

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

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

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)

(рівень вищої освіти)

Розроблення інтерфейсів для керування роботами
з використанням штучного інтелекту

(тема)

Виконав:

здобувач 2023 року навчання, групи КІТПВМ-23-2

Столяренко Н.В.

(прізвище, ініціали)

Спеціальність 174 Автоматизація,
комп'ютерно-інтегровані технології та робототехніка
освітньої програми Комп'ютерно-інтегровані
технологічні процеси і виробництва

(код і повна назва напрямку)

Тип програми освітньо-професійна

(повна назва освітньої програми)

Керівник доц. Хрустальова С. В.

(посада, прізвище, ініціали)

Допускається до захисту
зав. кафедри

(підпис)

Невлюдов І.Ш.

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет	Автоматики і комп'ютеризованих технологій
Кафедра	КІТАР
Рівень вищої освіти	другий (магістерський)
Спеціальність	174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Тип програми	освітньо-професійна
Освітня програма	Комп'ютерно-інтегровані технологічні процеси і виробництва
	(код і повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР

(підпис)

«_____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Столяренку Никіті Васильовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення інтерфейсів для керування роботами з використанням штучного інтелекту

затверджена наказом по університету від 22.11.2024 р. №1231 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 22.01.2025 р.

3. Вихідні дані до роботи _____

3.1 Інтерфейси з використанням штучного інтелекту;

3.2 Симулятор гра Mafia 2;

3.3 Мова програмування Python;

3.4 Система програмування Microsoft Visual Studio.

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

4.1 Вступ

4.2 Аналіз вихідних даних і літератури за темою дослідження

4.3 Дослідження взаємодії користувачів з інтерфейсом

4.4 Розроблення інтерфейсу керування роботами з використанням штучного інтелекту

4.5 Охорона праці

4.6 Висновки

4.7 Перелік джерел посилання

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційний матеріал представлений у форматі презентації PowerPoint (*.ppt) – 12 с. формату А4

6. Консультанти розділів роботи

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	23.11.2024	Виконано
2	Аналіз завдання	27.11.2024	Виконано
3	Огляд літератури з теми дослідження	04.12.2024	Виконано
4	Розробка інтерфейсу	11.12.2024	Виконано
5	Проведення тестування	18.12.2024	Виконано
6	Розрахунки з питань охорони праці	25.12.2024	Виконано
7	Оформлення пояснювальної записки	29.12.2024	Виконано
8	Подання кваліфікаційної роботи в ЕК	22.01.2025	Виконано

Дата видачі завдання 23.11.2024 року

Здобувач

(підпис)

Столяренко Н.В.

(прізвище, ініціали)

Керівник роботи

(підпис)

доц. Хрустальова С.В.

(посада, прізвище, ініціали)

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«22» січня 2025 р.

Столяренко Н.В.

РЕФЕРАТ

Пояснювальна записка: 77 с., 3 таб., 11 рис., 2 дод., 28 джерел.

ШТУЧНИЙ ІНТЕЛЕКТ, БЕЗДРОТОВА СИСТЕМА, ІНТЕРФЕЙС,
КОМУНІКАЦІЙНА СИСТЕМА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ,
АВТОМАТИЗАЦІЯ, WI-FI.

Об'єкт дослідження – процес керування роботами з використанням технологій штучного інтелекту, які дозволяють забезпечити ефективну, інтуїтивну та адаптивну взаємодію між людиною та роботизованими системами.

Предмет дослідження – програмний засіб для керування роботами з використанням штучного інтелекту.

Мета кваліфікаційної роботи – підвищення ефективності керування роботами за рахунок розроблення інтерфейсу на базі методів ШІ.

У роботі проводиться аналіз вихідних даних і літератури за темою дослідження. Розглядаються сучасні інтерфейси для керування роботами з акцентом на застосування ШІ. Проведений порівняльний аналіз переваг та недоліків існуючих підходів до інтеграції ШІ в управління роботами.

