУ "КПІ", 2023. 71 с.
36. Dassault Systèmes Virtual Worlds Help You Improve Real Life. URL: <https://www.3ds.com/> (дата звернення: 08.10.2025).
37. Siemens. URL: <https://www.siemens.com/ua/uk.html> (дата звернення: 08.10.2025).
38. Zhiying Wang, X.-L. Ding, Alberto Rovetta. Analysis of typical locomotion of a symmetric hexapod robot. *Robotica*, 2010, 28(06). 893-907p. DOI:10.1017/S0263574709990725.
39. Пічкур В.В., Капустян О.В., Собчук В.В. Теорія динамічних систем : навч. посіб. Луцьк : Вежа-Друк, 2020. 348 с. ISBN 978-966-940-301-8.
40. Чихладзе Е.Д., Берестянська С.Ю., Лисяков І.М. Динамічні розрахунки конструкцій : підручник. Харків: УкрДУЗТ, 2015. – 161 с. ISBN 978-617-654-021-2.
41. Бородавка Є.В., Терентьев О.О., Гончаренко Т.А. Динамічне програмування. Підручник. — Київ : Компринт, 2025. 178 с.
42. Математичні методи дослідження операцій : підручник / Є. А. Лаврова та інші. Суми : Сумський державний університет, 2017. 212 с. ISBN 978-966-657-730-9.
43. Бартіш М.Я., Дудзяний І.М. Дослідження операцій. Частина 2. Алгоритми оптимізації на графах : Підручник. Львів: Видавничий центр ЛНУ імені Івана Франка, 2007. 120 с. ISBN 966-613-496-9, ISBN 978-966-613-533-2. Ч. 2
44. Павленко П.М., Філоненко С.Ф., Чередніков О.М., Трейтяк В.В. Математичне моделювання систем і процесів: навч. пос. Київ: НАУ, 2017. 392 с.

45. Математичне моделювання та оптимізація процесів і систем. Частина 1 [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за освітньою програмою «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» спеціальності 121 Інженерія програмного забезпечення / КПІ ім. Ігоря Сікорського ; уклад.: О. В. Барабаш, О. В. Свинчук, А. П. Мусієнко. Електронні текстові дані (1 файл: 3,92 Мбайт). Київ : КПІ ім. Ігоря Сікорського, 2023. 160 с.

46. Сервопривід TowerPro MG996R. URL: <https://www.mini-tech.com.ua/ua/servomotor-mg996> (дата звернення: 06.11.2025).

47. Цифровий високоточний сервопривід MG996R Tower Pro з металевими шестернями для роботів і моделей. URL: <https://arduinokit.com.ua/ua/p2360686198-tsifrovoj-vysokotochnyj-servoprivod.html?srsltid=AfmBOopjclESEeeyjnUD0NhKdXP1-Aj6dps7yZUWOioOV9C8vuKfGKOK> (дата звернення: 06.11.2025).

48. Arduino Mega 2560. URL: <https://doc.arduino.ua/hardware/Mega2560> (дата звернення: 06.11.2025).

49. Arduino Mega 2560 R3. URL: <https://www.mini-tech.com.ua/ua/arduino-mega-2560> (дата звернення: 06.11.2025).

50. Arduino Mega 2560 Rev3 ORIGINAL made in Italy. URL: <https://diylab.com.ua/p107671870-arduino-mega-2560.html> (дата звернення: 06.11.2025).

51. ATmega2560. URL: <https://www.microchip.com/en-us/product/atmega2560> (дата звернення: 06.11.2025).

52. Джойстик PlayStation Геймпад PS1 PS2 джойстик PS2 Playstation 2 ps 2. URL: https://prom.ua/ua/p1326774637-dzhojstik-playstation-gejmpad.html#:~:text=%D0%A5%D0%B0%D1%80%D0%B0%D0%BA%D1%82%D0%B5%D1%80%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B8:%20*%20%D0%A2%D0%B8%D0%BF:%20%D0%94%D0%B6%D0%BE%D0%B9%D1%81%D1%82%D0%B8%D0%BA%20*%20%D0%9F%D1%96%D0%B4%D1%94%D

0%B4%D0%BD%D0%B0%D0%BD%D0%BD%D1%8F:%20%D0%B4%D1%80%D0%BE%D1%82%D0%BE%D0%B2%D0%B5,*%20%D0%95%D1%80%D0%B3%D0%BE%D0%BD%D0%BE%D0%BC%D1%96%D0%BA%D0%B0:%20%D0%92%D0%B8%D0%B4%D1%96%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9%20%D0%A0%D0%A3%D0%94%20*%20%D0%96%D0%B8%D0%B2%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F:%20USB. (дата звернення: 06.11.2025).

53. Bring Your Projects to Life with Arduino Software. URL: <https://www.arduino.cc/en/software/> (дата звернення: 06.11.2025).

54. Початок роботи з Ардуїно на Windows. URL: <https://doc.arduino.ua/guide/Windows> (дата звернення: 06.11.2025).

55. Камера А8 4G. URL: <https://primechoice.com.ua/ua/p2385272125-avtonomnaya-ohotnichya-kamera.html?srsltid=AfmBOoqTC7q&dD2EQ1m9wLwVSgpWwNNGnxBpsSrUR4r77hpTa18LT0g> (дата звернення: 06.11.2025).

56. Акумулятор 2s lipo. URL: <https://prom.ua/ua/Akkumulyator-2s-lipo.html> (дата звернення: 06.11.2025).

57. Акумулятор GOLDBAT LiPo 4000 мАг 7,4 В 80С 2S для радіокерованих машин та траків. URL: <https://prom.ua/ua/m7427578108310079399-akkumulyator-goldbat-lipo.html?p=2733475052> (дата звернення: 06.11.2025).

58. LM2596S-ADJ - DC-DC перетворювач TO-263. URL: <https://www.minitech.com.ua/ua/dc-dc-preobrazovatel-lm2596s-adj-smd> (дата звернення: 06.11.2025).

59. Регулятор напруги LM2596S 1.25-35V 3A знижуючий. URL: <https://radioland.com.ua/ua/products/rts-lm2596s-reguljator-naprjazhenija-lm2596s-1-25-35v-3a-ponizhajuschij> (дата звернення: 06.11.2025).

60. Інтерфейсна плата Pimoroni HAT Hacker. URL: https://mikronika.net/product/pimoroni-raspberry-pi-hat-hacker?srsltid=AfmBOop9KhHkRUMGUYkUqyiidg3_lsHHFToHF2IgvAM375c6bHHPmg (дата звернення: 19.11.2025).

61. Лазарев Ю. Ф. Моделювання динамічних систем у Matlab: електронний навчальний посібник. Київ : НТУУ «КПІ», 2011. 421 с.
62. Лазарев Ю. Ф. Довідник з MATLAB: електронний навчальний посібник з курсового і дипломного проектування. Київ : НТУУ «КПІ», 2013. 132 с.
63. Гоблик Н. М., Гоблик В. В. MATLAB в інженерних розрахунках. Комп'ютерний практикум: навчальний посібник. 3-тє вид., доповн. Львів : Видавництво Львівської політехніки, 2020. 190 с.: іл., табл., схеми, графіки. ISBN 978-966-941-510-3.
64. Simscape Multibody. URL: <https://uk.mathworks.com/products/simscape-multibody.html> (дата звернення: 19.11.2025).
65. K.A. Melsaac, J.P. Ostrowski. A geometric approach to anguilliform locomotion: modelling of an underwater eel robot [J]. Robotics and Automation, 2012.
66. STM32 для новачків: обираємо плати, середовища та перші кроки. URL: <https://dou.ua/forums/topic/52283/> (дата звернення: 19.11.2025).
67. STM32 32-bit Arm Cortex MCUs. URL: <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html> (дата звернення: 19.11.2025).
68. Мікроконтролери STM32 – вступ. URL: <https://itmaster.biz.ua/electronics/stm32/stm32-intro.html> (дата звернення: 19.11.2025).
69. STM плати. URL: <https://www.mini-tech.com.ua/ua/otladochnye-platy/stm-platy/> (дата звернення: 19.11.2025).
70. Мартинюк В.В., Корецька Л.О., Федула М.В., Савенко О.С. Кваліфікаційна робота : метод. рек. і настанови до виконання кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти за спец. 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка». Хмельницький : ХНУ, 2024. 25 с.

ДОДАТОК А (обов'язковий)

ПРОГРАМНИЙ КОД ДЛЯ КЕРУВАННЯ МОДЕЛЛЮ МРГ

```

//*****
// Includes
//*****
#include <PS2X_lib.h> //reference: http://www.billporter.info/
#include <Servo.h>
#include <math.h>

//*****
// Constant Declarations          常量声明
//*****

const int PS2_DAT = 2;           //gamepad port definitions          gamepad端口定义
const int PS2_ATT = 3;
const int PS2_CMD = 4;
const int PS2_CLK = 5;
const int RUMBLE = true;
const int PRESSURES = false;

const int COXA1_SERVO = 19;      //servo port definitions          伺服端口定义
const int FEMUR1_SERVO = 21;
const int TIBIA1_SERVO = 23;
const int COXA2_SERVO = 25;
const int FEMUR2_SERVO = 27;
const int TIBIA2_SERVO = 29;
const int COXA3_SERVO = 31;
const int FEMUR3_SERVO = 33;
const int TIBIA3_SERVO = 35;
const int COXA4_SERVO = 37;
const int FEMUR4_SERVO = 39;
const int TIBIA4_SERVO = 41;
const int COXA5_SERVO = 43;
const int FEMUR5_SERVO = 45;
const int TIBIA5_SERVO = 47;
const int COXA6_SERVO = 49;
const int FEMUR6_SERVO = 51;
const int TIBIA6_SERVO = 53;

const int COXA_LENGTH = 51;     //leg part lengths          腿部长度
const int FEMUR_LENGTH = 65;
const int TIBIA_LENGTH = 121;

const int TRAVEL = 30;         //translate and rotate travel limit constant          平移和旋转行程限制常数

const long A12DEG = 209440;    //12 degrees in radians x 1,000,000          12度弧度x 1000000
const long A30DEG = 523599;   //30 degrees in radians x 1,000,000          30度弧度x 1000000

const int FRAME_TIME_MS = 20; //frame time (20msec = 50Hz)          帧时间 (20毫秒=50赫兹)

const float HOME_X[6] = { 82.0, 0.0, -82.0, -82.0, 0.0, 82.0}; //coxa-to-toe home positions          髋至足位置
const float HOME_Y[6] = { 82.0, 116.0, 82.0, -82.0, -116.0, -82.0};
const float HOME_Z[6] = { -80.0, -80.0, -80.0, -80.0, -80.0, -80.0};

const float BODY_X[6] = { 110.4, 0.0, -110.4, -110.4, 0.0, 110.4}; //body center-to-coxa servo distances
const float BODY_Y[6] = { 58.4, 90.8, 58.4, -58.4, -90.8, -58.4};
const float BODY_Z[6] = { 0.0, 0.0, 0.0, 0.0, 0.0, 0.0};

const int COXA_CAL[6] = {-5, -5, -3, -4, -6, -4}; //加为逆时针

```

```

const int FEMUR_CAL[6] = {-2, 1, -6, -3, -5, 0}; // 加为上
const int TIBIA_CAL[6] = {8, 10, 10, 6, 0, 0}; // 加为外

//*****
// Variable Declarations 说明变量
//*****
int gamepad_error; //gamepad variables 游戏板变量
byte gamepad_type;
byte gamepad_vibrate; //游戏板振动

unsigned long currentTime; //frame timer variables 帧计时器变量
unsigned long previousTime;

int temp; //mode and control variables 模式和控制变量
int mode;
int gait; //步态
int gait_speed; //步态速度
int reset_position; //重置位置
int capture_offsets; //捕获偏移

float L0, L3; //inverse kinematics variables 反向运动学变量
float gamma_femur;
float phi_tibia, phi_femur; // phi 希腊字母表的第21个字母
float theta_tibia, theta_femur, theta_coxa; //膝盖θ

int leg1_IK_control, leg6_IK_control; //leg lift mode variables 抬腿模式变量
float leg1_coxa, leg1_femur, leg1_tibia;
float leg6_coxa, leg6_femur, leg6_tibia;

int leg_num; //positioning and walking variables 定位和行走变量
int z_height_LED_color;
int totalX, totalY, totalZ;
int tick, duration, numTicks;
int z_height_left, z_height_right;
int commandedX, commandedY, commandedR;
int translateX, translateY, translateZ; //平移
float step_height_multiplier;
float strideX, strideY, strideR; //跨步X
float sinRotX, sinRotY, sinRotZ;
float cosRotX, cosRotY, cosRotZ;
float rotOffsetX, rotOffsetY, rotOffsetZ; //旋转偏移X
float amplitudeX, amplitudeY, amplitudeZ; //振幅X
float offset_X[6], offset_Y[6], offset_Z[6]; //偏移量X
float current_X[6], current_Y[6], current_Z[6];

int tripod_case[6] = {1,2,1,2,1,2}; //for tripod gait walking 三脚架步态行走
int ripple_case[6] = {2,6,4,1,3,5}; //for ripple gait 纹波步态
int wave_case[6] = {1,2,3,4,5,6}; //for wave gait 波浪步态
int tetrapod_case[6] = {1,3,2,1,2,3}; //for tetrapod gait 四足步态

//*****
// Object Declarations 对象声明
//*****
PS2X ps2x; //PS2 gamepad controller PS2游戏板控制器

Servo coxal_servo; //18 servos 18舵机
Servo femur1_servo;
Servo tibial_servo;
Servo coxa2_servo;
Servo femur2_servo;

```

```

Servo femur2_servo;
Servo tibia2_servo;
Servo coxa3_servo;
Servo femur3_servo;
Servo tibia3_servo;
Servo coxa4_servo;
Servo femur4_servo;
Servo tibia4_servo;
Servo coxa5_servo;
Servo femur5_servo;
Servo tibia5_servo;
Servo coxa6_servo;
Servo femur6_servo;
Servo tibia6_servo;

//*****
// Initialization Routine      初始化例程
//*****
void setup()
{
  //start serial
  Serial.begin(9600);

  //attach servos
  coxa1_servo.attach(COXA1_SERVO, 500, 2500);
  femur1_servo.attach(FEMUR1_SERVO, 500, 2500);
  tibia1_servo.attach(TIBIA1_SERVO, 500, 2500);
  coxa2_servo.attach(COXA2_SERVO, 500, 2500);
  femur2_servo.attach(FEMUR2_SERVO, 500, 2500);
  tibia2_servo.attach(TIBIA2_SERVO, 500, 2500);
  coxa3_servo.attach(COXA3_SERVO, 500, 2500);
  femur3_servo.attach(FEMUR3_SERVO, 500, 2500);
  tibia3_servo.attach(TIBIA3_SERVO, 500, 2500);
  coxa4_servo.attach(COXA4_SERVO, 500, 2500);
  femur4_servo.attach(FEMUR4_SERVO, 500, 2500);
  tibia4_servo.attach(TIBIA4_SERVO, 500, 2500);
  coxa5_servo.attach(COXA5_SERVO, 500, 2500);
  femur5_servo.attach(FEMUR5_SERVO, 500, 2500);
  tibia5_servo.attach(TIBIA5_SERVO, 500, 2500);
  coxa6_servo.attach(COXA6_SERVO, 500, 2500);
  femur6_servo.attach(FEMUR6_SERVO, 500, 2500);
  tibia6_servo.attach(TIBIA6_SERVO, 500, 2500);

  //connect the gamepad      连接游戏板
  gamepad_error = ps2x.config_gamepad(PS2_CLK, PS2_CMD, PS2_ATT, PS2_DAT, PRESSURES, RUMBLE);
  if(gamepad_error == 0)      Serial.println("Controller attached");
  else if(gamepad_error == 1) Serial.println("No controller found");
  else if(gamepad_error == 2) Serial.println("Controller found but not accepting commands");
  else if(gamepad_error == 3) Serial.println("Controller refusing to enter Pressures mode");

  //verify the gamepad type      验证游戏板类型
  gamepad_type = ps2x.readType();
  if(gamepad_type == 0)      Serial.println("Unknown Controller type found");
  else if(gamepad_type == 1) Serial.println("DualShock Controller found");
  else if(gamepad_type == 2) Serial.println("GuitarHero Controller found");
  else if(gamepad_type == 3) Serial.println("Wireless Sony DualShock Controller found");

  //turn off gamepad vibration      关闭游戏板震动
  gamepad_vibrate = 0;
}

```

```

//clear offsets          清除偏移量
for (leg_num=0; leg_num<6; leg_num++)
{
  offset_X[leg_num] = 0.0;
  offset_Y[leg_num] = 0.0;
  offset_Z[leg_num] = 0.0;
}
capture_offsets = false;
step_height_multiplier = 1.0;

//initialize mode and gait variables      初始化模式和步态变量
mode = 0;
gait = 0;
gait_speed = 0;
reset_position = true;
leg1_IK_control = true;
leg6_IK_control = true;
}

//*****
// Main Program      主程序
//*****
void loop()
{
  //exit if no controller found or GuitarHero controller      如果找不到控制器或GuitarHero控制器, 则退
  if ((gamepad_error == 1) || (gamepad_type == 2))
  {
    Serial.println("Invalid Controller!");
    return;
  }

  //set up frame time      设置帧时间
  currentTime = millis();
  if ((currentTime - previousTime) > FRAME_TIME_MS)
  {
    previousTime = currentTime;

    //read controller and process inputs      读取控制器和过程输入
    ps2x.read_gamepad(false, gamepad_vibrate);
    process_gamepad();

    //reset legs to home position when commanded      根据指令将腿复位到原位
    if (reset_position == true)
    {
      for (leg_num=0; leg_num<6; leg_num++)
      {
        current_X[leg_num] = HOME_X[leg_num];
        current_Y[leg_num] = HOME_Y[leg_num];
        current_Z[leg_num] = HOME_Z[leg_num];
      }
      reset_position = false;
    }

    //position legs using IK calculations - unless set all to 90 degrees mode      使用IK计算定位
    if (mode < 99)
    {
      for (leg_num=0; leg_num<6; leg_num++)
        leg_IK(leg_num, current_X[leg_num]+offset_X[leg_num], current_Y[leg_num]+offset_Y[leg_num],

```

```

//reset leg lift first pass flags if needed      如果需要, 重置抬腿首次标志
if(mode != 4)
{
    leg1_IK_control = true;
    leg6_IK_control = true;
}

//battery_monitor();                          //battery monitor and output to LEDs      电池监视器和LED输出
print_debug();                                //print debug data                          打印调试数据

//process modes (mode 0 is default 'home idle' do-nothing mode)      进程模式 (模式0是默认的"主空闲"不执行任何操作模式)
if(mode == 1)                                //walking mode                                步行模式
{
    if(gait == 0) tripod_gait();              //walk using gait 0                          使用步态0行走
    if(gait == 1) wave_gait();                //walk using gait 1                          使用步态1行走
    if(gait == 2) ripple_gait();              //walk using gait 2                          使用步态2行走
    if(gait == 3) tetrapod_gait();            //walk using gait 3                          使用步态3行走
}
if(mode == 2) translate_control();            //joystick control x-y-z mode                  操纵手柄控制x-y-z模式
if(mode == 3) rotate_control();              //joystick control y-p-r mode                  操纵手柄控制y-p-r模式
if(mode == 4) one_leg_lift();                 //one leg lift mode                            单腿抬升模式
if(mode == 99) set_all_90();                  //set all servos to 90 degrees mode            将所有伺服设置为90度模式
}

*****
Process gamepad controller inputs             处理游戏手柄控制器输入
*****
id process_gamepad()

if(ps2x.ButtonPressed(PSB_PAD_DOWN))         //stop & select gait 0                        停止并选择步态0
{
    mode = 0;
    gait = 0;
    reset_position = true;
}
if(ps2x.ButtonPressed(PSB_PAD_LEFT))         //stop & select gait 1                        停止并选择步态1
{
    mode = 0;
    gait = 1;
    reset_position = true;
}
if(ps2x.ButtonPressed(PSB_PAD_UP))           //stop & select gait 2                        停止并选择步态2
{
    mode = 0;
    gait = 2;

    reset_position = true;
}
if(ps2x.ButtonPressed(PSB_PAD_RIGHT))        //stop & select gait 3                        停止并选择步态3
{
    mode = 0;
    gait = 3;
    reset_position = true;
}

if(ps2x.ButtonPressed(PSB_TRIANGLE))         //select walk mode                            选择步行模式
{
    mode = 1;
    reset_position = true;
}
if(ps2x.Button(PSB_TRIANGLE))                 //vibrate controller if walk button held      如果按下行走按钮, 则震动控制器
    gamepad_vibrate = 64;
else
    gamepad_vibrate = 0;
if(ps2x.ButtonPressed(PSB_SQUARE))           //control x-y-z with joysticks mode            使用操纵手柄模式控制x-y-z
{

```

```

else
  gameped_vibrate = 0;
if (ps2x.ButtonPressed(PSB_SQUARE)) //control x-y-z with joysticks mode          使用操纵手柄模式控制x-y-z
{
  mode = 2;
  reset_position = true;
}
if (ps2x.ButtonPressed(PSB_CIRCLE)) //control y-p-r with joysticks mode          使用操纵手柄模式控制y-p-r
{
  mode = 3;
  reset_position = true;
}
if (ps2x.ButtonPressed(PSB_CROSS)) //one leg lift mode                          单腿提升模式
{
  mode = 4;
  reset_position = true;
}
if (ps2x.ButtonPressed(PSB_START)) //change gait speed                          改变步态速度
{
  if (gait_speed == 0)
    gait_speed = 1;
  else
    gait_speed = 0;
}

if (ps2x.ButtonPressed(PSB_SELECT)) //set all servos to 90 degrees for calibration 将所有伺服设置为90度进行校准
{
  mode = 99;
}
if ((ps2x.ButtonPressed(PSB_L1)) || (ps2x.ButtonPressed(PSB_R1)))
{
  //capture offsets in translate, rotate, and translate/rotate modes          在平移、旋转和平移/旋转模式下捕获偏移
  capture_offsets = true;
}
if ((ps2x.ButtonPressed(PSB_L3)) || (ps2x.ButtonPressed(PSB_R3)))
{
  for (leg_num=0; leg_num<6; leg_num++) //clear offsets                          清除偏移量
  {
    offset_X[leg_num] = 0;
    offset_Y[leg_num] = 0;
    offset_Z[leg_num] = 0;
  }
  leg1_IK_control = true; //reset leg lift first pass flags                      重置抬腿首次标志
  leg6_IK_control = true;
  step_height_multiplier = 1.0; //reset step height multiplier                  重置步态高度倍增器
}
}

//*****
// Leg IK Routine
//*****
void leg_IK(int leg_number, float X, float Y, float Z)
{
  //compute target femur-to-toe (L3) length          计算目标股骨到脚趾(L3)长度
  L0 = sqrt(sq(X) + sq(Y)) - COXA_LENGTH;
  L3 = sqrt(sq(L0) + sq(Z));

  //process only if reach is within possible range (not too long or too short!)          仅当到达范围在可能范围内(不超长或太短!)时进行处理
  if ((L3 < (TIBIA_LENGTH+FEMUR_LENGTH)) && (L3 > (TIBIA_LENGTH-FEMUR_LENGTH)))
  {
    //compute tibia angle          计算胫骨角度
    phi_tibia = acos((sq(FEMUR_LENGTH) + sq(TIBIA_LENGTH) - sq(L3))/(2*FEMUR_LENGTH*TIBIA_LENGTH));
    theta_tibia = phi_tibia*RAD_TO_DEG - 23.0 + TIBIA_CAL[leg_number];
    theta_tibia = constrain(theta_tibia,0.0,180.0);

    //compute femur angle          计算股骨角度
    gamma_femur = atan2(Z,L0);
    phi_femur = acos((sq(FEMUR_LENGTH) + sq(L3) - sq(TIBIA_LENGTH))/(2*FEMUR_LENGTH*L3));
    theta_femur = (phi_femur + gamma_femur)*RAD_TO_DEG + 14.0 + 90.0 + FEMUR_CAL[leg_number];
    theta_femur = constrain(theta_femur,0.0,180.0);

    //compute coxa angle          计算髋关节角度
    theta_coxa = atan2(X,Y)*RAD_TO_DEG + COXA_CAL[leg_number];
  }
}

```

```

//compute femur angle          计算股骨角度
gamma_femur = atan2(Z,L0);
phi_femur = acos((sq(FEMUR_LENGTH) + sq(L3) - sq(TIBIA_LENGTH))/(2*FEMUR_LENGTH*L3));
theta_femur = (phi_femur + gamma_femur)*RAD_TO_DEG + 14.0 + 90.0 + FEMUR_CAL[leg_number];
theta_femur = constrain(theta_femur,0.0,180.0);

//compute coxa angle          计算髋关节角度
theta_coxa = atan2(X,Y)*RAD_TO_DEG + COXA_CAL[leg_number];

//output to the appropriate leg      输出到相应的分支
switch(leg_number)
{
  case 0:
    if(leg1_IK_control == true)          //flag for IK or manual control of leg      IK或手动控制腿的标志
    {
      theta_coxa = theta_coxa + 45.0;          //compensate for leg mounting      补偿支腿安装
      theta_coxa = constrain(theta_coxa,0.0,180.0);
      coxal_servo.write(int(theta_coxa));
      femur1_servo.write(int(theta_femur));
      tibial_servo.write(int(theta_tibia));
    }
    break;
  case 1:
    theta_coxa = theta_coxa + 90.0;          //compensate for leg mounting      补偿支腿安装
    theta_coxa = constrain(theta_coxa,0.0,180.0);
    coxal2_servo.write(int(theta_coxa));
    femur2_servo.write(int(theta_femur));
    tibial2_servo.write(int(theta_tibia));
    break;
  case 2:
    theta_coxa = theta_coxa + 135.0;          //compensate for leg mounting      补偿支腿安装
    theta_coxa = constrain(theta_coxa,0.0,180.0);
    coxal3_servo.write(int(theta_coxa));
    femur3_servo.write(int(theta_femur));
    tibial3_servo.write(int(theta_tibia));
    break;
  case 3:
    if(theta_coxa < 0)          //compensate for leg mounting      补偿支腿安装
      theta_coxa = theta_coxa + 225.0;          // (need to use different      需要使用不同的
    else          // positive and negative offsets      正负偏移量
      theta_coxa = theta_coxa - 135.0;          // due to atan2 results above!      由于上述atan2结果
    theta_coxa = constrain(theta_coxa,0.0,180.0);
    coxal4_servo.write(int(theta_coxa));
    femur4_servo.write(int(theta_femur));
    tibial4_servo.write(int(theta_tibia));
    break;
  case 4:
    if(theta_coxa < 0)          //compensate for leg mounting
      theta_coxa = theta_coxa + 270.0;          // (need to use different
    else          // positive and negative offsets
      theta_coxa = theta_coxa - 90.0;          // due to atan2 results above!
    theta_coxa = constrain(theta_coxa,0.0,180.0);
    coxal5_servo.write(int(theta_coxa));
    femur5_servo.write(int(theta_femur));
    tibial5_servo.write(int(theta_tibia));
    break;
  case 5:
    if(leg6_IK_control == true)          //flag for IK or manual control of leg
    {
  case 5:
    if(leg6_IK_control == true)          //flag for IK or manual control of leg
    {

```

```

    }
    if(theta_coxa < 0) //compensate for leg mounting
        theta_coxa = theta_coxa + 315.0; // (need to use different
    else // positive and negative offsets
        theta_coxa = theta_coxa - 45.0; // due to atan2 results above!)
    theta_coxa = constrain(theta_coxa,0.0,180.0);
    coxa6_servo.write(int(theta_coxa));
    femur6_servo.write(int(theta_femur));
    tibia6_servo.write(int(theta_tibia));
}
break;
}
}
}

//*****
// Tripod Gait 三脚架步态
// Group of 3 legs move forward while the other 3 legs provide support 一组三条腿向前移动, 另外三条腿提供支持
//*****
void tripod_gait()
{
    //read commanded values from controller 从控制器读取命令值
    commandedX = map(ps2x.Analog(PSS_RX),0,255,127,-127);
    commandedY = map(ps2x.Analog(PSS_RY),0,255,-127,127);
    commandedR = map(ps2x.Analog(PSS_LX),0,255,127,-127);

    //if commands more than deadband then process 如果命令超过死区, 则处理
    if((abs(commandedX) > 15) || (abs(commandedY) > 15) || (abs(commandedR) > 15) || (tick>0))
    {
        compute_strides();
        numTicks = round(duration / FRAME_TIME_MS / 2.0); //共分为2种情况
        for(leg_num=0; leg_num<6; leg_num++)
        {
            compute_amplitudes();
            switch(tripod_case[leg_num])
            {
                case 1: //move foot forward (raise and lower) 向前移动脚 (升高和降低)
                    current_X[leg_num] = HOME_X[leg_num] - amplitudeX*cos(M_PI*tick/numTicks);
                    current_Y[leg_num] = HOME_Y[leg_num] - amplitudeY*cos(M_PI*tick/numTicks);
                    current_Z[leg_num] = HOME_Z[leg_num] + abs(amplitudeZ)*sin(M_PI*tick/numTicks);
                    if(tick >= numTicks-1) tripod_case[leg_num] = 2;
                    break;
                case 2: //move foot back (on the ground) 把脚向后移 (在地上)
                    current_X[leg_num] = HOME_X[leg_num] + amplitudeX*cos(M_PI*tick/numTicks);
                    current_Y[leg_num] = HOME_Y[leg_num] + amplitudeY*cos(M_PI*tick/numTicks);
                    current_Z[leg_num] = HOME_Z[leg_num];
                    if(tick >= numTicks-1) tripod_case[leg_num] = 1;
                    break;
            }
        }
        //increment tick 增量记号
        if(tick < numTicks-1) tick++;
        else tick = 0;
    }
}

//*****
// Wave Gait 波浪步态
// Legs move forward one at a time while the other 5 legs provide support 两条腿一次向前移动一条腿, 另外5条腿提供支持
//*****
void wave_gait()
{
    //read commanded values from controller
    commandedX = map(ps2x.Analog(PSS_RX),0,255,127,-127);
    commandedY = map(ps2x.Analog(PSS_RY),0,255,-127,127);
}

```

```

//if commands more than deadband then process
if((abs(commandedX) > 15) || (abs(commandedY) > 15) || (abs(commandedR) > 15) || (tick>0))
{
  compute_strides();
  numTicks = round(duration / FRAME_TIME_MS / 6.0); //total ticks divided into the six cases      共分为6种情况
  for(leg_num=0; leg_num<6; leg_num++)
  {
    compute_amplitudes();
    switch(wave_case[leg_num])
    {
      case 1: //move foot forward (raise and lower)      向前移动脚 (升高和降低)
        current_X[leg_num] = HOME_X[leg_num] - amplitudeX*cos(M_PI*tick/numTicks);
        current_Y[leg_num] = HOME_Y[leg_num] - amplitudeY*cos(M_PI*tick/numTicks);
        current_Z[leg_num] = HOME_Z[leg_num] + abs(amplitudeZ)*sin(M_PI*tick/numTicks);
        if(tick >= numTicks-1) wave_case[leg_num] = 6;
        break;
      case 2: //move foot back one-fifth (on the ground)      把脚向后移五分之一 (在地上)
        current_X[leg_num] = current_X[leg_num] - amplitudeX/numTicks/2.5;
        current_Y[leg_num] = current_Y[leg_num] - amplitudeY/numTicks/2.5;
        current_Z[leg_num] = HOME_Z[leg_num];
        if(tick >= numTicks-1) wave_case[leg_num] = 1;
        break;
      case 3: //move foot back one-fifth (on the ground)      把脚向后移五分之一 (在地上)
        current_X[leg_num] = current_X[leg_num] - amplitudeX/numTicks/2.5;
        current_Y[leg_num] = current_Y[leg_num] - amplitudeY/numTicks/2.5;
        current_Z[leg_num] = HOME_Z[leg_num];
        if(tick >= numTicks-1) wave_case[leg_num] = 2;
        break;
      case 4: //move foot back one-fifth (on the ground)
        current_X[leg_num] = current_X[leg_num] - amplitudeX/numTicks/2.5;
        current_Y[leg_num] = current_Y[leg_num] - amplitudeY/numTicks/2.5;
        current_Z[leg_num] = HOME_Z[leg_num];
        if(tick >= numTicks-1)
          wave_case[leg_num] = 3;
        break;
      case 5: //move foot back one-fifth (on the ground)
        current_X[leg_num] = current_X[leg_num] - amplitudeX/numTicks/2.5;
        current_Y[leg_num] = current_Y[leg_num] - amplitudeY/numTicks/2.5;
        current_Z[leg_num] = HOME_Z[leg_num];
        if(tick >= numTicks-1) wave_case[leg_num] = 4;
        break;
      case 6: //move foot back one-fifth (on the ground)
        current_X[leg_num] = current_X[leg_num] - amplitudeX/numTicks/2.5;
        current_Y[leg_num] = current_Y[leg_num] - amplitudeY/numTicks/2.5;
        current_Z[leg_num] = HOME_Z[leg_num];
        if(tick >= numTicks-1) wave_case[leg_num] = 5;
        break;
    }
  }
  //increment tick
  if(tick < numTicks-1) tick++;
  else tick = 0;
}
}

//*****
// Ripple Gait      波纹步态
// Left legs move forward rear-to-front while right also do the same,      左腿从后向前移动, 右腿也一样,
// but right side is offset so RR starts midway through the LM stroke      但是右腿是偏移的, 所以RR在LM冲程的中间开始
//*****
void ripple_gait()
{
  //read commanded values from controller

```

```

commandedX = map(ps2x.Analog(PSS_RX),0,255,127,-127);
commandedY = map(ps2x.Analog(PSS_RY),0,255,-127,127);
commandedR = map(ps2x.Analog(PSS_LX),0,255,127,-127);

//if commands more than deadband then process
if((abs(commandedX) > 15) || (abs(commandedY) > 15) || (abs(commandedR) > 15) || (tick>0))
{
  compute_strides();
  numTicks = round(duration / FRAME_TIME_MS / 6.0); //total ticks divided into the six cases
  for(leg_num=0; leg_num<6; leg_num++)
  {
    compute_amplitudes();
    switch(ripple_case[leg_num])
    {
      case 1: //move foot forward (raise) 向前移动脚 (抬起)
        current_X[leg_num] = HOME_X[leg_num] - amplitudeX*cos(M_PI*tick/(numTicks*2));
        current_Y[leg_num] = HOME_Y[leg_num] - amplitudeY*cos(M_PI*tick/(numTicks*2));
        current_Z[leg_num] = HOME_Z[leg_num] + abs(amplitudeZ)*sin(M_PI*tick/(numTicks*2));
        if(tick >= numTicks-1) ripple_case[leg_num] = 2;
        break;
      case 2: //move foot forward (lower) 向前移动脚 (降低)
        current_X[leg_num] = HOME_X[leg_num] - amplitudeX*cos(M_PI*(numTicks+tick)/(numTicks*2));
        current_Y[leg_num] = HOME_Y[leg_num] - amplitudeY*cos(M_PI*(numTicks+tick)/(numTicks*2));
        current_Z[leg_num] = HOME_Z[leg_num] + abs(amplitudeZ)*sin(M_PI*(numTicks+tick)/(numTicks*2));
        if(tick >= numTicks-1) ripple_case[leg_num] = 3;
        break;
      case 3: //move foot back one-quarter (on the ground) 将脚向后移动四分之一 (在地面上)
        current_X[leg_num] = current_X[leg_num] - amplitudeX/numTicks/2.0;
        current_Y[leg_num] = current_Y[leg_num] - amplitudeY/numTicks/2.0;
        current_Z[leg_num] = HOME_Z[leg_num];
        if(tick >= numTicks-1) ripple_case[leg_num] = 4;
        break;
      case 4: //move foot back one-quarter (on the ground) 将脚向后移动四分之一 (在地面上)
        current_X[leg_num] = current_X[leg_num] - amplitudeX/numTicks/2.0;
        current_Y[leg_num] = current_Y[leg_num] - amplitudeY/numTicks/2.0;
        current_Z[leg_num] = HOME_Z[leg_num];
        if(tick >= numTicks-1) ripple_case[leg_num] = 5;
        break;
      case 5: //move foot back one-quarter (on the ground)
        current_X[leg_num] = current_X[leg_num] - amplitudeX/numTicks/2.0;
        current_Y[leg_num] = current_Y[leg_num] - amplitudeY/numTicks/2.0;
        current_Z[leg_num] = HOME_Z[leg_num];
        if(tick >= numTicks-1) ripple_case[leg_num] = 6;
        break;
      case 6: //move foot back one-quarter (on the ground)
        current_X[leg_num] = current_X[leg_num] - amplitudeX/numTicks/2.0;
        current_Y[leg_num] = current_Y[leg_num] - amplitudeY/numTicks/2.0;
        current_Z[leg_num] = HOME_Z[leg_num];
        if(tick >= numTicks-1) ripple_case[leg_num] = 1;
        break;
    }
  }
  //increment tick 增量记号
  if(tick < numTicks-1) tick++;
  else tick = 0;
}
}

//*****
// Tetrapod Gait 四足步态
// Right front and left rear legs move forward together, then right 右前腿和左后腿一起向前移动, 然后向右移动
// rear and left middle, and finally right middle and left fronts. 后腿和左中间, 最后是右中间和左前腿。

//*****
void tetrapod_gait()
{
  //read commanded values from controller
  commandedX = map(ps2x.Analog(PSS_RX),0,255,127,-127);
  commandedY = map(ps2x.Analog(PSS_RY),0,255,-127,127);
  commandedR = map(ps2x.Analog(PSS_LX),0,255,127,-127);

  //if commands more than deadband then process
  if((abs(commandedX) > 15) || (abs(commandedY) > 15) || (abs(commandedR) > 15) || (tick>0))
  {

```

```

compute_strides();
numTicks = round(duration / FRAME_TIME_MS / 3.0); //total ticks divided into the three cases
for(leg_num=0; leg_num<6; leg_num++)
{
    compute_amplitudes();
    switch(tetrapod_case[leg_num])
    {
        case 1: //move foot forward (raise and lower) 向前移动脚（升高和降低）
            current_X[leg_num] = HOME_X[leg_num] - amplitudeX*cos(M_PI*tick/numTicks);
            current_Y[leg_num] = HOME_Y[leg_num] - amplitudeY*cos(M_PI*tick/numTicks);
            current_Z[leg_num] = HOME_Z[leg_num] + abs(amplitudeZ)*sin(M_PI*tick/numTicks);
            if(tick >= numTicks-1) tetrapod_case[leg_num] = 2;
            break;
        case 2: //move foot back one-half (on the ground) 将脚向后移动一半（在地上）
            current_X[leg_num] = current_X[leg_num] - amplitudeX/numTicks;
            current_Y[leg_num] = current_Y[leg_num] - amplitudeY/numTicks;
            current_Z[leg_num] = HOME_Z[leg_num];
            if(tick >= numTicks-1) tetrapod_case[leg_num] = 3;
            break;
        case 3: //move foot back one-half (on the ground)
            current_X[leg_num] = current_X[leg_num] - amplitudeX/numTicks;
            current_Y[leg_num] = current_Y[leg_num] - amplitudeY/numTicks;
            current_Z[leg_num] = HOME_Z[leg_num];
            if(tick >= numTicks-1) tetrapod_case[leg_num] = 1;
            break;
    }
}
//increment tick
if(tick < numTicks-1) tick++;
else tick = 0;
}
}

//*****
// Compute walking stride lengths 计算步行步幅
//*****
void compute_strides()
{
    //compute stride lengths 计算步幅
    strideX = 90*commandedX/127;
    strideY = 90*commandedY/127;
    strideR = 35*commandedR/127;

    //compute rotation trig 计算旋转参数
    sinRotZ = sin(radians(strideR));
    cosRotZ = cos(radians(strideR));

    //set duration for normal and slow speed modes 设置正常和低速模式的持续时间
    if(gait_speed == 0) duration = 1000;
    else duration = 3240;
}

//*****
// Compute walking amplitudes 计算行走幅度
//*****
void compute_amplitudes()
{
    //compute total distance from center of body to toe 计算从身体中心到脚趾的总距离
    totalX = HOME_X[leg_num] + BODY_X[leg_num];
    totalY = HOME_Y[leg_num] + BODY_Y[leg_num];

    //compute rotational offset 计算旋转偏移
    rotOffsetX = totalY*sinRotZ + totalX*cosRotZ - totalX;
    rotOffsetY = totalY*cosRotZ - totalX*sinRotZ - totalY;

    //compute X and Y amplitude and constrain to prevent legs from crashing into each other 计算x和y幅值并进行的约束，以
    amplitudeX = ((strideX + rotOffsetX)/2.0);
    amplitudeY = ((strideY + rotOffsetY)/2.0);
}

```

```

amplitudeX = constrain(amplitudeX,-50,50);
amplitudeY = constrain(amplitudeY,-50,50);

//compute Z amplitude          计算Z振幅
if(abs(strideX + rotOffsetX) > abs(strideY + rotOffsetY))
    amplitudeZ = step_height_multiplier * (strideX + rotOffsetX) / 4.0;
else
    amplitudeZ = step_height_multiplier * (strideY + rotOffsetY) / 4.0;
}

//*****
// Body translate with controller (xyz axes)          带控制器的实体平移 (xyz轴)
//*****
void translate_control()
{
    //compute X direction move          计算X方向移动
    translateX = map(ps2x.Analog(PSS_RY),0,255,-2*TRAVEL,2*TRAVEL);
    for(leg_num=0; leg_num<6; leg_num++)
        current_X[leg_num] = HOME_X[leg_num] + translateX;

    //compute Y direction move
    translateY = map(ps2x.Analog(PSS_RX),0,255,2*TRAVEL,-2*TRAVEL);
    for(leg_num=0; leg_num<6; leg_num++)
        current_Y[leg_num] = HOME_Y[leg_num] + translateY;

    //compute Z direction move
    translateZ = ps2x.Analog(PSS_LY);
    if(translateZ > 127)
        translateZ = map(translateZ,128,255,0,TRAVEL);
    else
        translateZ = map(translateZ,0,127,-3*TRAVEL,0);
    for(leg_num=0; leg_num<6; leg_num++)
        current_Z[leg_num] = HOME_Z[leg_num] + translateZ;

    //lock in offsets if commanded          如果命令, 锁定偏移量
    if(capture_offsets == true)
    {
        for(leg_num=0; leg_num<6; leg_num++)
        {
            offset_X[leg_num] = offset_X[leg_num] + translateX;
            offset_Y[leg_num] = offset_Y[leg_num] + translateY;
            offset_Z[leg_num] = offset_Z[leg_num] + translateZ;
            current_X[leg_num] = HOME_X[leg_num];
            current_Y[leg_num] = HOME_Y[leg_num];
            current_Z[leg_num] = HOME_Z[leg_num];
        }
    }

    //if offsets were commanded, exit current mode          如果指令偏移, 退出当前模式
    if(capture_offsets == true)
    {
        capture_offsets = false;
        mode = 0;
    }
}

//*****
// Body rotate with controller (xyz axes)          身体随控制器旋转 (xyz轴)
//*****
void rotate_control()
{
    //compute rotation sin/cos values using controller inputs          使用控制器输入计算旋转正弦/余弦值
    sinRotX = sin((map(ps2x.Analog(PSS_RX),0,255,A12DEG,-A12DEG))/1000000.0);
    cosRotX = cos((map(ps2x.Analog(PSS_RX),0,255,A12DEG,-A12DEG))/1000000.0);
    sinRotY = sin((map(ps2x.Analog(PSS_RY),0,255,A12DEG,-A12DEG))/1000000.0);
    cosRotY = cos((map(ps2x.Analog(PSS_RY),0,255,A12DEG,-A12DEG))/1000000.0);
    sinRotZ = sin((map(ps2x.Analog(PSS_LX),0,255,-A30DEG,A30DEG))/1000000.0);
    cosRotZ = cos((map(ps2x.Analog(PSS_LX),0,255,-A30DEG,A30DEG))/1000000.0);

    //compute Z direction move          计算Z方向移动

```

```

translateZ = ps2x.Analog(PSS_LY);
if(translateZ > 127)
    translateZ = map(translateZ,138,255,0,TRAVEL);
else
    translateZ = map(translateZ,0,127,-2*TRAVEL,0);

for(int leg_num=0; leg_num<4; leg_num++)
{
    //compute total distance from center of body to toe
    totalX = HMMH_X[leg_num] + BODY_X[leg_num];
    totalY = HMMH_Y[leg_num] + BODY_Y[leg_num];
    totalZ = HMMH_Z[leg_num] + BODY_Z[leg_num];

    //perform 3 axis rotations 执行3轴旋转
    rotOffsetX = totalX*cosRotY*cosRotZ + totalY*sinRotX*sinRotY*cosRotZ + totalZ*cosRotX*sinRotY*cosRotZ + totalZ*sinRotX*sinRotZ - totalX;
    rotOffsetY = -totalX*cosRotY*sinRotZ - totalY*sinRotX*sinRotY*sinRotZ + totalY*cosRotX*cosRotZ + totalZ*cosRotX*sinRotY*sinRotZ + totalZ*sinRotX*cosRotZ - totalY;
    rotOffsetZ = totalX*sinRotY - totalY*sinRotX*cosRotY + totalZ*cosRotX*cosRotY - totalZ*cosRotX*cosRotY - totalZ;

    // Calculate foot positions to achieve desired rotation 计算脚的位置以实现所需的旋转
    current_X[leg_num] = HMMH_X[leg_num] + rotOffsetX;
    current_Y[leg_num] = HMMH_Y[leg_num] + rotOffsetY;
    current_Z[leg_num] = HMMH_Z[leg_num] + rotOffsetZ + translateZ;

    //lock in offsets if commanded 如果命令, 锁定偏移量
    if(capture_offsets == true)
    {
        offset_X[leg_num] = current_X[leg_num] + rotOffsetX;
        offset_Y[leg_num] = current_Y[leg_num] + rotOffsetY;
        offset_Z[leg_num] = current_Z[leg_num] + rotOffsetZ + translateZ;
        current_X[leg_num] = HMMH_X[leg_num];
        current_Y[leg_num] = HMMH_Y[leg_num];
        current_Z[leg_num] = HMMH_Z[leg_num];
    }
}

//if offsets were commanded, exit current mode 如果指令有效, 退出当前模式
if(capture_offsets == true)
{
    capture_offsets = false;
    mode = 0;
}
}

//.....
// One leg lift mode 单腿提升模式
// also can set a step height using capture offsets 还可以使用捕捉偏移量设置步高高度
//.....
void one_leg_lift()
{
    //read current leg servo 1 positions the first time 第一次读取当前腿伺服1位置
    if(leg1_IK_control == true)
    {
        leg1_coxa = coxa1_servo.read();
        leg1_femur = femur1_servo.read();
        leg1_tibia = tibia1_servo.read();
        leg1_IK_control = false;
    }

    //read current leg servo 6 positions the first time 第一次读取当前腿伺服6位置
    if(leg6_IK_control == true)
    {
        leg6_coxa = coxa6_servo.read();
        leg6_femur = femur6_servo.read();
        leg6_tibia = tibia6_servo.read();
        leg6_IK_control = false;
    }

    //process right joystick left/right axis 处理右摇杆手柄左/右轴
    temp = ps2x.Analog(PSS_RX);
    temp = map(temp,0,255,45,-45);
    coxa1_servo.write(constrain(int(leg1_coxa+temp),45,135));

    //process right joystick up/down axis 处理右摇杆手柄上/下轴
    temp = ps2x.Analog(PSS_RY);
    if(temp < 117) //if joystick moved up 如果摇杆柄向上移动
    {
        temp = map(temp,116,0,0,24); //move leg 1 移动腿1
        femur1_servo.write(constrain(int(leg1_femur+temp),0,170));
        tibia1_servo.write(constrain(int(leg1_tibia+4*temp),0,170));
    }
    else //if joystick moved down 如果摇杆柄向下移动
    {
        z_height_right = constrain(temp,140,255); // set Z step height 设置Z轴高度
        z_height_right = map(z_height_right,140,255,1,0);
    }

    //process left joystick left/right axis 处理左摇杆手柄左/右轴
    temp = ps2x.Analog(PSS_LX);
    temp = map(temp,0,255,45,-45);
    coxa6_servo.write(constrain(int(leg6_coxa+temp),45,135));

    //process left joystick up/down axis 处理左摇杆手柄上/下轴
    temp = ps2x.Analog(PSS_LY);

```

```

if(temp < 117) //if joystick moved up. 如果操纵杆向上移动
{
  temp = map(temp,116,0,0,24); //move leg 6 移动腿6
  femur6_servo.write(constrain(int(leg6_femur+temp),0,170));
  tibia6_servo.write(constrain(int(leg6_tibia+4*temp),0,170));
}
else //if joystick moved down 如果操纵杆向下移动
{
  z_height_left = constrain(temp,140,265); //set z step height 设置z台阶高度
  z_height_left = map(z_height_left,140,265,1,8);
}

//process z height adjustment
if(z_height_left>z_height_right)
  z_height_right = z_height_left; //use max left or right value 使用最大左或右值
// if(batt_LEDs > 2) z_height_LED_color=0; //use red LEDs if battery strong 如果电池电量充足, 则使用红色LED
// else z_height_LED_color=1; //use green LEDs if battery weak 如果电池电量不足, 则使用绿色LED
//LED_Bar(z_height_LED_color,z_height_right); //display z height 显示z高度
if(capture_offsets == true) //lock in z height if commanded 如果命令, 锁定z高度
{
  step_height_multiplier = 1.0 + ((z_height_right - 1.0) / 3.0);
  capture_offsets = false;
}
}

//.....
// Set all servos to 90 degrees
// Note: this is useful for calibration/alignment of the servos
// i.e: set COXA_CAL[6], FEMUR_CAL[6], and TIBIA_CAL[6] values in
// ..... constants section above so all angles appear as 90 degrees
//将所有伺服设置为90度

//注: 这对于伺服系统的校准/校准非常有用

//即: 在中设置髋关节校准(c), 股骨校准(f)和胫骨校准(t)值

//常数部分, 所以所有角度都显示为90度
//.....
void set_all_90()
{
  coxa1_servo.write(90+COXA_CAL[0]);
  femur1_servo.write(90+FEMUR_CAL[0]);
  tibia1_servo.write(90+TIBIA_CAL[0]);

  coxa2_servo.write(90+COXA_CAL[1]);
  femur2_servo.write(90+FEMUR_CAL[1]);
  tibia2_servo.write(90+TIBIA_CAL[1]);

  coxa3_servo.write(90+COXA_CAL[2]);
  femur3_servo.write(90+FEMUR_CAL[2]);
  tibia3_servo.write(90+TIBIA_CAL[2]);

  coxa4_servo.write(90+COXA_CAL[3]);
  femur4_servo.write(90+FEMUR_CAL[3]);
  tibia4_servo.write(90+TIBIA_CAL[3]);

  coxa5_servo.write(90+COXA_CAL[4]);
  femur5_servo.write(90+FEMUR_CAL[4]);
  tibia5_servo.write(90+TIBIA_CAL[4]);

  coxa6_servo.write(90+COXA_CAL[5]);
  femur6_servo.write(90+FEMUR_CAL[5]);
  tibia6_servo.write(90+TIBIA_CAL[5]);
}

//.....
// Print Debug Data 打印调试数据
//.....
void print_debug()
{
  //display elapsed frame time (ms) and battery voltage (V) 显示经过的时间 (ms) 和电池电压 (V)
  currentTime = millis();
  Serial.print(currentTime-previousTime);
  Serial.print(",");
  //Serial.print(float(batt_voltage)/100.0);
  Serial.print("\n");
}

```

ДОДАТОК Б
(обов'язковий)

Стаття подана на друк у журнал «ВИМІРЮВАЛЬНА ТА
ОБЧИСЛЮВАЛЬНА ТЕХНІКА В ТЕХНОЛОГІЧНИХ ПРОЦЕСАХ»

УДК 621.865:004.896

DOI:

БІГУС А.Р.

Студент АКІТРМ-24-1

КОРЕЦЬКА Л. О.

<https://orcid.org/0000-0002-4284-4936>

koretskal@khmnu.edu.ua

МАЙДАН П.С.

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

maidanps@gmail.com

МАКАРИШКІН Д.А.

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

makaryshkinde@khmnu.edu.ua

СОКОЛАН Ю.С.

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

sokolan.julia@gmail.com

Хмельницький національний університет

АНАЛІТИЧНЕ ДОСЛІДЖЕННЯ РОБОЧОЇ ХОДИ МОДЕЛІ МЕХАТРОННОГО РОБОТА-ГЕКСАПОДА

Біонічні роботи - це досить новий напрямок розвитку в галузі робототехніки. Це такий тип робота, що заснований на біологічних характеристиках конкретної істоти, тобто робот імітує зовнішню структуру істоти, рух і загальну поведінку.

На сучасному етапі розвитку науки і техніки існує три класифікації біонічних роботів: так звані роботи-гуманоїди, стандартні біонічні роботи та, навіть, біологічні роботи. Більшість з таких біонічних роботів використовуються для виявлення та передачі інформації в небезпечних для людей чи просто складних для досягнення областях.

Мехатронний робот-гексапод відрізняється від людини або стандартного колісного робота саме механічною структурою, для того, щоб адекватно працювати в змінному навколишньому середовищі. Конструкція повинна справлятися із суворими природними умовами, вільно переміщатися по нерівних поверхнях і повністю уникати перешкоди.

В роботі описано проблематику проєктування моделі мехатронного робота-гексапода, для чого виконано огляд та аналіз існуючих конструкцій роботів. Було проведено аналіз виконання процесу руху робота, виконано аналіз кількох типів ходи, які зазвичай використовуються мобільними роботом, більш детально описано виконання мехатронним роботом-гексаподом чотиринової ходи. Описано аналітичне дослідження стійкості ходи, а також виконано розроблення системи керування мехатронним роботом-гексаподом.

Ключові слова: мехатронний робот-гексапод, біоніка, робочий цикл, чотиринога хода, система керування.

BIHUS Artem, KORETSKA Liudmyla, MAIDAN Pavlo,

MAKARYSHKIN Denys, SOKOLAN Iuliia

Khmelnytskyi National University

ANALYTICAL RESEARCH OF THE WORKING PROCESS OF THE MECHATRONIC HEXAPOD ROBOT MODEL

Bionic robots are a relatively new development in the field of robotics. This type of robot is based on the biological characteristics of a specific creature, i.e., the robot imitates the creature's external structure, movement, and general behavior.

At the current stage of scientific and technological development, there are three classifications of bionic robots: so-called humanoid robots, standard bionic robots, and even biological robots. Most of these bionic robots are used to detect and transmit information in areas that are dangerous for humans or simply difficult to reach.

A mechatronic hexapod robot differs from a human or a standard wheeled robot in its mechanical structure, which allows it to function adequately in a changing environment. The design must be able to cope with harsh natural conditions, move freely over uneven surfaces, and completely avoid obstacles.

The paper describes the problems of designing a mechatronic hexapod robot model, for which a review and analysis of existing robot designs was performed. An analysis of the robot's movement process was carried out, an analysis of several types of gait commonly used by mobile robots was performed, and the four-legged gait of a mechatronic hexapod robot was described in more detail. An analytical study of gait stability was described, and a control system for a mechatronic hexapod robot was developed.

Keywords: mechatronic, hexapod robot, bionics, working cycle, quadrupedal gait, control system.

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

Біонічні роботи (БР) - це досить новий напрямок розвитку в галузі робототехніки, що заснований на біологічних характеристиках конкретної істоти, тобто робот імітує зовнішню структуру істоти, рух і загальну поведінку [1, 2].

На сучасному етапі розвитку науки і техніки існує три класифікації БР: так звані роботи-гуманоїди, стандартні БР та, навіть, біологічні роботи [1, 2]. Більшість з цих БР використовуються для виявлення та передачі інформації в небезпечних для людей чи просто складних для досягнення середовищах.

Мехатронний робот-гексапод (МРГ) відрізняється від людини або стандартного колісного мобільного робота саме механічною структурою, для того, щоб адекватно працювати в навколишньому середовищі. Конструкція, що проектується, повинна справлятися із суворими природними умовами, вільно переміщатися по нерівних поверхнях і повністю уникати встановлених контрольних точок. МРГ мають надзвичайно високу цінність у багатьох галузях промисловості і володіють досить великою здатністю до розвитку.

В останні роки, завдяки розвитку теорії біоніки та науки і техніки, дослідження і розробки інтелектуальних МРГ стали предметом певного занепокоєння дослідників. Було проведено безліч досліджень МРГ за кордоном, і були створені доволі різні конструкції МРГ, від чисто механічного руху до інтелектуального механічного руху інтелект і всебічні характеристики МРГ швидко розвивалися.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

У 1970-х роках минулого століття фірма виробник MOSHER створила 4-ногий механізм під назвою «Walking Truck», конструкція оснащена 4-ма ногами і приводиться в загальний рух за допомогою використання гідравліки. Давач положення може реалізувати функцію визначення місцезнаходження відносно людського тіла [3]. Щоб знизити ризик десантних операцій, ВПС США і Масачусетський технологічний інститут (MIT) створили конструкцію 6-ногого автоматизованого робота ARIEL, призначеного для виявлення мін на мілководді [4]. Кожна нога даного робота володіє двома ступенями свободи обертання, які можуть виконувати функцію перевертання і руху. Давач положення і компас також встановлені на корпусі МРГ, який закриває всі схеми машини і ПЛК.

Автоматичний інтелектуальний робот DANTE [5] - використовувався для вивчення будови вулкана Елібус в Антарктиді і надіслав велику кількість цінних зображень та інформації для аналізу даних.

Spider pot - маленький арахноподібний робот, створений Лабораторією реактивного руху НАСА [6] - використовуючи пару антен для виявлення перешкод і спритні задні лапи, може легко виконувати різні спеціальні дії. Дана конструкція також володіє усіма перевагами невеликого розміру і досить швидкого руху. Може уникати перешкод, справлятися з різними досить складними місцевостями і досліджувати важкодоступні місця. Конструкція робота, схожого на павука, також може бути використана для дослідження невеликих небесних тіл, крім того використовувався в якості обслуговуючого персоналу на Міжнародній космічній станції (МКС) і використовувався для ліквідації аварійних ситуацій, наприклад, виток повітря.

Дослідження і розробки конструкцій мобільних роботів в Китаї почалися пізно в порівнянні із іншими зарубіжними країнами приблизно в кінці 1980-х років минулого століття. Проте завдяки постійним зусиллям і дослідженням технологія даного обладнання в Китаї досягла досить суттєвих успіхів. У 2007 році дослідницька лабораторія Харбінського технологічного інституту (НІТ) розробила конструкцію мобільного робота НІТ-Spider101 на основі палички-комахи [7]. Як представлено на рисунку 1, шість ніг мобільного робота Spider101 рівномірно розподілені по боках корпусу, кожна окрема нога володіє 3-ма ступенями свободи, конструкція передачі паралельна чотирилангова передача, сервомеханізм встановлений на корпусі МРГ, що значно покращує загальну стійкість конструкції. Проте головним недоліком даної конструкції МРГ є те, що він може рухатись лише одним типом ходу і не може впоратися зі складними умовами навколишнього середовища.

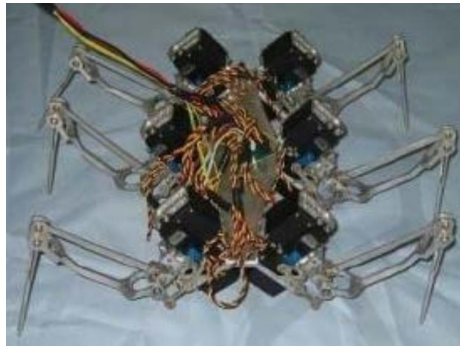


Рис.1. Загальний вигляд НІТ-Spider101

У 2012 році НІТ перепроєктував мобільного 6-ногого робота із центральною симетричною структурою і великим відношенням навантаження до ваги, що приводиться в рух електродвигуном. Нова конструкція МРГ володіє високою адаптивністю і великим співвідношенням навантаження до ваги в складних умовах навколишнього середовища. В даний час відношення навантаження до ваги даної конструкції МРГ може досягати 0,46, а кут підйому не більше 35° [7].

У 2018 році було створено конструкцію 4-ногого робота «Jueying», в Університеті Чжецзян та фірмою DEEP Robotics [8]. З основною функцією вправ і стрибків, конструкція мобільного робота також може ходити по більш пересіченій місцевості, такій як сніг, схили пагорбів чи, навіть, гравій. Час автономної роботи мобільного робота може досягати двох годин.

Наука робототехніка досить швидко розвивається в напрямку штучного інтелекту (ШІ), а технологія біонічних МР розвивається особливо швидко. Найбільша цінність використання робототехніки полягає в можливості виконувати завдання, які більшість людей просто не можуть виконати. Робототехніка висуває більш високі вимоги до матеріалу тіла при роботі в суворих природних умовах.

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Метою роботи є: дослідження робочої ходи моделі мехатронного робота-гексапода.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Грунтуючись на спостереженнях реальних істот із багатьма ногами і огляді літературних джерел [3-8], виконаємо аналіз кількох типів ходи, які зазвичай використовуються мобільними МРГ, і проведемо аналітичні дослідження із множинними рухами або так званим робочим циклом [9].

Початковий робочий цикл відноситься до частки періоду часу вмикання відносно загального періоду часу в імпульсному циклі [9]. У нашому випадку робочий цикл виражається як відношення кількості ніг МРГ, піднятих в циклі виконання руху, до кількості всіх ніг МРГ. Тобто формула робочого циклу виглядає наступним чином: кількість тільки піднятих ніг відноситься до кількості всіх ніг МРГ.

Згідно з формулою робочого циклу, можливо визначити, що бажаний діапазон робочого циклу знаходиться в проміжку від 0 до 6/6, і різні значення позначатимуть використання різного типу ходи МРГ.

При діапазоні робочого циклу від 0 до 3/6, МРГ володіє шістьма ногами, але більше трьох ніг одночасно не мають контакту із поверхнею, при виконанні руху менше як три ноги не стикаються із поверхнею, а також можливо, що жодна нога МРГ не стикається із поверхнею, як у вільному стані, в цьому стані МРГ рухається надзвичайно швидко, але не має загальної стійкості та сейсмостійкості.

При діапазоні робочого циклу 3/6, три ноги з загальних шести ніг МРГ в кожен момент часу не стикаються із поверхнею, при виконанні руху три ноги стикаються із поверхнею, даний стан називається рухом у триногій ході, в цьому стані МРГ відносно стабільний і швидкий. Тринога хода досить часто використовується саме в шестиногих МРГ.

При діапазоні робочого циклу в проміжку від 3/6 до 4/6, три з загальних шести ніг МРГ не стикаються із поверхнею, при виконанні руху три ноги стикаються із поверхнею, але в цей час одна з ніг збирається стикнутися із поверхнею в стадії виконання руху. У цьому стані МРГ не підходить для ні для аналізу ні для контролю виконання руху.

Коли діапазон робочого циклу дорівнює 4/6, дві із загальних шести ніг МРГ в кожен момент виконання руху не стикаються із поверхнею, а решта чотири ноги стикаються із поверхнею. Даний стан носить назву рух чотириновою ходою, МРГ більш стабільний, ніж МРГ з триновою ходою, але рух виконується із зменшеним рівнем швидкості.

При діапазоні робочого циклу в проміжку (від 4/6 до 5/6) дві з загальних шести ніг МРГ не стикаються із поверхнею, при гойданні чотири ноги стикаються із поверхнею, але в цей час одна з ніг збирається стикнутися із поверхнею в стадії виконання руху ногою, в даному стані МРГ не підходить ні для аналізу ні для контролю.

При діапазоні робочого циклу 5/6, одна з загальних шести ніг МРГ в кожен момент циклу не стикається із поверхнею, при виконанні руху п'ять ніг стикаються із поверхнею, даний стан носить назву рух п'ятиногій ходи, в даному стані МРГ більш стабільний, ніж МРГ із триновою і чотириновою ходою, але значно знижує швидкість руху МРГ в цілому.

При діапазоні робочого циклу в проміжку від 5/6 до 6/6 одна з шести ніг МРГ не стикається із поверхнею, при виконанні руху п'ять ніг стикаються із поверхнею, але при цьому одна з них збирається стикнутися із поверхнею в стадії виконання руху, в цьому стані ефективність руху МРГ досить низька, практично немає можливості виконувати рух.

Коли діапазон робочого циклу рівний одиниці (6/6), всі шість ніг МРГ стикаються з поверхнею в кожен момент циклу, і немає ніг, що виконують рух - стаціонарний стан.

З вищевказаного видно, що чим ближче значення діапазону до одиниці (6/6), тим більше опорних ніг МРГ стикаються із поверхнею, тим вища стійкість, проте тим нижчий рівень швидкості.

Відповідно до виконаного аналізу переваг і недоліків вказаних восьми типів ходи, обираємо для подальшого дослідження тип ходи із робочим циклом 4/6 [1, 2, 9].

З точки зору стабільності виконання руху, чотиринога хода відносно більш стабільна, ніж тринога хода, але з точки зору рівня швидкості чотиринога хода поступається триногій ході і має відносно нижчий рівень загальної швидкості.

Чотиринога хода рахується більш стабільною, тому що МРГ має як мінімум чотири ноги, що контактують із поверхнею в кожен момент руху, і разом з корпусом МРГ утворює стійку чотирикутну опорну конструкцію (рис. 4). У той же час дві ноги після підйому, виконання руху і приземлення стають ногами, що стикаються із поверхнею, а дві з чотирьох інших ніг, що стикаються із поверхнею, піднімаються, щоб стати ногами які виконують рух, утворюючи іншу чотирикутну опорну конструкцію і так далі. Така чотиринога хода може виконуватись шістьма ногами із різними типами комбінацій, що складніше контролювати, ніж виконання тринової ходи (рис. 2).



Рис.2. Фазова діаграма положення ніг чотиринової ходи МРГ (4/6)

Проаналізуємо виконання прямої ходи вперед, назад і поворот чотиринової ходи МРГ.

На рисунку 3 представлено схематичне зображення чотиринової ходи МРГ вперед і назад. Форма, утворена перетином двох еліпсів, представляє основний корпус МРГ, відрізок прямої лінії представляє ногу, коло чорного кольору на кінці лінії представляє підошву ноги, що стикається із поверхнею, а коло білого кольору на кінці лінії представляє підошву ноги, що виконує рух.

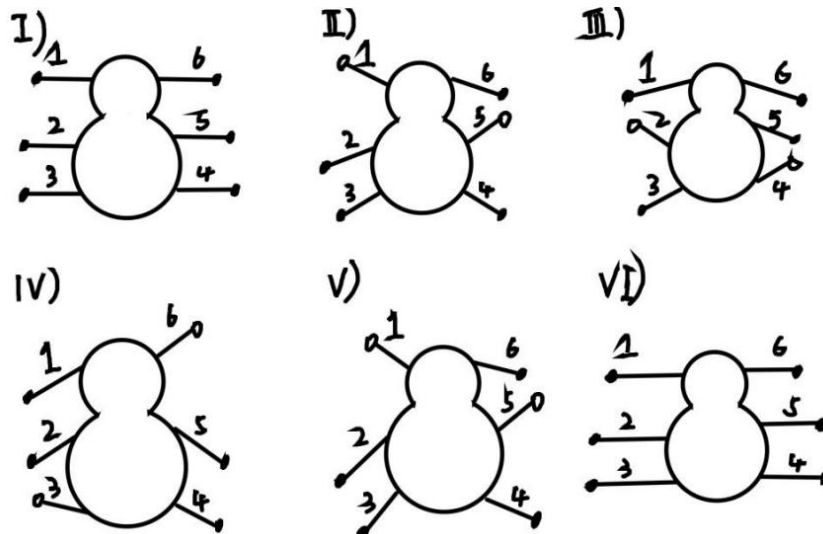


Рис.3. Схематичне зображення лінійного руху чотиринової ходи МРГ (4/6)

- перший етап (рис. 3) – вихідний стан МРГ, при цьому ноги 1 - 6 стикаються із поверхнею;
- другий етап – перша стадія виконання руху МРГ, в цей час ноги 2, 3, 4, 6 мають контакт із поверхнею, ноги 1, 5 виконують рух і поворот вперед на пів кроку, потім приземлення, весь корпус МРГ рухається вперед, ноги 2, 3, 4, 6 контактують з поверхнею, ноги 2, 4 будуть виконувати рух;
- третій етап - представляє другу стадію виконання руху МРГ, в цей час ноги 1, 3, 5, 6 мають контакт із поверхнею, ноги 2, 4 виконують рух і поворот вперед на крок, потім приземлення, весь корпус МРГ рухається вперед, ноги 1, 3, 5, 6 контактують з поверхнею, ноги 3, 5 будуть виконувати рух;

- четвертий етап - представляє третю стадію виконання руху МРГ, в цей час ноги 1, 2, 4, 6 мають контакт із поверхнею, ноги 3, 5 виконують рух і поворот вперед на крок, потім приземлення, весь корпус МРГ рухається вперед, ноги 1, 2, 4, 6 контактують з поверхнею, ноги 1, 5 будуть виконувати рух;

- п'ятий етап - четверта стадія виконання руху МРГ, при цьому ноги 2, 3, 4, 6 – контакт із поверхнею, ноги 1, 5 – виконання підйому і повороту вперед на пів кроку, потім приземлення, весь корпус МРГ рухається вперед, ноги 2, 3, 4, 6 – контакт з поверхнею;

- шостий етап – остання стадія виконання руху МРГ, зупинка руху, повернення у вихідний стан і завершення циклу виконання руху вперед МРГ.

Етапи виконання руху назад, є аналогічними як і в разі виконання руху вперед [9].

Дана конструкція МРГ використовує метод статичної граничної стійкості в методі статичного дослідження. Граничний метод статичної стійкості використовується для визначення стійкості МРГ. У русі, коли центр ваги всього корпусу знаходиться в замкнутій області, утвореній точками стопи ноги, що стикається із поверхнею, МРГ буде вважатись стійким, в іншому випадку МРГ буде перекидатися і ковзати.

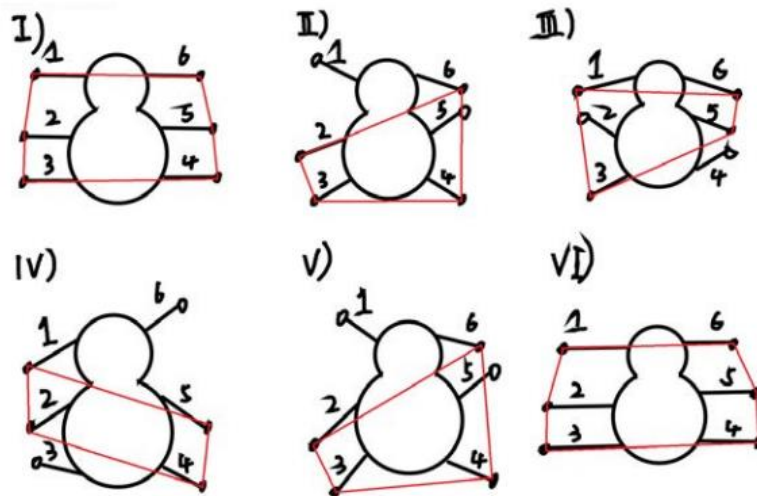


Рис.4. Виконання руху по прямій лінії, чотиринога хода (4/6)

Аналітичне дослідження показує, що центр ваги ходи 3-ох, 4-ох і 5 ніг знаходиться в багатокутнику, утвореному підшвами ніг, що стикаються із поверхнею, і всі три види ходи володіють певним рівнем стійкості (рис. 4). Порядок розміру – 3 ноги < 4 ноги < 5 ніг, але з точки зору рівнів швидкості 3 ноги краще, ніж 4 та 5 ніг. Оскільки конструкція вимагає не лише стабільності, але і певного рівня швидкості руху, в системі також переважає тринога хода. Інші види ходи будуть використовуватись в якості запасного виду ходи.

Дана конструкція МРГ потребує вісімнадцять суглобів (або механізм керування) для одночасної роботи для виконання поступального руху. У процесі руху кожен суглоб повинен керуватися впорядкованим чином, що є необхідною умовою для того, щоб МРГ міг плавно виконувати поступальний рух.

Отже, в даній моделі МРГ також необхідно розробити систему керування, яка відповідає біонічному роботі. На додаток до того, що МРГ повинен плавно виконувати рух, система керування (СК) також повинна мати систему камери.

СК МРГ умовно буде поділена на три частини-модулі: руху, камери і живлення. Плата керування і електропривід утворюють модуль руху для виконання руху МРГ, камера і модуль бездротової передачі 4G утворюють модуль камери, який може контролювати і передавати зображення в режимі реального часу під час виконання руху МРГ, а модуль живлення забезпечує живлення всього МРГ [1, 2, 9].

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ

І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

Описано проблематику проектування моделі мехатронного робота-гексапода, для чого виконано огляд та

аналіз існуючих конструкцій мехатронних роботів-гексаподів. Було проведено аналіз виконання руху, виконано дослідження варіантів виконання руху мехатронним роботом-гексаподом.

Більше детально аналіз циклів руху та проектування конструкції і системи керування, підбір основного обладнання та написання необхідного програмного забезпечення для роботи мехатронного робота-гексапода представлено в кваліфікаційній роботі магістра.

References

1. Strutynskiy V. B., Hurzhii A. M. Nazemni robotyzovani komplekсы: monohrafiia. Zhytomyr : PP «Ruta», 2023. 524 s.
2. Polishchuk M. M., Tkach M. M. Robototekhnichni systemy ta komplekсы: mobilni roboty dovilnoi oriantatsii: pidruchnyk dlia studentiv spetsialnosti «Informatsiini systemy ta tekhnolohii». Elektronni tekstovi dani (1 fail: 14,7 Mbait). Kyiv : KPI im. Ihoria Sikorskoho, 2023. 301 s. ISBN 978-966-990-076-0.
3. Quinn RD, Nelson GM, Bachmann RJ. Parallel Complementary Strategies for Implementing Biological Principles into Mobile Robots [J]. The International Journal of Robotics Research. 2013, 22(3). 169-186p.
4. Robo-zirka, cherepakho-bot i tekhnо-akula: vcheni stvoriuiut robotiv, yaki vmiiut plavaty, dlia doslidzhennia okeaniv. URL: <https://focus.ua/uk/technologies/736115-chas-mozhe-vzagali-ne-isnuvati-shcho-kazhut-fiziki> (data zvernennia: 15.09.2025).
5. M.C. Garcia-Lopez, E. Gorrostieta-Hurtado, Kinematic analysis for trajectory generation in one leg of a hexapod robot, Procedia Technology [J], 2012, 03. 342-350p.
6. NASA prodemonstruvalo mozhlyvosti robota-pavuka. URL: https://www.ukrinform.ua/rubric-other_news/1609760-nasa-prodemonstruvalo-moglivosti-robota-pavuka-video_1902746.html (data zvernennia: 15.09.2025).
7. Kharbinskyi tekhnolohichniy instytut, Mizhnarodnyi navchalnyi tsestr Veikhai. URL: <https://www.ncuk.ac.uk/uk/where-can-i-study/harbin-institute-of-technology-weihai-international-study-centre/> (data zvernennia: 15.09.2025).
8. DEEP Robotics. URL: <https://www.deeprobotics.cn/en> (data zvernennia: 15.09.2025).
9. Zhiying Wang, X.-L. Ding, Alberto Rovetta. Analysis of typical locomotion of a symmetric hexapod robot. Robotica, 2010, 28(06). 893-907p. DOI:10.1017/S0263574709990725.

Завідувачу кафедри АКІТгаР
канд. техн.наук, доценту Корецька Л.О.

Бігус Артем Романович

ПІБ здобувача вищої освіти

ФІТ, 2 курс, групи АКІТРм-24-1

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 (StrikePlagiarism 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.

05.12.2025

дата



підпис

Anti-Plagiarism v-15.258 (global version)

The maximum coincidence with one document 0.0%

Dictionaries check: en_US, ru_RU, ua_UA, Errors in the documents: 13%

ID: 253642 Title: МКР Метод розроблення моделі механічного робота-гексапода Added in a DB: 2025-12-18 Authors: Артем БІГУС Heads: Людмила КОРЕЦЬКА Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	83013	600	2297 (3%)	32 (5%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

Протокол аналізу звіту подібності експертом

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

Автор: Артем БІГУС

Співавтор:

Назва: Бігус антиплагіат

Експерт: Людмила КОРЕЦЬКА

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

Коефіцієнт подібності 1:1.5%

Коефіцієнт подібності 2:0.6%

Мікропробіли: 2

Заміна букв: 1

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2025-12-18 11:20:32.0

Після аналізу Звіту подібності констатую наступне:

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

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

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

Обґрунтування:

2025-12-18

Доцент Микола Федула

Дата

експерт

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

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

Назва: Метод розроблення моделі мехатронного робота-гексапода

Автор: Бігус Артем Романович

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

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

Науковий керівник: Корецька Людмила Олександрівна, кандидат технічних наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

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

Підтвердження:

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

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

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

3) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає 1,5% і адресується до 7 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи





Людмила КОРЕЦЬКА

Валерій МАРТИНЮК

Людмила КОРЕЦЬКА

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Бігус Артем Романович

Тема: Метод розроблення моделі мехатронного робота-гексапода

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

Обсяг кваліфікаційної роботи:

Кількість сторінок записки 86

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є розроблення моделі мехатронного робота-гексапода

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: У першому розділі описано проблематику проєктування моделі мехатронного робота-гексапода, для чого виконано огляд та аналіз існуючих конструкцій мехатронних роботів-гексаподів. У другому розділі виконано моделювання конструкції, проведено перевірку загальних параметрів і міцності ключових конструкцій мехатронного робота-гексапода. У третьому розділі проведено аналітичне дослідження стійкості ходи мехатронних роботів-гексаподів. Було виконано розроблення системи керування мехатронним роботом-гексаподом та підбір основних елементів керування. У четвертому розділі виконано моделювання роботи конструкції мехатронного робота-гексапода в програмному середовищі MATLAB. Розроблено програмне забезпечення для моделі мехатронного робота-гексапода для мікроконтролерів типу Arduino

4. Позитивні сторони роботи: висока практична цінність роботи, вся керуюча частина мехатронного робота-гексапода реалізується за допомогою використання сервоприводів Tower Pro MG996r, плати керування Arduino mega2560, пульта від PS2, камери Nanjing A8 та модуля живлення електроприводної частини LM2596S, що є досить поширеними на ринку і можуть бути легко замінені у випадку аварії чи поломки.

5. Негативні сторони роботи: у роботі недостатньо уваги приділяється аналізу існуючих технічних рішень та використанню інших мікроконтролерів у якості керуючих елементів.

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

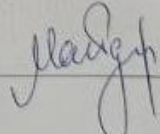
7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: відсутні

9. Оцінка дипломної роботи: добре (75 / 0)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Майдан Павло Сергійович, канд. техн. наук, доцент кафедри МАЕЕС

" 18 " 12 2025 р.

 (підпис)