

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій  
(повна назва)

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

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

Другий (магістерський)

(рівень вищої освіти)

Дослідження переміщення чотирилапого зооморфного робота «Робокіт»  
у невизначеному просторі

(тема)

Виконав:

студент 2 курсу, групи КТРСм-22-2

Долгуля А.В.

(прізвище, ініціали)

Спеціальності 151 Автоматизація та  
комп'ютерно-інтегровані  
технології

(код і повна назва спеціальності)

Тип програми Освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютеризовані та  
робототехнічні системи

(повна назва освітньої програми)

Керівник доцент Бронніков А.І.

(посада, прізвище, ініціали)

Допускається до захисту  
Зав. кафедри КІТАМ

(підпис)

Невлюдов І. Ш.

(прізвище, ініціали)

2023 р.

Я, як студент ХНУРЕ, розумію і підтримую політику  
закладу із академічної доброчесності. Я не наробив і не  
офертував незвального допущення під час підготовки  
кваліфікаційної роботи. Використання ідей, результатів  
і текстів інших авторів мають посилання на відповідне  
джерело.

## ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет \_\_\_\_\_ АКТ  
 Кафедра \_\_\_\_\_ КІТАР  
 Рівень вищої освіти \_\_\_\_\_ другий (магістерський)  
 Спеціальність \_\_\_\_\_ 151 Автоматизація та комп'ютерно-інтегровані технології  
 Тип програми \_\_\_\_\_ Освітньо-професійна  
 Освітня програма \_\_\_\_\_ Комп'ютеризовані та робототехнічні системи  
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАМ \_\_\_\_\_  
 (підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_ р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Долгулі Артему Вікторовичу  
 (прізвище, ім'я, по батькові)

1. Тема роботи Дослідження переміщення чотирилапого зооморфного робот  
 «Робокіт» у невизначеному просторі

Затверджена наказом по університету від 03.11.2023 № 1288Ст

2. Термін подання студентом роботи до екзаменаційної комісії \_\_\_\_\_

3. Вихідні дані до роботи нечітка логіка, MatLab, зооморфні роботи,  
 невизначений простір.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

Вступ; Аналіз переміщення роботів та їх орієнтація у просторі;

Вибір і обґрунтування середовища для моделювання;

Розробка програмного забезпечення для моделювання зооморфного робота;

Моделювання та експериментальні дослідження; Висновки; Додатки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій \_\_\_\_\_

*Графічний демонстративні матеріали в форматі PowerPoint (\*.ppt) формату*

---



---



---

6. Консультанти розділів роботи

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

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз технічного завдання	12.10 – 18.10.23	Виконано
2	Аналіз переміщення зооморфних роботів та їх орієнтація у просторі	20.10 – 09.11.23	Виконано
3	Вибір і обґрунтування основних елементів дослідження	23.11 – 01.12.23	Виконано
4	Розробка схеми компонентів робота	01.12 – 03.12.23	Виконано
5	Моделювання та експериментальні дослідження	04.12 – 20.12.23	Виконано
6	Подання роботи на перевірку Інтернет-сервісом Unichesk		
7	Оформлення пояснювальної записки		
8	Подання роботи на рецензію		
9	Подання роботи на підпис зав. кафедри		
11	Подання кваліфікаційної роботи в ЕК	за день до захисту	

Дата видачі завдання 10.10.2023

Студент \_\_\_\_\_  
(підпис)

Долгуля А.В  
(прізвище, ініціали)

Керівник роботи \_\_\_\_\_  
(підпис)

доцент Бронніков А.І  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 82с., 3 табл., 62 рис., 1 дод., 31 джерело.

### ЗООМОРФНІ РОБОТИ, НЕЧІТКА ЛОГІКА, МОДЕЛЮВАННЯ, ОРІЄНТАЦІЯ У ПРОСТОРИ.

Об'єкт дослідження – процес керування переміщенням зооморфного робота у невизначеному середовищі.

Предмет дослідження – методи керування зооморфним роботом «Робокіт».

Мета роботи – розробка методу управління переміщення зооморфного роботу на базі теорії нечітких множин для підвищення ефективності його керування, швидкості переміщення у невизначеному просторі.

У кваліфікаційній роботі проведено моделювання переміщення та виконання команд зооморфним роботом «Робокіт» у невизначеному просторі.

Для цього за допомогою додатку fuzzyLogicDesigner у програмному засобі MatLab було створено вхідні параметри відстані та поведінки для кожної кінцівки робота та описані правила, за якими відбувається модуляція.

Модуляція приведена у вигляді скріншотів у звіті.

Результати модуляції апробовані у науковій статті.

## ABSTRACT

Explanatory note: 82 p., 3 tabl., 62 pic., 1 application, 31 sources.

### ZOOMORPHIC ROBOTS, FUZZY LOGIC, MODELING, ORIENTATION IN SPACE.

The object of the research is the process of processing the movements of a zoomorphic robot in an unknown middle.

The subject of the research is methods of controlling the zoomorphic robot «Robokit».

The purpose of the work is to develop a method of controlling the movement of a zoomorphic robot based on the theory of fuzzy sets to increase the efficiency of its control and the speed of movement in an uncertain space.

To do this, using the fuzzyLogicDesigner application in the MatLab software, the input parameters of the distance and behavior for each limb of the robot were created and the rules according to which the modulation takes place were described.

The modulation is given in the form of screenshots in the report.

The results of the modulation are tested in a scientific article.

## ЗМІСТ

Перелік скорочень.....	7
Вступ.....	8
1 Аналіз переміщення роботів та їх орієнтація у просторі.....	9
1.1 Основні види переміщення роботів.....	9
1.2 Переваги та недоліки крокуючих механізмів.....	28
1.3 Види перешкод.....	29
1.4 Датчики орієнтації у просторі.....	31
1.5 Висновки до розділу.....	32
2 Вибір і обґрунтування середовища для моделювання.....	33
2.1 Вибір середовища для моделювання.....	34
2.2 Побудова схеми компонентів робота.....	41
2.3 Приклад використання нечіткої логіки.....	42
2.4 Приклад моделювання однієї ноги.....	45
2.5 Висновки до розділу.....	49
3 Розробка програмного модуля для зооморфного роботу.....	50
3.1 Розробка блок-схеми алгоритму.....	50
3.2 Розробка програмного забезпечення.....	53
3.3 Висновки до розділу.....	57
4 Моделювання та експериментальні дослідження.....	58
4.1 Моделювання та експериментальні дослідження.....	58
4.2 Висновки до розділу.....	75
5 Охорона праці.....	76
5.1 Охорона праці.....	76
Висновки.....	78
Перелік джерел посилання.....	79
Додаток А Апробація результатів наукових досліджень.....	83
Додаток Б Демонстраційний графічний матеріал.....	98

## ПЕРЕЛІК СКОРОЧЕНЬ

ЗЛГ – задня ліва гомілка;

ЗЛП – заднє ліве плече;

ЗПГ – задня права гомілка;

ЗПП – заднє праве плече;

ПЛГ – передня ліва гомілка;

ПЛП – переднє ліве плече;

ППГ – передня права гомілка;

ППП – переднє праве плече;

УЗ – ультразвуковий далекомір;

DARPA (англ. Defense Advanced Research Projects Agency) – агентство Міністерства оборони США, що відповідає за розробку нових технологій для використання в збройних силах США.

## ВСТУП

Сучасна робототехніка постійно вдосконалюється, і однією зі сфер, які отримують значний науковий і практичний інтерес, є роботи, здатні виконувати рухи та поведінку живих організмів [1]. Один із визначних представників таких роботів – «Робокіт». Ця технологічна розробка має великий потенціал у різних сферах, від пошуково-рятувальних операцій до вивчення біомеханіки та реабілітації.

Один з ключових аспектів у вивченні «Робокіт» – це його здатність переміщуватися в невизначеному просторі. Ця особливість важлива для роботів, які призначені для роботи в умовах, де поверхні непередбачувані та нерівні, а також важкодоступні для людини. У цьому звіті будуть представлені результати дослідження щодо переміщення «Робокіта» в невизначеному просторі та розглянуті важливі технологічні аспекти, що стоять за цією унікальною здатністю робота.

Метою роботи є розробка методу управління переміщення зооморфного роботу на базі теорії нечітких множин для підвищення ефективності його керування, швидкості переміщення у невизначеному просторі.

Для виконання поставленої мети треба вирішити такі питання:

- проаналізувати кінематику зооморфних роботів;
- проаналізувати датчики орієнтації у просторі;
- проаналізувати нечітку логіку;
- розробити схемі підключення компонентів робота;
- провести моделювання виконання команд роботом;
- оформлення пояснювальної записки керуючись методичними вказівками кваліфікаційної роботи магістра [2] та ДСТУ 3008-15 [3], результати модуляції апробовані у науковій статті [4].

# 1 АНАЛІЗ ПЕРЕМІЩЕННЯ РОБОТІВ ТА ЇХ ОРІЄНТАЦІЯ У ПРОСТОРИ

## 1.1 Основні види переміщення роботів

Як важливий базовий компонент чотириногих роботів, механічні лапи забезпечують роботам чудову маневреність і універсальність, які визначають основні характеристики, такі як адаптивність до роботи, швидкість ходьби та вантажопідйомність. Частіше за все кінцівки роботів схожі на промислові маніпулятори, тобто кажучи вони представляють собою стрижні з'єднані між собою в деяких точках для забезпечення згинання, яке приводиться завдяки серводвигунам. Існує багато представлень та топологій кінематики зооморфних роботів, однією із найпопулярніших є топологія шарнірних ніг [5], ця топологія зображена на рисунку 1.1.

Ця топологія виконує функцію подібну до колінного або ліктьового суглоба ноги тварини і має хороші характеристики.

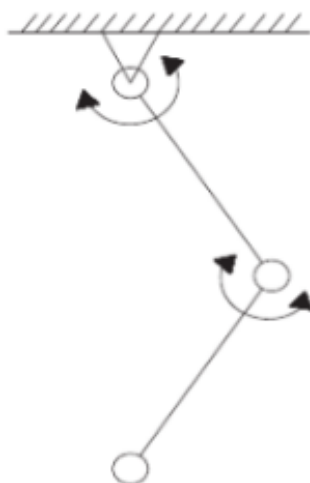


Рисунок 1.1 – Топологія шарнірних ніг

Шарнірні ноги можна розділити на шарнірні ноги типу ссавців та розтягнутого типу відповідно до їхньої конфігурації ноги, ці два типи зображено на рисунку 1.2. Розтягнутий тип означає, що перший сегмент ноги розташований у горизонтальному напрямку, а другий сегмент ноги – у вертикальному напрямку як стандартна поза. Порівнюючи ці два типи чотириногих роботів, роботи типу ссавців мають більш високу швидкість ходьби, менший крутний момент і меншу площу, в той час як роботи розтягнутого типу, мають більш високу стабільність, ширший діапазон рухів і більш високий рівень безпеки.

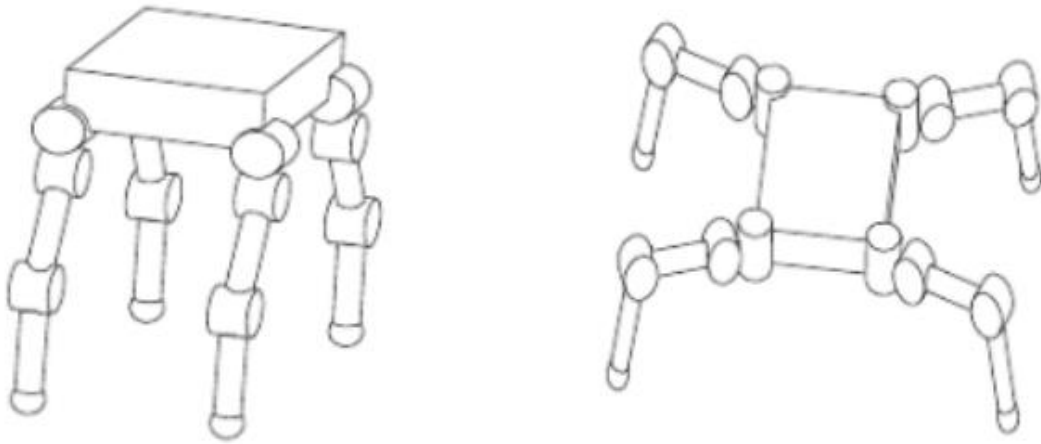


Рисунок 1.2 – Види шарнірних ніг зооморфних роботів: ліворуч – шарнірні ноги типу ссавців, праворуч – розтягнутого типу

Найбільш поширеним видом з цієї топології є шарнірні ноги типу ссавців. Boston Dynamics провела багато роботи з дослідження чотириноного робота і досягла плідних результатів. Вони розробили чотириноного робота WildCat, чотириноного робота LS3, чотириноного робота Spot і чотириноного робота SpotMini з використанням шарнірних ніг. WildCat і чотириногі роботи LS3 оснащені двотактним бензиновим двигуном, гідравлічним насосом і електричним сервоциліндром. Двотактний двигун використовується тому, що його щільність потужності вища, ніж у чотиритактного. Через повільну реакцію паливного двигуна та великі коливання крутного моменту та

швидкості вони оснащені гідравлічними баками для зберігання енергії, які забезпечують миттєву реакцію та зменшують коливання тиску масла. У розроблюваному роботі використовується кінематика топології шарнірних ніг типу ссавців. Ця топологія відповідає поставленій цілі розробці прототипу зооморфного робота Робокіт. Також головним плюсом є використання малої кількості серводвигунів на кінцівках робота, що дається взнаки на собівартості прототипу та енергоспоживанні.

Наступною топологією є «Крила» [6], представлена на рисунку 1.3. Крила в робототехніці можуть бути використані для створення зооморфних роботів, які імітують рухи тварин, таких як птахи, комахи чи навіть кажани. Використання крил у роботах може бути особливо ефективним у завданнях, пов'язаних з літанням або виконанням маневрів у повітрі.

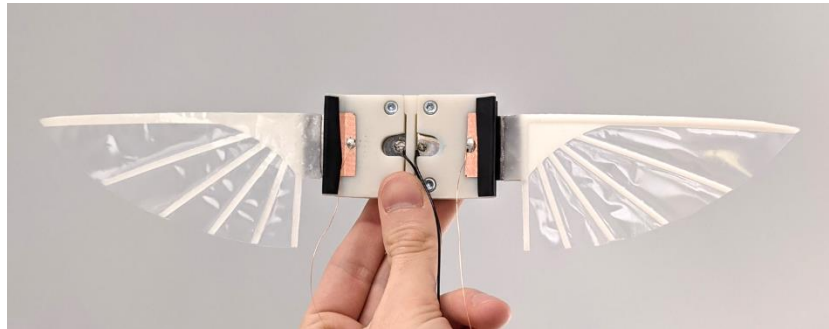


Рисунок 1.3 – Топологія «Крила»

При проектуванні зооморфних роботів з видом переміщення «Крила» звертають увагу на наступне:

- проектування крил. Крильця робота можуть бути здійснені за допомогою різних матеріалів, включаючи легкі та міцні полімери або металеві сплави. Форма крильця може бути оптимізована для досягнення максимальної ефективності під час літання;

- керування крилами. Кожне крило може бути оснащене власними актуаторами та сенсорами, щоб надати роботу можливість керування кожним

крилом незалежно. Це дозволяє роботам виконувати різноманітні маневри та реагувати на зміни у середовищі;

– моделювання руху комах. Крила можуть бути створені так, щоб імітувати рухи комах або птахів. Деякі роботи можуть використовувати не тільки крильця, але і інші структури, щоб наблизити робота до природного вигляду і руху;

– аеродинамічна оптимізація. Розробники можуть вдосконалювати форму та структуру крил, враховуючи принципи аеродинаміки. Це може включати оптимізацію форми крильця, кута атаки та інших параметрів для досягнення максимальної ефективності;

– забезпечення стабільності. Крила можуть бути використані для забезпечення стабільності робота під час літання. Деякі роботи можуть мати автономні системи стабілізації, які використовують дані з сенсорів та алгоритми для управління крильцями та підтримання стійкого руху в повітрі;

– перехідний режим. Деякі зооморфні роботи з крильцями можуть бути розроблені так, щоб здатні були до переходу між літанням і іншими типами руху, такими як ходьба або ползання, забезпечуючи їм більшу гнучкість в різних середовищах.

Прикладом такого робота є RoboBee зображений на рисунку 1.4 [7] від Harvard [8], який використовує крильця для літання в повітрі та вивчення аеродинамічних властивостей для розвитку більш ефективних методів літання для роботів.



Рисунок 1.4 – Робот RoboBee

За словами дослідників RoboBee, попередні спроби мініатюризувати роботів мало допомогли їм, оскільки малий розмір RoboBee змінює природу діючих сил. Інженерам довелося придумати, як будувати без роторних двигунів, шестерень, гайок і болтів, які нежиттєздатні в такому малому масштабі. У 2011 році вони розробили техніку, за якою вони вирізали рисунки з плоских аркушів, склали їх шарами та створювали форму. Клей використовувався для скріплення складених частин разом, аналогічно орігамі. Ця техніка замінила попередні, які були повільнішими та менш точними та використовували менш міцні матеріали. Виробничий процес, дозволяє швидко виготовляти прототипи RoboBee.

У мікромасштабі невелика кількість турбулентності може мати драматичний вплив на політ. Щоб подолати це, дослідникам довелося змусити RoboBee дуже швидко реагувати [8]. Для крил вони побудували «штучні м'язи», використовуючи п'єзоелектричний актуатор – тонку керамічну смужку, яка стискається, коли по ній пропускають електричний струм [9]. Тонкі пластикові шарніри служать з'єднаннями, які дозволяють обертальні рухи в крилах. Конструкція дозволяє роботам генерувати потужність, порівнянну з комахою такого ж розміру. Кожне крило можна окремо контролювати в реальному часі.

Кінцевою метою проекту є створення колоній повністю автономних і бездротових RoboBees. Станом на 2023 рік залишаються невирішеними дві проблеми. По-перше, робот занадто малий навіть для найменших інкапсульованих мікрочіпів, тобто роботи не можуть приймати рішення. Наразі RoboBee має вбудовані датчики зору, але дані потребують передачі до прив'язаної «підсистеми мозку» для інтерпретації. Триває робота над спеціалізованими апаратними прискорювачами з метою вирішення проблеми.

По-друге, дослідники не придумали, як отримати життєздатне джерело живлення на борту. Натомість роботи мають бути прив'язані крихітними шнурами, які забезпечують живлення та напрямки. Нещодавнім прогресом у сфері бортового керування живленням є демонстрація реверсивного,

енергоєфективного установки на звисах. Це дозволяє прототипу залишатися у високій точці огляду, зберігаючи енергію.

Наступною топологією є «Хвіст», яка зображена на рисунку 1.5 [10]. Хвіст у зооморфних роботах використовується для стабілізації та керування рухом. Цей елемент наводить на рухи тварин, таких як риби чи плазуни, які використовують хвост для управління та підтримання балансу.

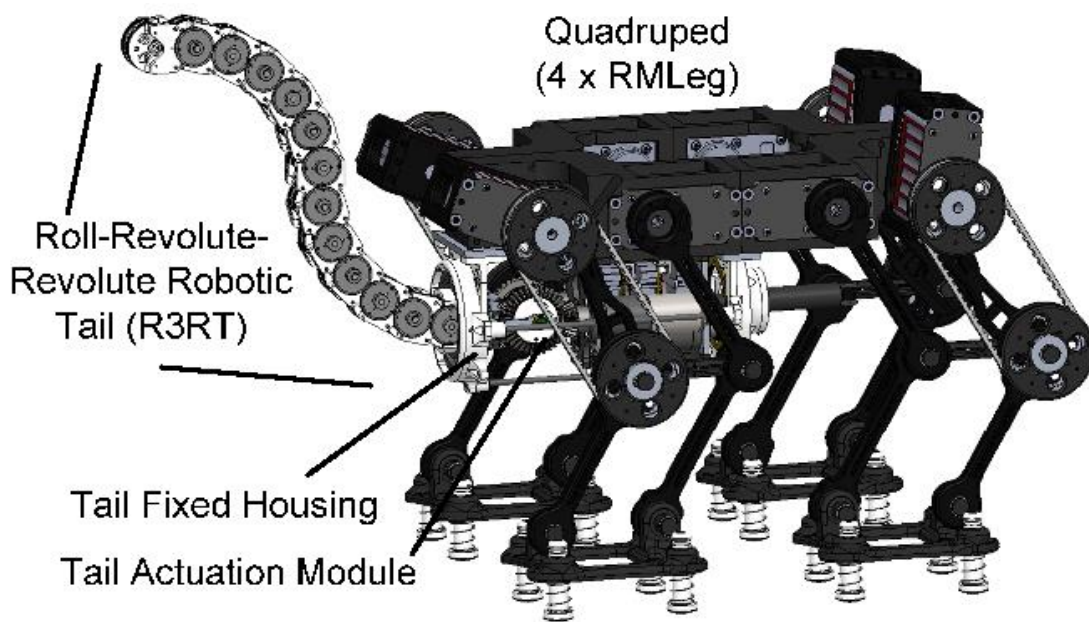


Рисунок 1.5 – Топологія «Хвіст»

Ключові аспекти використання хвоста в робототехніці:

- стабілізація руху. Хвіст може бути використаний для стабілізації робота під час руху, особливо в умовах, коли робот має обмежені можливості управління або коли він стикається зі змінами у середовищі;
- керування напрямком. Рухливий хвіст може бути використаний для зміни напрямку руху робота. Це дозволяє роботам ефективно реагувати на зміни у середовищі та виконувати маневри;
- імітація природних рухів. Хвіст може слугувати для імітації рухів тварин, таких як риби чи динозаври. Це може робити зооморфні роботи більш схожими на свої природні аналоги;

– сенсори та зворотний зв'язок. Хвіст може бути обладнаний сенсорами, які допомагають роботу отримувати інформацію про оточуюче середовище та власне положення. Ця інформація може використовуватися для адаптації руху робота до змін у середовищі;

– динамічне використання. Деякі роботи можуть мати хвости, які можуть змінювати свою форму або довжину в залежності від умов, що дозволяє їм бути більш адаптованими до різних ситуацій.

Одним із таких роботів є MIT Cheetah 3, зображений на рисунку 1.6 [11].

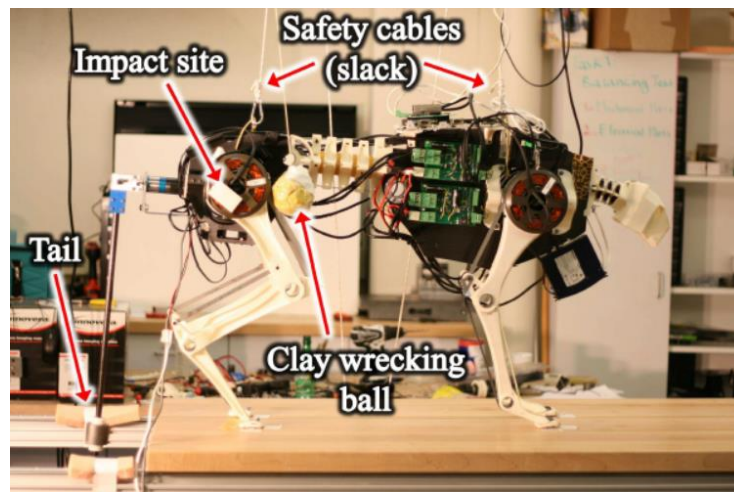


Рисунок 1.6 – Робот MIT Cheetah 3

Роботизований хвіст розроблено та впроваджено в MIT Cheetah, щоб перевірити його вплив на рівновагу за наявності зовнішніх перешкод. Маса хвоста 0,74 кг, а момент інерції відносно осі обертання двигуна 0,160 кг м<sup>2</sup>. На рисунку 1.6 зображено експериментальну установку; гепард Массачусетського технологічного інституту встановлюється стоячи, а глиняна руйнівна куля вдаряється в місце удару на задній частині тазу. Під час дослідів було видно що глиняна куля могла повалити робота, але хвіст ефективно стабілізує його.

Наступною топологією є «Плавання», зображена на рисунку 1.7 [12]. Зооморфні роботи, спроектовані для плавання, використовують різні принципи та дизайни, щоб імітувати рухи риб, водних комах, морських

ссавців або інших водяних тварин. Плавання має важливе значення в робототехніці для використання в різних сценаріях, таких як дослідження морського середовища, пошук та рятування на воді.

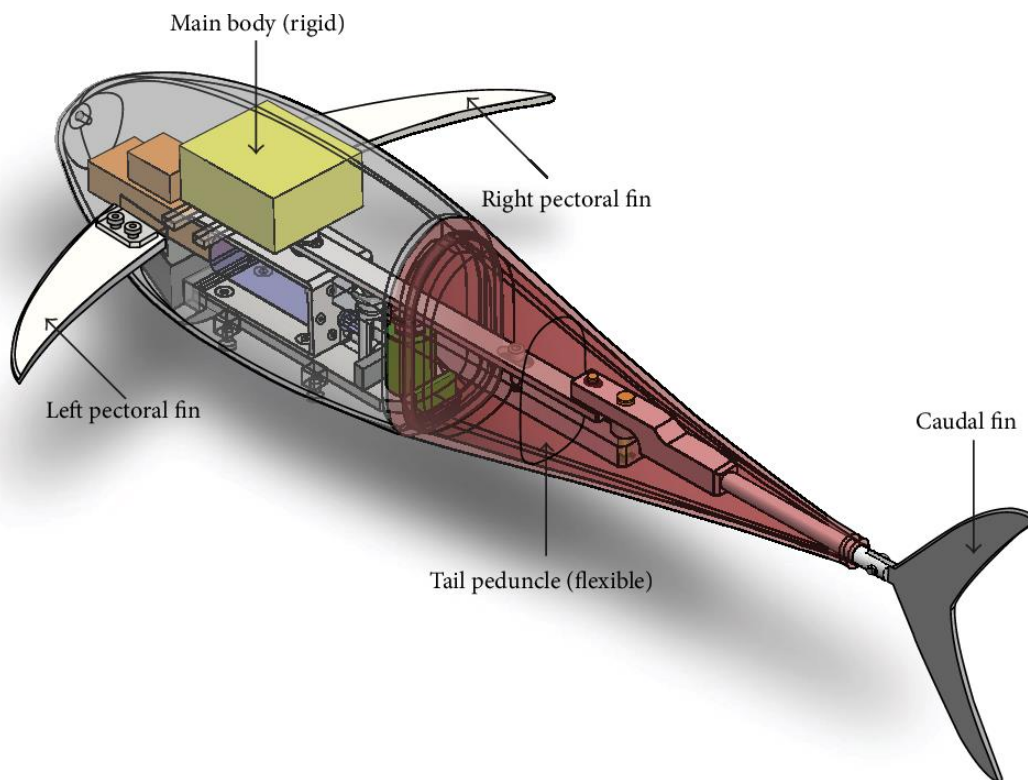


Рисунок 1.7 – Топологія «Плавання»

Під час проектування зооморфних роботів топології «Плавання», звертають увагу на:

- форма та структура Плаваючі роботи можуть мати різні форми та структури, від імітації риб до дизайнів, що наслідують структуру слизького тіла водяних істот. Розробники часто використовують гідродинамічні принципи для оптимізації форми робота для максимальної ефективності;

- актуатори та приводи. Плаваючі роботи можуть бути обладнані різними видами актуаторів та приводів, такими як електричні мотори, пневматичні системи чи підводні гвинти. Ці компоненти забезпечують рух та стабільність робота у водному середовищі;

– керування та навігація. Плаваючі роботи можуть використовувати різні системи для керування та навігації, включаючи гіростабілізатори, акселерометри, сенсори віддаленого виявлення та інші. Деякі роботи можуть використовувати системи штучного інтелекту для адаптації до змін у водному середовищі;

– безпроводний зв'язок. Зв'язок з плаваючими роботами може здійснюватися через безпроводові технології, такі як радіо чи акустичне зв'язку. Це важливо для передачі даних, використання дистанційного керування та взаємодії з іншими системами;

– задачі та застосування. Плаваючі роботи можуть виконувати різні завдання, включаючи збір даних про водне середовище, моніторинг забруднення води, дослідження морського життя, пошук та рятування в воді, а також допомогу у наукових дослідженнях;

– виклики та перспективи. Плаваючі роботи стикаються з викликами, такими як управління енергією, маневреність та комунікація у водному середовищі. Проте вони також мають великий потенціал для розв'язання проблем у морських науках та промисловості.

Одним із таких роботів є Robotuna, зображений на рисунку 1.8 [13].

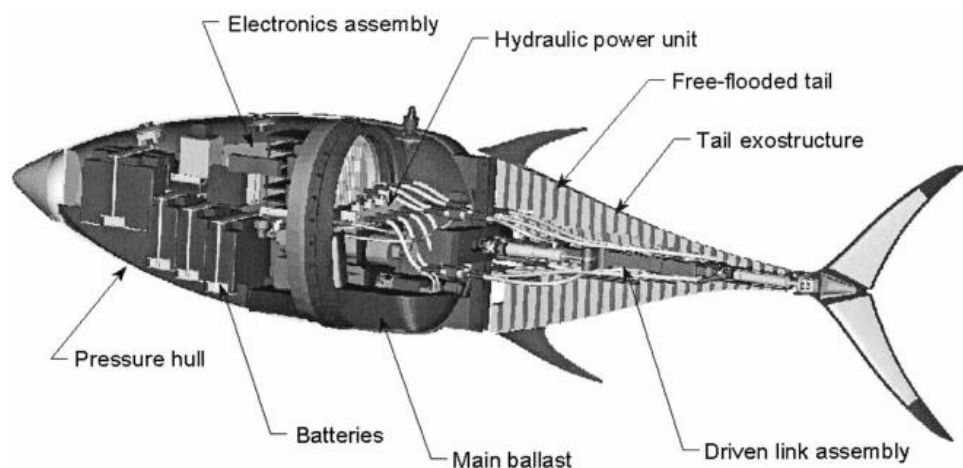


Рисунок 1.8 – Робот Robotuna

Проект розпочався в 1993 році. Мета розробників полягала в тому, щоб дослідити можливість побудови роботизованого підводного човна, який міг би відтворити те, як плавають тунці, і побачити, чи зможуть вони знайти кращу систему руху для автономних підводних апаратів. Експеримент був успішним, оскільки вони виявили, що їхня риба була маневренішою та споживала менше енергії, ніж інші роботизовані підводні човни.

Незважаючи на те, що перші результати були успішними, RoboTuna не зміг відтворити сплески прискорення, з якими справлявся справжній тунець. За словами дослідника Девіда Барретта, дослідники покращили дизайн за допомогою генетичного алгоритму, у якому найкращі системи «отримають віртуальне потомство». Ранні втілення працювали погано, але з розвитком системи можливості RoboTuna покращилися. Методи візуалізації показали, що система еволюціонувала таким чином, що RoboTuna використовувала вихори, які вона створювала. Помах його хвоста в один бік створював вихор, який потім використовувався змахом в інший бік, відштовхуючи його від створеного ним вихору. Ця техніка не тільки допомагає нормально плавати, але й пояснює вражаючу швидкість старту з місця справжнього тунця.

Ще однією з топологій є «Плазування», зображена на рисунку 1.9 [14]. Плазування може бути особливо ефективним для роботів, які працюють в умовах, де потрібно подолати нерівні поверхні або виконувати завдання у важкодоступних місцях.

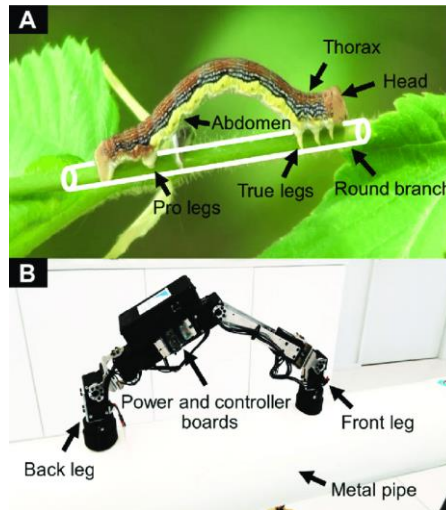


Рисунок 1.9 – Топологія «Плазування»

Ключові аспекти та приклади використання плазування у зооморфних роботах:

- структура та архітектура. Плазуючі роботи можуть мати гнучкі та адаптивні структури, які дозволяють їм подолати різні перешкоди та адаптуватися до форми поверхні;

- механізми переміщення. Вони можуть використовувати різні механізми для плазування, такі як гусениці, ноги з подібними до комах структурами або інші пристосування для пересування по поверхні;

- актуатори та сенсори. Плазуючі роботи зазвичай оснащені актуаторами, такими як електричні мотори або пневматичні системи, які дозволяють їм контролювати рух. Сенсори використовуються для збору інформації про навколишнє середовище та реагування на зміни;

- гнучкість та адаптивність. Гнучкість та адаптивність є ключовими характеристиками плазуючих роботів, оскільки вони можуть пристосовувати форму свого тіла до різних умов на поверхні;

- маневреність. Плазуючі роботи здатні виконувати різноманітні маневри для подолання перешкод, таких як лазіння по стінах, підняття вгору чи перетинання важкодоступних місць;

– застосування. Плазуючі роботи можуть мати застосування в пошуку та рятуванні, дослідженні важкодоступних місць, монтажі та обслуговуванні у важкодоступних середовищах, а також для виконання завдань в умовах, де інші типи руху були б неефективними.

Одним із таких роботів є Rhex. Rhex являє собою одну із самих вдалих конструкцій шестиногих роботів. Його ноги не мають суглобів і представляють собою дуго образні пружини, які обертаються навколо своєї осі, так що рухома робота забезпечує всього шість актуаторів. Гексапод Rhex дозволяє бігати зі швидкістю до 2,7 метра в секунду при довжині тіла близько півметра, долає майже 4 кілометри на сильно пересіченій місцевості на одній зарядці акумуляторів, підіймається на гірку з уклоном в 45 градусів, плаває і піддається під ступенем. Одна з версій робота Rhex зображена на рисунку 1.10 [15].

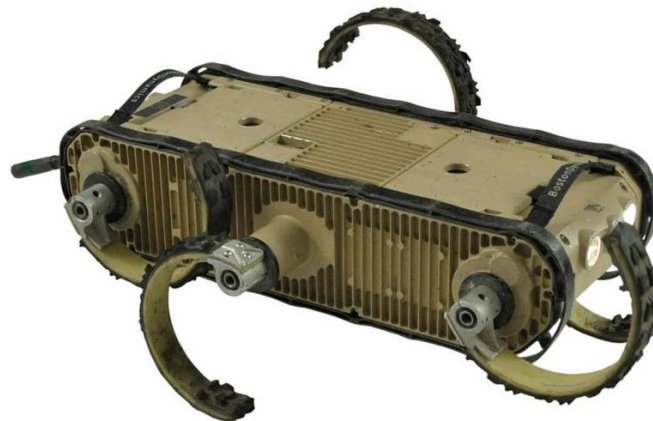


Рисунок 1.10 – Робот Rhex

Одне з останніх досягнень Rhex – стрибки. Стрибати робота навчили у лабораторіях Пенсільванського університету. Він перестрибує провалля шириною в 50 см з місця і 60 см з розбігу (вірніше, з розстрибу), робить зворотне сальто, застрибує на перешкоду заввишки 29 см, і чіпляється за уступ заввишки до 73 см, а потім підіймається на нього з висячого становища, правда це дається йому з великими труднощами – його ноги не призначені для лазіння.

Протягом свого розвитку Rhex придбав велику кількість можливостей у своєму поведінковому репертуарі. Фактично, це єдиний робот, який здатний виконувати таку різноманітну поведінку, як один автономний робот. Ця продуктивність пояснюється значною кількістю натхнення від вивчення біологічних систем, що призвело до низки принципів, що лежать в основі дизайну Rhex:

- використання ніг замість коліс або гусениць відкриває шлях для великої кількості видів поведінки;

- пасивна підтримка ніг долає обмеження, пов'язані з недостатнім спрацьовуванням, і допомагає спростити механічну конструкцію, забезпечуючи надійність;

- розпростерта поза, натхненна комахами, призводить до пасивної стабілізації бічного руху;

- контроль відкритий на рівні ходи, але замкнутий на рівні завдання. Стабільність забезпечується в результаті пасивної механіки, а не активного керування високою пропускною здатністю.

Незважаючи на великий список можливостей, в морфології Rhex все ще залишається безліч невирішених проблем. По-перше, гнучкість його конструкції на ніжках створює значний простір для додаткових дій. Крім того, його висока мобільність за природною місцевістю відкриває нові можливості для конкретних областей застосування, для яких компоненти для досягнення автономності все ще знаходяться в зачаточном стані. Нарешті, комерціалізація платформ Rhex вимагає серйозної розробки платформ, а також подальших поведених досліджень для підвищення її ефективності.

Наступною топологією є «М'язоподібна система» (MPS), яка зображена на рисунку 1.11 [16]. М'язоподібна система в робототехніці – це концепція проектування роботів, яка може імітувати будову та функцію м'язів та суглобів в тілі тварин. Вона намагається створити рухливі системи, що нагадують структуру та функцію м'язів у біологічних системах.

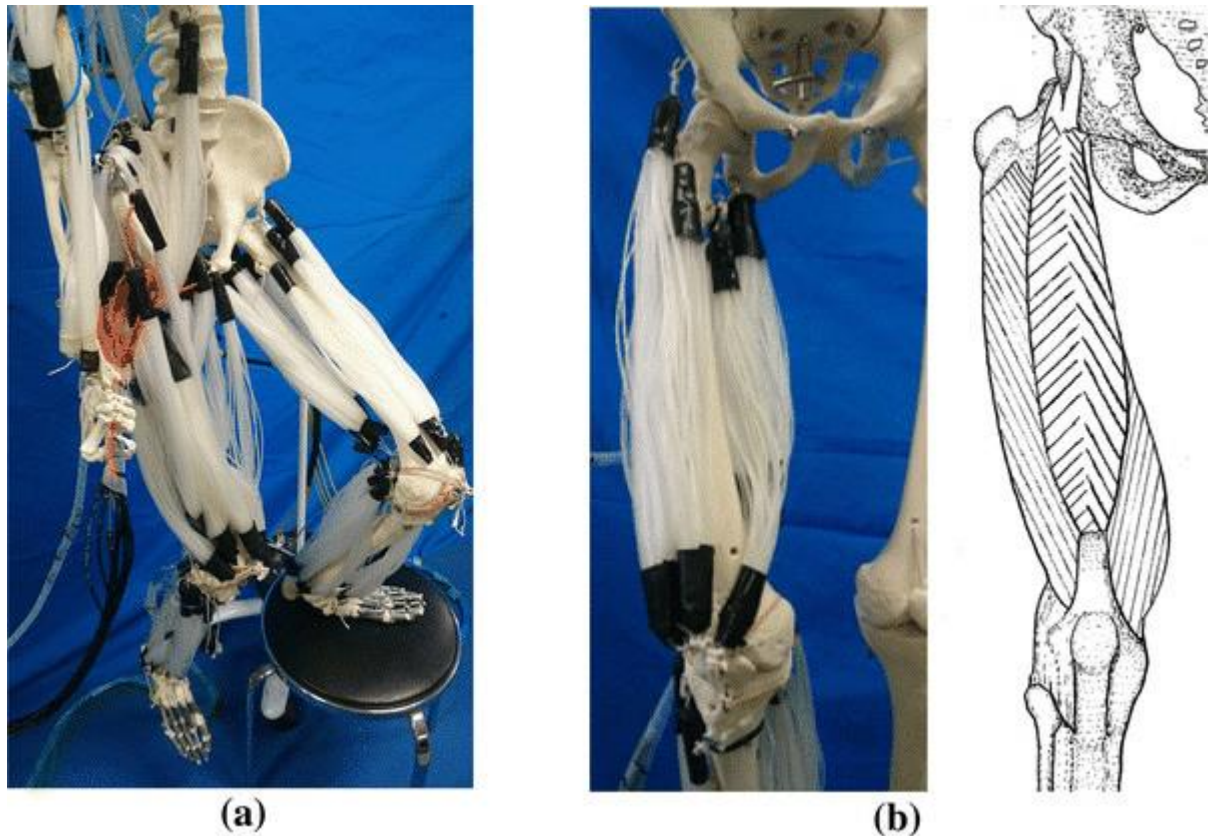


Рисунок 1.11 – Топологія «М'язоподібна система»

При проектуванні роботів по топології «М'язоподібна система» виділяють наступне:

- біоміметика. М'язоподібна система є однією з гілок біоміметики, де робототехніка бере в природі інспірацію для створення більш ефективних та адаптивних роботів;

- м'язи та актуатори. Замість традиційних механічних актуаторів, які використовуються у багатьох роботах, MPS використовує м'язоподібні структури. Це може включати штучні м'язи, виготовлені з еластомерів, пневматичних систем або полімерних матеріалів, які можуть стискатися та розслаблятися для відтворення рухів;

- гнучкість та адаптивність. М'язоподібна система дозволяє роботам бути більш гнучкими та адаптивними до змін в оточенні. Це особливо важливо для роботів, які повинні пристосовуватися до нерівних поверхонь або виконувати завдання в змінних умовах;

– імітація біологічних рухів. Роботи з м'язоподібною системою можуть імітувати біологічні рухи, що дозволяє їм більш ефективно взаємодіяти з навколишнім середовищем. Це може бути корисним в роботах для пошуку та рятування, дослідження важкодоступних місць чи хірургічних роботах;

– м'язоподібні сенсори. В м'язоподібних системах можуть бути включені сенсори, які слідкують за розтягненням та стисненням м'язів. Це надає роботу інформації про власне положення та реакцію на зовнішні впливи;

– відмінності від традиційних роботів. Традиційні роботи часто використовують електричні мотори та механічні суглоби. М'язоподібна система намагається подолати обмеження цих систем, створюючи більш гнучкі та легкі роботи;

– застосування. М'язоподібна система має потенціал в різних галузях, таких як медична робототехніка, дослідження природи та важкодоступних місць, а також для створення більш безпечних та гнучких роботів для взаємодії з людьми.

М'язоподібна система представляє собою інноваційний підхід до розробки роботів, який використовує біологічні концепції для створення більш ефективних та адаптивних механізмів руху.

Прикладом може бути маніпулятор Soft Gripper Harvard, зображений на рисунку 1.12 [17], який делікатно пересуває об'єкти.

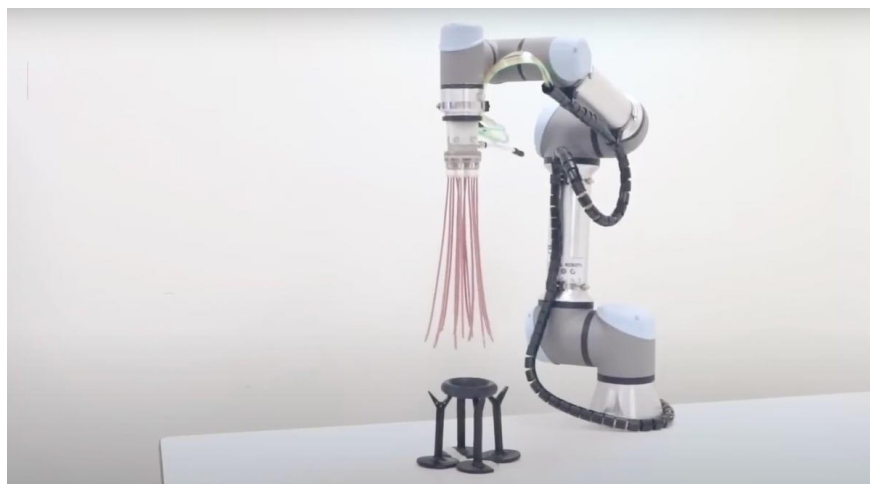


Рисунок 1.12 – Маніпулятор «Soft Gripper Harvard»

Більшість сучасних роботизованих захватів покладаються на вбудовані датчики, складні цикли зворотного зв'язку або розширені алгоритми машинного навчання в поєднанні з навичками оператора, щоб захоплювати крихкі об'єкти або об'єкти неправильної форми. Але дослідники з Гарвардської школи інженерії та прикладних наук імені Джона А. Полсона продемонстрували простіший спосіб.

Надихаючись природою, вони розробили новий тип м'якого роботизованого захвату, який використовує набір тонких щупалець, щоб заплутувати та ловити в пастку об'єкти, подібно до того, як медузи збирають оглушену здобич. Поодинокі окремі щупальця, або нитки, слабкі. Але разом сукупність ниток може захоплювати та надійно утримувати важкі об'єкти дивної форми. Захват покладається на просте надування, щоб охопити об'єкти, і не потребує датчиків, планування чи контролю зворотного зв'язку.

Сила та здатність до адаптації захоплення походять від його здатності заплутуватися в об'єкті, який він намагається схопити. Нитки довжиною футів являють собою порожнисті гумові трубки. Одна сторона трубки має товщу гуму, ніж інша, тому, коли трубка знаходиться під тиском, вона скручується, як косичка або як випрямлене волосся в дощовий день.

Локони зв'язуються в вузол і заплутуються один з одним і об'єктом, причому кожне заплутування збільшує міцність утримання. Хоча спільна фіксація сильна, кожен індивідуальний контакт слабкий і не пошкодить навіть найтендітніший предмет. Щоб звільнити об'єкт, нитки просто розгерметизують. Приклад захвату об'єкту роботом зображена на рисунку 1.13 [17].

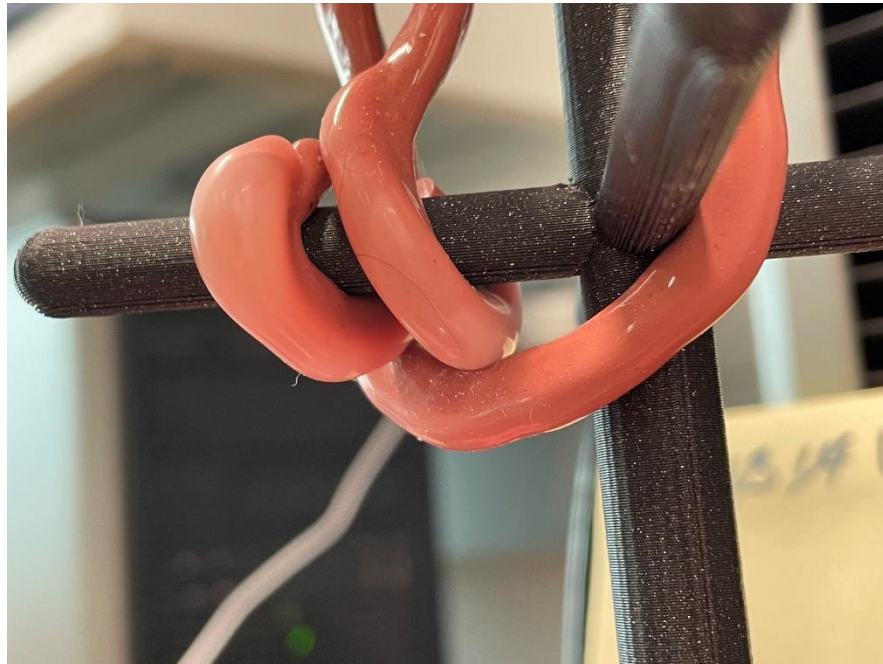


Рисунок 1.13 – Захват об'єкту роботом Soft Gripper Harvard

Дослідники використовували симуляції та експерименти, щоб перевірити ефективність захоплення, збираючи ряд об'єктів, включаючи різні кімнатні рослини та іграшки. Захват можна використовувати в реальних умовах для захоплення м'яких фруктів і овочів для сільськогосподарського виробництва та розповсюдження, делікатних тканин у медичних установах, навіть предметів неправильної форми на складах, таких як скляний посуд.

Також можна виділити топологію «Гусенична система», яка зображена на рисунку 1.14 [18].

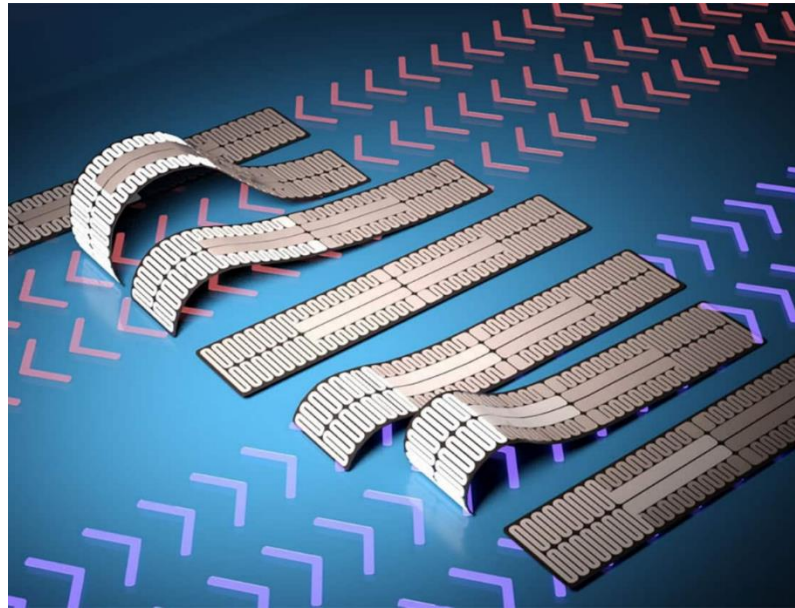


Рисунок 1.14 – Топологія «Гусенична система»

Гусенична система в робототехніці використовує конструкцію, аналогічну гусеницям, для забезпечення рухомого робота. Вона може мати ланки, що складаються з коліс або гусениць із сегментованою конструкцією, яка служить для подолання перешкод та руху в різних умовах.

Так, дослідники з Університету штату Північна Кароліна створили м'якого робота, який рухається, схоже на гусеницю. Як описано в дослідницькій роботі, розроблений ними робот складається з шару рідкокристалічних еластомерів і полідиметилсилоксану із вбудованим срібним нанодротом, який діє як нагрівач [19].

Таким чином, рідкокристалічний еластомер розроблено як термічний біморфний актуатор, який використовує чіткий коефіцієнт теплового розширення між сторонами рідкокристалічного еластомера та полідиметилсилоксану для створення деформації, яка добре контролюється, і, отже, руху. Оскільки нанодріт розділений на секції, які можна окремо нагрівати, точну деформацію можна досить жорстко контролювати, що забезпечує повзучий рух. На рисунку 1.15 [19], зображено переміщення цього робота.

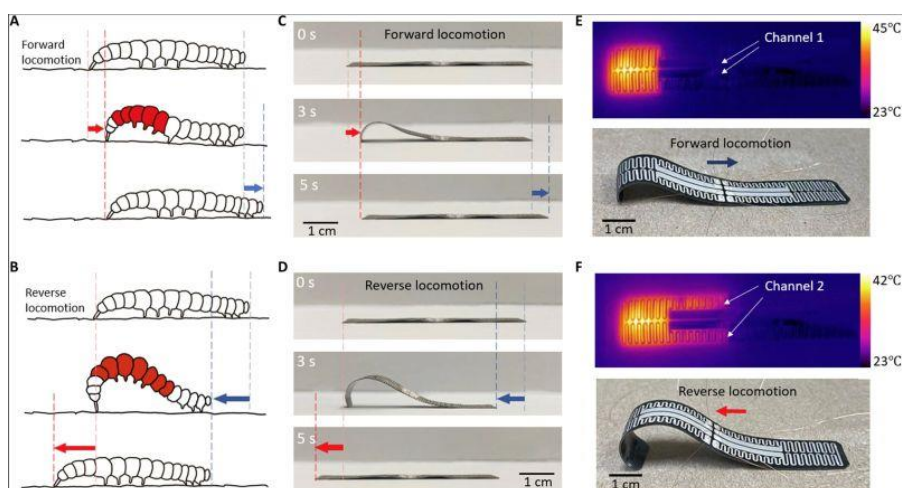


Рисунок 1.15 – Переміщення робота

Як видно з рисунку на ньому зображено:

- А – схема пересування гусениці вперед;
- В – схема зворотного руху гусениці;
- С – знімки повзаючого робота в одному циклі приведення в дію для зворотного пересування;
- D – знімки повзаючого робота в одному циклі приведення в дію для пересування вперед;
- Е – інфрачервоне зображення повзаючого робота зі струмом 0,05 А, поданого в канал 1, і похилий вид повзаючого робота;
- F – інфрачервоне зображення повзаючого робота зі струмом 30 мА, що подається в канал 2, і відповідний похилий вид повзаючого робота.

Головною особливістю цього робота є його здатність подолання вузьких місць, та інших схожих перешкод.

## 1.2 Переваги та недоліки крокуючих механізмів

Крокуючі механізми – це один із найбільш поширених типів механізмів руху для зооморфних роботів. Вони дозволяють роботам імітувати рухи тварин і пересуватися в складних умовах, таких як зарослі, стіни та інші перешкоди. Однак, як і будь-яка технологія, крокуючі механізми мають свої переваги та недоліки.

Однією з основних переваг крокуючих механізмів є їхня здатність проходження через зарослі, яка недоступна для колісних або гусеничних механізмів. Крокуючі роботи можуть легко підніматися по сходах, що дозволяє їм пересуватися в будівлях та інших обмежених просторах. Також, крокуючі механізми дозволяють роботам вести спостереження в різних умовах, включаючи труднодоступні місця, такі як гірські вершини або джунглі.

Однак, крокуючі механізми мають свої недоліки. Найбільш значущою проблемою є складність управління роботів з крокуючим механізмом. Ці механізми потребують більш складних алгоритмів управління, ніж звичайні механізми з колесами або гусеницями. Крокуючі роботи також потребують більшої кількості енергії для руху, що знижує час роботи на одному заряду батареї.

Однак, виробники зооморфних роботів продовжують займатися розвитком крокуючих механізмів, вдосконалюючи їхні функції та знижуючи їхню вартість. Це дозволяє створювати роботів з крокуючими механізмами, які здатні більш ефективно пересуватися в різних умовах. Наприклад, в наш час стали популярними роботи з крокуючими механізмами на основі м'язоподібних матеріалів, які відтворюють рух тварин на більш природний спосіб.

Також, крокуючі роботи можуть бути більш безпечним вибором для деяких видів діяльності, таких як пошукових і рятування робіт або медичних процедур. Вони можуть забезпечити точність та маневреність, що робить їх

корисними для дослідження складної місцевості, наприклад, підземних каналів або вулканів.

### 1.3 Види перешкод

Основними видами перешкод для зооморфних роботів є: сходи, пересічена місцевість, водне середовище, смуга перешкод. На відміні від колісних та гусеничних роботів, роботи які мають топологію шарнірних ніг без проблем перетинають пересічену місцевість та сходи. На рисунку 1.16 зображений робот BigDog [20], який пересувається лісом, тобто пересіченою місцевістю.



Рисунок 1.16 – Робот BigDog

Також, роботи такі роботи можуть пересуватися й вертикальними поверхнями. Так був розроблений робот Marvel, зображений на рисунку 1.17 [21]. Marvel призначений для виконання небезпечних робіт з обслуговування великих металевих конструкцій, таких як нафтові резервуари, мости та водонапірні башти. Він використовує комбінацію магнітних еластомерів та

електромагнітів для намагнічування та розмагнічування за необхідності, що дозволяє ногам робота триматися на поверхні або робити кроки. Перед роботою Marvel, подібно до кота, обережно перевіряє поверхню передніми лапами, щоб переконатися в її стійкості.

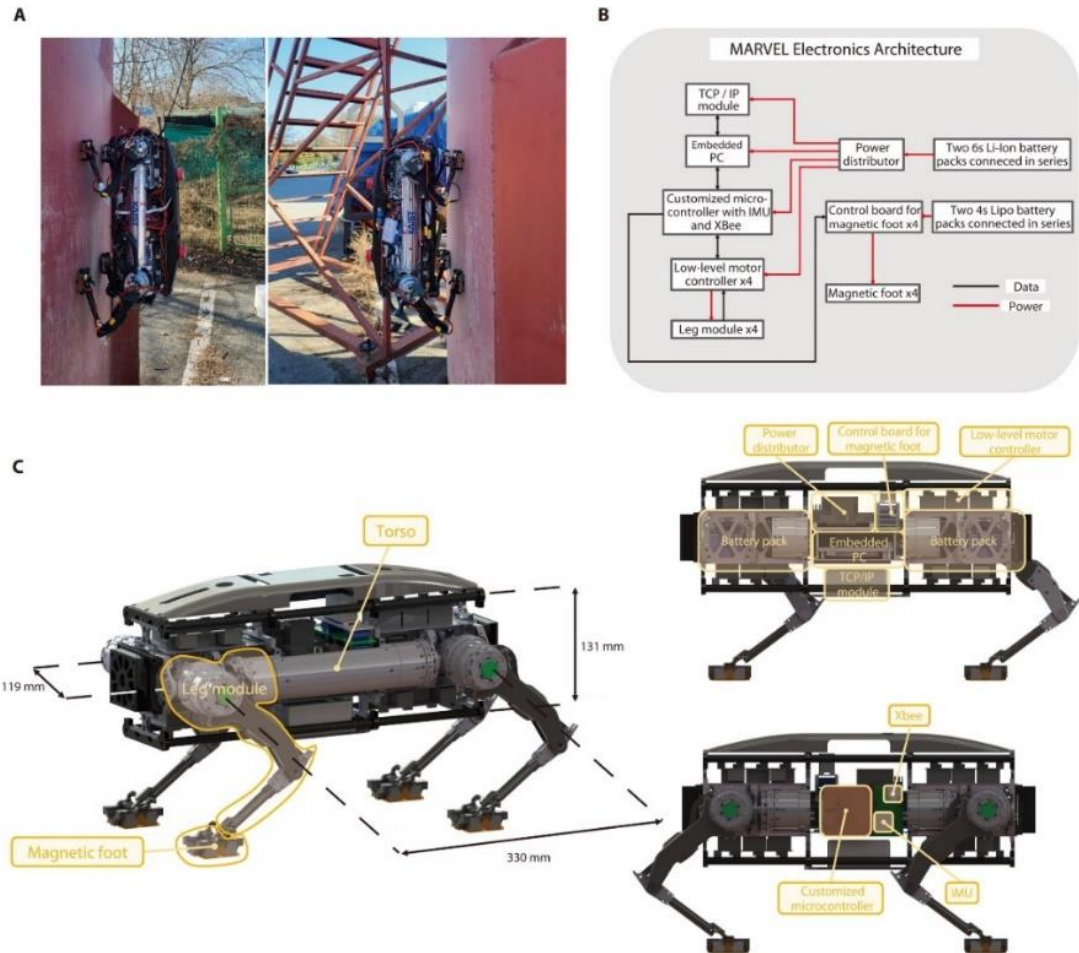


Рисунок 1.17 – Компоненти робота Marvel

У лабораторії роботу вдалося подолати щілини шириною 10 см та перешкоди висотою 5 см, а також переходити з горизонтальних на вертикальні поверхні та навпаки. Робот також може перевозити корисне навантаження вагою до 3 кілограмів. На рисунку 1.17 зображений робот Marvel та його компоненти.

## 1.4 Датчики орієнтації у просторі

Для орієнтації роботів у просторі можна виділити декілька основних датчиків: ультразвуковий датчик, гіроскоп та акселерометр, датчик лінії, також можна додати й камеру.

Ультразвуковий далекомір HC SR04 [22] – найвідоміший датчик для застосування в Arduino, Raspberry Pi, ESP8266 і ESP32 модулях. Дозволяє вимірювати відстань до об'єкта в діапазоні від 2 см до 400 см. Частіше всього цей датчик використовується при побудові роботів які повинні оминати перешкоди. HC SR04 має малі габарити та простий інтерфейс. Ультразвуковий датчик зображений на рисунку 1.18.

Ультразвуковий датчик відстані визначає відстань до об'єкта, вимірюючи час відображення звукової хвилі від об'єкта. Частота звукової хвилі знаходиться в межах частоти ультразвуку, що забезпечує концентрований напрямок звукової хвилі. Типовий ультразвуковий датчик відстані складається з двох мембран, одна з яких генерує звук, а інша реєструє відображене відлуння. Принцип роботи УЗ наступний: один з п'єзоелементів випромінює ультразвукову хвилю при подачі імпульсу тривалістю 15 мікросекунд, а інший п'єзоелемент приймає цю ж відображену хвилю від перешкоди. Потім змиритися час затримки від передачі до прийому хвилі, далі обчислюється відстань і передається сигнал на ногу Echo датчика, тривалістю пропорційною відстані до перешкоди. Нам залишається тільки подавати імпульс на датчик, прийняти його і обчислити відстань.



Рисунок 1.18 – Ультразвуковий датчик

Акселерометром є MPU6050 [23]. MPU6050 – трьох осьовий датчик прискорення (акселерометр) та кутової швидкості (гіроскоп). Використовується для отримання інформації про положення в просторі, русі, також може бути використаний як датчик вібрації та удару.

Акселерометр MPU 6050 зображений на рисунку 1.19.



Рисунок 1.19 – Акселерометр

### 1.5 Висновки до розділу

Дослідження переміщення зооморфних роботів є актуальною проблемою. Використання зооморфних роботів все частіше зустрічається в діяльності людини. Так як, робототехніка швидко розвивається, з'являються все більш досконалі роботи, які допомагають людині в різних сферах, таких як: обслуговування, дослідження, медицина, ремонтні роботи та ін. Покращення переміщення роботів зробить їх більш затребуваними у різних сферах діяльності людини.

## 2 ВИБІР І ОБГРУНТУВАННЯ ОСНОВНИХ ЕЛЕМЕНТІВ ДОСЛІДЖЕННЯ

Сучасний розвиток робототехніки відкриває перед нами нескінченні можливості для створення інтелектуальних мехатронних систем, які спроможні пристосовуватися до різноманітних умов і виконувати різні завдання. Однак, однією з ключових проблем є реалізація рухомих характеристик, які наближають роботів до біологічних організмів, зокрема, чотирилапих ссавців.

Данна дипломна робота присвячена дослідженню переміщення чотирилапого зооморфного робота «Робокіт» у невизначеному просторі. Вибір теми обґрунтовується актуальністю задачі розвитку зооморфних роботів, які можуть функціонувати в різних умовах і піднімати на новий рівень розуміння та імітацію рухів біологічних зразків.

Однією з ключових відмінностей біологічних організмів є їхня здатність до подолання нерівних поверхонь в навколишньому середовищі. Дослідження руху зооморфних роботів з топологією чотирилапих ссавців може відкрити нові можливості для розвитку зооморфних роботів, призначених для роботи в умовах, де попередні алгоритми були неефективними.

Цей розділ диплому розглядає важливий аспект дослідження, а саме вибір і обґрунтування основних елементів, які використовуються для моделювання та аналізу руху зооморфного робота «Робокіт». В ньому розглядається програмний засіб MatLab та засіб для моделювання нечіткої логіки – FuzzyLogicDesiner.

Мета даного дослідження полягає в створенні правил поведінки робота за допомогою нечіткою логіки, що буде являти собою моделювання переміщення зооморфного робота у невизначеному просторі. Також у розділі розкривається важливість вибору правильних параметрів і засобів для успішного виконання дослідження та реалізації поставлених цілей.

## 2.1 Вибір середовища для моделювання

Моделювання є однією з ключових стратегій досліджень у різних галузях науки і техніки, яка дозволяє аналізувати, прогнозувати та оптимізувати різноманітні процеси, системи та явища. Використання моделювання дозволяє вченим, інженерам, фахівцям та дослідникам вивчати складні проблеми, які б неможливо було вирішити аналітичним шляхом. Однак перед створенням моделі важливим є вибір середовища для її створення та аналізу.

У цьому розділі розглядаються важливі аспекти вибору середовища для моделювання зооморфного роботу «Робокіт» та обґрунтовується вибір MatLab як інструмента для вирішення завдання дослідження переміщення робота у невизначеному просторі. Так, середовище MatLab [24] має наступні особливості:

- можливість для математичного моделювання. Середовище MatLab володіє потужним набором функцій та бібліотек для математичного моделювання, обробки сигналів, обчислювальної математики та статистичного аналізу. Це середовище спрямоване на роботу з чисельними даними, що робить його ідеальним інструментом для обробки та аналізу результатів дослідження;

- легкість використання. MatLab має інтуїтивний інтерфейс, що дозволяє швидко навчитися його використовувати. Це особливо важливо в контексті досліджень, де час є обмеженим ресурсом. Простий синтаксис та графічний інтерфейс роблять його доступним для використання навіть для тих, хто не є професійним програмістом;

- загальна поширеність. MatLab є одним із найбільших поширених і використовуваних середовищ для наукових досліджень у різних областях, включаючи інженерію, фізику, біологію, економіку та інші. Це забезпечує доступність багатьох додаткових ресурсів, бібліотек та консультаційних послуг для вирішення специфічних завдань у вибраній галузі дослідження;

– підтримка для роботи з даними. MatLab підтримує роботу з різним типами даних, включаючи чисельні дані, тексти, графіки та сигнали. Це робить його ідеальним інструментом для аналізу та візуалізації результатів досліджень, а також для обробки інформації з різних джерел.

MatLab є потужним інструментом для моделювання роботів і робототехнічних систем. Він надає ряд можливостей для розробки та симуляції роботів у віртуальному середовищі. Способи, якими MatLab дозволяє моделювати роботів:

– розробка кінематичних та динамічних моделей робота. За допомогою MatLab можна створювати моделі руху роботів, включаючи їхню кінематику та динаміку. Це дозволяє аналізувати та прогнозувати робочі характеристики робота перед фізичною реалізацією;

– симуляція робочих операцій. MatLab надає можливість створювати віртуальні середовища, де можна симулювати роботи та їхні операції. Це дозволяє визначити ефективність робота у різних сценаріях;

– інтеграція з іншими інструментами. MatLab може бути інтегрований з різними інструментами і мовами програмування, що робить його універсальним інструментом для моделювання роботів у різних додаткових середовищах.

Так, для симуляції переміщення робота у невизначеному просторі використовується FuzzyLogicDesigner [25].

FuzzyLogicDesigner – це інтерактивний інструмент в середовищі MatLab, призначений для проектування та налаштування систем нечіткої логіки. За допомогою цього інструмента можна легко створювати та налаштовувати контролери на основі нечітких логічних правил для використання в системах керування, прийняття рішень та інших додатках.

Нечітка логіка – це форма багатозначної логіки, у якій значення істинності змінних може бути будь-яким дійсним числом від 0 до 1. Вона використовується для обробки концепції часткової істинності, де значення істинності може коливатися від повністю істинного до повністю хибного.

Навпаки, у булевій логіці істинними значеннями змінних можуть бути лише цілі числа 0 або 1.

Нечітка логіка базується на спостереженні, що люди приймають рішення на основі неточної та нечислової інформації. Нечіткі моделі або нечіткі множини є математичними засобами представлення невизначеності та неточності інформації (звідси термін нечіткість). Ці моделі мають здатність розпізнавати, представляти, маніпулювати, інтерпретувати та використовувати дані та інформацію, які є нечіткими та позбавленими певності [26][27].

Найбільш відомою системою є система, заснована на правилах Мамдані [14].

Він використовує такі правила:

- нечіткість усіх вхідних значень у нечіткій функції належності;
- виконайте всі відповідні правила в базі правил для обчислення функцій нечіткого виводу;
- дефазифікуйте функції нечіткого виведення, щоб отримати «чіткі» вихідні значення.

Фазифікація – це процес присвоєння числових даних системи нечітким множинам із деяким ступенем належності. Цей ступінь приналежності може бути де завгодно в межах інтервалу  $[0,1]$ . Якщо воно дорівнює 0, то значення не належить даному нечіткому набору, а якщо воно дорівнює 1, то значення повністю належить до нечіткого набору. Будь-яке значення від 0 до 1 означає ступінь невизначеності того, що значення належить до набору. Ці нечіткі множини зазвичай описуються словами, і тому, призначаючи вхідні дані системи нечітким множинам, ми можемо міркувати з ними лінгвістично природним способом.

Наприклад, на графіку представленому на рисунку 2.1 [28] значення виразів cold, warm і hot представлені функціями, що відображають температурну шкалу. Бал на цій шкалі має три «значення істинності» – по одному для кожної з трьох функцій. Вертикальна лінія на зображенні представляє певну температуру, яку вимірюють три стрілки (істинні значення).

Оскільки червона стрілка вказує на нуль, цю температуру можна інтерпретувати як «нежарко»; тобто ця температура має нульову приналежність до нечіткої множини «гаряче». Помаранчева стрілка (вказує на 0,2) може описати його як «трохи тепле», а синя стрілка (вказує на 0,8) «досить холодно». Таким чином, ця температура має 0,2 приналежності до нечіткої множини «тепла» та 0,8 приналежності до нечіткої множини «холодна». Ступінь приналежності, призначений для кожного нечіткого набору, є результатом фазифікації.

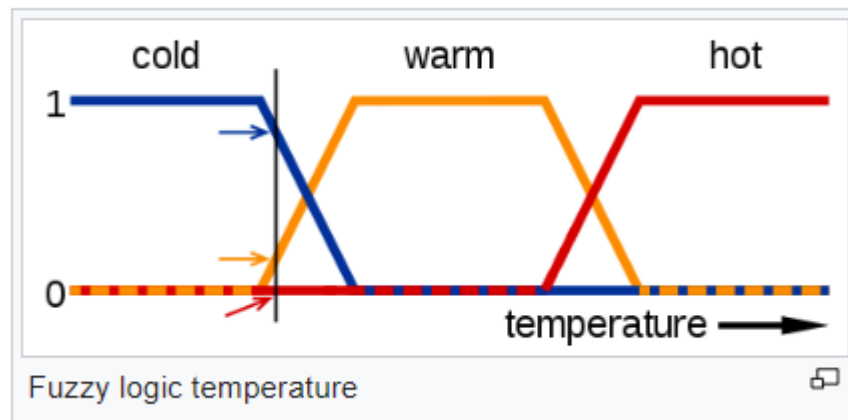


Рисунок 2.1 – Значення температури

Нечіткі множини часто визначаються як криві у формі трикутника або трапеції, оскільки кожне значення матиме нахил, при якому значення збільшується, пік, при якому значення дорівнює 1 (який може мати довжину 0 або більше), і нахил, при якому значення збільшується. Цінність знижується [29]. Їх також можна визначити за допомогою сигмоподібної функції [30]. Одним з найпоширеніших випадків є стандартна логістична функція, яка визначається як

$$S(x) = \frac{1}{1 + e^{-x}}, \quad (2.1)$$

який має наступну властивість симетрії:

$$S(x) + S(-x) = 1. \quad (2.2)$$

Звідси слідує що

$$(S(x) + S(-x)) \cdot (S(y) + S(-y)) \cdot (S(z) + S(-z)) = 1. \quad (2.3)$$

Нечітка логіка працює зі значеннями членства в такий спосіб, що імітує булеву логіку. Для цього мають бути доступні заміни на базові оператори AND, OR, NOT. І тому є кілька способів. Звичайна заміна називається Оператор Заде, представлені у таблиці 2.1.

Таблиця 2.1 – Оператор Заде

Логічне значення	Нечітке
AND(x,y)	MIN(x,y)
OR(x,y)	MAX(x,y)
NOT(x)	1-x

Для TRUE/1 і FALSE/0 нечіткі вирази дають той самий результат, що й логічні вирази.

Існують інші оператори, більш лінгвістичні за своєю природою, звані хеджами, які можна застосовувати. Зазвичай це прислівники, такі як «дуже» або «кілька», які змінюють значення набору за допомогою математичної формули.

Однак таблиця довільного вибору не завжди визначає функцію нечіткої логіки. Так було сформульовано критерій для визначення того, чи задана таблиця вибору визначає функцію нечіткої логіки, і запропоновано простий алгоритм синтезу функції нечіткої логіки на основі введених концепцій складових мінімуму і максимуму. Функція нечіткої логіки являє собою

диз'юнкцію складових мінімуму, де складовою мінімуму є кон'юнкція змінних поточної області, що перевищує або дорівнює значенню функції в цій області (праворуч від значення функції в нерівності, в тому числі значення функції).

Інший набір операторів AND/OR заснований на множенні:

$$x \text{ AND } y = x * y$$

$$\text{NOT } x = 1 - x$$

Hence,

$$x \text{ OR } y = \text{NOT}(\text{AND}(\text{NOT}(x), \text{NOT}(y)))$$

$$x \text{ OR } y = \text{NOT}(\text{AND}(1-x, 1-y))$$

$$x \text{ OR } y = \text{NOT}((1-x)*(1-y))$$

$$x \text{ OR } y = 1-(1-x)*(1-y)$$

$$x \text{ OR } y = x+y-xy$$

Враховуючи будь-які два з AND/OR/NOT, можна отримати третє. Узагальнення AND є прикладом t-норми.

Правила IF-THEN зіставляють вхідні чи обчислені значення істинності з бажаними вихідними значеннями істинності:

IF temperature IS very cold THEN fan\_speed is stopped

IF temperature IS cold THEN fan\_speed is slow

IF temperature IS warm THEN fan\_speed is moderate

IF temperature IS hot THEN fan\_speed is high

За певної температури нечітка змінна hot має певне значення істинності, яке копіюється в змінну high.

Якщо вихідна змінна зустрічається у кількох частинах THEN, значення з відповідних частин IF об'єднуються за допомогою оператора OR.

Нечітка логіка застосовувалася в багатьох областях, від теорії управління до штучного інтелекту.

Нейронна мережа приймає різноманітні цінні вхідні дані, надає їм різну вагу по відношенню один до одного та приймає рішення, яке зазвичай також має значення. Ніде в цьому процесі немає нічого подібного до послідовності рішень або-або, які характеризують нечітку математику, майже все комп'ютерне програмування та цифрову електроніку. У 1980-х роках дослідники розділилися щодо найефективнішого підходу до машинного навчання: дедуктивні моделі чи нейронні мережі. Перший підхід вимагає великих дерев рішень і використовує двійкову логіку, що відповідає апаратному забезпеченню, на якому він працює. Фізичні пристрої можуть бути обмежені двійковою логікою, але AI може використовувати програмне забезпечення для своїх обчислень. Нейронні мережі використовують цей підхід, що призводить до більш точних моделей складних ситуацій. Нейронні мережі незабаром знайшли свій шлях до безлічі електронних пристроїв [31].

Основні функції та можливості «FuzzyLogicDesigner» включають в себе наступне:

- створення нечітких логічних систем. FuzzyLogicDesigner дозволяє створювати нечіткі системи та визначити в них вхідні та вихідні змінні;
- додавання нечітких правил. Користувач може визначити нечіткі правила для системи, що описують, як система повинна відповідати на різні входи;
- візуалізація та графіки. FuzzyLogicDesigner надає графічний інтерфейс для відображення нечітких множин, правил та результатів вихідних змінних;
- налаштування параметрів нечітких функцій. Можна налаштувати форму нечітких функцій приналежності (трикутні, трапецевидні і т.д.);
- симуляція та тестування. Тестування та симулювання роботи нечіткої системи з використанням різних параметрів;
- генерація коду MatLab. Після створення та налаштування нечіткої логічної системи можна згенерувати код MatLab для подальшого використання в проекті.

FuzzyLogicDesigner робить процес проектування та налаштування нечітких логічних систем більш доступними та візуальними для користувачів MatLab, допомагаючи їм створювати ефективні системи керування та прийняття рішень, які враховують нечіткість та невизначеність вхідних даних.

## 2.2 Побудова схеми компонентів робота

Побудована схема компонентів робота зображена на рисунку 2.2.

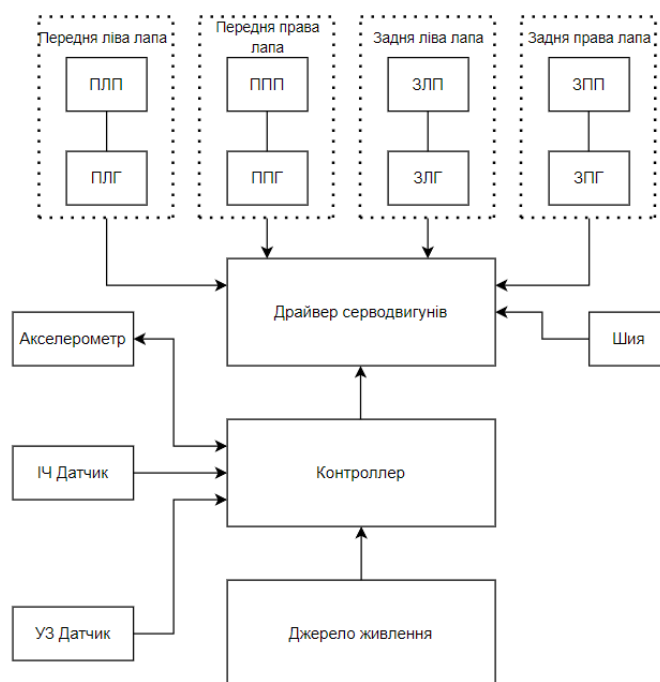


Рисунок 2.2 – Схема компонентів робота

Для розробки прототипу зооморфного робота Робокіт необхідні елементи такі як, плата курування, сервоприводуни, драйвера сервоприводів, інфрачервоний приймач, пульт керування, джерело живлення.

Слід звернути увагу на скорочення, вони представляють собою частини робота:

- ПЛП – переднє ліве плече;
- ПЛГ – передня ліва гомілка;
- ППП – переднє праве плече;

- ППГ – передня права гомілка;
- ЗЛП – заднє ліве плече;
- ЗЛГ – задня ліва гомілка;
- ЗПП – заднє праве плече;
- ЗПГ – задня права гомілка.

### 2.3 Приклад розрахунку нечіткої логіки

Так, можна взяти фактор безпечності на дорозі на основі двох факторів, а саме швидкості та відстані до іншого транспортного засобу.

Використаємо наступні фактори швидкості:

- низька швидкість (Low):  $\mu_{\text{Низька}}(60) = 1$  (якщо швидкість менше 60 км/год);
- середня швидкість (Medium):  $\mu_{\text{Середня}}(60) = 0.5$  (якщо швидкість 60 км/год);
- висока швидкість (High):  $\mu_{\text{Висока}}(60) = 0$  (якщо швидкість більше 60 км/год).

Використаємо наступні фактор відстані:

- безпечна відстань (Safe):  $\mu_{\text{Безпечна}}(10) = 1$  (якщо відстань більше 10 метрів);
- нормальна відстань (Moderate):  $\mu_{\text{Нормальна}}(10) = 0.5$  (якщо відстань 10 метрів);
- близька відстань (Close):  $\mu_{\text{Близька}}(10) = 0$  (якщо відстань менше 10 метрів).

Створимо правила, які занесені у таблицю 2.2.

Таблиця 2.2 – Правила нечіткої логіки

№	Швидкість	Відстань	Безпека
1	Низька	Безпечна	Висока
2	Низька	Нормально	Середня
3	Низька	Близька	Низька
4	Середня	Безпечна	Низька
5	Середня	Нормально	Середня
6	Середня	Близька	Висока
7	Висока	Безпечна	Середня
8	Висока	Нормально	Середня
9	Висока	Близька	Низька

Припустимо, що швидкість транспортного засобу дорівнює 80, а відстань до іншого автомобіля 12 метрів. Маємо:

$$\text{Швидкість} > 60 \text{ км/год, Відстань} > 10\text{м.} \quad (2.4)$$

Дивлячись на ці параметри, виконується правило 8 де значення безпеки середнє. Модуляція цього випадку зображена на рисунку 2.3.

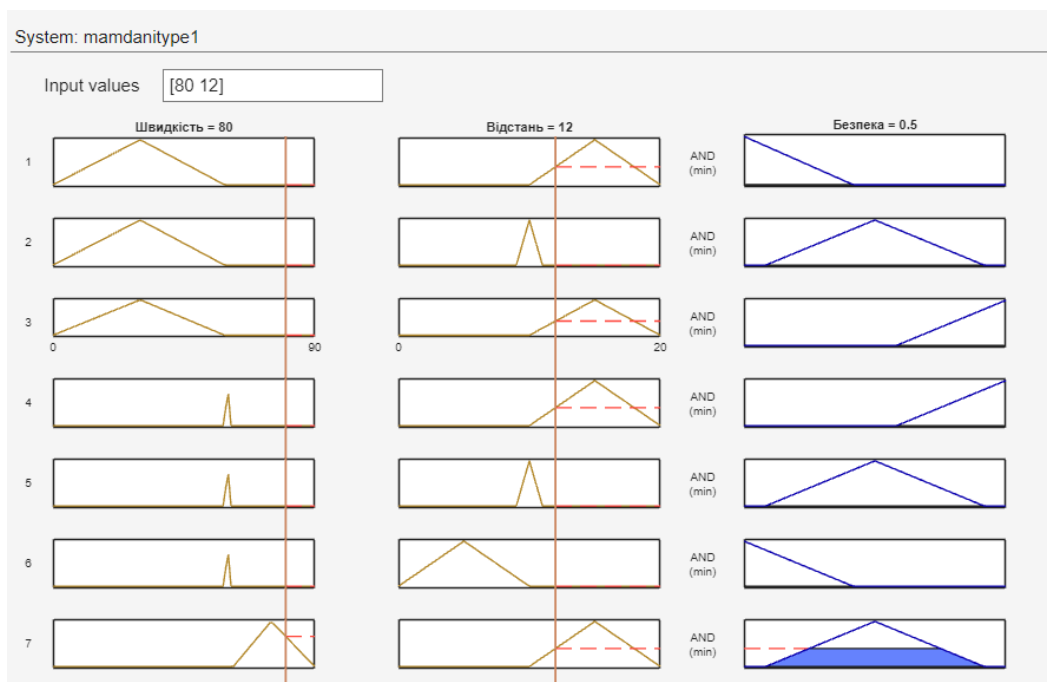


Рисунок 2.3 – Модуляція прикладу

Як видно з рисунку на виході ми отримали значення 0.5, що відповідає значенню безпеки «Середня».

Далі припустимо, що швидкість транспортного засобу дорівнює 80, а відстань до іншого автомобіля 2 метрів. Маємо:

$$\text{Швидкість} > 60 \text{ км/год, Відстань} < 10\text{м.} \quad (2.5)$$

Дивлячись на ці параметри, виконується правило 9 де значення безпеки низька. Модуляція цього випадку зображена на рисунку 2.4.

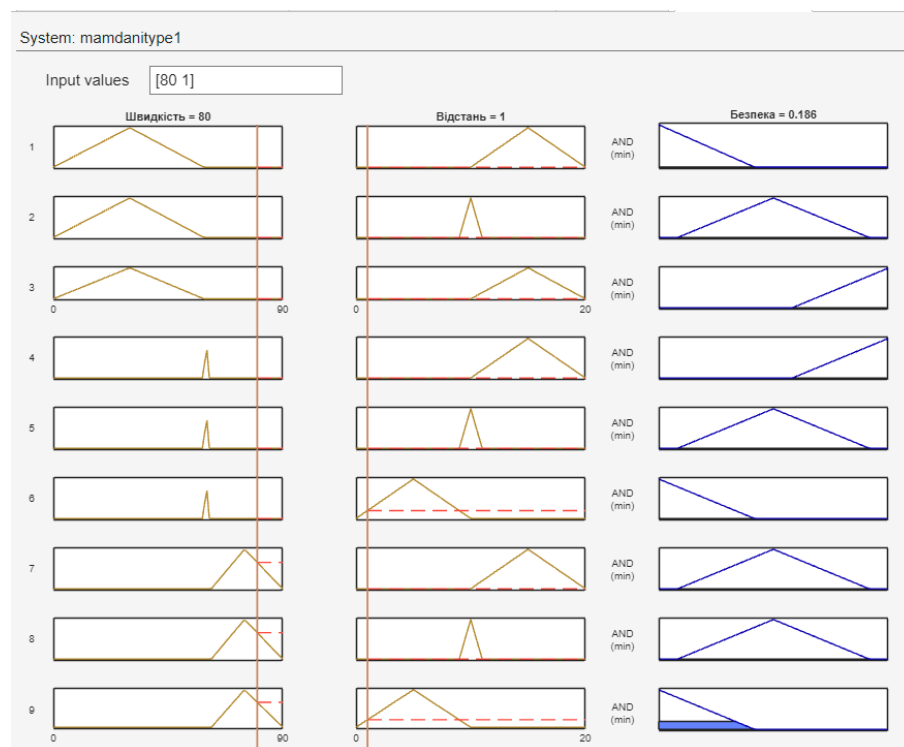


Рисунок 2.4 – Модуляція прикладу

Як видно з рисунку на виході ми отримали значення 0.186, що відповідає значенню безпеки «Низька».

Таким самим принципом виконуються і інші моделювання, це може бути значення температури, виробництво деталей та інше.

## 2.4 Приклад моделювання однією ноги

Моделювання ноги відбувається у FuzzyLogicDesigner, на вхід з правилами подається Input з параметрами, які представляють собою різну поведінку робота. В залежності від параметрів серводвигуни робота (ПЛП та ПЛГ) приймають різні положення, тобто обертаються на задану кількість градусів. Поведінки описані у блоці «Поведінка» який зображений на рисунку 2.5.

PROPERTY EDITOR: INPUT

Name: Поведінка

Range: [0 200]

Number of MFs: 20

Evenly Distribute MFs

Name	Type	Parameters
Початкове положення	Trapezoidal	[0.00610329 0.0469484 9.9061 10.0061]
Команда "Лежати"	Trapezoidal	[10.2 10.6573 19.5775 20.1]
Команда "Сидіти"	Trapezoidal	[20.59 20.7466 30.049 30.1]
Команда "Вияляння хвостом"	Triangular	[30.2 34.56 40.1]
Команда "Мурлякання"	Trapezoidal	[40.2 40.2081 49.6328 50.1]
Команда "Дати лапу"	Trapezoidal	[50.2 50.2448 60.0367 60.1]
Команда "Переміщення"	Triangular	[60.2 65 70.4406]
Команда "Охота"	Trapezoidal	[80.2 80.355 89.7797 90.1]
Команда "Розтягнутися"	Trapezoidal	[90.2 90.7589 99.5716 100.1]
Команда "Переміщення 2"	Triangular	[70.1958 75 80]
Команда "Дати лапу 2я 2"	Triangular	[100.6 105 110]
Команда "Витягнути задні ла..."	Triangular	[110.5 115 120]
Команда "Витягнути передні ..."	Triangular	[120.5 125 130]
Команда "Шипіння"	Triangular	[130.5 135 140]
Команда "Легти на правий бок"	Triangular	[140.5 145 150]
Команда "Легти на лівий бок"	Triangular	[150.5 155 160]
Команда "Витягнути задню л..."	Triangular	[160.5 165 170]
Команда "Витягнути задню п..."	Triangular	[170.5 175 180]
Команда "Витягнути передн..."	Triangular	[180.5 185 190]
Команда "Витягнути задню п..."	Triangular	[190.5 195 200]

Рисунок 2.5 – Поведінки робота

Правила описані у блоці «Mambani type 1» який представлений на рисунку 2.6, у цей блок додані усі правила які використовуються у проекті.

System: diplom\_m

Add All Possible Rules Clear All Rules

	Rule	Weight	Name
1	If Поведінка is Початкове положення then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 3, ППГ is Поворот 3, ЗЛП is Поворо...	1	rule1
2	If Поведінка is Команда "Лежати" then ПЛП is Поворот 5, ПЛГ is Поворот 5, ППП is Поворот 5, ППГ is Поворот 5, ЗЛП is Поворот 5, ...	1	rule2
3	If Поведінка is Команда "Сидіти" then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 3, ППГ is Поворот 3, ЗЛП is Поворот 3, ...	1	rule3
4	If Поведінка is Команда "Вилляння хвостом" then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 3, ППГ is Поворот 3, ЗЛП is П...	1	rule4
5	If Поведінка is Команда "Мурликання" and Відстань is not Відстань M then ПЛП is Поворот 1, ПЛГ is Поворот 1, ППП is Поворот 1, ППГ is Поворот 1, ЗЛП is Поворот 1, ЗЛГ is Поворот 3, ЗЛП is Поворот 3, ЗЛГ is Поворот 3, ЗЛП is Поворот 3, ЗЛГ is Поворот 3, Хвіст is Вилляти хвостом 1, Елемент is Мурликання	1	rule5
6		1	rule6
7	If Поведінка is Команда "Розтягнутися" then ПЛП is Поворот 1, ПЛГ is Поворот 1, ППП is Поворот 1, ППГ is Поворот 1, ЗЛП is Пово...	1	rule7
8	If Поведінка is Команда "Дати лапу" and Відстань is Відстань M then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 2, ППГ is ...	1	rule8
9	If Поведінка is Команда "Переміщення" and Відстань is Відстань M then ПЛП is Поворот 1, ПЛГ is Поворот 1, ППП is Поворот 1, ПП...	1	rule9
10	If Поведінка is Команда "Переміщення 2" and Відстань is Відстань M then ПЛП is Поворот 1, ПЛГ is Поворот 1, ППП is Поворот 1, П...	1	rule10
11	If Поведінка is Команда "Дати лапу 2я 2" then ПЛП is Поворот 2, ПЛГ is Поворот 2, ППП is Поворот 3, ППГ is Поворот 3, ЗЛП is Пов...	1	rule11
12	If Поведінка is Команда "Витягнути задні лапи" then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 3, ППГ is Поворот 3, ЗЛП...	1	rule12
13	If Поведінка is Команда "Витягнути передні лапи" then ПЛП is Поворот 5, ПЛГ is Поворот 5, ППП is Поворот 5, ППГ is Поворот 5, ЗЛП...	1	rule13
14	If Поведінка is Команда "Шипіння" and Відстань is Відстань Ш then ПЛП is Поворот 3, ПЛГ is Поворот 1, ППП is Поворот 3, ППГ is ...	1	rule14
15	If Поведінка is Команда "Легти на правий бок" then ПЛП is Поворот 1, ПЛГ is Поворот 1, ППП is Поворот 1, ППГ is Поворот 1, ЗЛП i...	1	rule15
16	If Поведінка is Команда "Легти на лівий бок" then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 3, ППГ is Поворот 3, ЗЛП is ...	1	rule16
17	If Поведінка is Команда "Витягнути задню ліву лапу" then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 3, ППГ is Поворот 3...	1	rule17
18	If Поведінка is Команда "Витягнути задню праву лапу" then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 3, ППГ is Поворо...	1	rule18
19	If Поведінка is Команда "Витягнути передню праву та задню ліву лапу" then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 4...	1	rule19
20	If Поведінка is Команда "Витягнути задню передню ліву та задню праву лапу" then ПЛП is Поворот 4, ПЛГ is Поворот 4, ППП is Пово...	1	rule20

Рисунок 2.6 – Правила проекту

Параметри які приймають блоки ПЛП та ПЛГ представлені на рисунку 2.7 та 2.8.

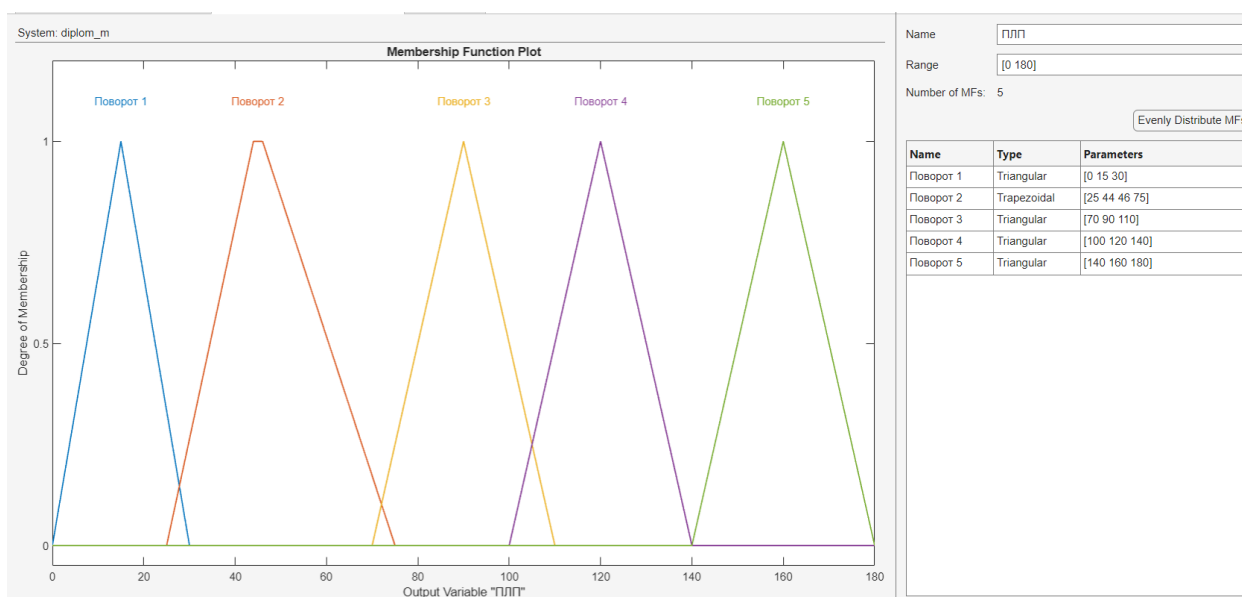


Рисунок 2.7 – Параметри блока ПЛП

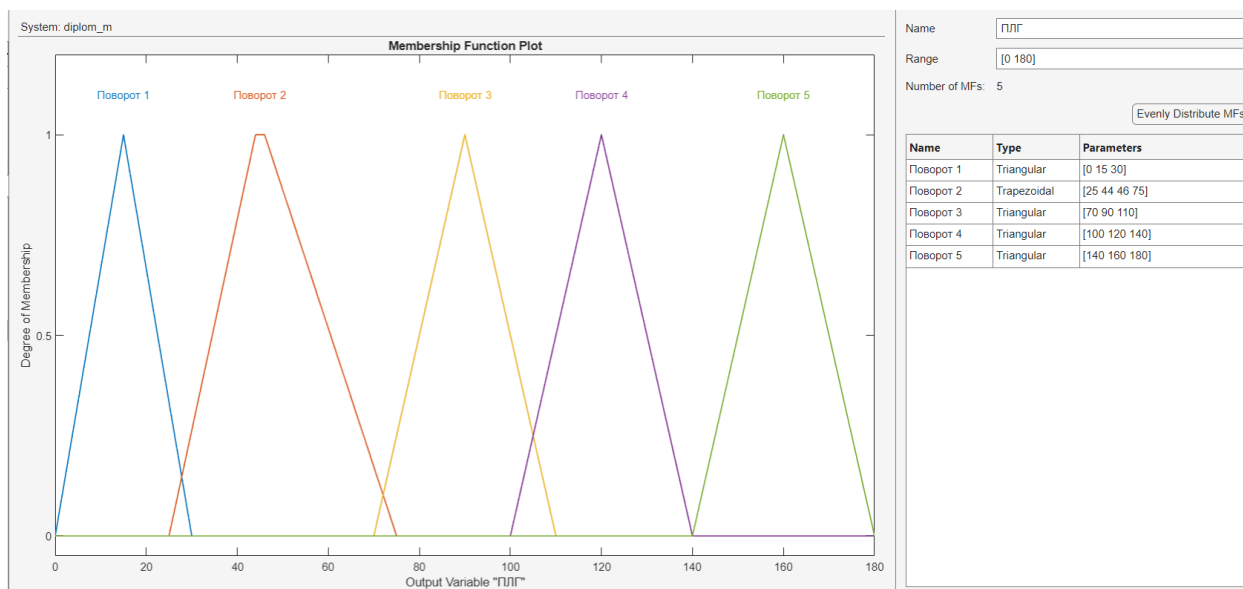


Рисунок 2.8 – Параметри блока ПЛГ

За допомогою функції «Rule interface» можна почати моделювання поведінки робота.

Так, якщо вхідним параметром буде «3», яке входить у проміжок «Початкове положення», то на виході ПЛП та ПЛГ буде 90. Приклад моделювання зображено на рисунку 2.9.

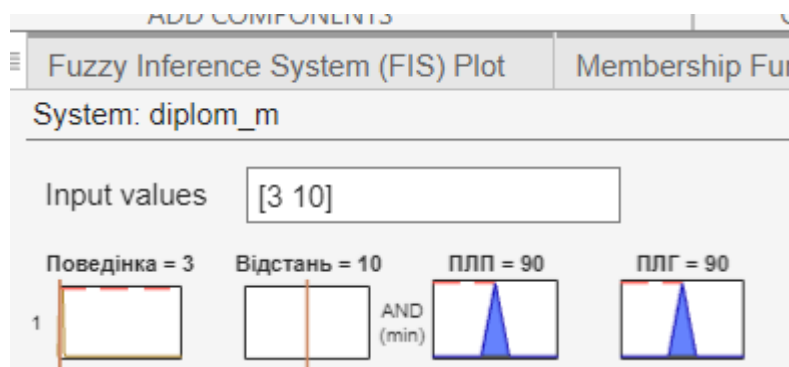


Рисунок 2.9 – Поведінка ПЛП та ПЛГ при вхідному параметрі «3»

Як видно з рисунку, на виході були отримані бажані значення, які дорівнюють куту повороту серводвигуна.

Так само відбувається моделювання і для інших поведінок, наприклад при вхідному параметрі «18», вихідним параметром ПЛП та ПЛГ буде 160. Приклад моделювання зображено на рисунку 2.10.

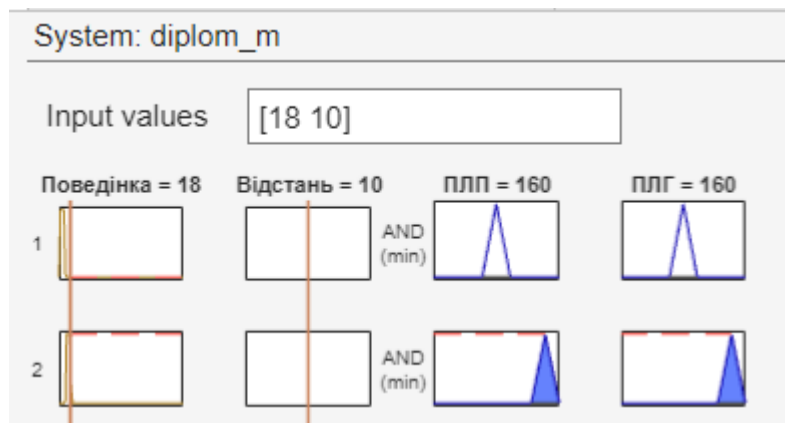


Рисунок 2.10 – Поведінка ПЛП та ПЛГ при вхідному параметрі «18»

Як видно з рисунку, на виході були отримані бажані значення.

Слід зазначити, що бажані значення налаштовуються при створенні правил. Приклад налаштування правила «Початкове положення», зображено на рисунку 2.11.

Fuzzy Inference System (FIS) Plot    Membership Function (MF) Editor    Rule Editor    Rule Inference

System: diplom\_m

Add All Possible Rules    Clear All Rules

Rule	Weight	Name
1 If Поведінка is Початкове положення then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 3, ППГ is Поворот 3, ЗЛП is Поворо...	1	rule1
2 If Поведінка is Команда "Лекати" then ПЛП is Поворот 5, ПЛГ is Поворот 5, ППП is Поворот 5, ППГ is Поворот 5, ЗЛП is Поворот 5, ...	1	rule2
3 If Поведінка is Команда "Сидіти" then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 3, ППГ is Поворот 3, ЗЛП is Поворот 3, ...	1	rule3
4 If Поведінка is Команда "Вилягти хвостом" then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 3, ППГ is Поворот 3, ЗЛП is П...	1	rule4
5 If Поведінка is Команда "Мурликання" and Відстань is not Відстань M then ПЛП is Поворот 1, ПЛГ is Поворот 1, ППП is Поворот 1, ...	1	rule5
6 If Поведінка is Команда "Скочити" then ПЛП is Поворот 5, ПЛГ is Поворот 5, ППП is Поворот 5, ППГ is Поворот 5, ЗЛП is Поворот 5, 3...	1	rule6
7 If Поведінка is Команда "Розтягнутися" then ПЛП is Поворот 1, ПЛГ is Поворот 1, ППП is Поворот 1, ППГ is Поворот 1, ЗЛП is Пове...	1	rule7
8 If Поведінка is Команда "Дати лапу" and Відстань is Відстань M then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 2, ППГ is ...	1	rule8
9 If Поведінка is Команда "Переміщення" and Відстань is Відстань M then ПЛП is Поворот 1, ПЛГ is Поворот 1, ППП is Поворот 1, ПП...	1	rule9
10 If Поведінка is Команда "Переміщення 2" and Відстань is Відстань M then ПЛП is Поворот 1, ПЛГ is Поворот 1, ППП is Поворот 1, П...	1	rule10
11 If Поведінка is Команда "Дати лапу 2a 2" then ПЛП is Поворот 2, ПЛГ is Поворот 2, ППП is Поворот 3, ППГ is Поворот 3, ЗЛП is Пов...	1	rule11
12 If Поведінка is Команда "Витягнути задні лапи" then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 3, ППГ is Поворот 3, ЗЛП...	1	rule12
13 If Поведінка is Команда "Витягнути передні лапи" then ПЛП is Поворот 5, ПЛГ is Поворот 5, ППП is Поворот 5, ППГ is Поворот 5, 3...	1	rule13
14 If Поведінка is Команда "Шалінути" and Відстань is Відстань Ш then ПЛП is Поворот 3, ПЛГ is Поворот 1, ППП is Поворот 3, ППГ is ...	1	rule14
15 If Поведінка is Команда "Летти на правий бок" then ПЛП is Поворот 1, ПЛГ is Поворот 1, ППП is Поворот 1, ППГ is Поворот 1, ЗЛП i...	1	rule15
16 If Поведінка is Команда "Летти на лівий бок" then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 3, ППГ is Поворот 3, ЗЛП i...	1	rule16
17 If Поведінка is Команда "Витягнути задню ліву лапу" then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 3, ППГ is Поворот 3...	1	rule17
18 If Поведінка is Команда "Витягнути задню праву лапу" then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 3, ППГ is Поворо...	1	rule18
19 If Поведінка is Команда "Витягнути передню праву та задню ліву лапу" then ПЛП is Поворот 3, ПЛГ is Поворот 3, ППП is Поворот 4...	1	rule19
20 If Поведінка is Команда "Витягнути задню передню ліву та задню праву лапу" then ПЛП is Поворот 4, ПЛГ is Поворот 4, ППП is По...	1	rule20

PROPERTY EDITOR RULE

Name: rule1

Weight: 1

Connection:  And     Or

If

Поведінка is Початкове н... and

Відстань is none

Then

ПЛП is Поворот 3

ПЛГ is Поворот 3

ППП is Поворот 3

ПЛГ is Поворот 3

ЗЛП is Поворот 3

ЗЛГ is Поворот 3

ЗЛП is Поворот 3

Рисунок 2.11 – Налаштування правила

## 2.5 Висновки до розділу

Fuzzy Logic Designer це інструмент в середовищі MATLAB. Інструмент надає інженерам та дослідникам зручний та інтуїтивний графічний інтерфейс для розробки, редагування та налаштування нечітких систем керування. Завдяки Fuzzy Logic Designer, можна легко визначати лінгвістичні змінні, терміни, правила та входи-виходи нечіткої системи.

Цей інструмент є потужним інструментом для моделювання та аналізу нечітких систем, що дозволяє ефективно вирішувати завдання в умовах невизначеності. Використання Fuzzy Logic Designer у дипломному проекті було ключовим для розробки та налаштування нечіткої системи керування, що дозволило досягти високої продуктивності та точності в рішеннях.

Використання нечіткої логіки стало важливою складовою роботи та сприяло досягненню успіху у розв'язанні складних завдань керування.

У контексті дослідження переміщення чотирилапого зооморфного робота «Робокіт» у невизначеному просторі, нечітка логіка використовується для моделювання нечіткості або невизначеності в різних аспектах руху робота. Наприклад, при визначенні траєкторії руху робота або при взаємодії з навколишнім середовищем можуть виникати ситуації, коли точні значення важко визначити. В таких випадках нечітка логіка дозволяє робити висновки на основі нечітких або розмитих даних.

У дипломній роботі нечітка логіка використовується для моделювання поведінки робота в умовах невизначеності простору, а також для прийняття рішень у ситуаціях, коли інформація є нечіткою або не повністю визначеною.

### 3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ДЛЯ ЗООМОРФНОГО РОБОТУ

#### 3.1 Розробка блок-схеми алгоритму

Розробка блок-схеми алгоритму роботи представляє собою виконання відповідної програми послідовності дій та операцій, які залежать від вхідних даних.

Розроблений алгоритм представлено на рисунку 3.1.

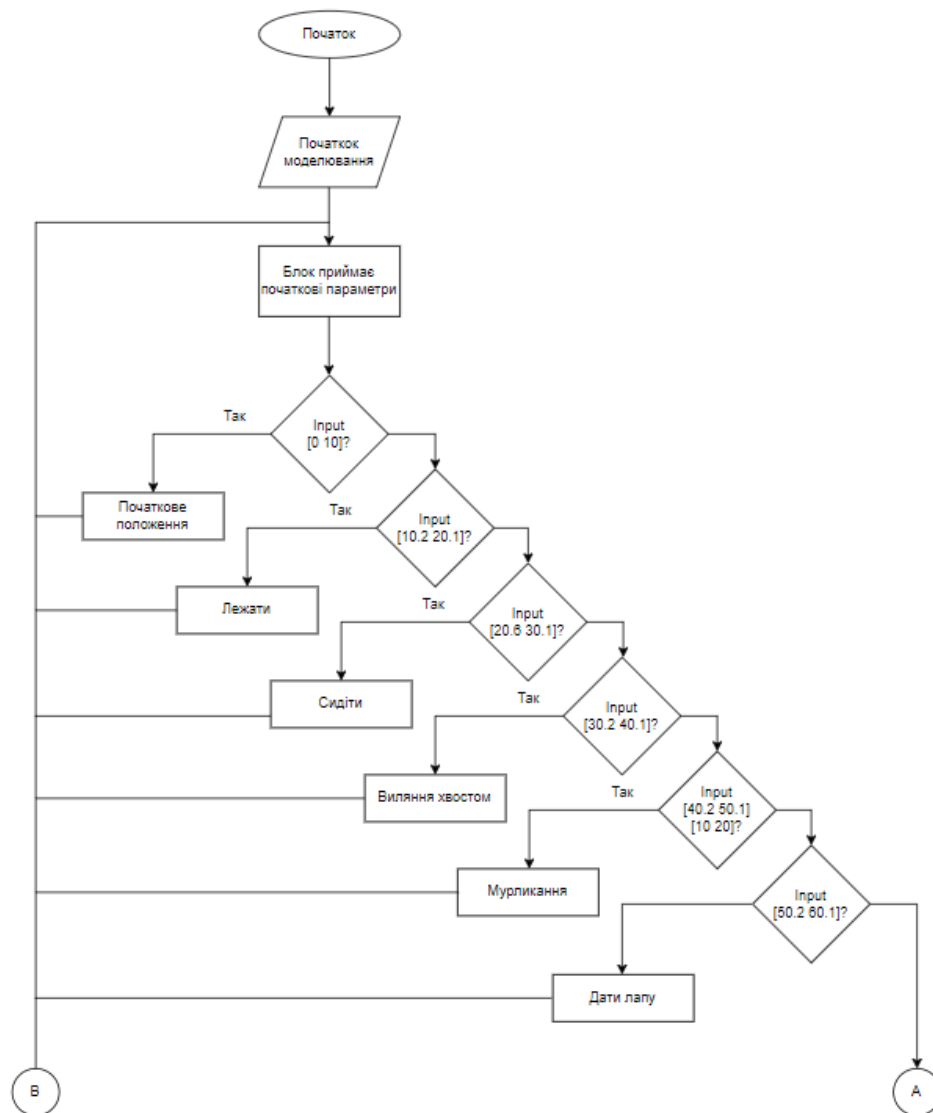


Рисунок 3.1 – Алгоритм роботи

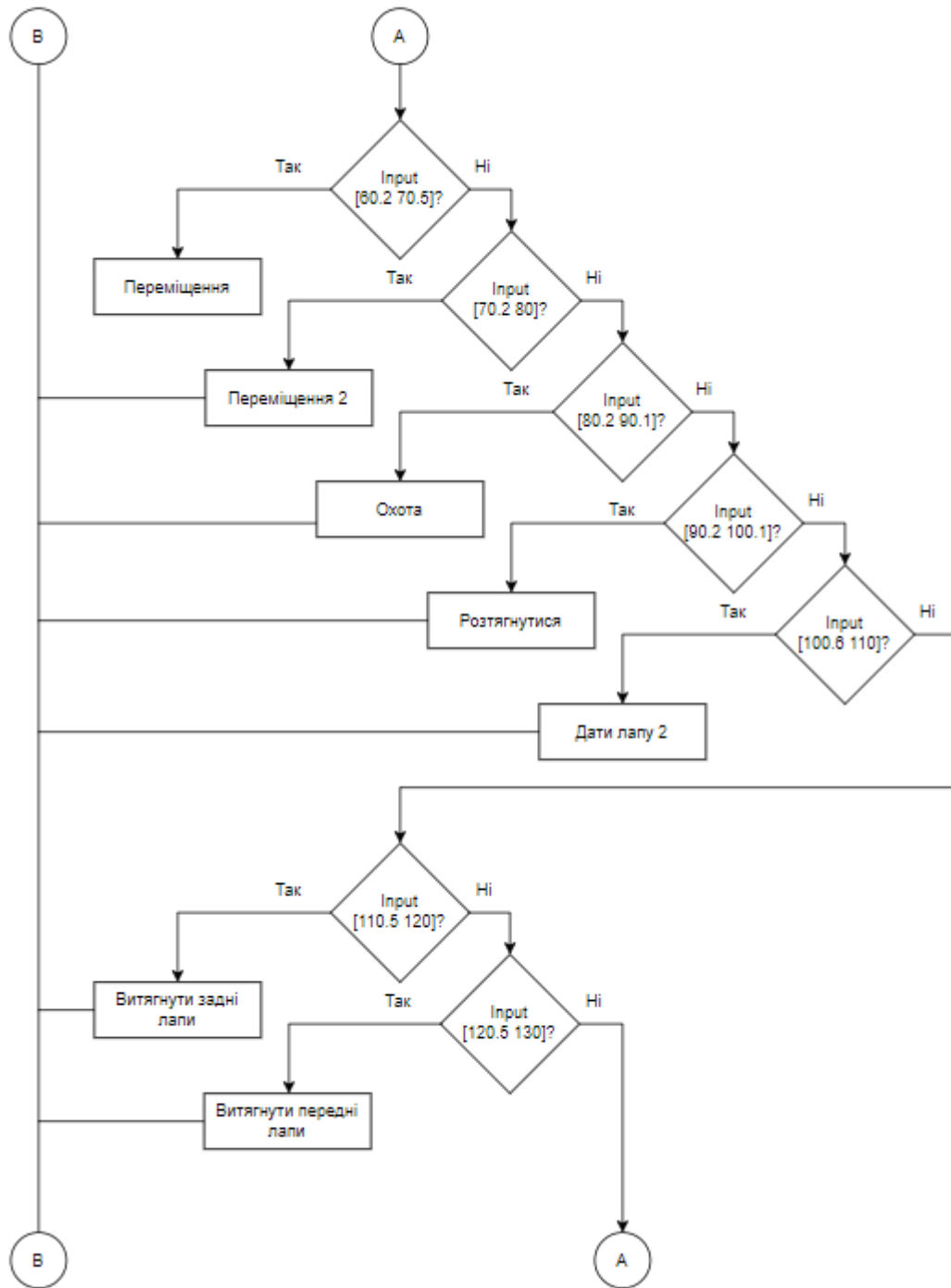


Рисунок 3.1, аркуш 2

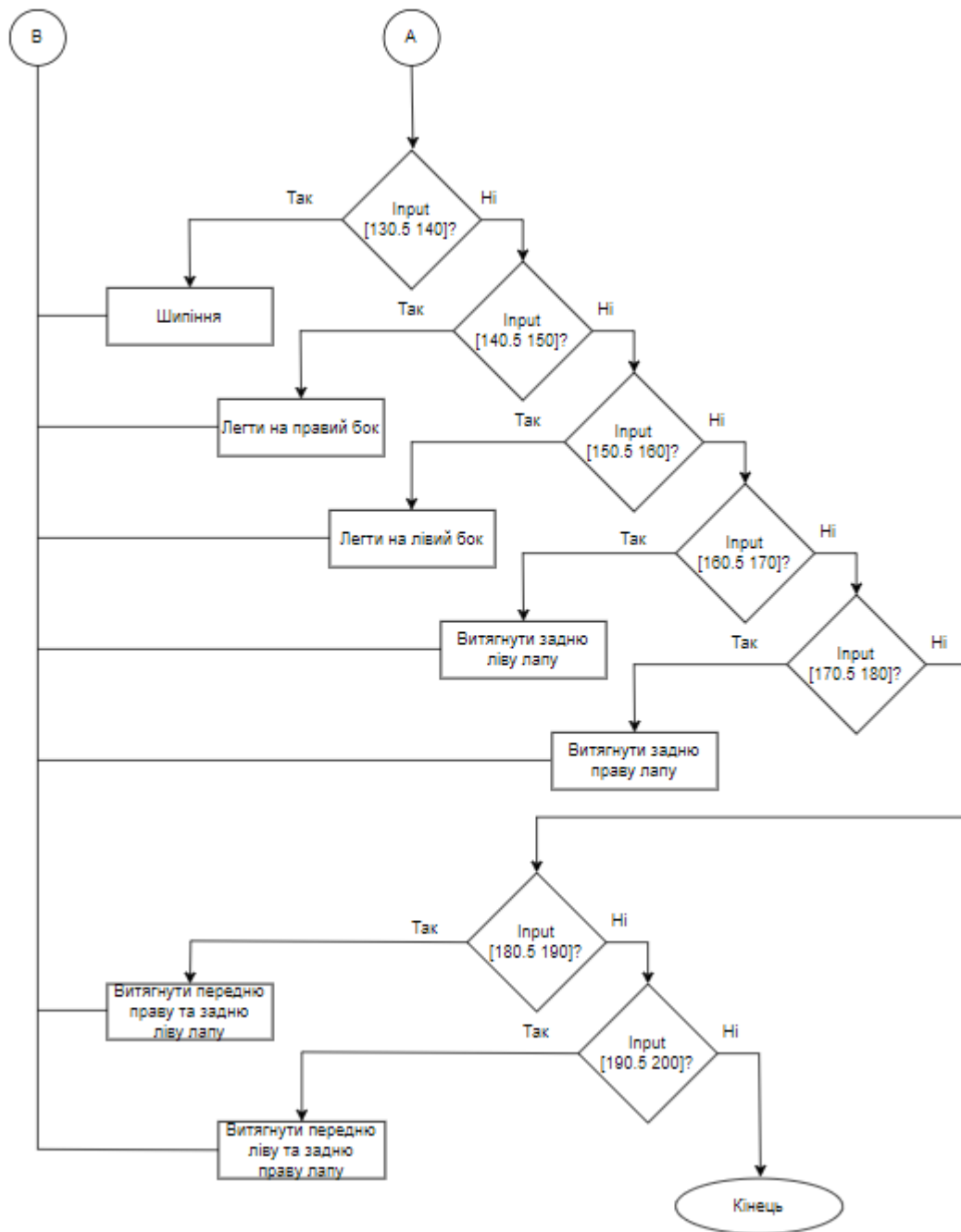


Рисунок 3.1, аркуш 3

### 3.2 Розробка програмного забезпечення

Розроблений код програми:

```
[System]
Name='diplom_m'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=11
NumRules=20
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='Поведінка'
Range=[0 200]
NumMFs=20
MF1='Початкове положення': 'trapmf', [0.00610328638497873
0.046948356807512 9.90610328638498 10.006103286385]
MF2='Команда "Лежати"': 'trapmf', [10.2 10.6572769953052 19.5774647887324
20.1]
MF3='Команда "Сидіти"': 'trapmf', [20.59 20.7466340269278
30.0489596083231 30.1]
MF4='Команда "Виляння хвостом"': 'trimf', [30.2 34.56 40.1]
MF5='Команда "Мурликання"': 'trapmf', [40.2 40.2080783353733
49.6328029375765 50.1]
MF6='Команда "Дати лапу"': 'trapmf', [50.2 50.2447980416157
60.0367197062424 60.1]
MF7='Команда "Переміщення"': 'trimf', [60.2 65 70.4406364749082]
MF8='Команда "Охота"': 'trapmf', [80.2 80.3549571603427 89.7796817625459
90.1]
MF9='Команда "Розтягнутися"': 'trapmf', [90.2 90.7588739290086
99.5716034271726 100.1]
MF10='Команда "Переміщення 2"': 'trimf', [70.1958384332925 75 80]
MF11='Команда "Дати лапу 2я 2"': 'trimf', [100.6 105 110]
MF12='Команда "Витягнути задні лапи"': 'trimf', [110.5 115 120]
MF13='Команда "Витягнути передні лапи"': 'trimf', [120.5 125 130]
MF14='Команда "Шипіння"': 'trimf', [130.5 135 140]
```

MF15='Команда "Легти на правий бок":'trimf',[140.5 145 150]  
 MF16='Команда "Легти на лівий бок":'trimf',[150.5 155 160]  
 MF17='Команда "Витягнути задню ліву лапу":'trimf',[160.5 165 170]  
 MF18='Команда "Витягнути задню праву лапу":'trimf',[170.5 175 180]  
 MF19='Команда "Витягнути передню праву та задню ліву лапу":'trimf',[180.5 185 190]  
 MF20='Команда "Витягнути задню передню ліву та задню праву лапу":'trimf',[190.5 195 200]

[Input2]

Name='Відстань'  
 Range=[0 20]  
 NumMFs=2  
 MF1='Відстань М':'trimf',[9.9 15 20]  
 MF2='Відстань Ш':'trimf',[0 5 10]

[Output1]

Name='ПЛП'  
 Range=[0 180]  
 NumMFs=5  
 MF1='Поворот 1':'trimf',[0 15 30]  
 MF2='Поворот 2':'trapmf',[25 44 46 75]  
 MF3='Поворот 3':'trimf',[70 90 110]  
 MF4='Поворот 4':'trimf',[100 120 140]  
 MF5='Поворот 5':'trimf',[140 160 180]

[Output2]

Name='ПЛГ'  
 Range=[0 180]  
 NumMFs=5  
 MF1='Поворот 1':'trimf',[0 15 30]  
 MF2='Поворот 2':'trapmf',[25 44 46 75]  
 MF3='Поворот 3':'trimf',[70 90 110]  
 MF4='Поворот 4':'trimf',[100 120 140]  
 MF5='Поворот 5':'trimf',[140 160 180]

[Output3]

Name='ППП'  
 Range=[0 180]  
 NumMFs=5  
 MF1='Поворот 1':'trimf',[0 15 30]  
 MF2='Поворот 2':'trapmf',[25 44 46 75]  
 MF3='Поворот 3':'trimf',[70 90 110]  
 MF4='Поворот 4':'trimf',[100 120 140]

MF5='Поворот 5':'trimf',[140 160 180]

[Output4]

Name='ППП'

Range=[0 180]

NumMFs=5

MF1='Поворот 1':'trimf',[0 15 30]

MF2='Поворот 2':'trapmf',[25 44 46 75]

MF3='Поворот 3':'trimf',[70 90 110]

MF4='Поворот 4':'trimf',[100 120 140]

MF5='Поворот 5':'trimf',[140 160 180]

[Output5]

Name='ЗПП'

Range=[0 180]

NumMFs=5

MF1='Поворот 1':'trimf',[0 15 30]

MF2='Поворот 2':'trapmf',[25 44 46 75]

MF3='Поворот 3':'trimf',[70 90 110]

MF4='Поворот 4':'trimf',[100 120 140]

MF5='Поворот 5':'trimf',[140 160 180]

[Output6]

Name='ЗПГ'

Range=[0 180]

NumMFs=5

MF1='Поворот 1':'trimf',[0 15 30]

MF2='Поворот 2':'trapmf',[25 44 46 75]

MF3='Поворот 3':'trimf',[70 90 110]

MF4='Поворот 4':'trimf',[100 120 140]

MF5='Поворот 5':'trimf',[140 160 180]

[Output7]

Name='ЗПП'

Range=[0 180]

NumMFs=5

MF1='Поворот 1':'trimf',[0 15 30]

MF2='Поворот 2':'trapmf',[25 44 46 75]

MF3='Поворот 3':'trimf',[70 90 110]

MF4='Поворот 4':'trimf',[100 120 140]

MF5='Поворот 5':'trimf',[140 160 180]

[Output8]

Name='ЗПГ'

```

Range=[0 180]
NumMFs=5
MF1='Поворот 1':'trimf',[0 15 30]
MF2='Поворот 2':'trapmf',[25 44 46 75]
MF3='Поворот 3':'trimf',[70 90 110]
MF4='Поворот 4':'trimf',[100 120 140]
MF5='Поворот 5':'trimf',[140 160 180]

```

```

[Output9]
Name='Хвіст'
Range=[0 180]
NumMFs=2
MF1='Виляти хвостом 1':'trapmf',[40 45 130 135]
MF2='Виляти хвостом 2':'trapmf',[0 60 120 180]

```

```

[Output10]
Name='Шия'
Range=[0 1]
NumMFs=1
MF1='Поворот Ш':'trimf',[0 0.5 1]

```

```

[Output11]
Name='Елемент'
Range=[0 100]
NumMFs=2
MF1='Мурликання':'trimf',[90 100 110]
MF2='Шипіння':'trimf',[0 50 100]

```

```

[Rules]
1 0,3 3 3 3 3 3 3 1 0 0 (1) : 1
2 0,5 5 5 5 5 5 5 4 1 0 0 (1) : 1
3 0,3 3 3 3 3 3 3 0 0 0 (1) : 1
4 0,3 3 3 3 3 3 3 1 0 0 (1) : 1
4 -1,1 1 1 1 3 3 3 3 1 0 1 (1) : 1
8 0,5 5 5 5 5 5 5 2 1 0 (1) : 1
9 0,1 1 1 1 5 5 5 5 0 1 0 (1) : 1
6 1,3 3 2 2 5 5 5 5 0 0 1 (1) : 1
7 1,1 1 1 1 1 1 1 1 1 1 (1) : 1
10 1,1 1 1 1 1 1 1 1 1 1 (1) : 1
11 0,2 2 3 3 5 5 5 5 0 0 0 (1) : 1
12 0,3 3 3 3 5 5 5 5 0 0 0 (1) : 1
13 0,5 5 5 5 3 3 3 3 0 0 0 (1) : 1
14 2,3 1 3 1 3 3 3 3 0 0 2 (1) : 1
15 0,1 1 1 1 3 3 3 3 0 0 0 (1) : 1

```

160,33331111000(1):1  
170,33334433000(1):1  
180,33333344000(1):1  
190,33444433000(1):1  
200,44333344000(1):1

Цей код був створений для виконання вхідних правил в FuzzyLogicDesigner.

### 3.3 Висновку до розділу

У цьому розділі було здійснено детальний огляд та вибір методів моделювання, які дозволяють відтворити реалістичне переміщення робота в умовах невизначеності середовища.

Програмне забезпечення, розроблене в рамках даного дослідження, виявилось ефективним і дозволяє враховувати різноманітні умови навколишнього середовища, що може виникнути у реальних умовах експлуатації. Високий рівень реалізації алгоритмів контролю та планування руху робота дозволяє досягти точного та стабільного переміщення навіть у складних умовах.

Також у цьому розділі було розроблено блок-схему алгоритму роботи програми, яка чітко відображає послідовність операцій, взаємодію компонентів та прийняті рішення в процесі моделювання робота в невизначеному середовищі. Вона є ключовим інструментом для розуміння та вдосконалення роботи програми, а також може використовуватися для подальшої розробки та вдосконалення алгоритму у майбутніх дослідженнях та розробці робота.

## 4 МОДЕЛЮВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

### 4.1 Моделювання та експериментальні дослідження

Моделювання представляє собою вибір вхідних параметрів, які відповідають командам робота. Вихідні параметри це кути обороту серводвигунів робота.

Моделювання команди «Початкове положення» представлено на рисунку 4.1.



Рисунок 4.1 – Моделювання команди «Початкове положення»

Як видно з рисунку, при вхідному параметрі 5, який знаходиться в межах від 0 до 10 (див. рисунок 2.3). Робот виконує команду «Початкове положення», вихідними параметрами стали значення кута повороту серводвигунів.

Наступною є команда «Лежати», моделювання якої зображено на рисунку 4.2.

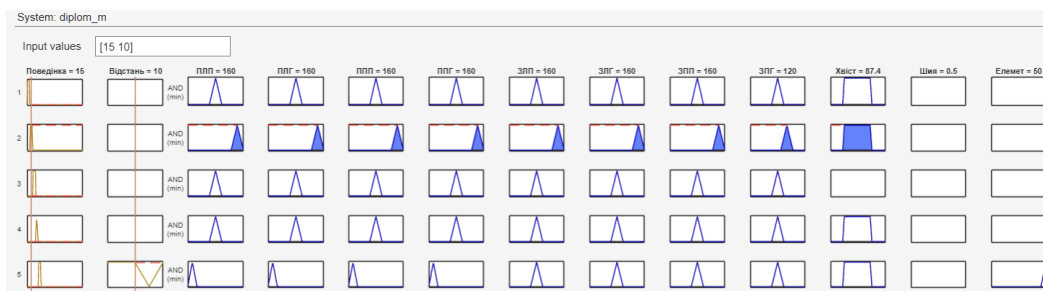


Рисунок 4.2 – Моделювання команди «Лежати»

Як видно з рисунку, при вхідному параметрі 15, який знаходиться в межах від 10 до 20 (див. рисунок 2.3). Робот виконує команду «Лежати», вихідними параметрами стали значення кута повороту серводвигунів.

Наступною командою є «Сидіти», моделювання якої представлено на рисунку 4.3.

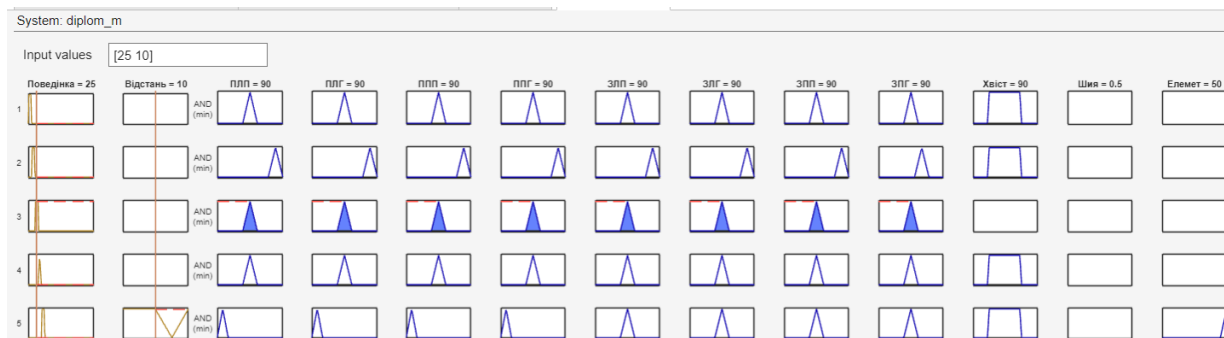


Рисунок 4.3 – Моделювання команди «Сидіти»

Як видно з рисунку, при вхідному параметрі 25, який знаходиться в межах від 20 до 30 (див. рисунок 2.3). Робот виконує команду «Сидіти», вихідними параметрами стали значення кута повороту серводвигунів.

Моделювання команди «Виляння хвостом» зображено на рисунку 4.4. Основною відмінністю цієї команди є те, що робот приймає початкове положення, а серводвигун хвоста робить декілька циклічних поворотів.

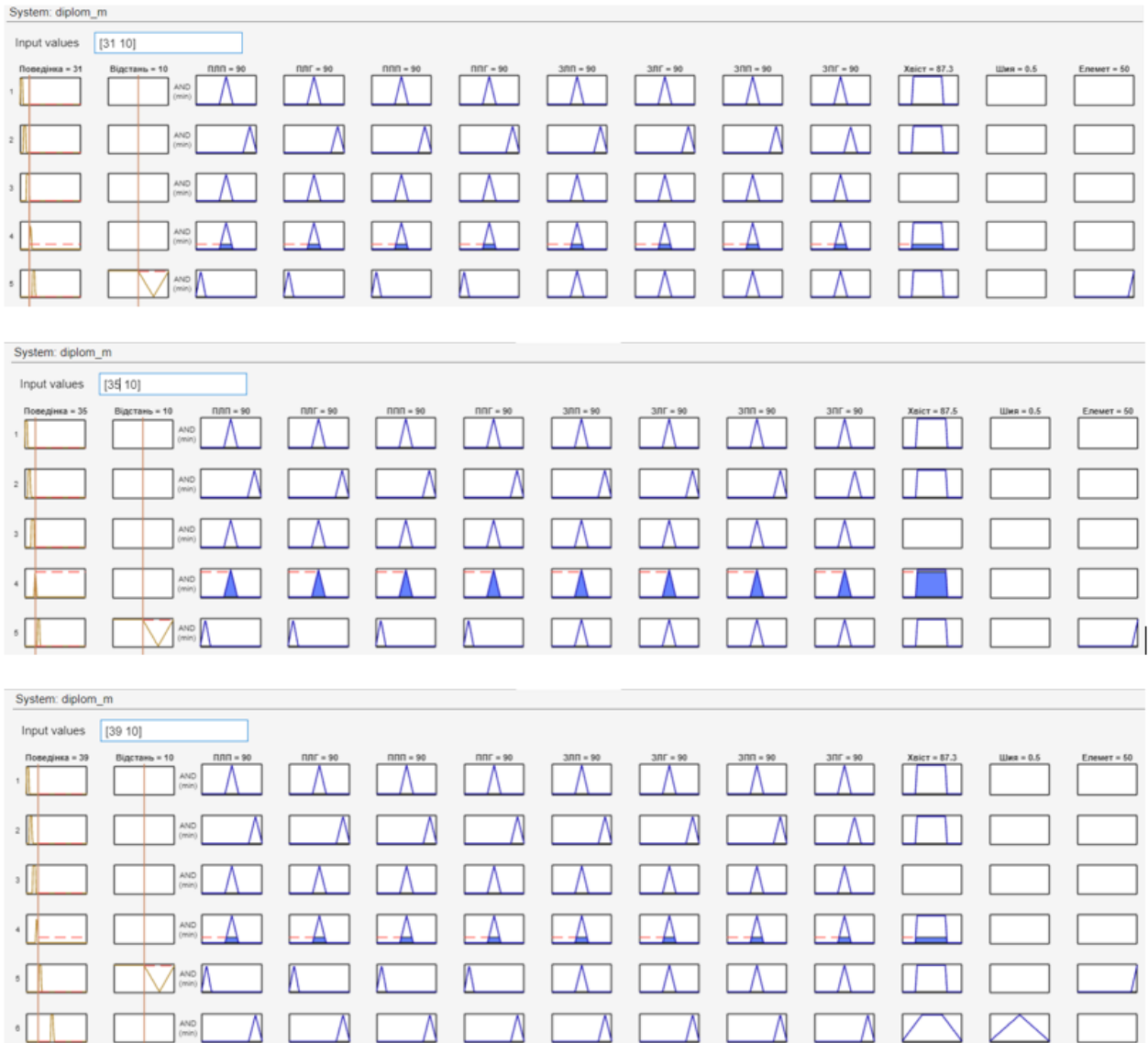


Рисунок 4.4 – Моделювання команди «Виляння хвостом»

Як видно з рисунку, коли на вхід поступає параметр у проміжку 30-40, робот приймає початкове положення, а хвіст виконує повороти.

Наступною командою є «Мурликання», моделювання якої зображено на рисунку 4.5.



Рисунок 4.5 – Моделювання команди «Мурликання»

Як видно з рисунку, при вхідному параметрі 45, який знаходиться в межах від 40 до 50 (див. рисунок 2.3), та вхідному параметрі «Відстань» від 0 до 10. Робот виконує команду «Мурликання», вихідними параметрами стали значення кута повороту серводвигунів та робота п'єзоелемента. Але, якщо вхідний параметр «Відстань» буде в межах від 10.1 до 20, то команда виконуватися не буде.

Моделювання команди «Дати лапу» зображено на рисунку 4.6.



Рисунок 4.6 – Моделювання команди «Дати лапу»

Як видно з рисунку, при вхідному параметрі 55, який знаходиться в межах 50-60, та вхідному параметрі «Відстань» в межах 10-20 робот виконує команду. Але якщо вхідний параметр «Відстань» не буде в межах 10-20 то робот не виконає команду. При виконанні цієї команди робот дає праву передню лапу.

Наступною Командою є «Переміщення», моделювання якої зображено на рисунку 4.7.

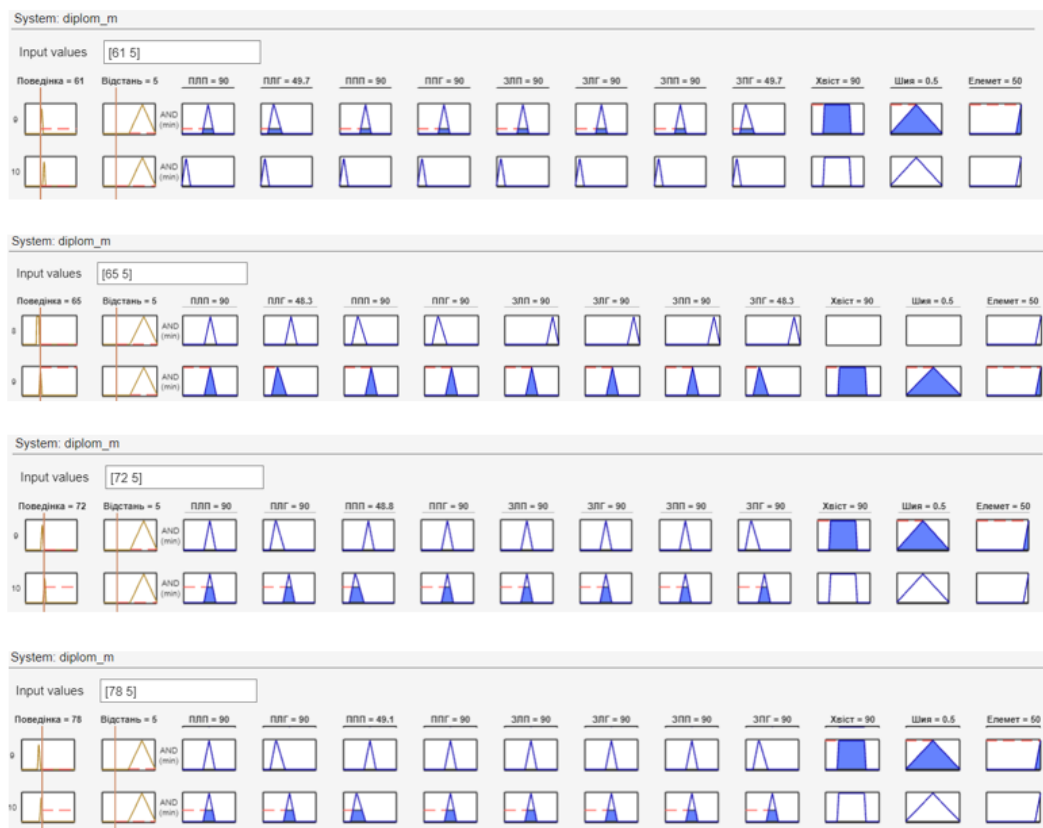


Рисунок 4.7 – Моделювання команди «Переміщення»

Як видно з рисунку. Переміщення робота відбувається якщо вхідні параметри в діапазоні 60-80. Головною відмінністю цієї команди є те, що правило розділено на дві частини. Перша частина виконується в діапазоні від 60 до 70, а інша від 70 до 80 при цих параметрах відбуваються оберти серводвигунів однієї пари кінцівок, а інша пара приймає початкове положення і навпаки. Це циклічне переміщення.

Наступною командою є «Охота». Моделювання команди зображено на рисунку 4.8.

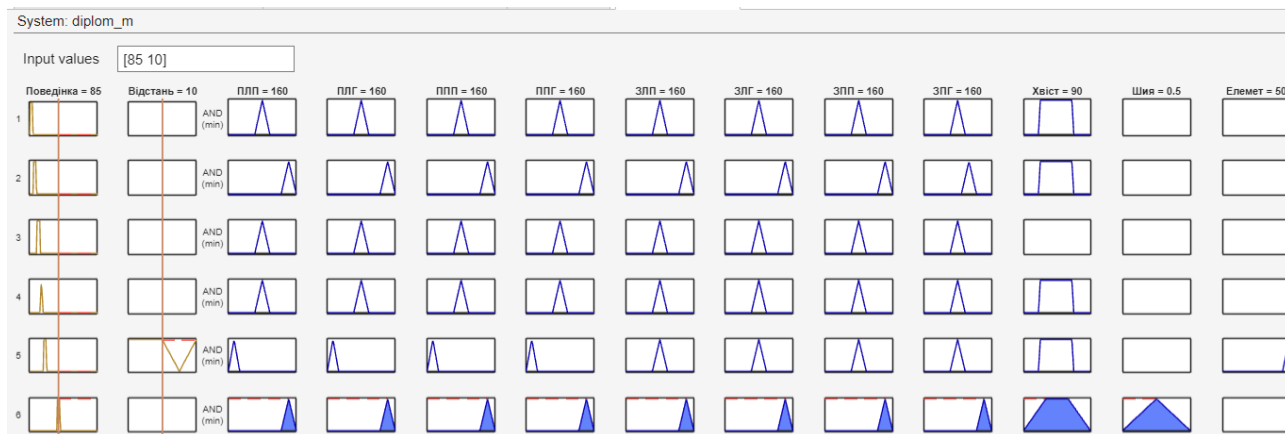


Рисунок 4.8 – Моделювання команди «Охота»

Як видно з рисунку, при вхідному параметрі 85, який знаходиться в межах 80.1-90 робот виконує команду «Охота». Під час цієї команди робот приймає положення «лежати», але рухає серводвигунами хвоста та голови.

Наступною командою є «Розтягнутися», моделювання команди зображено на рисунку 4.9.

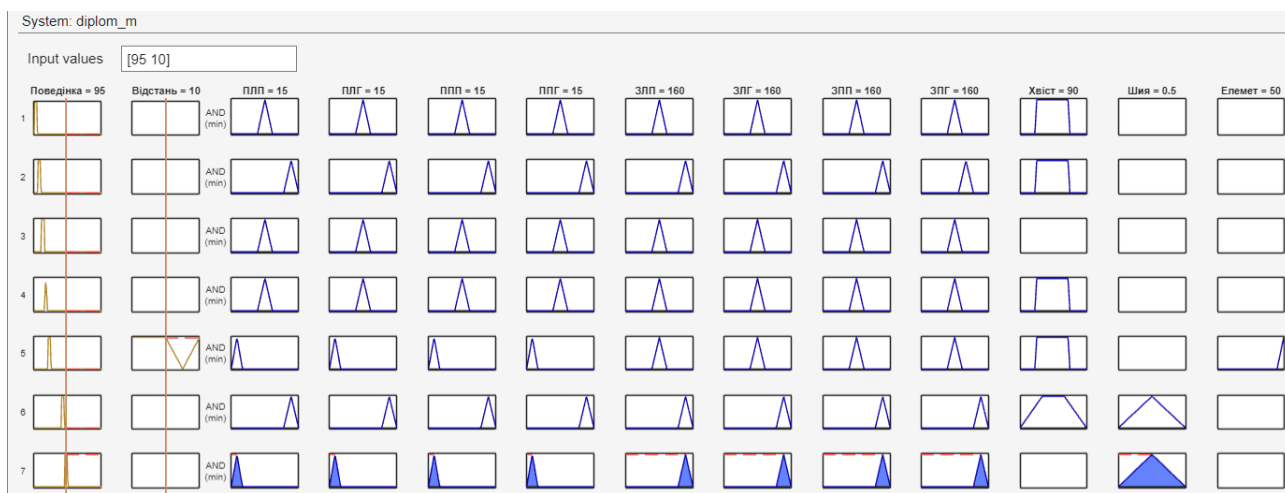


Рисунок 4.9 – Моделювання команди «Розтягнутися»

Як видно з рисунку, при вхідному параметрі 95, який знаходиться в межах 90,1-100 робот виконує команду «Розтягнути». Під час виконання цієї команди робот витягує передні лапи, а задні приймають початкове положення.

Наступною командою є «Дати лапу 2», моделювання якої зображено на рисунку 4.10.



Рисунок 4.10 – Моделювання команди «Дати лапу 2»

Як видно з рисунку при вхідному параметрі 105, який лежить в межах від 100,1-110 робот виконує команду «Дати лапу 2». Ця команда схожа з «Дати лапу», але відмінність полягає в тому, яку лапу протягне робот.

Наступною командою є «Витягнути задні лапи». Моделювання команди зображено на рисунку 4.11.

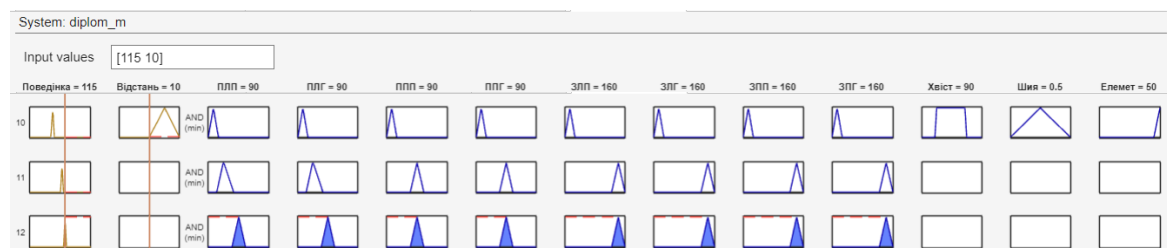


Рисунок 4.11 – Моделювання команди «Витягнути задні лапи»

Як видно з рисунку при вхідному параметрі 115, який лежить в межах від 110,5-120 робот виконує команду «Витягнути задні лапи». При цій команді робот встає на передні лапи і витягує задні.

Наступною командою є «Витягнути передні лапи». Моделювання команди зображено на рисунку 4.12.



Рисунок 4.12 – Моделювання команди «Витягнути передні лапи»

Як видно з рисунку при вхідному параметрі 125, який лежить в межах від 120,5-130 робот виконує команду «Витягнути передні лапи». При цій команді робот встає на задні лапи і витягує передні.

Наступною командою є «Шипіння», моделювання якої зображено на рисунку 4.13.



Рисунок 4.13 – Моделювання команди «Шипіння»

Як видно з рисунку при вхідному параметрі «Поведінка» 135, який знаходиться в межах 130,5-140 та «Відстань» 5, який знаходиться в межах 0-10 робот буде виконувати команду шипіння. Якщо при вхідному параметрі «Відстань» вона не буде дорівнювати проміжку 0-10, то команда виконуватися не буде. Ця ситуація зображена на рисунку 4.14.



Рисунок 4.14 – Моделювання команди «Шипіння» при іншому вхідному параметрі

Наступною командою є «Лягти на правий бік» моделювання команди зображено на рисунку 4.15.



Рисунок 4.15 – Моделювання команди «Лягти на правий бік»

Як видно з рисунку при вхідному параметрі 145, який знаходиться в межах 140,5-150. Під час цієї команди робот згинає праві кінцівки, та розгинає ліві. Слід зазначити, що для виконання наступних команд робота слід повернути в початкове положення.

Наступною командою є «Лягти на лівий бік» моделювання команди зображено на рисунку 4.16.



Рисунок 4.16 – Моделювання команди «Лягти на лівий бік»

Як видно з рисунку при вхідному параметрі 155, який знаходиться в межах 150,5-160. Під час цієї команди робот згинає ліві кінцівки, та розгинає праві. Ця команда схожа з попередньою.

Наступною командою є «Витягнути задню ліву лапу», моделювання якої зображено на рисунку 4.17.



Рисунок 4.17 – Моделювання команди «Витягнути задню ліву лапу»

Як видно з рисунку при вхідному параметрі 165, який знаходиться в межах 160,5-170. Під час цієї команди робот витягує задню ліву лапу з початкового положення.

Наступною командою є «Витягнути задню праву лапу», моделювання якої зображено на рисунку 4.18.

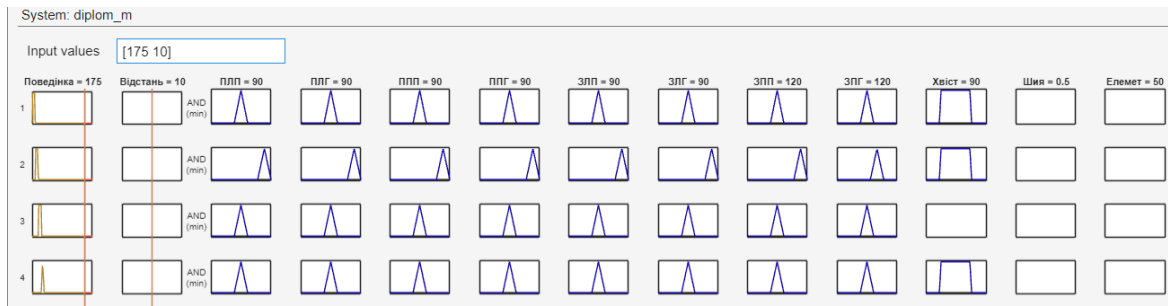


Рисунок 4.18 – Моделювання команди «Витягнути задню праву лапу»

Як видно з рисунку при вхідному параметрі 175, який знаходиться в межах 170,5-180. Під час цієї команди робот витягує задню праву лапу з початкового положення.

Наступною командою є «Витягнути передню праву та задню ліву лапу», моделювання якої зображено на рисунку 4.19.

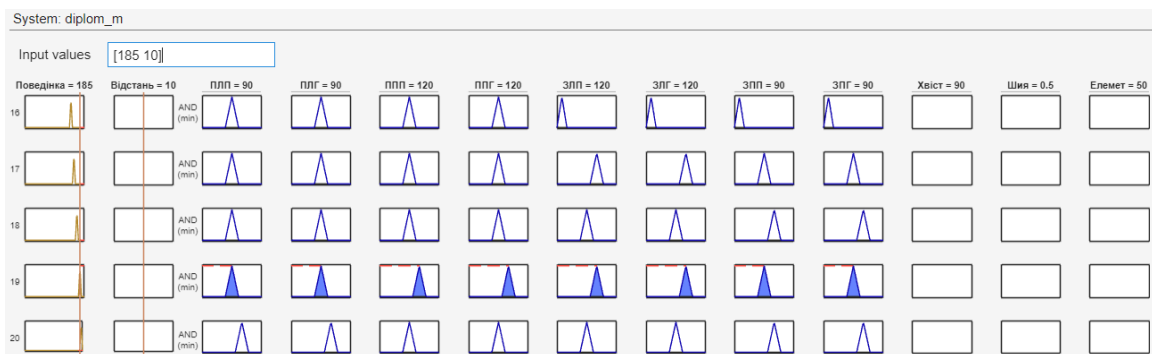


Рисунок 4.19 – Моделювання команди «Витягнути передню праву та задню ліву лапу»

Як видно з рисунку при вхідному параметрі 185, який знаходиться в межах 180,5-190. Під час цієї команди робот витягує задню ліву лапу та передню праву з початкового положення.

Остання команда це «Витягнути задню передню ліву та задню праву лапу», моделювання якої зображено на рисунку 4.20.

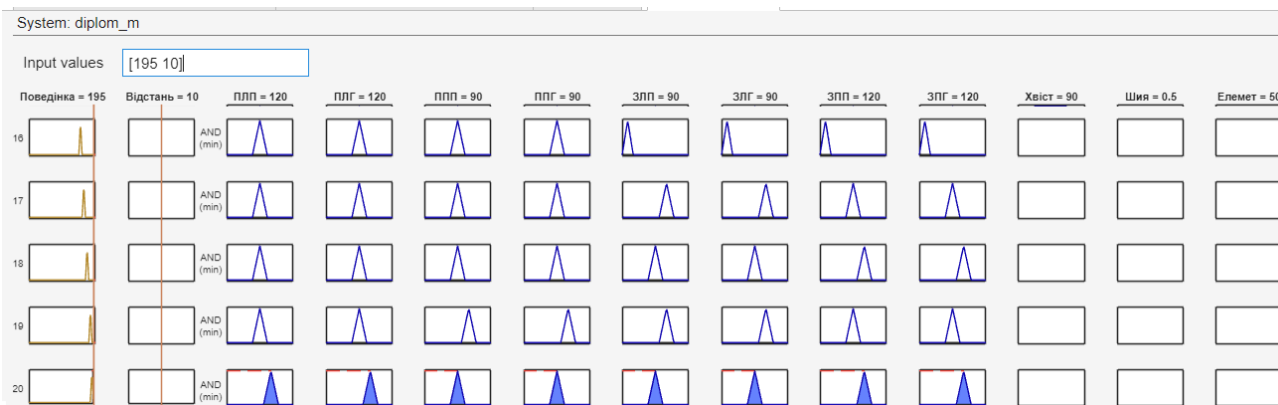


Рисунок 4.20 – Моделювання команди «Витягнути задню передню ліву та задню праву лапу»

Як видно з рисунку при вхідному параметрі 195, який знаходиться в межах 190,5-200. Під час цієї команди робот витягує задню праву лапу та передню ліву з початкового положення.

Залежність кінцівок від поведінки та відстані зображено на рисунку 4.21-4.31.

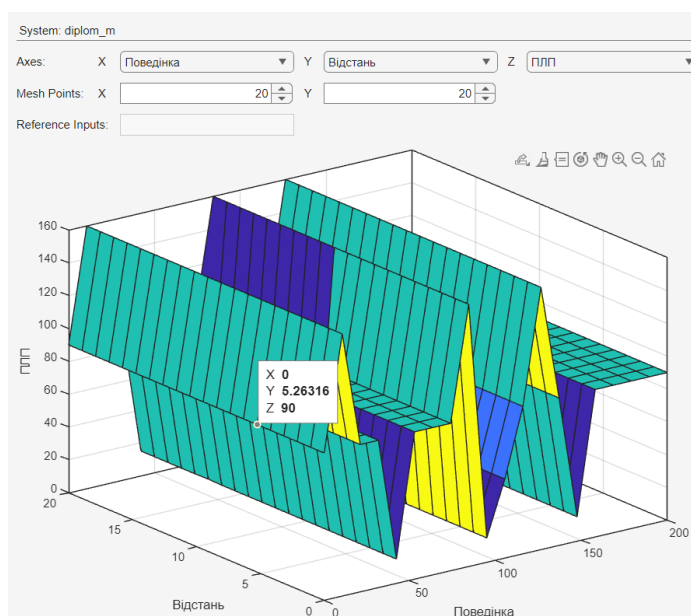


Рисунок 4.21 – Залежність ПЛП

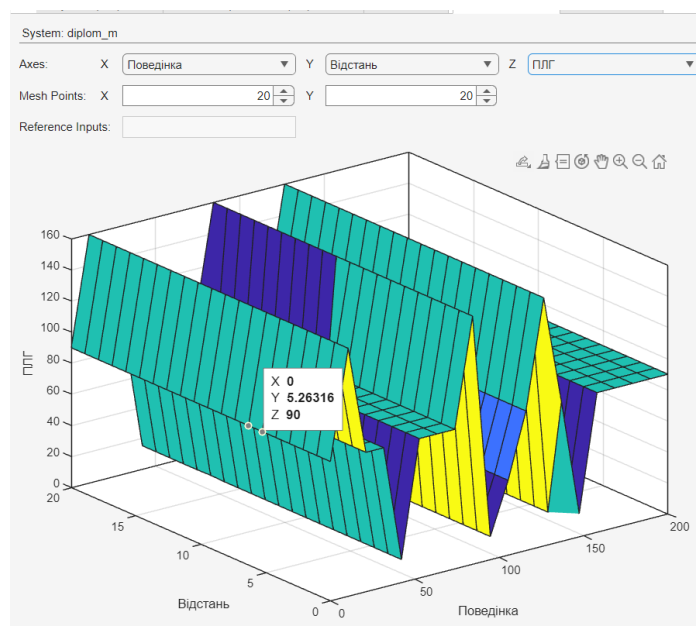


Рисунок 4.22 – Залежність PLLG

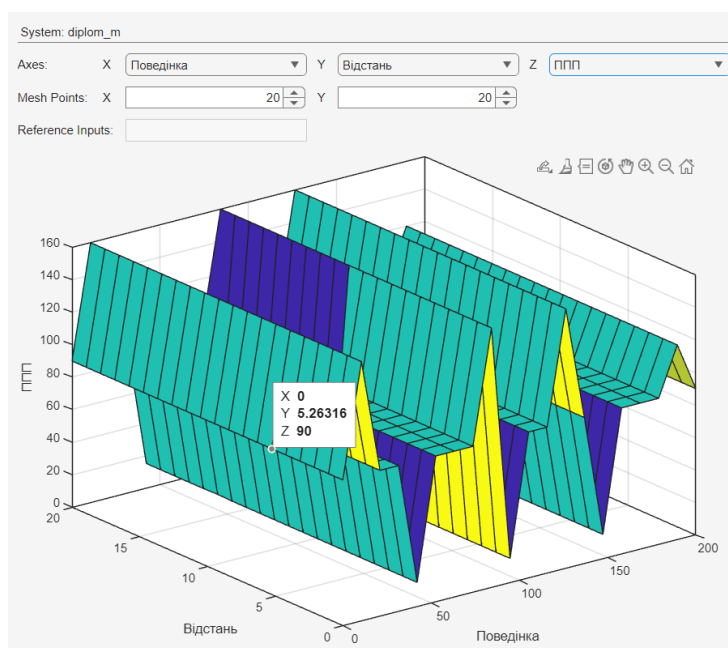


Рисунок 4.23 – Залежність PPLP

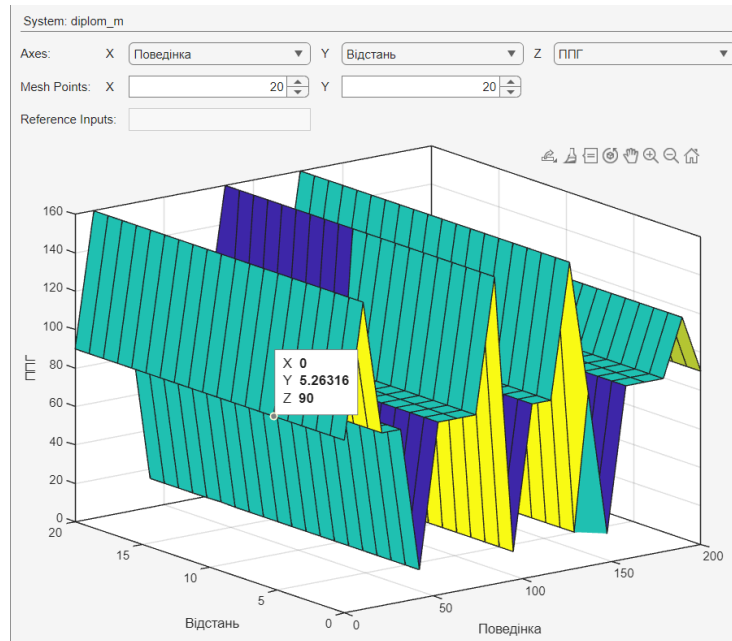


Рисунок 4.24 – Залежність ППГ

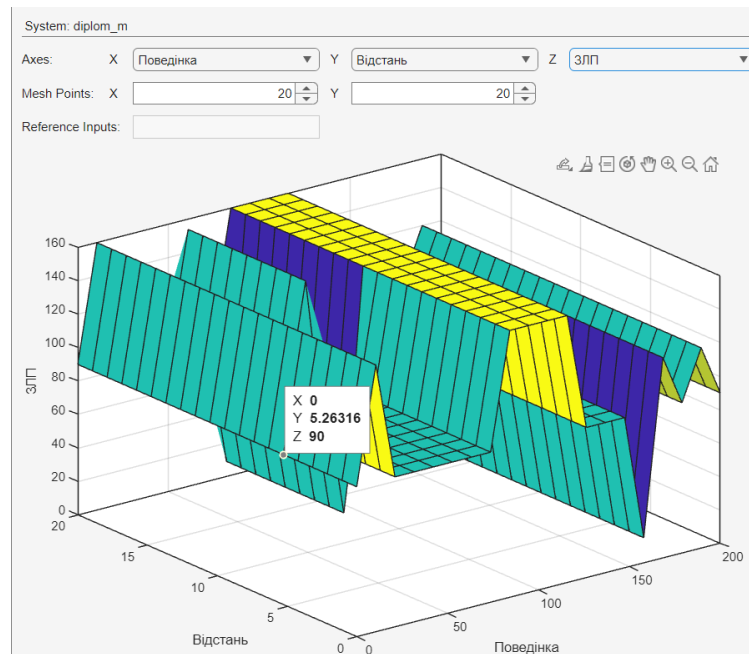


Рисунок 4.25 – Залежність ЗЛП

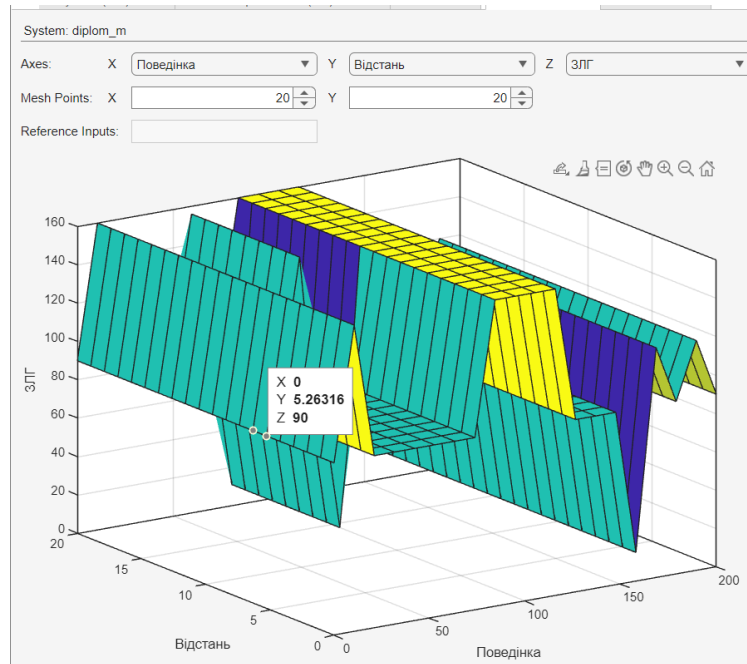


Рисунок 4.26 – Залежність ЗЛГ

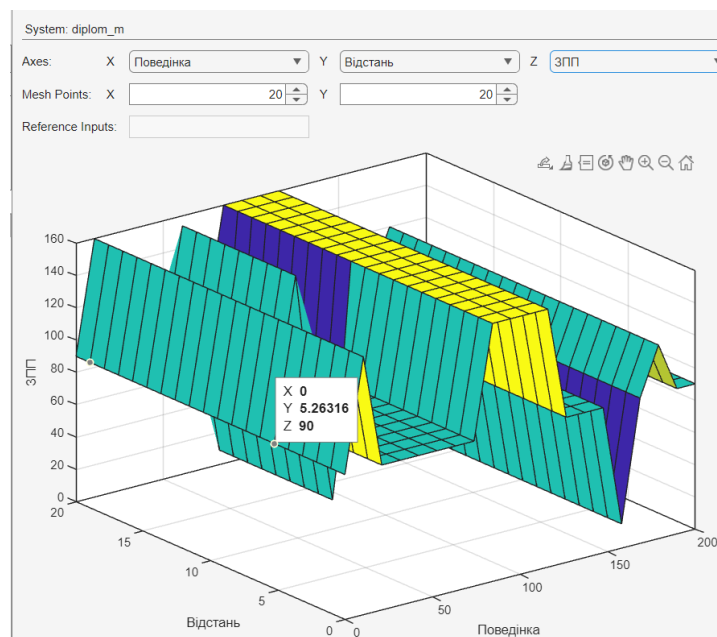


Рисунок 4.27 – Залежність ЗПП

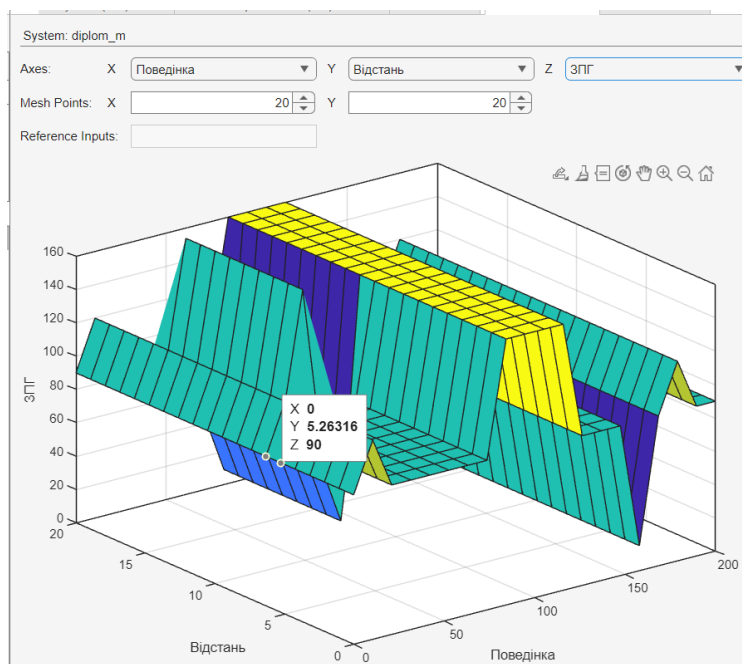


Рисунок 4.28 – Залежність ЗПГ

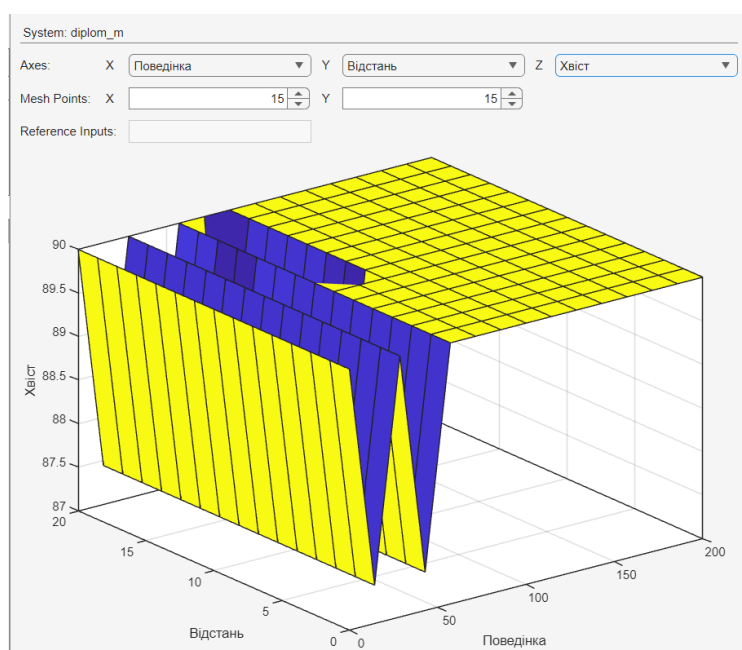


Рисунок 4.29 – Залежність Хвіст

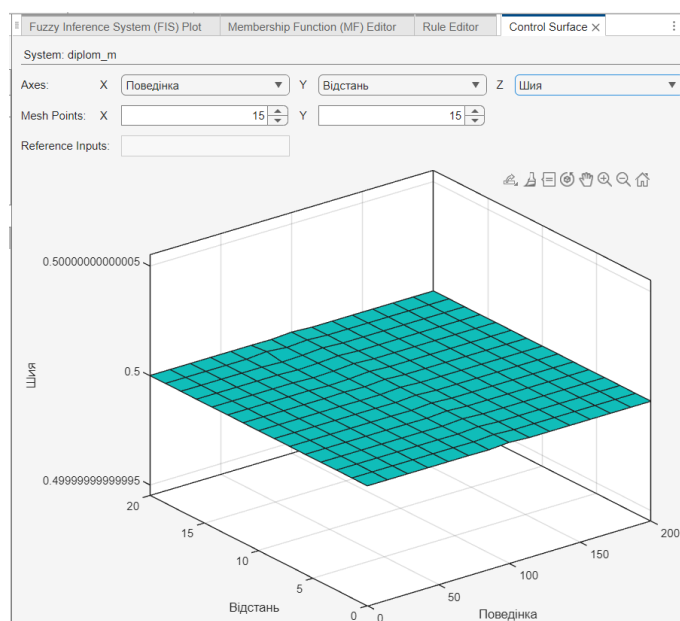


Рисунок 4.30 – Залежність Шия

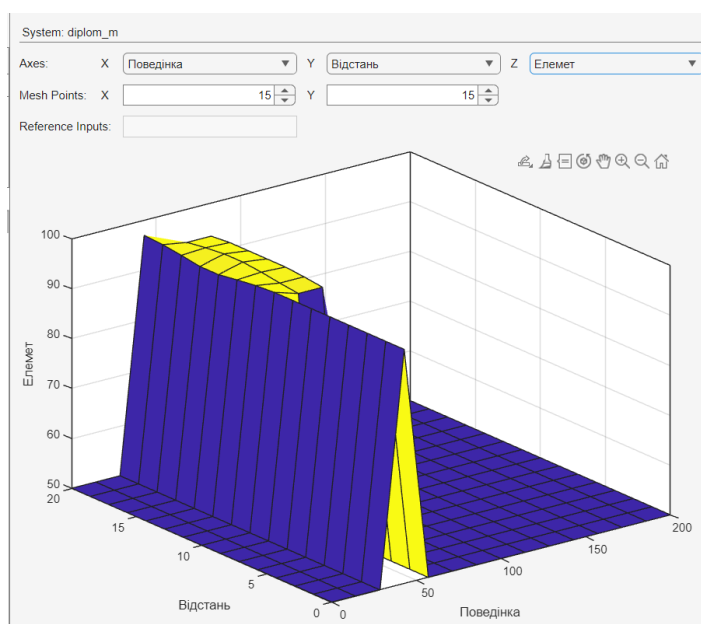


Рисунок 4.31 – Залежність Елемент

Як видно з цих рисунків, вони є графічним представленням відношення між вхідними та вихідними змінними у нечіткій системі. Це зображення є графіком, де одна вісь представляє вхідні змінні, інша вісь - вихідні змінні, а висота вказує на величину вихідного сигналу відповідно до вхідних умов.

## 4.2 Висновки до розділу

Цей розділ дипломної роботи було присвячено моделюванню роботи зооморфного робота «Робокіт» у невизначеному просторі за допомогою FuzzyLogicDesigner.

FuzzyLogicDesigner використовується для розробки інтерфейсу правил (Rule interface), який надає інформацію про моделювання нечітких систем через визначення та налаштування правил.

Rule interface дозволяє точно визначати форму та характеристики нечітких множин, що використовуються в правилах. Це важливо для визначення ступеня належності конкретного значення до кожної нечіткої множини, що формує базу для прийняття рішень.

Rule interface дозволяє легко конфігурувати правила, визначаючи логічні зв'язки між вхідними та вихідними змінними. Встановлення важливості кожного правила та налагодження ваг для визначення впливу кожного правила на вихідні результати забезпечує гнучкість та адаптивність системи.

Інтерфейс дозволяє візуалізувати структуру правил у зручній формі, що полегшує розуміння взаємодії між різними вхідними та вихідними змінними. Візуальний аналіз структури правил сприяє ефективній настройці та оптимізації нечіткої системи.

Rule interface дозволяє виконувати симуляції для тестування та аналізу роботи нечіткої системи в різних умовах. Здатність відстежувати зміни в вихідних значеннях при зміні вхідних параметрів допомагає вдосконалювати та оптимізувати правила для досягнення бажаних результатів.

Так, Rule interface FuzzyLogicDesigner у Matlab є потужним інструментом для детального моделювання та аналізу нечітких систем. Визначення та оптимізація правил через цей інтерфейс забезпечують глибоке розуміння та контроль над поведінкою системи в умовах невизначеності.

## ОХОРОНА ПРАЦІ

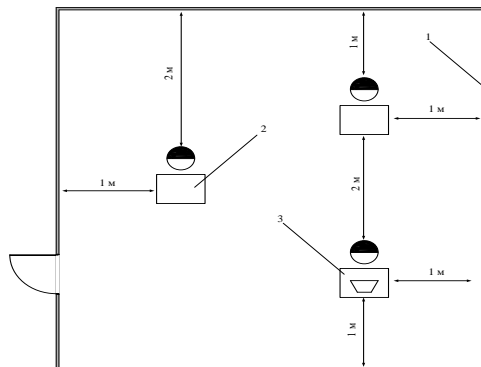
### 5.1 Охорона праці

Згідно до ДСН (Державних санітарних норм) 3.3.6-042-99 робота в лабораторії з персональним комп'ютером (ПК) по категорії робіт відноситься до легких (категорія Іа – легкі фізичні роботи з енерговитратами до 139 Вт або 120 ккал/г). Робота ведеться сидячи, не вимагає систематичного фізичного напруження і переміщення важких предметів.

У робочій зоні приміщення згідно з ДСН 3.3.6.042-99 повинні бути встановлені оптимальні поєднання параметрів мікроклімату. Для категорії робіт І а в приміщенні повинні дотримуватися такі норми мікроклімату:

- температура повітря не повинна перевищувати 22-24 °С в літній період і 23-25 °С в зимовий період;
- відносна вологість повинна бути в межах 40-60%;
- швидкість руху повітря не повинна перевищувати 0,1 м/с.

Приміщення, в яких здійснюється експлуатація персональних комп'ютерів, обов'язково повинні мати одночасно і природне, і штучне освітлення, яке відповідає вимогам чинної нормативної документації. Так, на рисунку 5.1 зображено приклад розташування робочих місць у приміщенні.



1 – вікно; 2 – робоче місце; 3 – комп'ютеризоване робоче місце з ПК та ЖК – монітором

Рисунок 5.1 – План виробничого приміщення

У таблиці 5.1 наведено виміряні показники необхідних параметрів для приміщень.

Таблиця 5.1 – Виміряні параметри

Параметр	Фактичне значення	Значення по СН-245-71 чи державному стандарту	Висновок
Шум, дБ	50	46	В межах допустимих значень
Освітленість (загальна), Лк	350	200-400	В межах допустимих значень
Значення К.П.О., %	1,5	1,6	В межах допустимих значень
Загазованість (концентрація і вид газу), мг/м <sup>3</sup>	Азот 4,8 Аміак 12 Озон 0,05 Свинець 0,01 Хлор 0,7	Азот 5 Аміак 20 Озон 0,1 Свинець 0,01 Хлор 1	В межах допустимих значень
Температура повітря взимку, влітку, °С	21-23 22-24	21-24 22-25	В межах допустимих значень
Відносна вологість, %	45	40-60	В межах допустимих значень
Швидкість руху повітря, м/с	0,15	0,1-0,2	В межах допустимих значень

Як видно з рисунку 5.1, виробниче приміщення відповідає вимогам ДСН 3.36.037-99, а саме приміщення має як природне, так і штучне освітлення, яке не перевищує норму. Розташування монітору запобігає потраплянню на нього відблисків світла. Також, як видно з таблиці 5.1 у приміщенні високий рівень шуму, для його зменшення слід замінити систему охолодження на комп'ютері. Для запобігання підвищенню концентрації шкідливих елементів слід робити вологе прибирання.

## ВИСНОВКИ

В ході виконання кваліфікаційної роботи були проаналізовані сучасні зооморфні роботи та їх конструкції. Проаналізовано та обґрунтовано вибір програмного засобу, який використовується для моделювання зооморфного робота «Робокіт». Була побудована схема підключення елементів.

В рамках кваліфікаційної роботи було проаналізовано програмний засіб MatLab та додаток fuzzyLogicDesigner. Також було проаналізовано правила Мамдані. На їх основі було розроблені правила для поведінки робота.

Також у кваліфікаційній роботі було представлено виконання правил роботом у за допомогою RuleInterface у додатку fuzzyLogicDesigner.

Вихідними параметрами під час виконання правила є кут оберту валу серводвигуна. Який дозволяє роботу рухатись або робити інші дії.

Головним плюсом цієї модуляції є її простота, та гнучкість. Наприклад, можна на вхід подати відповідні параметри, які необхідні у виконанні поставленої задачі, та отримати на виході ті параметри які необхідні.

При подальшій розробці та вдосконаленні зооморфного робота «Робокіт», ці модуляції можуть використовуватися для вдосконалення програмного забезпечення проекту. Слід зазначити, що код представлений у кваліфікаційній роботі не є кодом програмного забезпечення робота, але цей код містить у собі параметри, які можуть використовуватися у подальшому вдосконаленні проекту зооморфного робота «Робокіт».

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Євсєєв В.В. 3D Проектування і моделювання кінцівок зооморфного робота / В.В. Євсєєв В.В., Н.В. Замірець, В.І. Роменський, В.Е. Салієва. // Технологія приладобудування. – 2’2017. – с. 10–13. – 12.10.2023р.
2. Методичні вказівки з підготовки та захисту кваліфікаційної роботи [Електронний ресурс] – режим доступу: [https://tapr.nure.ua/wp-content/uploads/2022/04/mv\\_\\_mag\\_151.pdf](https://tapr.nure.ua/wp-content/uploads/2022/04/mv__mag_151.pdf).
3. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.
4. Долгуля А.В. Дослідження переміщення чотирилапого зооморфного робота «Робокіт» у невизначеному просторі [Електронний ресурс] – режим доступу: <https://openarchive.nure.ua/server/api/core/bitstreams/a58be3b9-8dac-425d-8cfd-ea3762bf9dda/content>.
5. Z. Yuhai, F. Huashan [Текст]/Analysis and research of quadruped robot’s legs: A comprehensive review – May 2019р.
6. These robot wings use artificial muscle to flap like an insect [Електронний ресурс] – режим доступу: <https://www.popsci.com/technology/disembodied-robot-flies-laza-system/>.
7. Agile and versatile climbing on ferromagnetic surfaces with a quadrupedal robot [Електронний ресурс] – режим доступу: <https://www.science.org/doi/10.1126/scirobotics.add1017>.
8. RoboBees: Robotic insects make first controlled flight [Електронний ресурс] – режим доступу: <https://phys.org/news/2013-05-robobees-robotic-insects-flight-video.html>.
9. Meet RoboBee, a bug-sized, bio-inspired flying robot [Електронний ресурс] – режим доступу: <https://www.latimes.com/science/sciencenow/la-sci-sn-flying-robot-robobee-smallest-ever-20130502-story.html>.

10. Maneuvering and stabilizing control of a quadrupedal robot using a serpentine robotic tail [Электронный ресурс] – режим доступа: <https://www.semanticscholar.org/paper/Maneuvering-and-stabilizing-control-of-a-robot-a-Rone-Ben-Tzvi/86706b1c460d383257c790390aede7b36bf732f3>.

11. Tail Design for Maneuverability [Электронный ресурс] – режим доступа: <https://biomimetics.mit.edu/research/6339b35c-124a-43da-8749-c2a777566065>.

12. RoboTuna [Электронный ресурс] – режим доступа: <https://www.semanticscholar.org/topic/RoboTuna/1825209>.

13. MIT's Robotic Fish Takes First Swim [Электронный ресурс] – режим доступа: <https://news.mit.edu/1994/robotuna>.

14. An Inchworm-Inspired Crawling Robot [Электронный ресурс] – режим доступа: [https://www.researchgate.net/figure/iCrawl-robot-is-inspired-by-the-structure-and-locomotion-behavior-of-an-inchworm-A\\_fig1\\_346695687](https://www.researchgate.net/figure/iCrawl-robot-is-inspired-by-the-structure-and-locomotion-behavior-of-an-inchworm-A_fig1_346695687).

15. Робот RHex [Электронный ресурс] – режим доступа: <http://edurobots.ru/dictionary/robot-rhex/>.

16. Redundant musculoskeletal robot with thin McKibben muscles [Электронный ресурс] – режим доступа: [https://www.researchgate.net/figure/Redundant-musculoskeletal-robot-with-thin-McKibben-muscles-b-Robotic-and-human\\_fig17\\_308043606](https://www.researchgate.net/figure/Redundant-musculoskeletal-robot-with-thin-McKibben-muscles-b-Robotic-and-human_fig17_308043606).

17. Tentacle robot can gently grasp fragile objects [Электронный ресурс] – режим доступа: <https://seas.harvard.edu/news/2022/10/tentacle-robot-can-gently-grasp-fragile-objects>.

18. Soft Robot Moves Like a Caterpillar [Электронный ресурс] – режим доступа: <https://www.asme.org/topics-resources/content/soft-robot-moves-like-a-caterpillar>.

19. CATERPILLAR-LIKE SOFT ROBOT WITH DISTRIBUTED PROGRAMMABLE THERMAL ACTUATION [Электронный ресурс] – режим доступа: <https://hackaday.com/2023/03/26/caterpillar-like-soft-robot-with-distributed-programmable-thermal-actuation/>.

20. BigDog | Boston Dynamics [Електронний ресурс] – режим доступу: [www/ URL: https://www.bostondynamics.com/bigdog](https://www.bostondynamics.com/bigdog) - 13.10.2023р.
21. Ультразвукової далекомір hc-sr04, клас робототехніки [Електронний ресурс] – режим доступу до ресурсу: <https://ua.waykun.com/articles/arduino-ultrazvukovoi-dalekomir-hc-sr04-klas.php> - 10.11.2023р.
22. Робота з Arduino і MPU6050 [Електронний ресурс] – режим доступу: <https://arduino-diy.com/arduino-MPU6050-dlya-opredeleniya-ugla-naklona> - 11.11.2023р.
23. Документація MatLab [Електронний ресурс] – режим доступу: <https://www.mathworks.com/help/>.
24. Novák, V.; Perfilieva, I.; Močkoř, J. (1999). Mathematical principles of fuzzy logic. Dordrecht: Kluwer Academic. [Електронний ресурс] – режим доступу: [https://www.researchgate.net/publication/228832263\\_Mathematical\\_Principles\\_of\\_Fuzzy\\_Logic](https://www.researchgate.net/publication/228832263_Mathematical_Principles_of_Fuzzy_Logic).
25. What is Fuzzy Logic? [Електронний ресурс] – режим доступу: <https://web.archive.org/web/20181111173944/https://mechanicalsite.com/157/what-is-fuzzy-logic>.
26. Babuška, Robert (1998). Fuzzy Modeling for Control. Springer Science & Business Media. [Електронний ресурс] – режим доступу: [https://books.google.com.ua/books?id=-nzcAAAQBAJ&redir\\_esc=y](https://books.google.com.ua/books?id=-nzcAAAQBAJ&redir_esc=y).
27. Mamdani, E. H. (1974). "Application of fuzzy algorithms for control of simple dynamic plant". Proceedings of the Institution of Electrical Engineers. [Електронний ресурс] – режим доступу: <https://digital-library.theiet.org/content/journals/10.1049/piee.1974.0328>.
28. FuzzyLogicDesigner [Електронний ресурс] – режим доступу: [https://en.wikipedia.org/wiki/Fuzzy\\_logic](https://en.wikipedia.org/wiki/Fuzzy_logic).

29. Xiao, Zhi; Xia, Sisi; Gong, Ke; Li, Dan (1 December 2012). "The trapezoidal fuzzy soft set and its application in MCDM" [Электронный ресурс] – режим доступа:

<https://www.sciencedirect.com/science/article/pii/S0307904X12000510?via%3Dihub>.

30. Wierman, Mark J. "An Introduction to the Mathematics of Uncertainty: including Set Theory, Logic, Probability, Fuzzy Sets, Rough Sets, and Evidence Theory" [Электронный ресурс] – режим доступа: [https://www.creighton.edu/fileadmin/user/CCAS/programs/fuzzy\\_math/docs/MOU.pdf](https://www.creighton.edu/fileadmin/user/CCAS/programs/fuzzy_math/docs/MOU.pdf).

31. Elkan, Charles (1994). "The paradoxical success of fuzzy logic" [Электронный ресурс] – режим доступа: [https://www.researchgate.net/publication/3281863\\_The\\_paradoxical\\_success\\_of\\_fuzzy\\_logic](https://www.researchgate.net/publication/3281863_The_paradoxical_success_of_fuzzy_logic).