Отримані результати роботи можуть використовуватись для інтерактивного керування автономними роботами у складних середовищах, таких як логістика або промислове виробництво, підвищуючи ефективність і знижуючи ризик людських помилок.

Також, отримані результати роботи можна віднести до Цілі сталого розвитку 9 "Промисловість, інновації та інфраструктура", а саме до п. 9.4 "Сприяти прискореному розвитку високо- та середньовисокотехнологічних секторів переробної промисловості, які формуються на основі використання ланцюгів "освіта – наука – виробництво" та кластерного підходу за напрямками: розвиток інноваційної екосистеми; розвиток інформаційно-телекомунікаційних технологій (ІК); високотехнологічне машинобудування; розвиток біоінженерної галузі."

ABSTRACT

Explanatory note: 77 p., 3 tab., 11 fig., 2 appendices, 28 sources.

ARTIFICIAL INTELLIGENCE, WIRELESS SYSTEM, INTERFACE,
COMMUNICATION SYSTEM, SOFTWARE, AUTOMATION, WI-FI.

The object of research is a process of controlling robots using artificial intelligence technologies that allow for effective, intuitive and adaptive interaction between humans and robotic systems.

The subject of research is a software tool for controlling robots using artificial intelligence. The goal of the qualification work is to increase the efficiency of robot management due to the development of an interface based on AI methods.

The study involves analyzing initial data and literature on the research topic. Modern interfaces for robot control are reviewed with an emphasis on the application of AI. A comparative analysis of the advantages and disadvantages of existing approaches to AI integration in robot control is conducted.

The results of this work can be applied to interactive control of autonomous robots in complex environments, such as logistics or industrial production, improving efficiency and reducing the risk of human error.

Additionally, the outcomes of this work align with Sustainable Development Goal 9, "Industry, Innovation, and Infrastructure," specifically target 9.4 "Promote accelerated development of high- and medium-high-tech sectors of the processing industry, based on the use of education–science–production chains and a cluster approach in the following areas: development of an innovation ecosystem; development of information and communication technologies (ICT); high-tech mechanical engineering; development of the bioengineering sector."

ЗМІСТ

Перелік скорочень	8
Вступ.....	9
1 Аналіз методі, технологій та алгоритмів керування автономними роботизованими системами.....	11
1.1 Аналіз завдання на кваліфікаційну роботу	11
1.2 Класифікація підходів та огляд основних технологій.....	12
1.3 Аналіз сучасних інтерфейсів для керування роботами.....	20
1.4 Компоненти автономних роботизованих систем.....	27
1.5 Алгоритми ШІ в автономних роботизованих системах.....	32
1.6 Висновки до першого розділу.....	40
2 Розроблення автономної роботизованої системи.....	41
2.1 Структура автономної роботизованої системи.....	41
2.2 Алгоритм роботи автономної роботизованої системи	43
2.3 Розрахунок стійкості автономної роботизованої системи.....	45
2.4 Висновки до другого розділу.....	50
3 Розроблення інтерфейсу керування роботами з використанням ШІ.....	51
3.1 Програмні засоби для створення інтерфейсу керування роботами...	51
3.2 Моделювання взаємодії користувачів з інтерфейсом	52
3.3 Розроблення програмного забезпечення	54
3.4 Дослідження трудомісткості розроблення програмного продукту	63
3.5 Висновки до третього розділу	68
4 Охорона праці	69
4.1 Аналіз умов праці на робочому місці	69
4.2 Промислова безпека на робочому місці	69
4.3 Виробнича санітарія і гігієна праці.....	70
4.4 Пожежна безпека приміщення	72
Висновки.....	73
Перелік джерел посилання.....	74
Додаток А Код програми	77
Додаток Б Демонстраційний матеріал	77

ПЕРЕЛІК СКОРОЧЕНЬ

- ШІ – Штучний інтелект;
- API – Інтерфейс прикладного програмування;
- AR – Доповнена реальність;
- DL – Глибинне навчання;
- GPU – Графічний процесор;
- GUI – Графічний інтерфейс користувача;
- ML – Машинне навчання;
- NLP – Обробка природної мови;
- RPA – Роботизована автоматизація процесів;
- VR – Віртуальна реальність;
- Wi-Fi – Wireless Fidelity.

ВСТУП

На сьогодні робототехніка стала добре розвиненою галуззю. Роботи активно використовуються у виробництві, медицині, сільському господарстві, логістиці та навіть у побуті. Ефективне управління цими системами є критичним для підвищення продуктивності, зменшення людських помилок і оптимізації робочих процесів.

Оскільки роботи стають все складнішими і виконують дедалі більше завдань, традиційні інтерфейси керування стають менш ефективними. Використання штучного інтелекту (ШІ) дозволяє спростувати ці інтерфейси, роблячи управління більш інтуїтивним, особливо для користувачів без спеціальних технічних знань.

Інтерфейси, що використовують ШІ, можуть автоматично адаптуватися під потреби та поведінку користувача, що робить взаємодію з роботом більш зручною. Можуть навчатися на основі взаємодії з користувачами та накопиченого досвіду, постійно вдосконалюючись. Це дозволяє їм ставати більш ефективними з часом і забезпечувати більш персоналізований підхід до керування.

Автономні роботи, які здатні виконувати складні завдання без постійного контролю з боку людини, стають все більш популярними. Для таких систем необхідні інтерфейси, здатні обробляти великі обсяги даних, які генерують роботи, і приймати рішення в реальному часі, що робить ШІ незамінним елементом.

Розроблення інтерфейсів з використанням ШІ дозволить компаніям залишатися на передовій інновацій та зберігати конкурентоспроможність у сфері робототехніки та автоматизації. Це сприятиме впровадженню нових рішень, які можуть стати стандартом у майбутньому [1].

Виходячи з цього тема кваліфікаційної роботи «Розроблення інтерфейсів для керування роботами з використанням штучного інтелекту» є актуальною.

Об'єкт дослідження – процес керування роботами з використанням технологій штучного інтелекту, які дозволяють забезпечити ефективну, інтуїтивну та адаптивну взаємодію між людиною та роботизованими системами.

Предмет дослідження – програмний засіб для керування роботами з використанням штучного інтелекту, зокрема алгоритми адаптації, автоматизації та

підвищення ефективності взаємодії користувача з роботизованими системами.

Мета кваліфікаційної роботи – підвищення ефективності керування роботами за рахунок розроблення інтерфейсу на базі методів ШІ.

Для досягнення цієї мети планується розв'язати наступні задачі:

- провести аналіз існуючих методів, засобів, технологій та алгоритмів керування роботизованими системами;
- розробити архітектуру інтерфейсу та алгоритмів взаємодії ШІ з роботами для управління і навігації в динамічному середовищі;
- розглянути технології та алгоритми ШІ, що використовуються для автоматизації й адаптації інтерфейсів;
- провести порівняльний аналіз існуючих сучасних інтерфейсів для керування роботами з акцентом на застосування ШІ;
- розробити інтерфейс для керування роботами для управління і навігації в динамічному середовищі.

Робота виконується згідно з [2-4], як складова наукових досліджень, які здійснюються на кафедрі КІТАР Харківського національного університету радіоелектроніки.

Відповідно Указу президента України №722/2019 від 30.09.2019 року "Про Цілі сталого розвитку (ЦСР) України на період до 2030 року" [5], питання, що розглядаються в кваліфікаційній роботі відповідають Цілі сталого розвитку 9 "Промисловість, інновації та інфраструктура", а саме завданню 9.4 "Сприяти прискореному розвитку високо- та середньовисокотехнологічних секторів переробної промисловості, які формуються на основі використання ланцюгів "освіта – наука – виробництво" та кластерного підходу за напрямками: розвиток інноваційної екосистеми; розвиток інформаційно-телекомунікаційних технологій (ІК); застосування ІК в АПК, енергетиці, транспорті та промисловості; високотехнологічне машинобудування; створення нових матеріалів; розвиток фармацевтичної та біоінженерної галузей".

Отримані результати роботи опубліковано у Збірнику студентських наукових статей "Automation and Development of Electronic Devices" назва статті "Розробка інтерфейсів для керування роботами з використанням штучного інтелекту" [6].

1 АНАЛІЗ МЕТОДІВ, ТЕХНОЛОГІЙ ТА АЛГОРИТМІВ КЕРУВАННЯ АВТОНОМНИМИ РОБОТИЗОВАНИМИ СИСТЕМАМИ

1.1 Аналіз завдання на кваліфікаційну роботу

Виходячи із завдання, на кваліфікаційну роботу можна виділити основні вимоги до інтерфейсу для керування роботами з використанням методів ШІ.

Для цього необхідно вирішити такі завдання:

- розглянути існуючі технології та алгоритми штучного інтелекту, що використовуються для автоматизації й адаптації інтерфейсів;

- провести аналіз основних параметрів які впливають на покращення взаємодії з користувачем, автоматизації процесів та підвищення продуктивності робототехнічних систем;

- провести інтеграцію ШІ в процес управління роботами для управління і навігації в динамічному середовищі;

- розробити інтерфейс для керування;

- провести перевірку розробленої моделі на наукову новизну.

Основні вимоги :

- автоматичне підлаштування інтерфейсу під потреби та поведінку користувача;

- здатність системи на основі ШІ самостійно приймати рішення або пропонувати найбільш ефективні варіанти дій користувачу;

- здатність системи навчатися на основі зібраних даних про дії користувача, що дозволить покращувати управління роботами з часом;

- система має забезпечувати можливість гнучкого налаштування для різних сценаріїв використання;

- використання сучасних технологій (актуальні версії мов програмування і фреймворків).

1.2 Класифікація підходів та огляд основних технологій

Автономні роботизовані системи – це системи, які можуть самостійно виконувати завдання без постійного контролю з боку людини, використовуючи сенсори, алгоритми штучного інтелекту та інші технології для адаптації до оточення, прийняття рішень і виконання поставлених цілей. Такі системи зазвичай мають здатність до самонавчання, що підвищує їхню ефективність у виконанні складних завдань та пристосуванні до нових умов [7].

Розглянемо алгоритм роботи автономної роботизованої системи.

Після ввімкнення система перевіряє всі компоненти, сенсори та рівень заряду акумулятора. Встановлюються вихідні параметри роботи та конфігурація.

Сенсори збирають інформацію про навколишнє середовище та стан робота. Ці дані передаються до обчислювального блоку для подальшої обробки.

В обчислювальному блоці за допомогою ШІ та алгоритмів обробки даних визначаються критичні ситуації, такі як наявність перешкод або зміна навколишнього середовища.

Модуль прийняття рішень аналізує оброблені дані та формує команди для виконання необхідних дій.

Модуль навігації визначає оптимальний маршрут, щоб уникнути перешкоди, обрати безпечний шлях та досягти цілі. Рух робота контролюється контролером руху.

Робот виконує заплановані дії, які можуть включати переміщення, маніпуляції з об'єктами або інші завдання.

Система відслідковує виконання команд і за потреби коригує рух.

Через інтерфейс користувача надається інформація про виконання завдань, стан робота та можливі помилки. Користувач також може вносити коригування в роботу системи через GUI.

На основі зібраної інформації та дій робота система оновлює свої алгоритми та моделі ШІ, щоб покращити майбутню роботу та підвищити адаптивність.

Автономні роботизовані системи мають широкий спектр застосувань у багатьох галузях завдяки своїм можливостям працювати без постійного нагляду з

боку людини, адаптуватися до змін у навколишньому середовищі та ефективно виконувати складні завдання [8].

Самокеровані автомобілі, вантажівки для доставки автоматизують процеси транспортування вантажів та пасажирів, підвищуючи безпеку і ефективність руху.

Використання автономних транспортних засобів у логістичних центрах та на заводах для переміщення матеріалів між різними станціями знижує затрати часу та людського ресурсу.

Автономні системи перевіряють стан доріг, мостів, будівель, що дозволяє виявляти проблеми на ранніх етапах. Здійснюють автоматизовані виробничі процеси, включаючи зварювання, збірку, сортування, пакування. Такі роботи часто працюють у середовищах з високим рівнем ризику.

Використовуються у сільському господарстві. Виконують завдання, такі як висівання, збирання врожаю, моніторинг стану культур. Вони здатні адаптуватися до змін у природному середовищі та мінімізувати витрати ресурсів.

Перевагами автономних роботизованих систем є ефективність, безпека, адаптивність та мінімальні потреби у великій кількості людського ресурсу, особливо в рутинних завданнях. Завдяки цим перевагам автономні роботизовані системи поступово стають невід'ємною частиною багатьох галузей, допомагаючи оптимізувати процеси, підвищувати продуктивність і забезпечувати безпеку.

Хоча автономні роботизовані системи мають численні переваги, вони також мають низку недоліків і обмежень, які впливають на їх широке впровадження та використання.

Розробка, впровадження та обслуговування автономних систем потребує значних фінансових ресурсів. Вартість високотехнологічного обладнання, сенсорів, а також потужних обчислювальних модулів є досить високою. Програмування автономних роботизованих систем і їх інтеграція з існуючими інфраструктурами вимагає спеціалізованих знань та досвіду в галузі штучного інтелекту, машинного навчання та сенсорних технологій.

Автономні системи, особливо ті, що мають мережеве з'єднання, піддаються ризику кібератак, що може призвести до втрати контролю над роботом або витоку конфіденційних даних. Залежать від сенсорів, які збирають інформацію про оточення.

Якщо сенсори не можуть точно фіксувати інформацію через погані умови (наприклад, дощ, туман), система може діяти некоректно. Умови, що впливають на точність сенсорів, можуть спричинити помилки у функціонуванні системи або її повну зупинку [9].

Для подолання цих проблем потрібно розвивати технології захисту від кібератак, знижувати вартість виробництва, покращувати якість сенсорів і алгоритмів ШІ, а також розробляти відповідні законодавчі норми. Відкриття нових методів зниження витрат на впровадження, підвищення безпеки та вдосконалення технологій також сприятиме поширенню цих систем у майбутньому.

Інноваційні підходи до розробки інтерфейсів для керування роботами з використанням штучного інтелекту можна класифікувати за кількома основними напрямками. До цих напрямків відносяться підходи, що забезпечують автоматизацію, адаптацію до поведінки користувача, підвищення ефективності та точності керування, а також інтуїтивність інтерфейсу [10].

Класифікація інноваційних підходів:

- адаптивні інтерфейси на основі ШІ;
- інтерфейси на основі природної мови;
- інтерфейси з використанням розпізнавання жестів;
- доповнена реальність (AR) для керування роботами.

До адаптивних інтерфейсів на основі ШІ відносяться машинне та глибоке навчання.

Машинне навчання (ML) – це підгалузь штучного інтелекту, яка надає системам здатність вчитися та покращувати свої дії на основі досвіду без явного програмування. У контексті розробки інтерфейсів для керування роботами, ML використовується для створення адаптивних систем, які можуть реагувати на поведінку користувача, аналізувати дані в режимі реального часу та автоматично коригувати свою роботу.

Підходи машинного навчання поділяються на навчання з учителем, навчання без учителя та навчання з підкріпленням [11].

Метод навчання з учителем (англ. