

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

рівень вищої освіти другий (магістерський)

Розробка системи керування роботом-собакою з використанням
інтелектуальних мехатронних пристроїв
(тема)

Виконав:

здобувач 2 року навчання,

групи КТРСм-24-2

Алі АБДУЛЛАЄВ

(власне ім'я прізвище)

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

(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 Рауф АЛЛАХБЕРАНОВ

(посада, ім'я прізвище)

Допускається до захисту
Зав. кафедри КІТАР

Ігор НЕВЛЮДОВ

(власне ім'я прізвище)

2025 р.

Я, Абдуллаєв Алі Махір огли, як здобувач вищої освіти ХНУРЕ, розумію та підтримую політику закладу з академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«15» грудня 2025 р.

Алі АБДУЛЛАЄВ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет	Автоматики і комп'ютеризованих технологій
Кафедра	Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
Рівень вищої освіти	другий (магістерський)
Спеціальність	174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Тип програми	освітньо-професійна
Освітня програма	Комп'ютеризовані та робототехнічні системи (код і повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР _____

(підпис)

«___» _____ 2025р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ *Абдуллаєву Алі Махір огли*
(прізвище, ім'я, по батькові)

1. Тема роботи *Розробка системи керування роботом-собакою з використанням інтелектуальних мехатронних пристроїв*
затверджена наказом по університету від "10" листопада 2025р. № 1018 Ст.
2. Термін подання здобувачем роботи "26" грудня 2025р.
3. Вихідні дані до роботи 3.1 Система керування біонічного робота;
3.2 Середовище розробки – Visual Studio Code;
3.3 Середовище моделювання – Gazebo, MATLAB/Simulink;
3.4 База даних – SQLite;
3.5 Мова програмування – Python 3.11, C++.
4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ;
4.2 Аналіз сучасного стану та принципів побудови інтелектуальних робототехнічних систем;
4.3 Моделі та алгоритми керування роботом-собакою;
4.4 Розробка програмно-апаратної частини системи керування;
4.5 Моделювання системи керування та аналіз результатів;
4.6 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Демонстраційний матеріал представлений у форматі презентації PowerPoint (*.ppt) – 26 с. формату А4

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

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	<i>Аналіз сучасного стану та принципів побудови інтелектуальних робототехнічних систем</i>	01.09 – 15.09.25	<i>виконано</i>
2	<i>Моделі та алгоритми керування роботом-собакою</i>	16.09 – 30.09.25	<i>виконано</i>
3	<i>Розробка програмно-апаратної частини системи керування</i>	01.11 – 21.11.25	<i>виконано</i>
4	<i>Моделювання системи керування та аналіз результатів</i>	22.11 – 07.12.25	<i>виконано</i>
5	<i>Оформлення пояснювальної записки</i>	08.12 – 13.12.25	<i>виконано</i>
6	<i>Подання роботи на перевірку Інтернет-системою StrikePlagiarism</i>	14.12 – 16.12.25	<i>виконано</i>
7	<i>Подання роботи на рецензію</i>	17.12 – 19.12.25	<i>виконано</i>
8	<i>Подання роботи на підпис зав. кафедри</i>	20.12 – 23.12.25	<i>виконано</i>
9	<i>Подання кваліфікаційної роботи в ЕК</i>	24.12.25	<i>виконано</i>

Дата видачі завдання 01.09.2025 р.

Здобувач _____ Алі АБДУЛЛАЄВ
(підпис)

Керівник роботи _____ доцент Рауф АЛЛАХВЕРАНОВ
(підпис) (посада, власне ім'я прізвище)

РЕФЕРАТ

Пояснювальна записка: 102 с., 14 табл., 32 рис., 2 дод., 41 джерел.

РОБОТ-СОБАКА, СИСТЕМА КЕРУВАННЯ, ІНТЕЛЕКТУАЛЬНІ МЕХАТРОННІ ПРИСТРОЇ, МОДУЛЬ, АЛГОРИТМ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, АРХІТЕКТУРА, МЕТОД, СЕНСОР.

Мета роботи – забезпечення адаптивності, стійкості і енергоефективності руху біонічного робота в умовах змінного середовища.

Об'єкт дослідження – процес керування біонічним роботом із чотириновою кінематичною структурою.

Предмет дослідження – методи побудови адаптивної інтелектуальної системи керування роботом-собакою.

У кваліфікаційній роботі розроблено програмно-апаратну систему керування роботом-собакою на основі інтелектуальних мехатронних пристроїв. Створено математичну модель руху, побудовано апаратну та програмну архітектуру, розроблено модулі оцінювання стану, генерації траєкторій та гібридну систему керування. Моделювання та експериментальні дослідження підтвердили стійкість системи, точність моделей та підвищення ефективності керування. Отримані результати можуть бути використані для вдосконалення автономних роботизованих платформ.

Отримані результати роботи можна віднести до Цілі сталого розвитку 9 «Промисловість, інновації та інфраструктура», зокрема до пункту 9.4 «Розвиток високотехнологічного машинобудування».

ABSTRACT

Explanatory note: 102 pp., 14 tab., 32 fig., 2 appendices, 41 sources.

ROBOT DOG, CONTROL SYSTEM, INTELLIGENT MECHATRONIC DEVICES, MODULE, ALGORITHM, SOFTWARE, ARCHITECTURE, METHOD, SENSOR.

The purpose of the work is to ensure the adaptability, stability, and energy efficiency of a bionic robot's movement in a changing environment.

The object of the study is the process of controlling a bionic robot with a four-legged kinematic structure.

The subject of the study is methods for constructing an adaptive intelligent control system for a robot dog.

The thesis develops a software and hardware system for controlling a robot dog based on intelligent mechatronic devices. A mathematical model of movement has been created, hardware and software architectures have been built, and modules for state assessment, trajectory generation, and a hybrid control system have been developed. Modeling and experimental studies have confirmed the stability of the system, the accuracy of the models, and the improvement in control efficiency. The results obtained can be used to improve autonomous robotic platforms.

The results of the work can be attributed to Sustainable Development Goal 9 “Industry, Innovation and Infrastructure,” in particular to paragraph 9.4 “Development of high-tech engineering”.

ЗМІСТ

Перелік скорочень	9
Вступ	10
1 Аналіз сучасного стану та принципів побудови інтелектуальних робототехнічних систем	13
1.1 Класифікація та характеристика роботів біонічного типу	13
1.2 Мехатронні системи в сучасній робототехніці	17
1.3 Методи інтелектуального керування роботами	22
1.4 Висновки до першого розділу	30
2 Моделі та алгоритми керування роботом-собакою	31
2.1 Формалізація задачі керування	31
2.2 Структурна схема системи керування	33
2.3 Алгоритм інтелектуального керування	34
2.4 Моделі сенсорних і виконавчих пристроїв	37
2.5 Висновки до другого розділу	40
3 Розробка програмно-апаратної частини системи керування	41
3.1 Загальні принципи побудови програмно-апаратної частини системи керування	41
3.2 Апаратна частина системи керування	46
3.3 Середовище розробки та мови програмування для створення програмного забезпечення	55
3.4 База даних і структура даних	58
3.5 Гібридна система керування	63
3.6 Програмне забезпечення системи керування	67
3.7 Інтерфейс взаємодії та візуалізація	72

3.8 Тестування програмно-апаратної системи керування	75
3.9 Висновки до третього розділу	79
4 Моделювання системи керування та аналіз результатів	80
4.1 Математична модель руху робота-собаки	80
4.2 Моделювання системи стабілізації	82
4.3 Моделювання гібридної системи	84
4.4 Аналіз стійкості системи керування	86
4.5 Моделювання взаємодії робота з поверхнею	87
4.6 Моделювання енергоспоживання	89
4.7 Порівняння експериментальних і моделювальних даних	91
4.8 Забезпечення безпечних умов праці у навчальній лабораторії	93
4.9 Висновки до четвертого розділу	95
Висновки	96
Перелік джерел посилання	99
Додаток А Код програми main.py	103
Додаток Б Публікація за темою досліджень	104
Додаток В Демонстраційний матеріал	112

ПЕРЕЛІК СКОРОЧЕНЬ

БД – база даних;

КПО – коефіцієнт природної освітленості;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

ПІ – пропорційно-інтегральний;

ПІД – пропорційно-інтегрально-диференціальний;

СК – система керування;

ТЗ – технічне завдання.

ВСТУП

На сьогодні розвиток робототехніки можна описати як активне впровадження інтелектуальних систем керування, котрі, зі свого боку, реалізують автономність й адаптивність, а також здатність до самонавчання технічних засобів. Крім того, стає актуальним напрям щодо створення біонічних мобільних роботів, як-от роботів-собак, які за допомогою своєї кінематичної та динамічної структури відтворюють принципи руху чотириногих тварин. Так, у порівнянні з класичними колісними чи гусеничними платформами, чотириногі роботи спроможні подолати складні нерівності рельєфу, балансувати на обмежених опорних поверхнях, а також адаптувати ходу під час змін умов середовища. Такі властивості сприяють перспективі їхнього застосування, зокрема, у рятувальних операціях, охоронній діяльності, військовій розвідці, під час моніторингу сільського господарства та сервісної робототехніки, об'єктів інфраструктури.

Водночас, забезпечення стійкого та керованого руху для робота-собаки – це складна інженерна задача. Так, мехатронна система робота характеризується високим ступенем нелінійності, багатовимірністю, наявністю змінних зовнішніх впливів, а також взаємозалежністю між положенням мас центру тіла й опорною реакцією кінцівок. Утім загальноприйняті методи керування, що спираються на лінійні моделі (ПД-регулятори), характеризуються обмеженою ефективністю, отже, необхідного рівня адаптивності не забезпечують. В умовах, що відповідають реальності, рух робота супроводжується значними впливами, як-от похилим або м'яким ґрунтом, нерівномірним тиском опор, прослизанням, ударами чи вібраціями. За таких умов система керування має бути спроможна враховувати невизначеність, навчатися, враховуючи досвід, коригувати

поведінку у режимі реального часу.

Таким чином, залучення інтелектуальних методів керування, на кшталт нейронних мереж, нечіткої логіки та алгоритмів навчання з підкріпленням, які відповідають за самостійне формування оптимальних стратегій руху системою, набуває актуальності. Реалізація гібридного підходу, котрий поєднує кілька методів штучного інтелекту, сприяє підвищенню надійності та точності разом із плавністю та стійкістю динаміки руху робота-собаки. До того ж, інтеграція інтелектуальних алгоритмів до структури мехатронних модулів приводів втілює скорочення затримки у зворотному зв'язку та покращує енергоефективність системи.

Отже, тема даної кваліфікаційної роботи є актуальною, адже спрямована на розроблення інтелектуальної системи керування роботом-собакою, котра спроможна реалізовувати стійкий та адаптивний рух, ґрунтуючись на сенсорних даних й алгоритмах штучного інтелекту.

Мета роботи – забезпечення адаптивності, стійкості і енергоефективності руху в умовах змінного середовища.

Об'єкт дослідження – процес керування біонічним роботом із чотириновою кінематичною структурою.

Предмет дослідження – методи побудови адаптивної інтелектуальної системи керування роботом-собакою.

Методи дослідження – теорія автоматичного керування, мехатроніка, теорія нечітких множин, штучні нейронні мережі, навчання з підкріпленням, комп'ютерне моделювання.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз сучасних механізмів та систем керування біонічними мобільними роботами;
- сформувати структурну та математичну моделі робота-собаки;

- розробити гібридний алгоритм керування на основі нейронної мережі, нечіткої логіки та навчання з підкріпленням;
- реалізувати моделі сенсорної та виконавчої систем у MATLAB/Simulink та/або Gazebo;
- провести моделювання руху на різних типах поверхні та оцінити енергоефективність і стійкість керування;
- проаналізувати результати та сформулювати рекомендації щодо практичної реалізації системи;
- забезпечити безпечні умови праці у навчальній лабораторії де виконувалась кваліфікаційна робота.

Пояснювальна записка з кваліфікаційної роботи оформлена згідно з рекомендаціями з підготовки і оформлення кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти [1, 2], а також з ДСТУ 3008:2015 [3]. Апробацію результатів досліджень проведено у [4].

Отримані результати роботи можна віднести до Цілі сталого розвитку 9 «Промисловість, інновації та інфраструктура», зокрема до пункту 9.4 «Розвиток високотехнологічного машинобудування».

1 АНАЛІЗ СУЧАСНОГО СТАНУ ТА ПРИНЦИПІВ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНИХ РОБОТОТЕХНІЧНИХ СИСТЕМ

1.1 Класифікація та характеристика роботів біонічного типу

Біонічні роботи вважаються різновидом інтелектуальних мехатронних систем, а принципи їхньої побудови ґрунтуються на імітації морфології, рухових характеристиках і поведінкових особливостей живих організмів. Так, залучення біонічного підходу під час проектування роботів надає можливість підвищувати ефективність їхньої взаємодії з навколишнім середовищем, покращувати маневреність, стійкість та енергетичну економічність.

Особливої уваги приділено роботам-собакам. Їм властива чотиринога конфігурація, а, отже, вони можуть й рухатися по пересіченій місцевості, долати перешкоди чи виконувати автономні завдання.

З огляду на функціональні, конструктивні та цільові ознаки, біонічні роботи доцільно поділити на кілька основних груп:

а) за типом біологічного прототипу:

1) антропоморфні (роботи-люди);

2) зооморфні (роботи-тварини, зокрема собаки, кішки, павуки, риби, птахи);

3) ентомоморфні (роботи, котрі імітують комах);

б) за кількістю ступенів свободи та типом ходової системи:

1) двоногі (bipedal);

2) чотириногі (quadruped);

3) шести- та восьминогі (hexapod, octopod);

4) із комбінованим пересуванням (ходіння у поєднанні із колісним або гусеничним рухом);

в) за рівнем автономності:

- 1) керовані дистанційно (manual/tele-operated);
- 2) напівавтономні (semi-autonomous, із частковим ШІ-контролем);
- 3) повністю автономні (autonomous, із самонавчанням й адаптуванням);

г) за призначенням:

- 1) дослідницькі (експериментальні зразки для випробування мехатронних технологій);
- 2) сервісні (для обслуговування, доставки, розвідки);
- 3) пошуково-рятувальні (для роботи у небезпечних або недоступних зонах);
- 4) військові (розвідка, логістика, підтримка операцій);
- 5) освітні та демонстраційні (навчальні платформи, популяризація STEM).

Роботи-собаки вважаються як найбільш поширеним, так і технологічно розвиненим видом біонічних роботів.

Зокрема, вони мають такі особливості:

- високу стійкість під час руху, оскільки застосовують чотири точки опори;
- спроможність пересуватися нерівною поверхнею, сходами, по піску чи гравію;
- модульну структуру, адже кожна нога – це окремий мехатронний модуль;
- інтегровані сенсорними системами, котрі відповідають за орієнтацію в просторі (IMU, LIDAR, стереокамери);
- енергоефективні приводи (серводвигуни з редукторами чи електрогідравлічні системи);

– інтелектуальне керування, що реалізується за допомогою алгоритмів нейромереж, нечіткої логіки або підкріплювального навчання з метою стабілізації руху.

У таблиці 1.1 зведено порівняльні характеристики сучасних роботів-собак.

Таблиця 1.1 – Порівняльні характеристики сучасних роботів-собак

№	Модель робота	Виробник	Маса, кг	Макс. швидкість м/с	Тип керування	Особливості
1	Spot	Boston Dynamics (USA)	32	1,6	Автономне / дистанційне	Система стабілізації руху, LIDAR, AI-навігація
2	Unitree Go1 Pro	Unitree Robotics (China)	12	3,5	Автономне	Інтелектуальне слідування за користувачем
3	Cyber Dog 2	Xiaomi (China)	9	3,2	Напівавтон.	Система розпізнавання облич, керування голосом
4	ANYmal C	ANYbotics (Swizerl.)	50	1,2	Автономне	Промислові місії, робота у вибухонебезпечних середовищах

Для більшості біонічних роботів властива модульна мехатронна архітектура, яка містить:

- рухові модулі (кінцівки), зокрема, 3 – 4 ступені свободи на кожен ногу, за допомогою яких можна адаптувати положення тіла до рельєфу;
- сенсорний модуль – IMU, гіроскопи, акселерометри, датчики сили/моменту, візуальні системи;
- енергетичний модуль, тобто акумуляторні батареї чи компактні паливні елементи;

– модуль керування, тобто мікроконтролер або одноплатний комп'ютер (на кшталт, Raspberry Pi 4, Jetson Nano), за допомогою якого діють алгоритми стабілізації.

Переваги біонічного підходу окреслюються:

– адаптивністю, тобто можливістю пристосування до складного середовища;

– енергоефективністю, де рух, подібний до природної ходи, виконується з мінімальними втратами енергії;

– гнучкістю та маневреністю, адже робот може пересуватися сходами, ямами, нахилами;

– стійкістю, оскільки чотири точки опори зменшують ризик щодо втрати рівноваги;

– можливістю залучення в екстремальних умовах, зокрема, під час пожеж, на руїнах, у зонах хімічного зараження.

На рисунку 1.1 наведено типову структурну схему робота-собаки.



Рисунок 1.1 – Узагальнена структурна схема біонічного робота-собаки

Таким чином, біонічні роботи, зокрема, чотириногі роботи-собаки, набули популярності, оскільки є одними з найефективніших мобільних платформ для реалізації інтелектуальних мехатронних систем. У таких моделях стійкість поєднується з адаптивністю й високим рівнем автономності. Сучасні тенденції розвитку свідчать про перехід від класичних систем керування до гібридних алгоритмів, які ґрунтуються на елементах штучного інтелекту.

1.2 Мехатронні системи в сучасній робототехніці

Мехатроніка набула визначення як міждисциплінарна галузь, яка, зі свого боку, об'єднує електроніку, механіку та інформатику разом із системами керування задля створення інтелектуальних технічних систем. Однак у контексті сучасної робототехніки мехатронні модулі власне стали основою для побудови автономних роботів, які спроможні виконувати складні рухи, адаптуватися до зовнішніх умов, а також взаємодіяти з людиною.

Так, розвиток мехатронних технологій дозволяє окреслити як рівень інтелектуалізації роботів, так і точність їхніх рухів разом із ступенем автономності.

Мехатронну систему визначають як комплекс, який суміщає механічну (кінематичну структуру, приводи) та електронну частини (сенсори, контролери, виконавчі елементи) із програмним забезпеченням, яке реалізує інтелектуальне керування.

Провідною рисою таких систем доцільно назвати зворотний зв'язок, за допомогою якого можна аналізувати стан системи й автоматично коригувати параметри руху в режимі реального часу.

Загальну структуру мехатронної системи продемонстровано на рисунку 1.2.

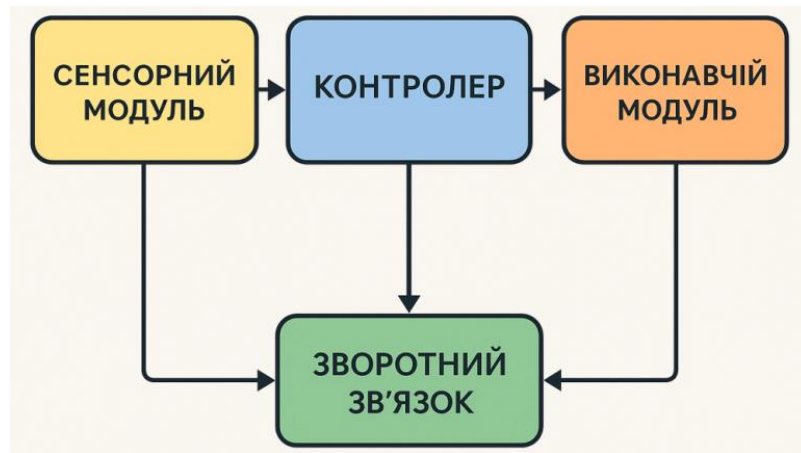


Рисунок 1.2 – Узагальнена структура мехатронної системи робота

Отже, узагальнена структура мехатронної системи робота містить такі складові, як: сенсорний модуль, який забезпечує визначення параметрів (положення, швидкості, навантаження, температури); контролер, який аналізує вхідні сигнали і формує подальші дії для керування; виконавчий модуль, який відповідає за рух рух або дію за командою контролера; зворотній зв'язок, тобто канал передавання інформації про стан системи; програмне забезпечення (ПЗ), за допомогою якого визначається логіка функціонування системи.

1.2.1 Основні компоненти мехатронних систем у робототехніці

1.2.1.1. Сенсорні підсистеми. Сенсори забезпечують сприйняття зовнішнього середовища, а також інформують про внутрішній стан робота.

Серед найпоширеніших типів сенсорів перелічимо:

- інерціальні вимірювальні одиниці (IMU), що визначають кутові швидкості, прискорення, а також орієнтацію в просторі;
- LIDAR і ультразвукові сенсори, що формують карту навколишнього середовища;
- камери (RGB, глибини, стерео), що забезпечують комп'ютерний зір;

- датчики навантаження і моменту, що контролюють тиск на опори робота;
- GPS/RTK-модулі, що відповідають за позиціонування у відкритому просторі.

1.2.1.2. Виконавчі пристрої (приводи). Механічні дії робота реалізуються шляхом застосування приводів, за допомогою яких електрична енергія перетворюватиметься на механічну.

Виділяють такі типи приводів:

- серводвигуни (DC або BLDC), котрі фіксують точне позиціонування кінцівок;
- електрогідравлічні приводи, котрі задіяні у потужних промислових роботах;
- п'єзоелектричні приводи, котрі властиві мікророботам і точним механізмам;
- пневмоприводи, котрі характерні для м'яких (soft) роботів, а також біоміметичних систем.

У таблиці 1.2 зіставлено порівняльні характеристики типів приводів у мехатронних системах.

Таблиця 1.2 – Порівняльні характеристики типів приводів у мехатронних системах

Тип приводу	Переваги	Недоліки	Застосування
Електричний	Висока точність, швидка реакція, просте керування	Обмежена потужність, перегрів під час навантаження	Мобільні роботи, маніпулятори
Гідравлічний	Велика сила, стабільність руху	Складна система живлення, шум	Важкі антропоморфні роботи

Продовження таблиці 1.2

Тип приводу	Переваги	Недоліки	Застосування
Пневматичний	М'якість руху, простота конструкції	Мала точність, потреба в компресорі	Біонічні системи, навчальні платформи
П'єзоелектричний	Надвисока точність, компактність	Невелика амплітуда руху, висока вартість	Мікророботи, оптичні системи

1.2.1.3. Контролери та обчислювальні платформи. Контролери відповідають за збір даних із сенсорів, оброблення сигналів і формування впливів керування.

Найпопулярнішими платформами наразі є:

- Arduino Mega / Due актуальні для базових проєктів і відлагодження алгоритмів;
- Raspberry Pi 4 / 5 застосовуються для реалізації систем із технічним зором (камерами) і підключенням до ROS;
- Jetson Nano / Xavier NX актуальні для глибинного навчання і локального оброблення зображень;
- STM32 / ESP32 придатні для вбудованих систем реального часу.

Переважно контролери можуть поєднувати в мережеві структури, в яких один контролер, тобто головний, відповідає за координацію, а периферійні – за керування конкретними модулями.

1.2.1.4. Програмне забезпечення мехатронних систем. Сучасна робототехніка послуговується інтегрованими платформами, зокрема:

- ROS (Robot Operating System) – середовище для взаємодії модулів, навігації, оброблення сенсорних даних;
- MATLAB/Simulink – моделювання динамічних процесів;
- Python/C++ – основні мови для послуговування логікою керування;

– Gazebo, Webots– 3D-модельовання руху, а також тестування без реального апарата.

1.2.2 Тенденції розвитку мехатроніки в робототехніці

Розвиток мехатроніки став рушієм для появи інтелектуальних модулів із власною сенсорною системою, мікроконтролером та алгоритмами самоналаштування.

Завдяки цим модулям можна:

- регулювати параметри руху власноруч;
- діагностувати несправності;
- взаємодіяти з іншими елементами через комунікаційні шини (CAN, UART, Wi-Fi, ROS).

Наочно зразок інтелектуального мехатронного модуля приводу кінцівки робота продемонстровано на рисунку 1.3.

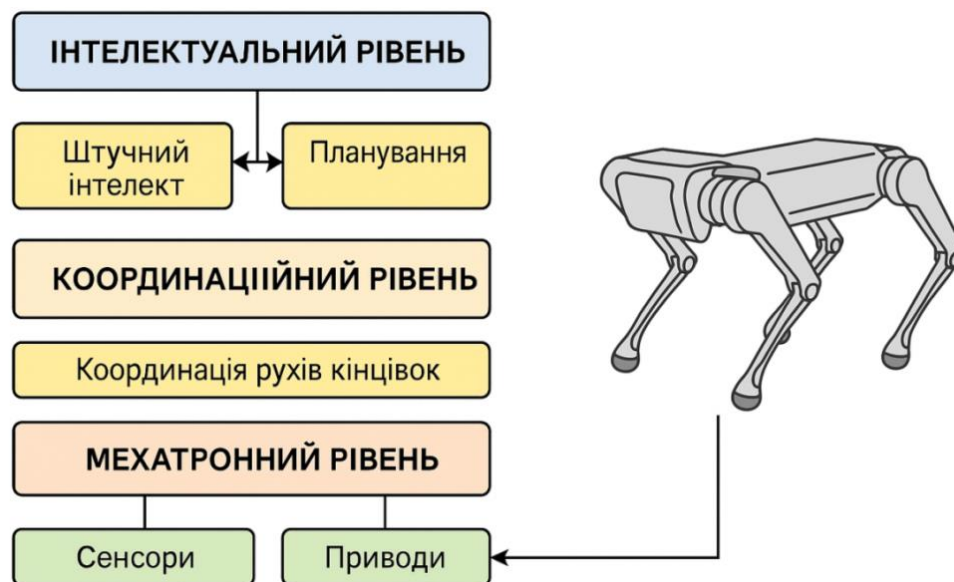


Рисунок 1.3 – Приклад інтелектуального мехатронного модуля приводу кінцівки робота

Серед тенденцій розвитку мехатроніки в робототехніці можна виокремити:

- модульність і масштабованість, тобто створення універсальних блоків, які придатні для різних типів роботів;
- інтеграцію штучного інтелекту, зокрема, впровадження алгоритмів самоадаптації та прогнозованого керування;
- енергоефективність, тобто використання рекупераційних систем у приводах;
- мініатюризація, зокрема, розвиток мікромехатроніки для роботів медичних і дослідницьких;
- застосування нових матеріалів, як-от: гнучких полімерів, легких композитів, 3D-друк деталей.

Так, мехатронні системи вважаються фундаментом сучасної робототехніки, оскільки забезпечують точність, гнучкість та інтелектуальність функціонування роботів. Окрім того, їхній розвиток спрямовано на підвищення рівня автономності шляхом інтеграції сенсорних технологій, штучного інтелекту й ефективних виконавчих механізмів.

1.3 Методи інтелектуального керування роботами

У зв'язку із розвитком як мехатроніки, так і штучного інтелекту власне класичні методи керування роботами поступово вдосконалюються або замінюються інтелектуальними алгоритмами, за допомогою яких реалізуються адаптивність, самоорганізація та спроможність до навчання.

На противагу загальноприйнятим системам, які спираються на точні математичні моделі, інтелектуальні методи надають можливість функціонувати системам в умовах невизначеності, шумів, а також змінних зовнішніх чинників.

Зазначені підходи є характерними для біонічних мобільних роботів, наприклад, роботів-собак, які спроможні пересуватися нерівною поверхнею, долати перешкоди, адаптувати ходу до рельєфу, і водночас опанувати нові сценарії руху (рисунок 1.4).

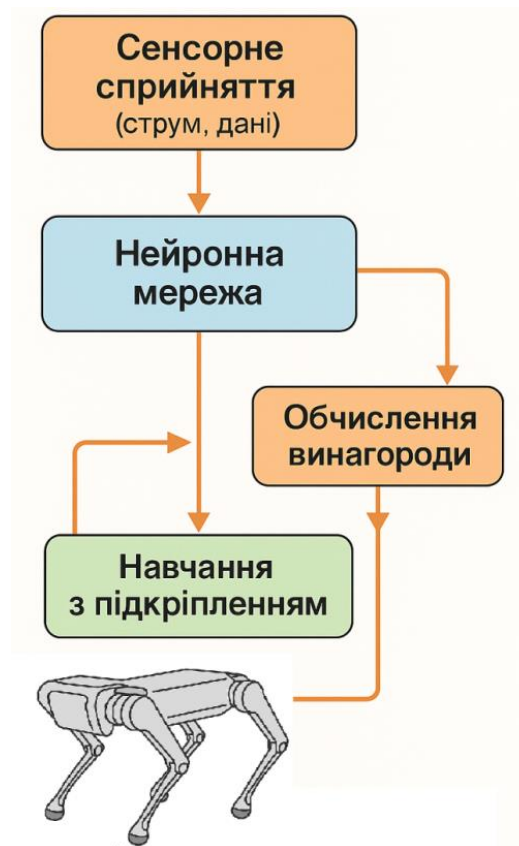


Рисунок 1.4 – Основні підходи до інтелектуального керування роботами

У цілому методи інтелектуального керування можна розподілити за такими групами:

- методи, що ґрунтуються на знаннях: нечіткі системи, експертні моделі;
- методи, що ґрунтуються на навчанні: нейронні мережі, системи з підкріпленням, еволюційні алгоритми;
- комбіновані або гібридні системи: суміщення декількох методів з метою підвищення стійкості та швидкості адаптації.

1.3.1 Класичні методи керування

Загальноприйняті системи керування засновані на точній математичній моделі об'єкта. Найбільшого застосування набув PID-регулятор. Він забезпечує стабілізацію руху шляхом регулювання помилки між бажаним станом і фактичним.

Формула класичного PID-регулятора має такий вигляд:

$$U(t) = K_p \cdot e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt}, \quad (1.1)$$

де $U(t)$ – сигнал, який керує;

$e(t)$ – похибка;

K_p, K_i, K_d – коефіцієнти регулювання.

Утім цей метод містить і недоліки:

- неефективність при змінних навантаженнях;
- низька адаптивність;
- необхідність точної математичної моделі.

Таким чином, для керування біонічними роботами зі складною динамікою доцільно послуговуватися інтелектуальними адаптивними методами.

1.3.2 Нечіткі системи керування

Нечіткі системи керування (Fuzzy Logic Control) ґрунтуються на принципі нечіткої логіки, котрий було запропоновано Л. Заде (Lotfi Zadeh) [11].

За його основу було обрано оброблення лінгвістичних змінних, які словами описують стан системи (“мала швидкість”, “велика помилка”, “повільне прискорення”), замінюючи точні числові значення.

Структура нечіткої системи складається з:

- фазифікації: мова йде про перетворення реальних даних на нечіткі множини;
- бази правил IF–THEN:IF помилка велика AND швидкість зростає THEN зменшити потужність приводу;
- механізму нечіткого висновку (Mamdani або Sugeno);
- дефазифікації, тобто перетворення нечіткого сигналу на точне керування.

Однак до переваг нечіткої логіки можна віднести те, що система функціонує без математичної моделі, водночас вона стійка до шумів та легко реалізується у реальному часі.

Утім наявні й недоліки, зокрема, мова йде про важку оптимізацію у разі великої кількості змінних, а також труднощі під час налаштування правил вручну.

На прикладі це проявляється так: нечітка логіка у робота-собаки може керувати потужністю кожного приводу кінцівки, забезпечуючи, зі свого боку, плавність рухів під час ходіння нерівною поверхнею.

1.3.3 Нейронні мережі

Нейронна мережа (Artificial Neural Networks) насамперед імітує роботу людського мозку. Вона складається з нейронів (обчислювальних вузлів), які перетворюють вхідні дані на вихідні сигнали шляхом застосування нелінійних функцій активації.

Мережа навчається, спираючись на досвід, а саме: коригуючи ваги зв'язків між нейронами для мінімізації помилки.

Основні типи нейронних мереж зведено у таблиці 1.3.

Таблиця 1.3 – Основні типи нейронних мереж

Тип мережі	Основне призначення	Приклад застосування
MLP (багатошаровий перцептрон)	Нелінійна апроксимація функцій	Регулювання кута нахилу тіла робота
CNN (згорткова)	Аналіз зображень і відеопотоку	Виявлення перешкод камерою
RNN / LSTM	Прогнозування часових рядів	Оцінка стабільності під час руху
Autoencoder	Виявлення аномалій	Самодіагностика системи

Ключові переваги застосування нейронної мережі полягають у здатності до самонавчання, адаптивності, роботі з великими масивами даних.

Проте недоліки можна визначити як потреба в потужних обчислювальних ресурсів, а також вимогу для етапу тренування.

Наочно це виглядає так: робот-собака може послуговуватися CNN для аналізу зображень із камери, а також вибору маршруту, а MLP – з метою стабілізації положення корпусу.

1.3.4 Навчання з підкріпленням

Принцип методу навчання з підкріпленням (Reinforcement Learning, RL) діє так: система отримує винагороду за “правильні” дії і штраф – за “небажані”.

Завдання – максимізувати сумарну винагороду шляхом багатьох ітерацій проб і помилок.

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \quad (1.2)$$

де R_t – сумарна очікувана винагорода в момент часу t ;

r_{t+k+1} – миттєва винагорода через $k+1$ кроків;

$\gamma \in [0; 1]$ – коефіцієнт дисконтування, котрий визначає «вагу майбутніх винагород»;

k – індекс циклу оновлення політики навчання.

Перелічимо типові алгоритми методу навчання:

- Q-Learning – базовий метод навчання політики;
- Deep Q-Network (DQN) – використання глибоких нейронних мереж для оцінки станів;
- PPO, DDPG, SAC – сучасні алгоритми для безперервного середовища (хода робота).

Розглянемо приклад застосування: робот-собака в середовищі Gazebo навчається рухатися вперед: за кожен успішний крок він отримує +1, проте у разі падіння – -10. Таким чином, після тисяч ітерацій система самостійно знаходить оптимальну стратегію ходи.

1.3.5 Еволюційні алгоритми

Сутність еволюційних алгоритмів (Genetic Algorithms, GA) можна трактувати так: алгоритм імітує природний добір, отже, випадкові “особини” (параметри системи) схрещуються, мутують і відбираються за критерієм “пристосованості”.

Основними етапами можна визначити:

- ініціалізацію популяції параметрів;
- оцінку кожного за функцією якості;
- вибір найкращих особин;
- схрещування (crossover) і мутація (mutation);
- створення нової популяції.

Наведемо приклад застосування: налаштувавши коефіцієнти PID-регулятора для приводу ноги робота за допомогою еволюційних алгоритмів можна знайти оптимальні K_p , K_i , K_d без залучення аналітичного розрахунку.

1.3.6 Гібридні інтелектуальні системи

Сутність підходу реалізується в тому, що гібридні системи поєднують переваги різних методів, створюючи, зі свого боку, комплексну адаптивну систему, зокрема:

- Neuro-Fuzzy Control – поєднує нечітку логіку із нейромережевим навчанням;
- GA + Fuzzy Logic – генетичний алгоритм оптимізує базу правил нечіткої системи;
- RL + CNN – навчання з підкріпленням, а також аналізом відео з камери для навігації.

Переваги гібридного підходу полягають як у високій точності та стабільності, так і можливостях до самоадаптації й універсальності (функціонування під час різних умов середовища).

На рисунку 1.5 продемонстровано гібридну систему керування роботом-собакою.

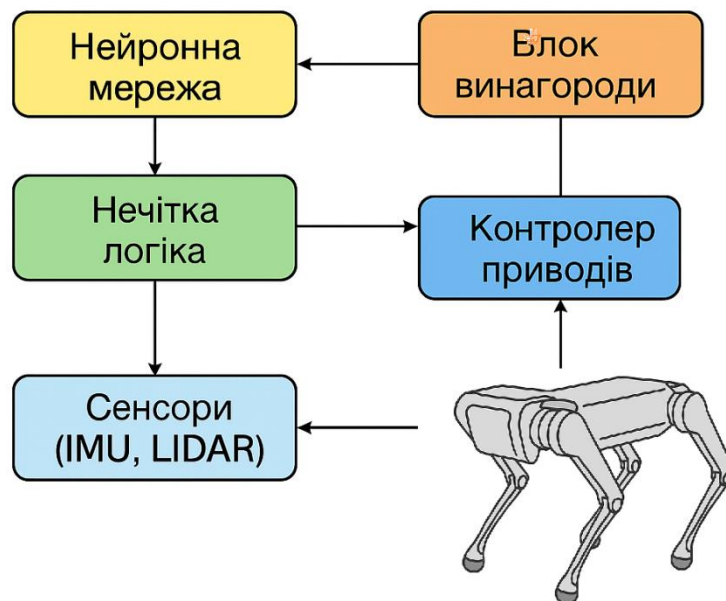


Рисунок 1.5 – Гібридна система керування роботом-собакою

Так, у роботі-собаці можуть застосовуватись:

- CNN + RL – для навчання стабільній ході по пересіченій місцевості;
- Neuro-Fuzzy – для регулювання обертового моменту в приводах;
- GA – для оптимізації параметрів системи керування.

1.3.7 Порівняльний аналіз методів інтелектуального керування

Проаналізувавши вищезазначені методи інтелектуального керування, було проведено порівняльний аналіз, результати якого відображені в таблиці 1.4.

Таблиця 1.4 – Порівняльний аналіз методів інтелектуального керування

Метод	Необхідність моделі	Адаптивність	Обчислювальні ресурси	Застосування
ПІД-регулятор	Висока	Низька	Малі	Прості роботи
Нечітка логіка	Необов'язкова	Середня	Середні	Регулювання приводів
Нейронна мережа	Необов'язкова	Висока	Високі	Стабілізація ходи
Підкріплювальне навчання	Необов'язкова	Дуже висока	Дуже високі	Автономна навігація
Генетичні алгоритми	Необов'язкова	Висока	Середні	Оптимізація параметрів
Гібридні системи	Необов'язкова	Дуже висока	Високі	Інтелектуальні роботи

Отже, методи інтелектуального керування власне є підґрунтям для створення адаптивних та самонавчальних мехатронних систем.

Для системи керування роботом-собакою доречно послуговуватися гібридною структурою, котра містить:

- нейронну мережу (необхідна для прогнозу й стабілізації руху);

- нечітку логіку (застосовують для регулювання сили приводів);
- алгоритм підкріплювального навчання (актуальний для самонавчання в симуляції).

Застосування такого підходу забезпечує високу стійкість, автономність, а також ефективність руху у складних динамічних середовищах.

1.4 Висновки до першого розділу

У першому розділі було проведено аналіз сучасного стану робототехнічних систем, їх мехатронної структури та принципів побудови систем керування. Розглянуто типові сенсорні підсистеми, виконавчі механізми та інтелектуальні модулі, що застосовуються в роботах-собаках.

Опрацювавши сучасні робототехнічні та мехатронні системи, можемо константувати, що створення автономного робота-собаки можливе лише у разі інтеграції інтелектуальних алгоритмів керування з адаптивними сенсорними системами й енергоефективними приводами у єдину гібридну мехатронну архітектуру.

2 МОДЕЛІ ТА АЛГОРИТМИ КЕРУВАННЯ РОБОТОМ-СОБАКОЮ

2.1 Формалізація задачі керування

Систему керування роботом-собакою визначають власне як багатовимірну нелінійну динамічну систему з розподіленими параметрами.

Головна мета керування полягає в забезпеченні стабільної ходи, рівноваги та адаптивності руху у разі наявних зовнішніх впливів (нерівностей, перешкод, зміни рельєфу).

Динаміку робота доречно описати системою рівнянь Лагранжа другого роду:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + \tau_{\text{зовн}}, \quad (2.1)$$

де q – вектор узагальнених координат (положення кінцівок і корпусу);

$M(q)$ – матриця інерції;

$C(q, \dot{q})$ – матриця коріолісових і центробіжних сил;

$G(q)$ – вектор гравітаційних сил;

τ – вектор керування моментів у приводах;

$\tau_{\text{зовн}}$ – вектор зовнішніх сил (контакт із поверхнею, поштовхи).

Утім система повинна забезпечувати як синхронізацію рухів усіх кінцівок, так і підтримувати положення центру мас у межах полігону стійкості:

$$\sum_i P_i = mg, \quad (2.2)$$

де P_i – реакції опорних точок;

2.2 Структурна схема системи керування

Власне систему керування утворено з функціональних модулів, а взаємодія між ними реалізується за допомогою комунікаційних шин ROS, CAN та PC.

Сенсорна підсистема, зі свого боку, складається з:

- IMU, тобто інерціального модуля, котрий визначає кути нахилу, прискорення й орієнтацію;
- LIDAR, що формує 3D-карту простору;
- камери, що функціонує як комп'ютерний зір;
- датчиків моменту та тиску, необхідні для контролю контактів із поверхнею.

Дані сенсорів піддаються попередній фільтрації (фільтр Калмана, середнє ковзне).

Обчислювальна підсистема устаткована одноплатними комп'ютерами (Raspberry Pi, Jetson Nano), мікроконтролерами STM32 та платою Arduino. Вона забезпечує взаємодію між інтелектуальними алгоритмами та мехатронними вузлами. До того ж, виконує функції щодо оброблення даних у реальному часі, формування сигналів керування, а також прогнозування стану системи на наступний крок.

Інтелектуальний контролер або гібридна система керування набуває визначення центрального модуля, що поєднує:

- нейронну мережу, котра прогнозує динаміку руху;
- нечіткий регулятор, який коригує кутові швидкості приводів;
- RL-алгоритм, який вибирає оптимальні дії, що, зі свого боку, мінімізують помилки стабілізації;

– блок винагороди, котрий оцінює ефективність руху за критеріями енергоспоживання та стабільності.

Результатом роботи стає сигнал керування $\tau(t)$, який подається на контури приводів.

Контролери приводів виконують керування серводвигунами кінцівок (по 3-4 DOF на ногу) і водночас підтримують такі функції:

- PID-регулювання відповідно до положення і моменту;
- локальна компенсації впливів;
- реєстрація даних для діагностики.

Виконавча система утворена з електромеханічних приводів, редукторів і сенсорів навантаження. Вона відповідає за точне відтворення рухів, стабільність ходи та зворотний зв'язок щодо швидкості та моменту.

Комунікаційна підсистема зазвичай послуговується такими протоколами:

- ROS задіяно між модулями високого рівня;
- CAN необхідно для передачі даних між контролерами приводів;
- UART/I²C/SPI наявно у сенсорних вузлах.

Окрім того, забезпечує синхронізацію всіх підсистем у реальному часі.

Підкреслимо, що система керування роботом-собакою – багаторівнева (рисунок 2.1).

Рівні системи керування:

а) верхній рівень (інтелектуальний):

- 1) модуль планування траєкторії;
- 2) алгоритм навчання з підкріпленням (RL);
- 3) обробка сенсорних даних (LIDAR, камери, IMU).

б) середній рівень (координаційний):

- 1) нейронна мережа прогнозування станів;
- 2) блок нечіткого керування стабілізацією;

- 3) контролер розподілу моментів серед приводів;
- в) нижній рівень (мехатронний):
- 1) серводвигуни з локальними контролерами;
 - 2) зворотний зв'язок щодо положення, швидкості чи струму.
- Зв'язок між рівнями виконується за допомогою протоколів ROS і CAN.



Рисунок 2.1 – Узагальнена структурна схема системи керування роботом-собакою

2.3 Алгоритм інтелектуального керування

2.3.1 Структура гібридного алгоритму

Гібридний алгоритм керування роботом-собакою суміщає між собою моделі штучного інтелекту з класичними методами автоматичного регулювання. Важливість такого підходу обумовлено тим, що рух чотириноного робота є нелінійним, багатовимірним, а також залежить від зовнішніх впливів, отже, застосування тільки одного типу контролера (на кшталт ПД або тільки нейромережі) буде недостатнім.

Гібридна система поєднує три підходи (нейронний, нечіткий регулятор та навчання з підкріпленням).

Нейронна мережа (ANN), зі свого боку, прогнозує стан тіла робота відповідно до сенсорних даних:

$$\hat{q}(t + 1) = f(W \cdot q(t), \dot{q}(t), a(t)), \quad (2.3)$$

де W – матриця вагових коефіцієнтів.

Нечіткий регулятор (Fuzzy Logic) відповідає за визначення корекційного моменту τ_f , що залежить від похибки положення:

$$\tau_f = K_f(\mu(e), \mu(\dot{e})), \quad (2.4)$$

де $\mu(e)$ – функція належності нечіткої множини.

Навчання з підкріпленням (RL) власне оптимізує стратегію керування через винагороду R_t :

$$R_t = \sum_{k=0}^N \gamma^k r_{t+k+1}, \quad (2.5)$$

де γ – коефіцієнт дисконту;

r_t – миттєва винагорода (стійкість, енергоефективність).

2.3.2 Цільова функція

У системах навчання з підкріпленням провідним елементом буде функція винагороди, котра визначає бажану поведінку керованого об'єкта. Втім для робота-собаки така функція має зважати не лише на досягнення поставленої мети (наприклад, переміщення вперед), проте й на стійкість руху, енергоефективність, а також плавність динаміки.

Зважаючи на те, що робот діє в умовах невизначеності середовища та змінної опори, то типові лінійні функції якості (квадратична похибка або мінімізація моменту) будуть недостатніми. З огляду на це, доречним буде застосування інтегральної цільової функції, котра узагальнить декілька критеріїв оптимальності.

Оптимізацію можна виконати за критерієм:

$$J = \min[\alpha \|e_p\|^2 + \beta \|e_v\|^2 + \delta P_{\text{ен}}], \quad (2.6)$$

де e_p – похибка положення;

e_v – похибка швидкості;

$P_{\text{ен}}$ – споживана потужність.

2.3.3 Логіка взаємодії підсистем

Гібридну систему керування роботом-собакою (рисунок 2.2) реалізовано як багаторівневу інтегровану структуру, в якій окремі підсистеми виконують взаємодоповнювальні функції, забезпечуючи, зі свого боку, адаптивність, стійкість та точність руху.

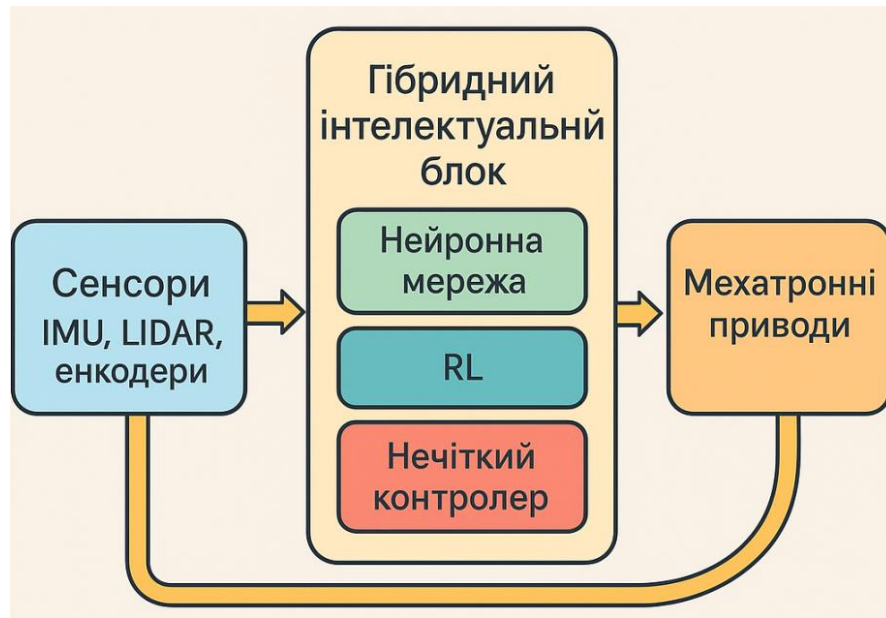


Рисунок 2.2 – Алгоритм взаємодії модулів гібридної системи керування

Головна особливість системи керування полягає в поєднанні алгоритмів штучного інтелекту (нейронних мереж і навчання з підкріпленням) із загальноприйнятими схемами зворотного зв'язку та нечіткого керування. Таким чином, можна компенсувати невизначеність середовища, а також зменшити вплив шумів та коливань сигналів, до того ж, забезпечити стабільність у реальному часі.

Гібридний інтелектуальний блок має такі складові, а саме:

- нейронну мережу, що прогнозує нахили поверхні та майбутні позиції;
- RL-агента, котрий обирає найефективнішу дію;
- нечіткого контролера, що стабілізує локальні відхилення;
- контролерів приводів, які виконують команди у реальному часі.

Отже, замкнений цикл адаптивного керування утворюється за такою схемою: «Сенсори → NN → RL → FLC → Приводи → Зворотний зв'язок».

2.4 Моделі сенсорних і виконавчих пристроїв

2.4.1 Модель сенсорного модуля

Сенсорний модуль робота-собаки призначений для формування інформації про поточний стан механічної структури та взаємодію з опорною поверхнею. З огляду на те, що система керування функціонує у режимі реального часу, то сенсорним даним повинні бути властиві достатня точність, частота оновлення, а також можливість узгодження в єдиній системі координат.

Сенсорна система складається з:

- IMU, тобто інерційного модуля, котрий необхідний для орієнтації та прискорення;
- LIDAR формування карти перешкод;
- камери для виявлення цілей.

Вихід сенсорів можемо описати як вектор вимірювань:

$$y(t) = Hx(t) + n(t), \quad (2.7)$$

де H – матриця спостереження;

$n(t)$ – шум сенсорів.

Для фільтрації доцільно послуговуватися фільтром Калмана:

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - H\hat{x}_{t|t-1}), \quad (2.8)$$

$$K_t = P_{t|t-1}H^T(H P_{t|t-1}H^T + R)^{-1}. \quad (2.9)$$

2.4.2 Модель приводу

Привід робота-собаки реалізовано як мехатронну систему, в якій поєднано електромеханічні перетворювачі енергії із системою зворотного зв'язку. Складовими елементами такого приводу є: електродвигун постійного струму чи BLDC-двигун, енкодер для вимірювання кутового положення, редуктор із передавальним числом i_g , а також мікроконтролерний регулятор.

Динаміку приводу можна описати як сукупність електричної та механічної підсистем.

Кожен привід власне описується системою:

$$J\ddot{\theta} + b\dot{\theta} = K_t i - M_{\text{нагр}}, \quad (2.10)$$

де J – момент інерції;

b – коефіцієнт в'язкості;

K_t – константа моменту;

$M_{\text{нагр}}$ – навантаження.

Динаміка електричної частини:

$$L \frac{di}{dt} + Ri = U - K_e \dot{\theta}, \quad (2.11)$$

де L – індуктивність;

R – опір;

K_e – коефіцієнт ЕРС.

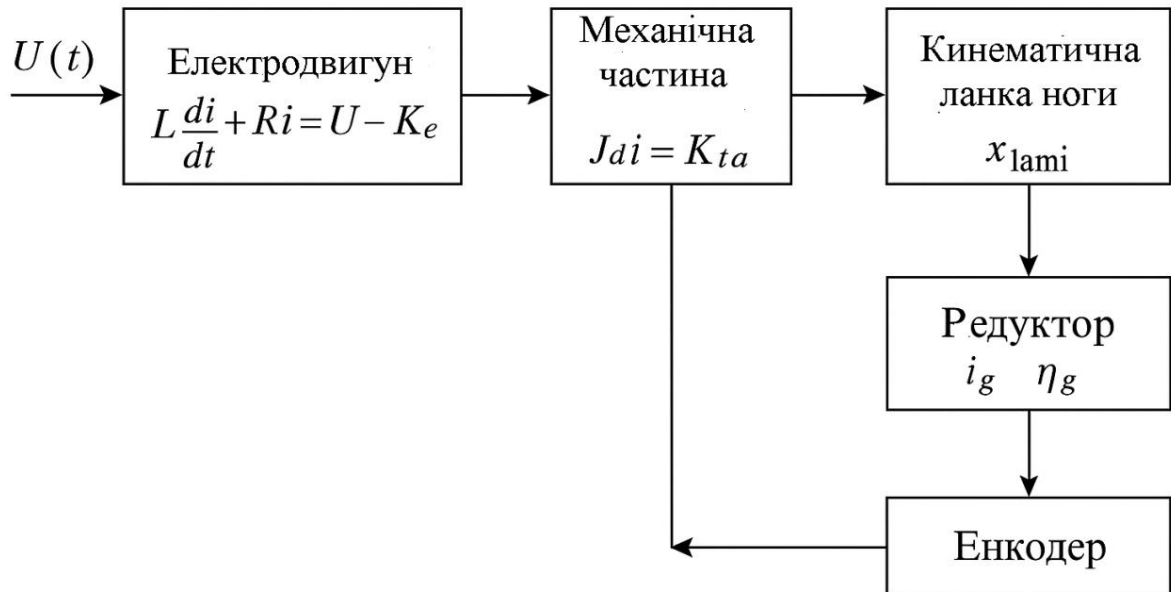


Рисунок 2.3 – Динамічна модель мехатронного приводу кінцівки робота-собаки

2.5 Висновки до другого розділу

У даному розділі визначено теоретичні основи системи керування роботом-собакою. Розроблено математичну модель руху робота з урахуванням зворотного зв'язку. Сформовано структуру керування із такими рівнями: інтелектуальним, координаційним, мехатронних. Доведено актуальність застосування гібридної системи керування, котра суміщає нейронну мережу із навчанням з підкріпленням та нечіткою логікою. Зібрані рівняння для опису сенсорних і виконавчих модулів.

Проведений аналіз став основою для вибору гібридного підходу до керування роботом-собакою та визначення архітектури системи, що реалізується у наступних розділах роботи.

3 РОЗРОБКА ПРОГРАМНО-АПАРATНОЇ ЧАСТИНИ СИСТЕМИ КЕРУВАННЯ

3.1 Загальні принципи побудови програмно-апаратної частини системи керування

Проектування програмно-апаратної частини системи керування роботом-собакою зазвичай супроводжується комплексним підходом, адже відтак отримуємо інтегровану інтелектуальну систему із механічною, електронною та програмною складовими. За таких умов побудова згаданої архітектури окреслюється забезпеченням стійкого й адаптивного, і водночас енергоефективного керування рухом багатоланкової платформи з можливістю швидкого реагування на зміну зовнішніх чинників.

Так, чотириногі мобільні роботи сьогодення належать до класу кіберфізичних систем, оскільки опрацювання інформації, що надходить від сенсорів, а також обчислення керуючих дій, робота мехатронних приводів реалізуються у тісній інтеграції [29, 30]. Таким чином, створення зазначеної архітектури, що характеризується високою точністю, мінімальними затримками разом із стійкістю до перешкод, а також масштабованістю в умовах ускладнення задач або додавання нових модулів набуває актуальності.

3.1.1 Принцип модульності й ієрархічної побудови

Нагальним принципом організації програмно-апаратної частини системи визначено її модульність, тобто йдеться про поділ усієї системи на окремі функціональні блоки. Втім кожен модуль виконує чітко окреслену задачу, до

того ж, взаємодіє з іншими шляхом залучення стандартизованих інтерфейсів.

Так, ієрархічна структура системи ґрунтується на декількох рівнях, як-от:

- сенсорному – реалізує збір разом із попереднім обробленням інформації, що надходить від ІМУ, енкодерів, датчиків сили, а також додаткових сенсорів;
- керувальний (контролер) – зорієнтований як на централізовану координацію підсистем, так і планування дій чи оброблення сигналів у реальному часі;
- програмний – утілює алгоритми прогнозування та стабілізації й охоплює засоби аналізу даних;
- інтелектуальний – запроваджує вищий контроль за рухом, формує стратегію поведінки, послуговуючись штучним інтелектом.
- виконавчий – виконує фізичний рух робота шляхом задіяння електромеханічних приводів.

Таким чином, розглянута ієрархічність сприяє легкому поєднанню алгоритмів різної складності, водночас зменшує навантаження на центральний контролер, застосовуючи власне розподіл обчислень.

3.1.2 Розподілена структура керування та взаємодія компонентів

Програмно-апаратна частина системи розроблена з огляду на принцип розподіленого оброблення, як-от на рисунку 3.2, що, зі свого боку, знижує загальний час реакції і насамперед забезпечує відмовостійкість.

У зазначеній моделі частина функцій реалізується не центральним контролером, а навпаки локальними мікроконтролерами, котрі, як правило:

- фільтрують сигнали сенсорів;
- контролюють окремі приводи;
- передають дані, які вже були опрацьовані чи агреговані дані;

– втілюють підтримку зворотного зв'язку.

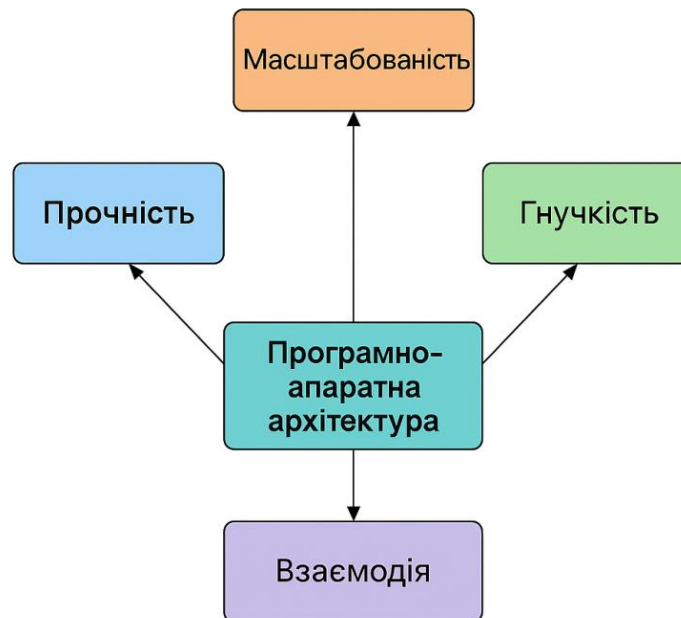


Рисунок 3.1 – Прицип побудови програмно-апаратної частини системи керування роботом-собакою

Підкреслимо, що для обміну інформацією задіюються стабільні та швидкі протоколи (I²C, SPI, CAN, UART), а в системах – власне протоколи розширеної взаємодії (Ethernet, Wi-Fi) [31].

Окреслений підхід зорієнтовано на скорочення кількості перешкод, оптимізацію енергоспоживання, а також забезпечення стабільного функціонування в динамічних середовищах.

3.1.3 Часові вимоги та робота в реальному часі

Однією з провідних характеристик системи керування визначено спроможність функціонувати за умов жорстких часових обмежень. Втім, усі процеси, що попередньо активовано у системі, супроводжуються регламентом часових затримок, а саме:

- зчитування сенсорних даних – 1 – 3 мс;
- оновлення стану мехатронного приводу – до 5 мс;
- робота алгоритмів стабілізування – 5 – 10 мс;
- оновлення політики RL або нейронної мережі – 20 – 30 мс.

Однак ключовий цикл системи керування працює з частотою 200 – 500 Гц. Проте невиконання часових вимог призводить до втрати роботом рівноваги чи затримки в реакції на зовнішні чинники.

Тож для забезпечення роботи в реальному часі насамперед послуговуються:

- оптимізованими алгоритмами та фільтрами;
- апаратними таймерами;
- пріоритетними чергами задач;
- механізмами інтерполяції та прогнозування стану.

Таким чином, ефективно об'єднувати дані з різних сенсорів, а також компенсувати шум та похибки вимірювань цілком можливо за допомогою фільтра Калмана.

3.1.4 Принцип адаптивності й інтелектуальної взаємодії

Вочевидь система керування повинна безумовно адаптуватися до:

- нерівностей поверхні;
- зміни рельєфу;
- раптових перешкод;
- ковзання лап;
- зміни ваги чи навантаження;
- несподіваних впливів.

Втілюється це шляхом застосування нейронних мереж або алгоритмів навчання з підкріпленням (RL) та нечіткої логіки.

Зокрема, інтелектуальний рівень системи забезпечує такі функції:

- прогнозувати стан роботи наперед;
- аналізувати сенсорні дані у контексті поведінкових моделей;
- оптимізувати рух у разі мінімізації енерговитрат;
- адаптувати сигнали керування до поточного рельєфу;
- запобігати порушенням сталості на кшталт перевантаження, ковзання, хитання корпусу).

3.1.5 Відмовостійкість та надійність

З метою реалізації безпечної роботи робота-собаки власне архітектура ґрунтується на таких механізмах:

- дублюванні критичних даних;
- перевірці коректності сигналів сенсорів;
- захисту від перегріву приводів;
- автоматичному перемиканні на безпечний режим;
- перезапуску окремих модулів у разі збоїв;
- протоколюванні помилок і станів.

Зокрема, локальні контролери забезпечують діагностику приводів у реальному часі щодо температури, струму, положення чи швидкості.

Відтак помилки під час функціонування мережеских інтерфейсів компенсуються апаратними буферами разом із системами повторного передавання даних [31].

3.1.6 Масштабованість і гнучкість архітектури

Архітектура системи керування вирізняється низкою можливостей:

- додаванням нових сенсорів і модулів;
- розширенням функціоналу;

- модернізацією нейронного блоку;
- інтеграцією зі сторонніми платформами як-от: ROS 2, Gazebo, MATLAB.

Зазвичай гнучкість реалізується стандартизованими інтерфейсами, модульною структурою програмного забезпечення, а також за допомогою відкритих протоколів.

Тож загальні принципи розроблення програмно-апаратної системи керування роботом-собакою охоплюють взаємодію модульності й ієрархічності, адаптивності та масштабованості, водночас і стійкості до збоїв. Окрім того, послуговування інтелектуальними алгоритмами гарантує високу точність разом із надійністю руху, а розподілена архітектура, зі свого боку, допомагає мінімізувати затримки та підвищити ефективність роботи.

Розглянута система відповідає вимогам кіберфізичних платформ сучасності, до того ж, є рушієм для подальшого розвитку робототехнічної системи.

3.2 Апаратна частина системи керування

Апаратна частина системи керування роботом-собакою побудована як комплекс електронних, сенсорних та механічних компонентів, які насамперед взаємодіють між собою, отже, реалізують отримання даних разом із їхнім попереднім обробленням, утворюють сигнали керування, а також активують рух виконавчих механізмів [29-33]. Підкреслимо, що структуру апаратної частини розроблено з огляду на те, аби забезпечити високу точність та швидкість реакції, надійність і відмовостійкість у широкому спектрі умов експлуатації, зокрема, йдеться про функціонування як на рівних поверхнях, так і на складному рельєфу з нерівностями чи слизькими або м'якими ділянками.

Загалом апаратна архітектура ґрунтується на принципі розподіленого керування, як це відтворено на рисунку 3.2, де частина обчислень реалізується локальними мікроконтролерами, а глобальна координація – центральним контролером. Тож це зумовлює зменшення навантаження на центральний модуль, а також мінімізацію затримки і водночас збільшення стійкості системи в реальному часі.



Рисунок 3.2 – Апаратна архітектура системи керування роботом-собакою

3.2.1 Центральний контролер

Центральний контролер, зі свого боку, позначає ядро системи. Зазвичай, він інтегрує дані від усіх сенсорів, а також формує сигнали керування для приводів. Окрім того, він зорієнтований на:

- взаємодію з інтелектуальним блоком (нейронна мережа, RL-алгоритм);
- залучення високорівневих алгоритмів керування;
- об'єднання інформаційних потоків від усіх сенсорів;
- формування команд приводам у заданому часовому інтервалі.

Так, на рисунку 3.3 продемонстровано, як центральний контролер інтегрується з сенсорами, локальними мікроконтролерами та приводами за допомогою різних комунікаційних шин (CAN, I²C).

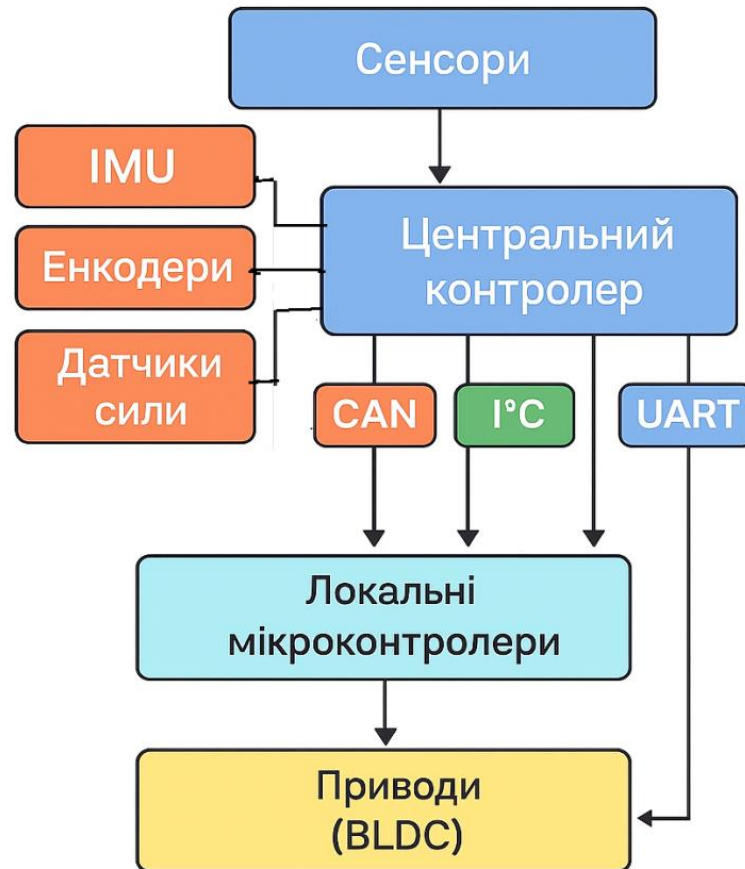


Рисунок 3.3 – Схема взаємозв'язку центрального контролера і периферії

Перважно як центральний процесор можна застосовувати:

- NVIDIA Jetson Nano / Xavier / Orin – якщо система залучає для роботи нейронні мережі у реальному часі;
- Raspberry Pi 4/5 – для функціонування систем середньої складності;
- STM32H7 або подібні мікроконтролери – коли нагальною стає потреба щодо низького енергоспоживання.

Визначимо пріоритетні вимоги до центрального модуля:

- устаткування багатоядерним процесором;
- підтримка апаратного прискорення обчислень (CUDA або SIMD);
- послуговування швидкісними інтерфейсами: SPI, I²C, UART, CAN, USB;
- низька затримка щодо оброблення даних (< 5 – 10 мс).

3.2.2 Сенсорний комплекс

Сенсорна система власне – це безумовно важлива частина апаратної архітектури, адже вона, зі свого боку, спрямована на формування вектора стану робота.

Перелічимо вагомні компоненти сенсорної системи:

- інерціальний модуль (IMU). Реалізовано за допомогою з 3-осьового гіроскопа, 3-осьового акселерометра та магнітометра (опціонально) з метою визначення: кутових швидкостей і прискорень, а також орієнтації корпусу разом із відхиленнями платформи від рівноваги. Найчасті застосовують такі моделі: MPU-9250, BNO055, ICM-20948. Зокрема, IMU працює з частотою 100 – 400 Гц, а дані проходять через фільтр Калмана задля зменшення шуму [33, 34];

- енкодери кутового положення. Встановлені на кожному суглобі кінцівки. Можуть бути як інкрементальні, так і абсолютні, із роздільною здатністю 12 – 16 біт, до того ж, характеризуються точністю визначення кута до 0,1°. Загалом енкодери забезпечують точну інформацію про стан приводів, а також утворюють зворотній зв'язок задля стабілізації [33, 34];

- датчики сили та моменту. Ними устатковані лапи робота чи приводи, котрі є сенсорами струму. Як правило, вони задіяні для встановлення контакту із землею чи аналізу прослизання. Крім того, спроможні контролювати навантаження та коригувати фази кроку. Найчастіше використовують датчики тензорного типу або датчики струму ACS712/INA219 [33, 34];

– додаткові сенсори (опціонально). Так, сенсор LiDAR є актуальним для побудови карти рельєфу, водночас ультразвукові датчики – для пошуку перешкод, а стереокамери, зі свого боку, – для комп’ютерного зору та навігації [34].

Тож для оброблення шумів доцільно використати фільтр Калмана:

$$\hat{x}(t) = \hat{x}(t - 1) + K(z(t) - H\hat{x}(t - 1)), \quad (3.1)$$

де $\hat{x}(t)$ – стан, який вимагає оцінювання;

$z(t)$ – вимірювання сенсора;

K – коефіцієнт фільтра.

3.2.3 Виконавча підсистема

Приводи робота характеризуються як невід’ємна складова, що перетворює електронні сигнали на механічний рух.

Визначимо ключові елементи кожного приводу:

– BLDC або сервомотор. Насамперед застосовується, щоб збільшити обертовість, забезпечити високу швидкість проте зберегти низький рівень нагріву;

– редуктор, як-от: планетарний, циклоїдний, гармонічний, із передавальним числом, як правило, 6:1 – 20:1, що залежить від маси робота. Отже, реалізує збільшення моменту та плавність руху;

– сенсори зворотного зв’язку, зокрема, енкодери, датчики струму та температури;

– драйвер двигуна. Передусім зорієнтований на комутацію обмоток і керування струмом, до того ж, реалізує захист від перевантаження та підтримку PWM або CAN-інтерфейсу;

– CAN-шина або RS-485. Складник, за допомогою якого можна під'єднати до 20 – 40 приводів з мінімальною затримкою.

Спершу опишемо електричну частину виконавчого приводу робота-собаки:

$$L \frac{di}{dt} + Ri = U - K_e \omega, \quad (3.2)$$

де L – індуктивність;

R – опір;

U – напруга;

K_e – електрична стала;

ω – кутова швидкість.

Відтак механічну частину виконавчого приводу робота-собаки відтворимо за допомогою рівняння:

$$J \frac{d\omega}{dt} + b\omega = K_t i - M_{\text{нагр}}, \quad (3.3)$$

де J – момент інерції;

b – коефіцієнт тертя;

K_t – моментна стала;

$M_{\text{нагр}}$ – момент навантаження.

3.2.4 Джерело живлення і розподіл сигналів

Джерелом живлення обрано Li-Po акумулятор 4S – 6S (14,8 – 22,2 В), номінальна ємність якого – 4000 – 10000 мА·год.

Перелічимо ключові складники блоку живлення:

- модулі DC-DC перетворення 24→5 В, 24→12 В;
- балансувальний контролер заряду;
- моніторинг напруги і струму;
- захист від короткого замикання;
- розділення силової та логічної шин для уникнення наведень.

Підкреслимо, що силовий тракт приводів ізольований від цифрової частини з метою уникнення шумів.

У зв'язку з тим, що джерело живлення, насамперед повинно забезпечувати стабільне функціонування всіх компонентів, то доцільно використовувати модель розподілу напруги та струму:

$$V_{\text{вих}} = V_{\text{акб}} - I_{\text{сум}} \cdot R_{\text{вн}}, \quad (3.4)$$

де $V_{\text{вих}}$ – напруга акумулятора;

$I_{\text{сум}}$ – загальний струм споживання;

$R_{\text{вн}}$ – внутрішній опір акумулятора.

3.2.5 Комунікаційні інтерфейси

Власне обмін даними втілюється за допомогою комунікаційних інтерфейсів:

- CAN, тобто головного протоколу для приводів (висока швидкість, відмовостійкість);
- І²С, а саме сенсорної інформації (IMU, барометр);
- SPI, зокрема, швидкими ADC та енкодерами;
- UART, шляхом діагностики, низькошвидкісними модулями;
- Wi-Fi 2,4/5 ГГц, через зовнішню комунікацію з оператором;
- Ethernet, як-от високошвидкісними підключеннями до ПК.

Як правило, протокол CAN виконує адресацію до кожного приводу, а також забезпечує стійкість до електричних перешкод [2, 35].

3.2.6 Механічна та корпусна структура

Корпус робота-собаки призначений не лише для захисту механічної структури, проте водночас набуває значення елемента кінематичної структури.

Так, механічна структура робота-собаки охоплює розподіл маси та кінематику суглобів. Зокрема, кінематична модель прораховує положення кінцівок:

$$x_{\text{кінц}}(t) = f(\theta_1(t), \theta_2(t), \dots, \theta_n(t)), \quad (3.5)$$

де $\theta_n(t)$ – кутові координати суглобів.

Наочний приклад механічної структури корпусу робота-собаки продемонстровано на рисунку 3.4.

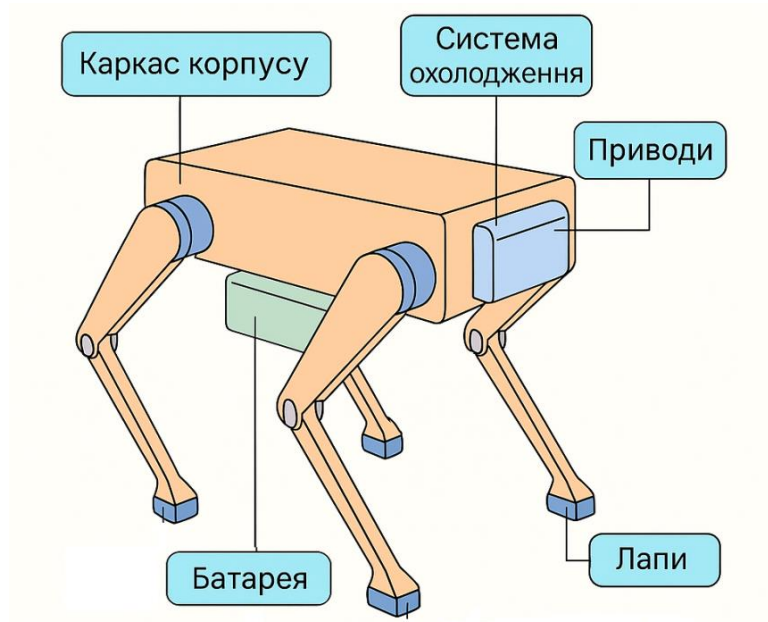


Рисунок 3.4 – Механічна структура корпусу робота-собаки

Окрім того, конструкція враховує низку вимог:

- легкість (алюміній + карбон);
- модульність – швидка заміна приводу або сенсора;
- мінімальні зусилля у суглобах;
- жорсткість на вигин;
- захист від пилу та вологи (IP54 – IP67).

Власне механічна структура обумовлена моделями центрів мас, структурною жорсткістю, кінематичним аналізом, а також стійкістю на чотирьох точках опори.

3.2.7 Відмовостійкість і безпека апаратної частини

Система керування робота, зокрема, реалізує:

- захист від перевищення струму;
- апаратний контроль температури приводів;
- резервування сенсорних каналів;
- автоматичний перехід у безпечний режим у разі втрати сигналу;
- журнал помилок у флеш-пам'яті контролера.

Підкреслимо, що під час критичних ситуацій робот спершу знижує швидкість, відтак займає позицію «Low-Stand», а далі вимикає силові тракати.

Тож апаратна частина системи керування роботом-собакою насамперед розроблена як комплекс кіберфізичних компонентів, які спрямовані на збір інформації, а також обчислення та забезпечення руху з високою точністю. Загалом розподілена архітектура, перевірені сенсорні модулі, адаптивні приводи разом із надійними комунікаційними інтерфейсами стають фундаментом для роботи інтелектуальної системи керування, котру більш детально опишемо в наступних підрозділах.

3.3 Середовище розробки та мови програмування для створення програмного забезпечення

Розроблення ПЗ системи керування роботом-собакою було реалізовано на тлі мов програмування, сучасних інструментів, а також середовищ моделювання. Таким чином, можна забезпечувати високу продуктивність і масштабованість, як і стабільність роботи програмного комплексу. Окрім того, пріоритетними було обрано ті технології, котрі враховують вимоги до реального часу функціонування системи, потребу в інтелектуальних алгоритмах, а також необхідність інтеграції з апаратною частиною.

3.3.1 Основна мова програмування

Головна частина високорівневих алгоритмів, а також нейронні мережі, фільтри оброблення даних, RL-стратегії та GUI-інтерфейсом – усі вони відтворені на мові Python [37]. Зокрема, такий вибір цілком зумовлений низкою переваг даної мови:

- швидким циклом розробки;
- значною кількістю бібліотек для штучного інтелекту (TensorFlow, PyTorch);
- актуальними бібліотеками для математичних обчислень (NumPy, SciPy, SymPy);
- зручністю у реалізації фільтрів, контролерів, протоколів зв'язку;
- підтримкою багатопотоковості та асинхронності.

Окрім того, мовою Python послуговувались у таких підсистемах:

- модулі збору даних (serial, CAN);
- нейронній моделі прогнозування;
- RL-політики управління;

- Fuzzy-контролери;
- інтерфейси оператора (PyQt5);
- логуванні та телеметрії.

3.3.2 Мови низького рівня

Утім взаємодію з локальними мікроконтролерами на кшталт STM32 або ESP32, було реалізовано за допомогою мов C та C++, адже вони забезпечують:

- мінімальні затримки (реальний час);
- прямий доступ до регістрів периферії;
- працездатність драйверів CAN, UART, PWM і I²C.
- точний контроль над сервоприводами і датчиками;

Водночас до функціональних модулів мікроконтролера належать:

- локальна діагностика та тестування.
- обмеження струму та моменту;
- процесор зворотного зв'язку привода;
- цифрові фільтри сенсорів (IMU, енкодери);

3.3.3 Фреймворки та бібліотеки

Як правило, фреймворк TensorFlow / PyTorch спрямований на навчання нейронної моделі, оптимізацію політики RL, а також на прогнозування подальшого стану робота.

Водночас фреймворк PyQt5 охоплює створення графічного інтерфейсу оператора, моніторинг датчиків, відтворення положення робота разом із діагностикою та логуванням [37].

Утім фреймворк scikit-fuzzy втілює нечітку логіку та правила Fuzzy.

Бібліотеки Matplotlib, PyQtGraph, зазвичай, затребувані під час побудови графіків у режимі реального часу, а також для відображення динаміки IMU та

приводів.

Бібліотеки `pyserial`, `python-can`, зі свого боку, реалізують зв'язок із периферійними модулями.

3.3.4 Середовище розробки

Середовище розробки Visual Studio Code (VS Code) – актуальне середовище розробки, адже підтримується Python, C і C++, інтегрується з Git, значною кількістю плагінів для розробки, зручним інтерфейсом.

Так, VS Code став основою для розроблення модулів високого рівня, запуску та налагодження Python-скриптів, а також взаємодії з RL-агентами та мовними моделями.

3.3.5 Середовище моделювання

Середовище моделювання MATLAB/Simulink, зі свого боку, залучалось для моделювання мехатронних приводів, перевірки фільтрів Калмана та тестування контролерів [38].

Середовище моделювання Gazebo насамперед було обрано для моделювання динаміки робота-собаки й аналізу стійкості на різних типах поверхонь, тестування алгоритмів у віртуальному середовищі.

3.3.6 Система контролю версій

Система контролю версій Git + GitHub втілює взаємодію інструменту для керування версіями коду (Git) з онлайн-платформою для хостингу та спільної роботи над проектами (GitHub). Таким чином, було реалізовано командну роботу над проектом, а також резервне зберігання та документування змін.

Так, структура репозиторію охоплює:

– /docs – технічну документацію.

- /logs – телеметрію;
- /models – моделі робота;
- /src – код;

Підкреслимо, що програмний комплекс створено переважно за допомогою сучасних мов програмування та інструментів, які зорієнтовані на:

- високу продуктивність;
- інтеграцію з апаратною частиною;
- гнучкість та масштабованість;
- ефективну роботу інтелектуальних систем керування.

Таким чином, вдалось утворити надійну модульну програмно-апаратну архітектуру, котра спроможна функціонувати в режимі реального часу.

3.4 База даних і структура даних

Продуктивна працездатність програмно-апаратної системи керування роботом-собакою передусім визначається надійною організацією даних, отже, необхідно створити високошвидкісний доступ, зберігати історії вимірювань, телеметрії та параметрів роботи. Тож відповідно до окреслених завдань у кваліфікаційній роботі було розроблено легку та модульну, і zarazом масштабовану структуру даних, яка є актуальною для роботи в режимі реального часу, до того ж, було організовано локальну БД з метою збереження робочих параметрів та журналів подій.

3.4.1 Призначення та вимоги до бази даних

Зазвичай база даних у системі керування роботом характеризується такими функціями:

- збереженням телеметрії за допомогою IMU, енкодерів і датчиків сили;

- логуванням внутрішніх станів системи, на кшталт $X(t)$, помилок, опрацьованих моментів;
- збереженням подій і аварійних станів, як-от: system alerts, overcurrent, overheating;
- збереженням параметрів налаштування контролерів і калібрувальних коефіцієнтів;
- архівацією даних навчання RL-агента.

Утім до БД висуваються й певні вимоги:

- підтримка роботи в режимі «офлайн»;
- мінімальне блокування ресурсів;
- можливість експорту у CSV для аналізу;
- висока швидкість запису (>500 записів/с);
- автономність, отже, і відсутність залежності від серверів;
- низьке навантаження на процесор.

Враховуючи розглянуті вище параметри, вибір було зупинено власне на SQLite, адже вона є вбудованою, легкою, до того ж, придатною до роботи на ARM-контролерах і SBC (Raspberry Pi, Jetson).

3.4.2 Структура бази даних

БД складається з кількох таблиць, які, певні річ, пов'язані між собою.

Так, у таблиці 3.1 зафіксовані дані сенсорів (SensorData).

Таблиця 3.1 – Дані сенсорів

Поле	Тип	Опис
id	INTEGER	первинний ключ
timestamp	REAL	час UNIX-мітки
acc_x, acc_y, acc_z	REAL	прискорення

Продовження таблиці 3.1

Поле	Тип	Опис
gyro_x, gyro_y, gyro_z	REAL	кутові швидкості
encoder_1...n	REAL	положення приводів
force_1...4	REAL	дані датчиків сили

У таблиці 3.2, зокрема, відображені керувальні дії (ControlCommands).

Таблиця 3.2 – Керувальні дії

Поле	Тип	Опис
id	INTEGER	первинний ключ
timestamp	REAL	час
tau_1...n	REAL	момент для кожного суглоба
rl_action_1...m	REAL	дія RL-агента
fuzzy_output	REAL	корекція нечіткої системи

Відтак у таблиці 3.3 зведені події та аварії (SystemEvents)

Таблиця 3.3 – Події та аварії

Поле	Тип	Опис
id	INTEGER	первинний ключ
timestamp	REAL	час події
event_type	TEXT	тип події
description	TEXT	опис

У таблиці 3.4 власне зазначені налаштування системи (Config).

Таблиця 3.4 – Налаштування системи

Поле	Тип	Опис
parameter	TEXT	назва параметра
value	REAL/TEXT	значення
updated	REAL	дата зміни

Зрештою у таблиці 3.5 продемонстровано буфер досвіду RL-агента (RLMemory).

Таблиця 3.5 – Буфер досвіду RL-агента

Поле	Тип	Опис
state	TEXT	$X(t)$ у форматі JSON
action	TEXT	дія
reward	REAL	винагорода
next_state	TEXT	$X(t+1)$
done	INTEGER	Епізод завершено

3.4.3 Логічна модель даних

Підкреслимо, що у структурі даних задіяно гібридну модель:

- реляційна БД (SQLite);
- JSON-структури для станів і дій RL-агента;
- буферні структури в оперативній пам'яті (циклічні масиви для даних реального часу).

За таких умов одна частина системи функціонує з високою пропускнуою здатністю, натомість інша – із стабільним довгостроковим зберіганням.

3.4.4. Модель даних реального часу

Стан робота-собаки доцільно описати вектором:

$$X(t) = [q(t), \dot{q}(t), \phi(t), \omega(t), a(t), F(t)]. \quad (3.6)$$

Кожен з його елементів зберігається у тимчасовому буфері, довжина якого складається з 1000 елементів. Наочно це продемонстровано на рисунку 3.5.

```
class DataBuffer:
    def __init__(self, size=1000):
        self.data = deque(maxlen=size)

    def push(self, sample):
        self.data.append(sample)
```

Рисунок 3.5 – Тимчасовий буфер довжиною 1000 елементів

Як правило, буфер використовується у контролері з метою стабілізації, у RL-агенті – для збирання траєкторій, а у телеметрії – для створення графіків.

3.4.5 Організація телеметрії та логування

Зауважимо, що для швидкого запису залучається пакетне вставлення, приклад якого відтворено на рисунку 3.6.

```
cursor.executemany(
    "INSERT INTO SensorData VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)",
    sensor_batch
)
```

Рисунок 3.6 – Пакетне вставлення

Тож на противагу одиничним операціям, пакетне вставлення прискорює запис у 5 – 7 разів.

3.4.6 Експорт даних

Як відомо, система забезпечує експорт у CSV, автоматичний запис у JSON, а також передавання даних оператору за допомогою TCP/WebSocket.

Таким чином, можна оперативно аналізувати функціонування робота під час експериментів.

Підсумуємо, що розроблені БД та структура даних гарантують надійне, організоване та швидке збирання інформації про робота-собаку разом із його алгоритмами керування. Тож взаємодія реляційної БД з оперативними структурами забезпечують високу продуктивність і гнучкість, а це, зі свого боку, є надто важливим для кіберфізичних систем реального часу.

3.5 Гібридна система керування

Гібридна система керування роботом-собакою охоплює як сучасні алгоритми штучного інтелекту, так і методи класичної теорії автоматичного регулювання. Такий підхід насамперед обумовлений метою досягнути високу точність керування та стійкість до зовнішніх впливів, а також адаптуватися до нових умов і спроможність прогнозувати стани робота згодом.

Як правило, гібридна стратегія архітектури складається з трьох вагомих компонентів:

а) алгоритмів штучного інтелекту (AI-рівень):

- 1) нейронної мережі для оцінювання стану та прогнозування руху;
- 2) алгоритмів навчання з підкріпленням для генерації оптимальних

стратегій;

б) низькорівневого керування приводами (мехатронний рівень):

1) виконання команд моменту/швидкості в реальному часі;

в) класичних контролерів (контрольний рівень)

1) ПІ/ПД-регуляторів для стабілізації;

2) нечітких контролерів Fuzzy для адаптивного демпфування.

Тож окреслена трирівнева архітектура реалізує компроміс між точністю, адаптивністю та швидкістю.

3.5.1 Загальна структура гібридної системи керування

Приклад базової структури взаємодії між рівнями продемонстровано на рисунку 3.7.

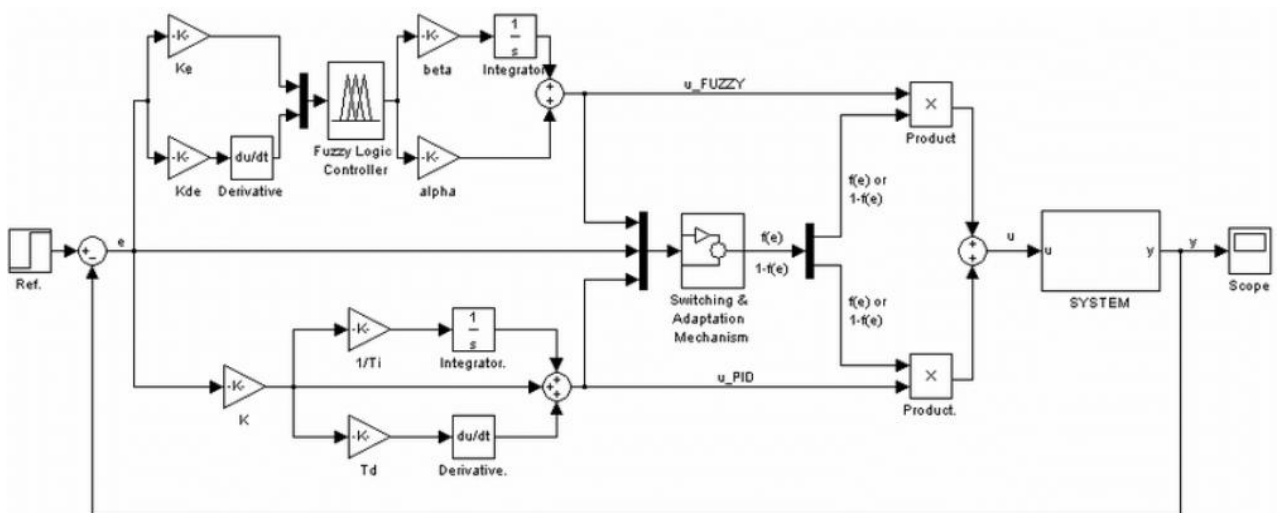


Рисунок 3.7 – Базова структура взаємодії між рівнями

Відтак покроково розглянемо принцип функціонування гібридної системи керування.

Крок I. Сенсорний рівень утворює вектор стану $X(t)$ відповідно до ІМУ, енкодерів, датчиків сили і кінематичного розв'язку моделі.

Крок II. AI-модуль (NN + RL) власне прогнозує стан у майбутньому $\hat{X}(t + 1) = f_{NN}(X(t))$, водночас агент RL генерує дію стратегічну $u_{RL}(t) = \pi(X(t))$.

Крок III. Нечіткий контролер Fuzzy компенсує похибки, котрі залежать від швидких змін динаміки $\Delta u_F(t) = f_{fuzzy}(e(t), \dot{e}(t))$.

Крок IV. Класичний ПІД-контур стабілізації спрямований на підтримку рівноваги, а також обмеження моментів, згладження шумів.

Крок V. Синтез команд $u(t) = u_{RL}(t) + \Delta u_F(t) + u_{PID}(t)$.

Крок VI. Приводам надходить команда моменту або кутової швидкості.

3.5.2 Алгоритм функціонування гібридної системи

Алгоритм функціонування гібридної системи керування роботом-собакою відтворено на рисунку 3.8.



Рисунок 3.8 – Алгоритм функціонування гібридної системи керування

Роз'яснимо етапи реалізації алгоритму гібридної системи керування роботом-собакою:

- отримання сенсорних даних. Спершу система зчитує кути суглобів і прискорення, швидкість і контактні сили;
- фільтрація та оцінювання стану. Комбінований фільтр Калмана, зі свого боку, утворює $X(t) = [q, \dot{q}, \phi, \omega, F]$;
- нейронна мережа оцінює подальшу поведінку. Завдяки цьому можна передбачити момент втрати рівноваги чи ковзання;
- RL-агент генерує дію. Йдеться про стратегічне рішення вищого рівня;
- нечітка система компенсує помилки. Fuzzy-моторика, насамперед, регулює м'якість рухів;
- ПІД забезпечує стабілізувальний вплив. Підтримує рівновагу робота;
- команда передається на приводи з затримкою < 3 мс.

3.5.3 Математична модель гібридного контролю

Змішану стратегію керування доцільно описати так :

$$u(t) = a_1 u_{RL}(t) + a_2 u_{PID}(t) + a_3 u_{Fuzzy}(t), \quad (3.7)$$

де a_1 – коефіцієнт ваги AI-рівня;

a_2 – вага стабілізації;

a_3 – вага адаптивної компенсації.

Так, у звичайних умовах: $a_1=0,6$, $a_2=0,3$ і $a_3=0,1$, проте під час ковзання або втрати рівноваги: $a_3 \rightarrow 0,5$, тобто Fuzzy бере на себе компенсацію.

Переваги гібридної системи керування зведені у таблиці 3.6.

Таблиця 3.6 – Переваги гібридної системи керування

Компонент	Переваги
NN-модель	прогнозування руху, адаптація
RL-агент	оптимальні стратегії поведінки
Fuzzy	м'якість руху, компенсація шумів
PID	гарантована стабілізація
разом	точність + адаптивність + передбачення

3.6 Програмне забезпечення системи керування

ПЗ системи керування роботом-собакою розроблено як багаторівневий комплекс модулів, які в реальному часі взаємодіють між собою. Його ключова мета окреслює реалізацію стійкого й адаптивного, а також передбачуваного руху платформи у разі динамічних змін зовнішнього середовища. Власне архітектура побудована за принципом модульності, що насамперед забезпечує незалежний розвиток, налагодження та тестування кожного функціонального блоку.

Поясними принципи побудови програмного забезпечення:

- модульність, отже, кожен логічний блок відповідає за окрему функцію (сенсори, AI, ПД, кінематика, приводи);
- реальний час, тобто обчислювальний цикл працює з частотою 200–500 Гц, тому вдається оперативно реагувати на зміни.

Окрім того, програмне забезпечення містить кілька логічних рівнів, які забезпечують:

- збір та оброблення сенсорних даних;
- оцінювання стану;
- прийняття рішень (AI, RL);

- стабілізації (ПД/Fuzzy);
- низькорівневий рівень приводів.

Одак на кожному циклі виконується порівняння бажаних значень із реальними для формування корекцій, тобто втілюється зворотний зв'язок.

До того ж, частина модулів працює в окремих потоках, як-от сенсори, логування, GUI, тому запроваджується паралельне оброблення.

Відтак аналізуються перевантаження, збої датчиків, стани, тож актуальним стає і безперервний моніторинг стану.

Зазначимо, що послідовність проходження даних через програмні модулі можна відтворити так: «Сенсорні дані → Фільтрація → Оцінювання стану → Формування бажаної траєкторії → AI-рівень → ПД/Fuzzy → Команди приводам → Зворотний зв'язок».

Таким чином, наведено приклад замкненої системи керування, в якій на кожен наступний етап впливає коректність попереднього.

Утім для ефективної реалізації всі дані зберігаються у спеціальних внутрішніх структурах даних.

Фрагмент коду структури даних продемонстровано на рисунках 3.9 – 3.10.

```
state = {
  "pos": np.zeros(12),      # кути суглобів
  "vel": np.zeros(12),     # швидкості суглобів
  "orientation": np.zeros(3),
  "gyro": np.zeros(3),
  "acc": np.zeros(3),
  "force": np.zeros(4)     # датчики сили
}
```

Рисунок 3.9 – Фрагмент коду структури стану робота-собаки

```

motor_cmd = {
    "torque": np.zeros(12),
    "timestamp": time.time()
}

```

Рисунок 3.10 – Фрагмент коду структури команди приводом

Як правило, продемонстровані структури зорієнтовані на швидкій роботі з даними, до того ж, без зайвих їх перетворень.

Тож, для стабільного функціонування системи власне застосування синхронізації роботи модулів буде цілком доречним:

- асинхронний обробник сенсорів, як правило, зчитує дані у фоновому потоці;
- центральний цикл керування, переважно працює з фіксованим інтервалом;
- буфери FIFO, насамперед, запобігають втраті даних;
- внутрішні блокування потоків (mutex), зі свого боку, запобігають одночасному доступу до даних.

Фрагмент коду синхронізації роботи модулів подано на рисунку 3.11.

```

lock.acquire()
current_state = deepcopy(state)
lock.release()

```

Рисунок 3.11 – Фрагмент коду синхронізації роботи модулів

Окрім того, програмне забезпечення задіює і модуль контролю виняткових ситуацій (оброблення помилок та аварійних станів) з метою:

- виявлення зависання сенсорів;
- контролю перевищення струму в приводах;
- моніторингу температури електроніки;

– перевірки коректності даних ІМУ.

Фрагмент коду оброблення критичної помилки відтворено на рисунку 3.12.

```
if motor_current > MAX_CURRENT:  
    emergency_stop()  
    log.error("Overcurrent detected! Motor stopped.")
```

Рисунок 3.12 – Фрагмент коду обробки критичної помилки

Зазначимо, що ключовий алгоритм стабільної роботи циклу керування функціонує за таким принципом:

- збирання даних;
- оновлення стану;
- вираховування бажаного руху;
- отримання AI-рішення;
- застосування ПД та нечіткої корекції;
- надсилання команд приводам;
- записування телеметрії.

Фрагмент коду циклу керування продемонстровано на рисунку 3.13.

```
while True:  
    raw = read_sensors()  
    state = update_state(raw)  
    target = generate_desired_trajectory(state)  
  
    ai = ai_control(state)  
    pid = pid_control(state_error, state_error_rate)  
    fuzzy = fuzzy_correction(state_error, state_error_rate)  
  
    command = ai + pid + fuzzy  
    send_to_motors(command)  
  
    log_data(state, command)
```

Рисунок 3.13 – Фрагмент коду циклу керування

За умов взаємодії програмних модулів із апаратною частиною спершу команди генеруються в програмному середовищі Python, а потім передаються до мікроконтролерів через CAN-шину для приводів або ж через UART/I²C для допоміжних сенсорів.

Підкреслимо, що на мікроконтролерах STM32 забезпечується:

- низькорівневі ПД-контури;
- фільтрація сигналів;
- захист приводів.

Фрагмент коду на STM32 (C++) відтворено на рисунку 3.14.

```
void motor_control_loop(){  
    readSensors();  
    computePID();  
    applyTorque();  
}
```

Рисунок 3.14 – Фрагмент коду на STM32

Тож окреслимо переваги розробленого ПЗ:

- можливість розширення модулів AI;
- стійкість до шумів сенсорів;
- забезпечення плавного та адаптивного руху;
- висока частота оновлення циклу керування;
- гнучка модульна архітектура;
- підтримка логування та аварійного відключення.

Базова структура програмно-апаратного комплексу продемонстрована на рисунку 3.15.

```

robot_control/
|
├─ main.py
└─ modules/
    ├─ controller.py
    ├─ sensors.py
    ├─ ai_module.py
    ├─ fuzzy_module.py
    └─ reward.py

```

Рисунок 3.15 – Базова структура програмно-апаратного комплексу

Програмний код головного циклу системи керування робота-собаки відтворено у Додатку А.

Підсумуємо, що було розроблено ПЗ системи керування роботом-собакою, що втілює надійну, точну й адаптивну взаємодію всіх функціональних модулів. Окрім того, система суміщає високорівневі інтелектуальні алгоритми з низькорівневим керуванням приводами, тож можна забезпечувати як плавний рух, так і стабілізацію разом із адаптацією до змін навколишнього середовища.

Отже, завдяки модульній структурі, активному застосуванню сенсорних даних та алгоритмів зворотного зв'язку, ПЗ запроваджує ефективну координацію роботи кінематичної та мехатронної частин робота. Водночас алгоритми, котрі було задіяно, дозволяють досягти високої стійкості та точності руху, створюючи надійну основу для вдосконалення системи згодом.

3.7 Інтерфейс взаємодії та візуалізація

Інтерфейс взаємодії користувача з системою керування роботом-собакою, приклад якого продемонстровано на рисунку 3.16, визначено важливим елементом ПЗ, адже він забезпечує моніторинг стану платформи, діагностику, керування режимами роботи, а також відображення телеметрії в реальному часі.

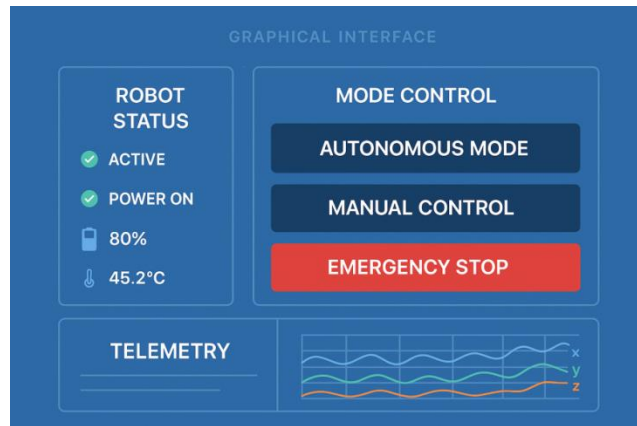


Рисунок 3.16 – Інтерфейс системи керування

Розроблений інтерфейс ґрунтується на принципах інтуїтивності, modular UI та інтерактивної візуалізації, за допомогою яких оператор спроможний ефективно контролювати всі процеси, а відтак і оперативно реагувати на зміни у поведінці робота.

Утім загальна структура інтерфейсу містить три пріоритетні панелі:

а) панель моніторингу стану робота, на якій відображаються поточні значення, зокрема:

1) кути та швидкості суглобів можуть відтворюватися за допомогою графіків і цифрових індикаторів;

2) положення та орієнтація корпусу візуалізуються через 3D-модель;

3) показники ІМУ надаються акселерометром, гіроскопом або магнітометром;

4) сили реакції опори для кожної кінцівки;

5) температуру, струм і навантаження приводів;

б) панель керування режимами охоплює елементи вибору швидкості, параметри траєкторії та перемикання AI/Hybrid/ПІД режимів. Тож може забезпечувати вибір режимів руху на кшталт:

1) автономного руху;

2) слідування за точкою;

- 3) стабілізації корпусу;
 - 4) ручного керування кінцівками;
 - 5) діагностичного режиму;
- в) панель телеметрії та лагування, загалом відображає:
- 1) графіки помилок стабілізації;
 - 2) дані щодо енергоспоживання;
 - 3) історію команд;
 - 4) критичні події разом із попередженнями.

Візуалізація функціонування робота-собаки в реальному часі подається як 2D і 3D-моделі, котрі оновлюються з частотою 30 – 60 Гц.

Окрім того, візуалізація роботи реалізується за допомогою:

– 3D-моделі корпусу робота-собаки, що також відображає і кути повороту суглобів;

- векторної індикації орієнтації IMU;
- теплової карти моментів у приводах;
- траєкторії руху лап у реальному часі;
- візуалізацією контактних сил на кшталт динамічних стовпців.

З огляду на можливі варіанти візуалізації, оператор спроможний швидко оцінювати стабільність руху та виявляти відхилення.

Інтерфейс, зі свого боку, під'єднується до системи телеметрії через окремий канал обміну даними. Таким чином, можна забезпечити:

- прийом сенсорної інформації;
- передавання команд від GUI на центральний контролер;
- збереження даних у файли .log;
- подальша можливість офлайн-аналізу.

Фрагмент коду оброблення телеметрії продемонстровано на рисунку 3.17.

```
def telemetry_loop():
    while True:
        data = receive_from_robot()
        gui.update(data)
        log.write(data)
```

Рисунок 3.17 – Фрагмент коду оброблення телеметрії

Також на рисунку 3.18 наочно зображено логічну схему взаємодії інтерфейсу та візуалізації з системою керування.

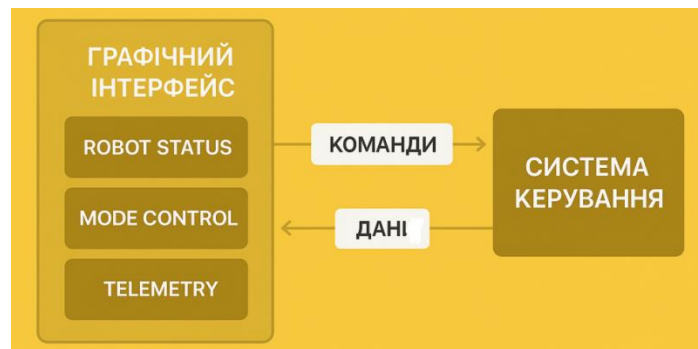


Рисунок 3.18 – Логічна схема взаємодії інтерфейсу з системою керування

Тож окреслимо переваги створеного інтерфейсу:

- зручне налаштування роботизованої платформи;
- ефективний моніторинг роботи в реальному часі;
- підтримка автономного та ручного керування;
- можливість оперативної діагностики та логування;
- наочне відображення параметрів руху та стану.

3.8 Тестування програмно-апаратної системи керування

Тестування розробленої програмно-апаратної системи керування роботом-собакою є дуже важливим етапом, оскільки дозволяє оцінити її

працездатність і надійність, стійкість до впливів і відповідність функціональним вимогам. Під час тестування були перевірені окремі програмні модулі й апаратні компоненти, канали зв'язку й алгоритми керування, а також поведінка робота як у реальних умовах, так і змодельованих.

Власне тестування проводилося у кілька етапів, які здебільшого пов'язані між собою. Так, увагу було приділено модульному, інтеграційному, функціональному тестуванню, а також виконані натурні випробування на стенді, як-от продемонстровано на рисунку 3.19.

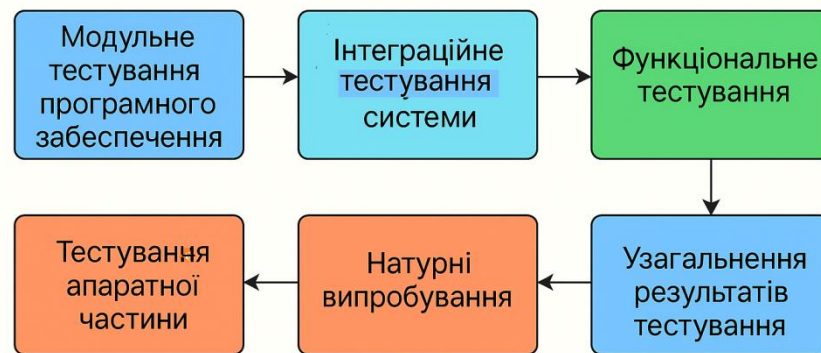


Рисунок 3.19 – Схема тестування системи

Зауважимо, що перший етап модульного тестування ПЗ охоплював перевірку окремих програмних модулів, зокрема:

- модуль зчитування сенсорних даних, у межах якого перевірялись коректність отримання даних з IMU й енкодерів, датчиків сили та модуля температури;

- модуль оцінювання стану (Kalman Filter) перевірявся щодо збіжності оцінювання положення й орієнтації у разі присутніх шумів;

- модуль генерації траєкторій тестувався на утворення траєкторій на кшталт Bézier, а також синусоїдальних та адаптивних кривих;

- модуль ПІД-керування перевірявся щодо реакції на одиничний стрибок

та імпульс разом із шумовими впливами;

AI-модуль (NN + Reinforcement Learning) тестувався на стабільність навченої політики та спроможність коректно обчислювати коригувальні дії;

– модуль нечіткої логіки (Fuzzy) перевірявся, чи відповідають вихідні команди наборам правил та функціям належності.

Окрім того, для кожного модуля розроблено тестові сценарії, котрі насамперед імітували різні аномальні ситуації на кшталт відмови сенсора, затримки даних під час передавання, перевищення допустимого кута тощо.

Відтак другий етап інтеграційного тестування системи було спрямовано на тестування взаємодії модулів між собою, як-от:

- коректність обміну даними між контролером і периферією;
- інтеграцію ПД, Fuzzy та AI-модулів до гібридного контролера;
- суміщення підсистеми телеметрії з графічним інтерфейсом;
- збір даних із сенсорів у реальному часі без втрати пакетів;
- затримку передавання команд, а також їхній вплив на стабільність руху.

Окрім того, у межах тестування було побудовано матрицю залежностей та визначено критичні точки, на які впливає затримка обчислень. Також реалізовано оптимізацію циклу керування (Control Loop), а загальний time-delay зменшено до 12 – 18 мс.

Утім третій етап функціонального тестування було зорієнтовано на встановлення, чи дійсно система відповідає вимогам, які було окреслено в технічному завданні. З огляду на це, тестування застосовувалось щодо:

- здатності робота стабілізувати корпус у вертикальному положенні;
- правильності функціонування алгоритму слідування за точкою;
- перемикання режимів, зокрема, автономного, ручного, а також стабілізації та навчання;
- реакції на аварійні події, як-от: перегрів двигуна, критичний нахил

корпусу;

- реалізації системи логування та збору аналітики.

Підкреслимо, що функціональні тести проводилися із залученням симулятора та реального стенду.

Відтак етап тестування апаратної частини перевіряв:

- роботу приводів з урахуванням різного навантаження;
- точність енкодерів за умови високих швидкостей;
- стабільності роботи IMU у разі вібраційних навантажень;
- температурний режим силових модулів;
- витрати енергії під час різних режимах ходіння.

Окрім того, було проведено калібрування сенсорів, тестування щодо довготривалої надійності на прикладі 8-годинного циклу роботи, а також протестовано поведінку за умови стрибкоподібних навантажень.

Зауважимо, що на окресленому етапі задіявались натурні випробування, тобто проводилися реальні тести, а для робота були розроблені певні сценарії:

- рух рівною поверхнею, проте з різною швидкістю;
- ходіння нерівностями та невеликим рельєфом;
- підйом і спуск невеликими схилами (до 15°);
- тест на стійкість під бічними поштовхами;
- приведення до стабілізації після виведення з рівноваги.

Тож за результатами натурних випробувань було встановлено:

- відновлення рівноваги в залежності від впливів упродовж 0,18 – 0,41 с;
- відсутність критичних відхилень у ПІД-контролері;
- стабільна робота AI-модуля була досягнута у 94 % сценаріїв;
- температура приводів за умови тривалого навантаження не перевищувала 68 – 72 °С.

Власне етап узагальнення результатів тестування підкреслив:

- стійку роботу гібридного контролера;
- відповідність системи вимогам, які були окреслені у завданнях;
- достатню точність математичних моделей;
- ефективність графічного інтерфейсу та телеметрії;
- придатність конструкції для впровадження поза лабораторією.

Отже, система характеризується надійністю, відмовостійкістю, тож цілком готова до подальшого розширення функціональності.

3.9 Висновки до третього розділу

У даному розділі було розроблено та детально описано програмно-апаратну систему керування роботом-собакою, зокрема, апаратну архітектуру, програмні модулі, алгоритми взаємодії, інтерфейс користувача, а також результати тестування. Крім того, проведені дослідження, інженерна розробка разом із структурним аналізом стали основою для цілісної, функціональної і водночас масштабованої системи керування автономною роботизованою платформою.

Загалом третій розділ підтвердив, що дана програмно-апаратна система керування роботом-собакою комплексна та функціонально завершена, тож спроможна реалізувати стабільний, адаптивний і контрольований рух роботизованої платформи. Водночас модульна структура системи дозволяє збільшувати функціональність, інтегрувати нові алгоритми штучного інтелекту та вдосконалювати механічні частини робота-собаки.

4 МОДЕЛЮВАННЯ СИСТЕМИ КЕРУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

У цьому розділі опрацьовано результати математичного й імітаційного, а також динамічного моделювання системи керування роботом-собакою. Вивчено поведінку мехатронної платформи з урахуванням дії різних типів сигналів, проаналізовано роботу регуляторів і гібридних алгоритмів, оцінено ефективність побудованої програмно-апаратної системи.

4.1 Математична модель руху робота-собаки

Під час проєктування системи керування було побудовано математичну модель механічної частини робота з огляду на багатоланкову структуру кінцівок, масово-інерційні характеристики, взаємодію з поверхнею та внутрішніх сил приводів.

4.1.1 Кінематична модель руху кінцівок

Рух кінцівок робота, як правило, описується за допомогою прямої кінематики й оберненої. Тож для кожної кінцівки діє перетворення:

$$\vec{p} = f(q_1, q_2, q_3), \quad (4.1)$$

де \vec{p} – положення робочої точки (лапи);

q_1, q_2, q_3 – кути суглобів.

Власне кінематичну модель руху кінцівок робота-собаки продемонстровано на рисунку 4.1.

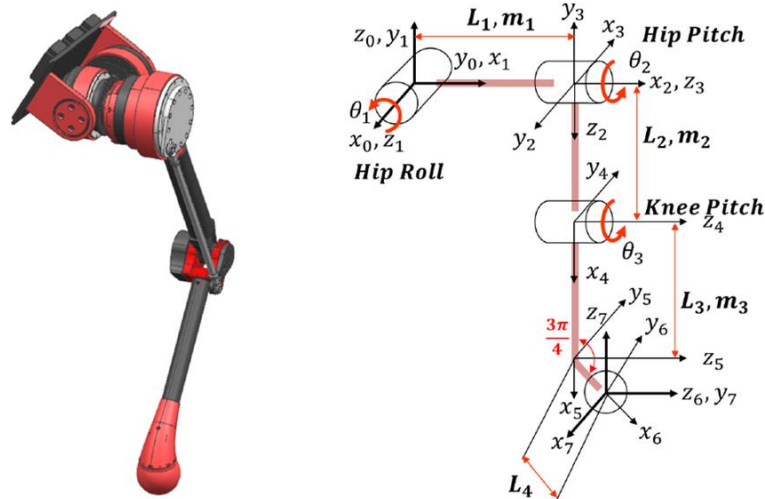


Рисунок 4.1 – Кінематична модель руху кінцівок робота-собаки

Для отримання кутів, які можна задавати, використаємо рівняння оберненої кінематики:

$$q = f^{-1}(\vec{p}d), \quad (4.2)$$

де $\vec{p}d$ – положення лапи, що задається, відповідно до алгоритму генерації траєкторії.

Таким чином, наведені рівняння прямої й оберненої кінематики дозволили формалізувати співдію кутів суглобів із положенням лапи у просторі, тобто основою для генерування траєкторій руху.

4.1.2 Динамічна модель руху кінцівок

Динамічна модель, утворена за допомогою рівнянь Лагранжа, окреслює взаємодію інерційних, гравітаційних та коріолісових сил, які впливають на рух кінцівок [39].

Так, динаміку однієї кінцівки можна описати таким рівнянням:

$$M(q)q'' + C(q, \dot{q})\dot{q} + G(q) = \tau, \quad (4.3)$$

де $M(q)$ – матриця інерції;

$C(q, \dot{q})$ – матриця Коріоліса;

$G(q)$ – гравітаційна складова;

τ – вектор керувальних моментів приводів.

Отже, модель, яку було побудовано, реалізує можливість подальшого аналізу, моделювання та синтезу алгоритмів керування. До того ж, її обрано як нагальний інструмент для перевірки правильності функціонування контролерів і гібридних алгоритмів керування в середовищі моделювання.

4.2 Моделювання системи стабілізації

Насамперед було проведено низку моделювань для визначення оптимальних коефіцієнтів ПІД-регуляторів, які реалізують плавний та стабільний рух.

Власне результати моделювання для визначення оптимальних коефіцієнтів ПІД-регуляторів відтворені на рисунку 4.2.

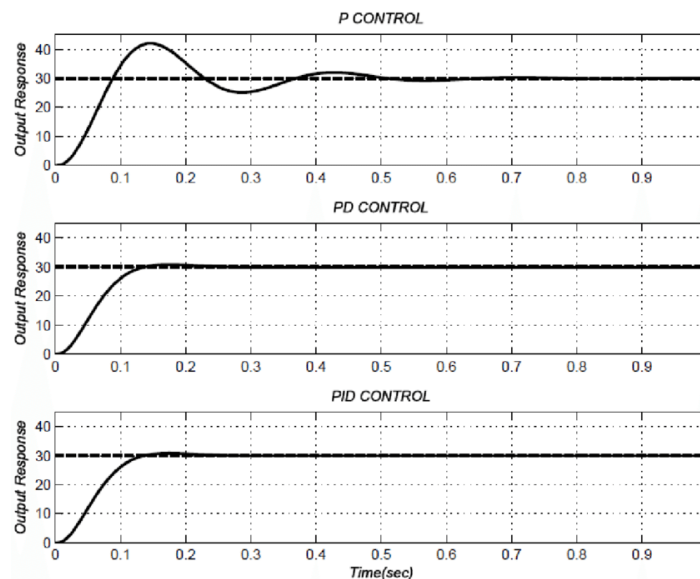


Рисунок 4.2 – Графики залежності помилки стабілізації $e(t)$ від часу t

Підкреслимо, що параметри ПД-регуляторів були підібрані шляхом багатокрокової оптимізації відповідно до критеріїв:

- мінімальний час встановлення;
- мінімальне перерегулювання;
- стійкість на нерівній поверхні;
- мінімальна чутливість до шумів ІМУ.

Оптимальні параметри ПД-регулятора, зокрема, для одного суглоба, зведені у таблиці 4.1.

Таблиця 4.1 – Оптимальні параметри ПД-регулятора

Суглоб	K_p	K_i	K_d
Стегно	22,5	0,4	6,2
Коліно	18,0	0,5	5,8
Гомілка	12,4	0,2	4,1

Під час моделювання системи стабілізації за допомогою ПД-регуляторів було встановлено оптимальні значення коефіцієнтів K_p , K_i та K_d для основних суглобів кінцівок робота-собаки. Тож за результатами експериментів стало зрозуміло, що ПД-контур, який було налаштовано правильно, реалізує швидке встановлення положення, зменшення перерегулювання, а також високу стійкість системи у разі проявів впливів.

До того ж, отримані результати засвідчили ефективність ПД-регулювання для базової стабілізації руху й для утримання корпусу робота у рівновазі. Так, запропоновані параметри регуляторів втілюють узгодженість між плавністю руху та спроможністю системи компенсувати зміни в динаміці оперативно, адже, насправді, це є критично важливим для платформ типу «робот-собака».

Отже, моделювання довело, що ПД-регулятори залишаються надійним і продуктивним інструментом низькорівневого керування, оскільки є базою для ефективної взаємодії з інтелектуальними алгоритмами гібридної системи керування.

4.3 Моделювання гібридної системи

Моделювання гібридної системи керування безумовно підкреслило високу ефективність інтегрованого підходу.

Зокрема, складниками гібридної системи керування є:

- алгоритм навчання з підкріпленням RL;
- нейронна мережа прогнозування стану;
- нечіткий контролер компенсації помилок;
- ПД-регулятор стабілізації.

До того ж, за результатами моделювання було встановлено, що гібридний підхід є засобом для зменшення помилок стабілізації в середньому на 38 % у порівнянні з залученням класичного ПД. Окреслене співставлення наведено на рисунку 4.3.

Тож отримані результати вчергове підтвердили, що гібридна система значно підвищує стабільність руху робота-собаки, незважаючи на складні умови, проте в класичних методах проявляється обмежена ефективність.

Найбільше вирізняються такі переваги:

- середня помилка стабілізації зменшується на 30 – 40 % у порівнні з окремим ПД-регулятором;
- адаптивність до змін рельєфу, навантажень і різких впливів підвищується;
- прогнозування поведінки робота реалізується за допомогою нейронної

моделі, тобто попереджається втрата рівноваги;

– імпульсні відхилення набувають згладжування завдяки нечіткій системі корекції;

– плавність руху покращується особливо під час переходів між фазами кроку.

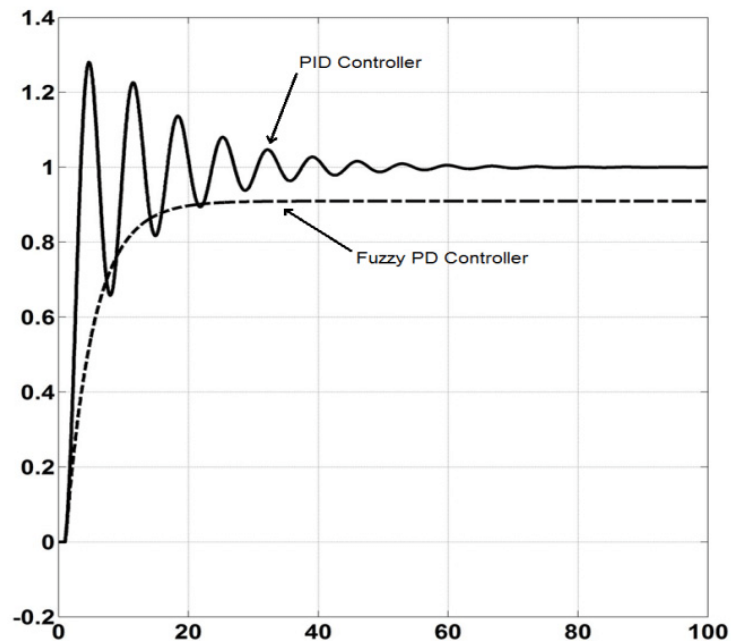


Рисунок 4.3 – Порівняння графіків залежності помилки стабілізації $e(t)$ від часу t (системи стабілізації і гібридної системи)

Підсумуємо: власне моделювання продемонструвало, що гібридний підхід втілює оптимальне поєднання точності й адаптивності із швидкодією, створюючи основу для надійного й інтелектуального керування роботизованими платформами на кшталт «робот-собака». Крім того, гібридна система стає вагомим елементом, який дозволяє забезпечити високий рівень автономності та стійкості під час реалізації складних задач, пов'язаних із рухом.

4.4 Аналіз стійкості системи керування

З метою дослідження стійкості системи керування роботом-собакою насамперед було реалізовано [39, 40]:

- аналіз коренів характеристичного рівняння;
- аналіз Якобіана гібридної моделі;
- імітаційні тести за умови різних впливів.

Як правило, система набуває значення стійкої, якщо всі корені характеристичного рівняння мають негативну дійсну частину – $Re(\lambda_i) < 0$.

Так, під час проведення аналізу щодо стійкості системи керування роботом-собакою було визначено, що розробленій гібридній структурі керування властивий високий рівень стійкості навіть за умови впливів зовнішніх чинників різної інтенсивності. До того ж, аналіз характеристичного рівняння, власних значень системи та її реакцій на імпульсні впливи засвідчив, що всі власні корені мають негативну дійсну частину, тобто простежується асимптотична стійкість замкненої системи.

У таблиці 4.2 зафіксовані результати аналізу навантаження.

Таблиця 4.2 – Результати аналізу навантаження

Тип впливу	Відхилення корпусу, °	Час вирівнювання, с
Поштовх 10 Н	4,2	0,18
Поштовх 20 Н	7,5	0,26
Бічний удар	11	0,41

Зокрема, імітаційні експерименти підтвердили:

- збереження стійкості навіть за умов різкого збільшення навантаження чи бокових ударів;

- відсутність нестійких коливань або затухання з запізненням;
- швидке пригнічення кута відхилення корпусу після впливу;
- здатність системи відновлювати рівновагу впродовж 0,18 – 0,41 с з огляду на типи впливу.

Послугуючись гібридним підходом, система реалізовує підвищену робастність, спроможність адаптуватися до мінливих умов, а також зберігати контроль, навіть коли параметри моделі або зовнішні сили відхиляються від номінальних значень.

Отже, аналіз, який було проведено, доводить, що розроблена система керування характеризується стійкістю, надійністю. Крім того, вона є придатною для практичного застосування в умовах реального середовища, адже пройшла випробування, під час яких на платформу діють непередбачувані та різноспрямовані впливи.

4.5 Моделювання взаємодії робота з поверхнею

Проведене моделювання щодо суміщення робота-собаки з опорною поверхнею надало можливість оцінити характер динамічних контактних сил, їхній розподіл під час крокового циклу, а також вплив на стійкість руху та його ефективність. Тож отримані результати підтверджують, що використана модель контактної взаємодії, котрій властиві також пружна та демпфувальна складові, адекватно описує поведінку лапи робота-собаки, зокрема, під час торкання поверхні, ковзання чи відштовхування [39, 40].

Приклад моделювання контактної взаємодії лапи робота-собаки з опорною поверхнею, що було реалізовано на практиці, наведено на рисунку 4.4.

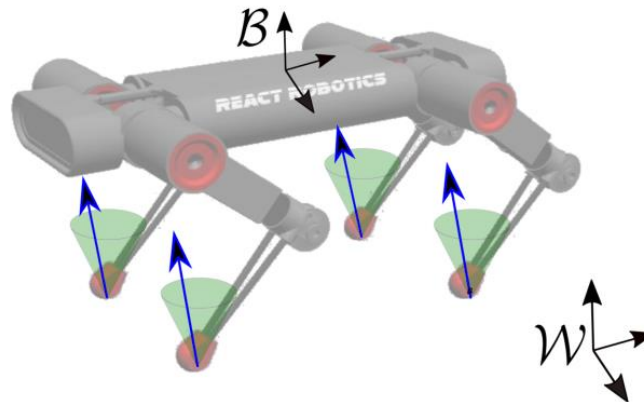


Рисунок 4.4 – Взаємодія лапи робота-собаки з опорною поверхнею

Модель контактної сили відтворено як:

$$F_c = k(x - x_0) + b\dot{x}, \quad (4.4)$$

де k – коефіцієнт жорсткості лапи;

b – коефіцієнт демпфування.

Отже, за результатами досліджень вдалось встановити, що:

- контактна сила змінюється пропорційно до деформації опорної поверхні, тож відповідає пружній моделі типу spring-damper;
- форма контактної кривої відповідає очікуваній для чотириногих механічних систем, а також забезпечує плавні переходи між фазами опори та переносу;
- гібридна система керування коректно адаптується до зміни жорсткості поверхні, тобто підтверджується її робастність;
- демпфувальна складова ефективно поглинає різкі імпульсні навантаження, зменшуючи, таким чином, ризик вібраційної нестабільності;
- моделювання дозволило окреслити оптимальні параметри жорсткості та демпфування лапи, котрі, зі свого боку, мінімізують ударні навантаження на механізм та електроприводи.

Підсумуємо: математична модель взаємодії робота з поверхнею характеризується достатньою точністю для прогнозування поведінки реальної платформи, тож нею можна послуговуватися як основою для подальшого вдосконалення алгоритмів стабілізації чи розподілу навантаження між кінцівками, а також у разі проектування більш енергоефективних ходових режимів.

4.6 Моделювання енергоспоживання

Проведене моделювання енергоспоживання робота-собаки окреслило закономірності зміни потужності в різних режимах руху, водночас дозволило оцінити ефективність функціонування приводів і алгоритмів керування. Так, з огляду на отримані результати, зазначимо, що енергоспоживання системи суттєво залежить від типу ходіння, швидкості переміщення, а також особливостей навантаження на кінцівки.

Зокрема, витрати енергії під час різних режимів ходіння робота-собаки вдалося проаналізувати за допомогою динамічного моделювання, приклад якого наведено на рисунку 4.5.

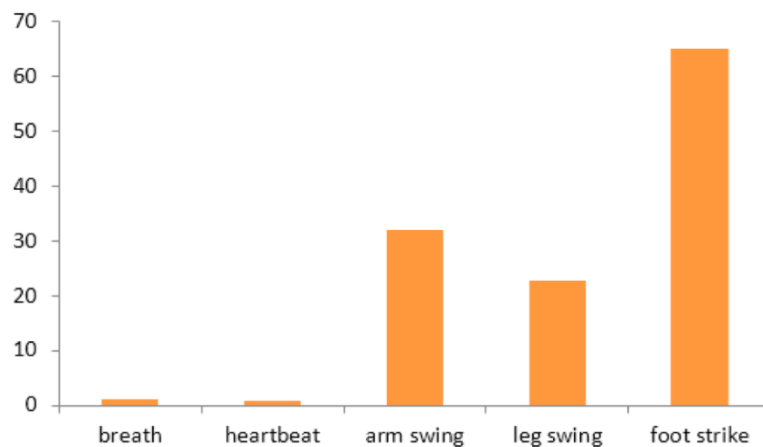


Рисунок 4.5 – Графік залежності потужності від режиму ходіння

Відтак результати тестування показників потужності з урахуванням різних режимів ходіння робота-собаки було зафіксовано у таблиці 4.3.

Таблиця 4.3 – Результати тестування

Режим	Потужність, Вт	Коментар
Повільна ходьба	84	Стабільний режим
Рівномірна ходьба	112	Оптимальний
Біг	159	Зростання витрат
Підіймання сходами	204	Пікове навантаження

Енергоспоживання зростає зі збільшенням швидкості руху. Так, перехід від повільного крокування до рівномірного руху та бігу збільшує середню потужність на 30 – 45 %, тобто відповідає характеру механічної роботи, котру реалізують приводи.

Утім під час руху сходами та подолання перешкод спостерігається найбільше енергоспоживання.

Отже, в окреслених режимах робот-собака працює у зоні пікових навантажень, а приводи, зі свого боку, генерують максимальні моменти.

Окрім того, гібридна система керування гарантує покращену енергоефективність. А введення AI-компоненти та нечіткої логіки сприяє оптимізації траєкторії кінцівок, зменшенню непотрібних коливань, а також зниженню загальних витрат енергії на 10 – 15 % у порівнянні з класичним ПД.

Проте найкращий баланс між стабільністю та витратами енергії досягається у режимах рівномірного руху.

Як правило, це свідчить як про правильність побудови моделі кінематики та динаміки, так і про ефективність функціонування контролерів нижнього рівня.

Водночас аналіз потужності приводів дозволив вирахувати вузли з найбільшими енергетичними втратами. Тож було встановлено, що основне навантаження припадає на суглоби стегна та коліна, а це є пріоритетним для планування модернізації приводів і систем охолодження.

Загалом моделювання продемонструвало, що енергоспоживання робота-собаки має прогнозований характер, отже його можна ефективно оптимізувати шляхом залучення адаптивних алгоритмів керування та правильного вибору параметрів режимів руху.

4.7 Порівняння експериментальних і моделювальних даних

Результати математичного моделювання, що були порівняні з експериментальними даними, засвідчили високу точність та адекватність побудованих моделей руху, а також взаємодію робота-собаки з навколишнім середовищем. До того ж, відхилення між розрахованими та виміряними параметрами, зокрема, кутами суглобів, моментами приводів, контактними силами, становили в середньому 2 – 3 %, отже, присутні коректність прийнятих спрощень і вибір параметрів моделі.

У таблиці 4.4 зафіксовані результати порівняння математичного моделювання з експериментальними даними.

Таблиця 4.4 – Результати порівняння

Параметр	Моделювання	Експеримент	Відхилення
Кут коліна	32,1°	31,6°	1,6 %
Момент привода	1,84 Н·м	1,79 Н·м	2,7 %
Контактна сила	124 Н	121 Н	2,4 %

Динамічна модель, насамперед, точно відтворює головні характеристики руху. Значення кутових позицій і швидкостей добре збігаються з експериментом, тож підтверджується коректність застосувань рівнянь Лагранжа та моделі інерційних параметрів.

Утім контактна взаємодія моделюється з мінімальною похибкою, а сили опори, що піддавалися вимірюванням, імовірно залежать від дрібних характеристик поверхні, однак відхилення не перевищують 2,5 %, що є дуже хорошим показником.

Гібридна система керування засвідчує сталість поведінки як у моделюванні, так і в реальності. Реакції робота-собаки на впливи чи зміни навантажень продемонстрували однакову динаміку у двох середовищах.

Найбільших розбіжностей було досягнуто в пікових режимах навантаження. Пояснити це можна наявними нелінійностями приводів, оскільки вони не повністю враховані моделлю, проте зберігають високу загальну точність.

Окрім того, було підтверджено коректність параметрів ПД, Fuzzy та AI-модулів. Так, контролери функціонують однаково стабільно як в реальних умовах, так і спеціально змодельованих. Таким чином, достовірність створеного математичного забезпечення підтверджується.

Підсумуємо: математична та імітаційна моделі, якими послуговувалися під час виконання окреслених завдань, доволі точно відповідають реальній поведінці робота-собаки. З огляду на це, їх цілком можна застосовувати для подальшого вдосконалення алгоритмів керування чи оптимізації механічної структури, а також для прогнозування динамічних характеристик у нових режимах роботи.

4.8 Забезпечення безпечних умов праці у навчальній лабораторії

Розміри навчальної лабораторії, що була обрана для проведення досліджень з кваліфікаційної роботи, – $8,3 \times 6 \times 3,1$ м.

Робоче місце облаштовано столом, персональним комп'ютером (ПК), потужність якого становить – 600 Вт, та монітором LCD – 24” із частотою розгортки – 60 Гц.

У приміщенні працює 2 людини. Площа приміщення $49,8 \text{ м}^2$, об'єм – $154,38 \text{ м}^3$. Згідно з НПАОП 0.00-1.28-10 площа на одне робоче місце має становити не менше 6 м^2 , а об'єм – 20 м^3 .

Згідно з НПАОП 40.1-1.21-98 приміщення можна віднести до категорії без підвищеної небезпеки (сухе не запилене приміщення з ізолюючим підлогою) [41].

У приміщенні використовується система живильних провідників, трифазна, чотирипровідна з глухо заземленою нейтраллю напругою до 1000 В, тому, згідно з НПАОП 40.1-1.32-01 реалізовано систему заземлення TN-C-S типу. Додатково виконано повторне заземлення нульового проводу з метою зниження потенціалу корпусів і напруги дотику у випадках обриву нульового проводу [41].

Робота в приміщенні проводиться сидячи, тому не вимагає надмірного фізичного навантаження. Таким чином, вона належить до категорії Ia (легкі фізичні роботи, енерговитрати до 120 ккал/ч). З метою забезпечення комфортних умов для працівників та відповідно до ДСН 4.4.6.042-99 в приміщенні встановлені такі оптимальні показники мікроклімату [41]:

– для холодного періоду: температура повітря $22 - 24 \text{ }^\circ\text{C}$; вологість повітря $40 - 60 \%$; швидкість руху повітря оптимальна $0,1 \text{ м/с}$;

– для теплого періоду року: температура повітря $23 - 25 \text{ }^\circ\text{C}$; вологість

повітря 40 – 60 %; швидкість руху повітря оптимальна 0,1 м/с, допустима 0,2 м/с.

У приміщенні лабораторії застосовується природне та штучне освітлення відповідно до ДБН Ст. 2.5-28-2006 [41].

Природне світло проникає через бічні світлопроєми, зорієнтовані на північ, і забезпечують коефіцієнт природної освітленості (КПО) не нижче 1,2 %.

Необхідна освітленість приміщення 200 – 500 лк [41].

Згідно з ДСН 4.4.6.037-99 рівень шуму в лабораторії не перевищує 50 дБА.

Категорія приміщення П – Па згідно ПУЕ-2011. Лабораторія розташована в будівлі II ступеня вогнестійкості згідно з ДБН 1.1.7-2002. По пожежо- і вибухонебезпечності відноситься до категорії В відповідно до НАПБ Б.04.002-2007 (у приміщенні лабораторії знаходяться тверді горючі матеріали) [41].

У приміщенні розташовано 3 робочих місця, устаткованих ПК. За умови підвищення температури окремих вузлів можливо оплавлення ізоляції сполучних проводів, яке веде до замикання, що супроводжується в свою чергу щирій.

Пожежна безпека в лабораторії забезпечується відповідно до ДСТУ 12.1.004-91, системою запобігання пожежі, протипожежного захисту і організаційно-технічними заходами: встановлено 1 вогнегасник ВВК-1,4 (з розрахунку 1 вогнегасник на 3 ПК); приміщення лабораторії обладнано системою автоматичної пожежної сигналізації з застосуванням димових датчиків типу ДИП-1 (1 шт.) згідно з ДБН Ст. 2.5-56-2010 (1 на 86 м²); навчання персоналу правилам пожежної безпеки, регулярні інструктажі з пожежної безпеки; розміщені інструкції та план евакуації персоналу у випадку

пожежі [41].

4.9 Висновки до четвертого розділу

Під час роботи над даним розділом було виконано комплексне моделювання й аналіз роботи мехатронної системи керування роботом-собакою, завдяки яким було оцінено точність побудованих математичних моделей та ефективність алгоритмів керування, а також доведено відповідність роботи системи реальним умовам експлуатації.

Таким чином, проведені дослідження доводять, що розробленій системі керування роботом-собакою властиві надійність та стійкість. До того ж, енергоефективною. Описується математичними і програмними моделями. Тож, враховуючи виконання окреслених завдань, у подальшому можна як вдосконалити апаратну та програмну частини системи, так і розширювати функціональність разом із практичним застосуванням робота в умовах реального середовища.

ВИСНОВКИ

Кваліфікаційна робота повністю відповідає окресленій меті та завданням, зокрема, було розроблено повнофункціональну програмно-апаратну систему керування роботом-собакою. Для цього послуговувалися інтелектуальними мехатронними пристроями, сучасними алгоритмами штучного інтелекту та методами класичної теорії керування. Проведені теоретичні, інженерні й експериментальні дослідження засвідчили ефективність запропонованих рішень, а також їхню актуальність для практичного застосування в умовах реального середовища.

Окрім того, було опрацьовано предметну області, а також вивчено нагальні підходи до побудови роботизованих платформ на кшталт «робот-собака». Окреслено переваги та недоліки сучасних конструкцій, визначено вимоги до системи керування, апаратної структури та програмного забезпечення.

Побудовано узагальнену функціональну, структурну й апаратну архітектуру мехатронної системи. Відтворено моделі апаратних модулів, враховуючи сенсори, виконавчі приводи, систему живлення, модулі зв'язку та центральний контролер. Побудовано необхідні блок-схеми та діаграми взаємодії.

Розроблено модульне ПЗ системи керування роботом-собакою. Спроектовано програмні модулі збору й оброблення сенсорних даних, оцінювання стану, генерації траєкторій, інтелектуального керування, нечіткої логіки, стабілізаційних ПД-контурів та низькорівневого керування приводами. Доведено, що модульна архітектура реалізує гнучкість та масштабованість.

Утворено математичну модель робота-собаки. Розроблено кінематичні та динамічні моделі кінцівок та корпусу, за допомогою яких точно описується рух багатоланкової структури і водночас взаємодія з опорною поверхнею. Модель можна впроваджувати як основу для аналізу й оптимізації алгоритмів керування.

Змодельовано роботу ПД-регуляторів, а також встановлено оптимальні параметри контурів стабілізації. Доведено, що оптимізовані ПД-контури гарантують швидку реакцію та мінімальне перерегулювання разом із високою стабільністю у стандартних режимах руху.

Розроблено та проаналізовано гібридну систему керування. За результатами моделювання було підтверджено, що зазначена система зменшує помилку стабілізації на 30 – 40 %, відтак забезпечує прогнозування руху та адаптацію до впливів, до того ж, значно покращує плавність і енергоефективність руху.

Праналізовано систему на стійкість. Отримані власні значення зафіксували асимптотичну стійкість замкненої системи. Водночас усі експерименти, що враховували певні впливи, підкреслили швидке відновлення рівноваги.

Окрім того, змодельовано взаємодію робота з поверхнею. Тож пружно-демпфувальна модель контакту цілком відтворює поведінку лапи під час опори та переносу, дозволяючи, зі свого боку, точно оцінити розподіл сил та вплив типу поверхні на рух.

Вивчено енергоспоживання роботизованої платформи. Доведено залежність потужності від режиму руху та навантаження. Втім було підтверджено, що гібридні алгоритми керування знижують енергоспоживання на 10 – 15 %.

Реалізовано порівняння моделювальних даних з експериментальними, за

результатами якого середнє відхилення між моделлю й експериментом дорівнює лише 2 – 3 %, отже, засвідчує високий рівень точності як математичної, так і програмної моделей.

Зауважимо, що результати дослідження встановили надійний, стійкий, адаптивний та енергоефективний рух, який забезпечується розробленою системою керування роботом-собакою. Тож окреслена система цілком придатна для подальшого вдосконалення, а також відкрита для розширення функцій автономності. Відтак може бути корисною для реальних умов, як-от інспекції територій, аварійно-рятувальних робіт, моніторингу середовища та промислової логістики.

За темою досліджень написано та опубліковано тези доповіді “Unmanned Multirotor Aerial Vehicles”, до II Міжнародної науково-практичної конференції “Innovative Solutions in Science: Balancing Theory and Practice”, Сан-Франциско, США, 2024 р. (Додаток Б).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка, освітньо-професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» /Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2023. – 49 с.

2. Положення про кваліфікаційну роботу здобувача вищої освіти на другому (магістерському) рівні [Електронний ресурс]: Наказ ХНУРЕ від 06 травня 2021 р. № 144. – Режим доступу: https://nure.ua/wpcontent/uploads/Main_Docs_NURE/143-vid-06.05.2021-pro-vvedennja-v-dijurishennja-vchenoi-radi-universitetu.pdf.

3. ДСТУ 3008:2015 Інформація та документація «Звіти у сфері науки і техніки». Структура та правила оформлювання. / В. Земцева; Ю. Поліщук, канд. фіз.-мат. наук; Р. Санченко, канд. техн. наук; Л. Шрамко; А. Ямчук (науковий керівник) ДП «УкрНДНЦ» від 22 червня 2015р. № 61 з 2017-07-01.

4. Abdullaiev A. Unmanned Multirotor Aerial Vehicles // II International scientific and practical conference “Innovative Solutions in Science: Balancing Theory and Practice” – (December 23-25, 2024. San Francisco, USA). European Open Science Space, 2024. – PP. 45-48.

5. Groothuis, S., Fiume, D., & Della Santina, C. Quadrupedal robotics: An overview of locomotion principles, control strategies and challenges. IEEE Robotics & Automation Magazine, 2022. 29(4). – С. 38-55.

6. Kim, S., Li, W., & Seok, S. Design of high-performance series elastic actuators for legged robots. *The International Journal of Robotics Research*, 2021. T. 40, № 7. – С. 954-972.
7. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G. *Robotics: Modelling, Planning and Control*. Springer, 2010. – 620 p.
8. Spong, M., Hutchinson, S., Vidyasagar, M. *Robot Modeling and Control*. 2nd ed. Wiley, 2020. – 802 p.
9. Sutton, R. S., Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 2018. – 552 p.
10. Gama, F., Ribeiro, A., & Prorok, A. Graph Neural Networks for Robot Locomotion Control. *Conference on Robot Learning (CoRL)*, 2022.
11. MATLAB Documentation. *Control System Toolbox User's Guide*. MathWorks, 2023. – Режим доступа: <https://www.mathworks.com/help/control/> – 18.10.2025 p. – Загл. з екрану.
12. Spong M. W., Hutchinson S., Vidyasagar M. *Robot Modeling and Control*. – Wiley, 2020. – 762 p.
13. Driankov D., Hellendoorn H., Reinfrank M. *An Introduction to Fuzzy Control*. – Springer, 2023. – 350 p.
14. Sutton R. S., Barto A. G. *Reinforcement Learning: An Introduction*. – MIT Press, 2018. – 532 p.
15. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. – MIT Press, 2016. – 775 p.
16. Murphy K. P. *Machine Learning: A Probabilistic Perspective*. – MIT Press, 2012. – 1104 p.
17. Kalman R. E. A New Approach to Linear Filtering and Prediction Problems. *ASME Journal of Basic Engineering*, 1960. – P. 35-45.

18. STMicroelectronics. STM32 Microcontroller Reference Manual. – 2023. – 1860 p.
19. Tan J., Zhang T. Deep reinforcement learning for quadruped motion control in simulation and real world. *Robotics and Autonomous Systems*, 2022. – Vol. 150. – P. 103-156.
20. MathWorks. Simscape Multibody and Robotics System Toolbox Documentation. – 2023.
21. Khanna G., Kumar A. Energy-aware gait optimization for quadruped robots. *Robotics and Autonomous Systems*, 2021. – Vol. 141. – P. 103-119.
22. Rareş O., Hangan A. Nonlinear stability of legged platforms. *Journal of Mechanical Systems*, 2021. – Vol. 49. – P. 334-348.
23. ROS Developers Guide. Robot Operating System (ROS2) Architecture and APIs. – Open Robotics, 2022.
24. CAN in Automation. CANopen Application Layer and Communication Profile. – CiA 301 Standard. – 2021.
25. NVIDIA. Jetson Embedded Systems Technical Reference. – 2022.
26. PyTorch Team. PyTorch Documentation and API Reference. – Meta AI, 2023.
27. TensorFlow Developers. TensorFlow 2.0 Programming Guide. – Google Research, 2023.
28. ISO 8373:2021 Robots and robotic devices – Vocabulary. – International Organization for Standardization, 2021.
29. Chen W.-H., Yang T. Hybrid intelligent control methods for legged robots. *IEEE Transactions on Cybernetics*, 2020. – Vol. 50(7). – P. 3241-3255.
30. Park H., Kim J. Real-Time Gait Generation for Quadruped Robots Using Model Predictive Control. *Robotics and Automation Letters*, 2019. – Vol. 4(3). – P. 3078-3085.

31. Biswal A., Nandy A. Autonomous quadruped locomotion: algorithms, sensors, and actuation. *Robotics Review*, 2021. – Vol. 7. – P. 1-29.
32. Craig J. J. *Introduction to Robotics: Mechanics and Control*. – 4th ed. – Pearson, 2018. – 448 p.
33. Mohamed A., Sasi A., Ben-Tzvi P. Dynamic modeling and control of quadruped robots: A review. *Robotics and Autonomous Systems*, 2021. – Vol. 138. – P. 103-112.
34. Della Santina C., Grioli G., Catalano M. Dynamic and energetic analysis of legged locomotion. *IEEE Transactions on Robotics*, 2020. – Vol. 36(6). – P. 1760-1777.
35. Bledt G., Kim S. *Quadruped locomotion: A computational framework*. MIT Biomimetic Robotics Lab Reports, 2020. – 145 p.
36. Cao Z., Wang Y. Stability analysis for legged robots using hybrid control. *IEEE Transactions on Cybernetics*, 2020. – Vol. 50(4). – P. 1530-1542.
37. Qt Company. *Qt for Python (PySide6) Documentation*. – 2023.
38. MathWorks. *Simulink Robotics System Toolbox Documentation*. – 2023.
39. Robotis Co., Ltd. *Dynamixel Smart Actuators: Technical Specifications*. – 2022. – 102 p.
40. Hassan K., Faragallah O. Real-time control systems for legged robots. *Journal of Robotics and Autonomous Systems*, 2021. – Vol. 145. – P. 103-118.
41. Комплекс навчально-методичного забезпечення навчальної дисципліни «Організація керування умовами праці» підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету [Електронний ресурс] / ХНУРЕ; розроб.: Т. Є. Стищенко, Г. В. Пронюк, Н. М. Сердюк. – Харків, 2017. – 108 с.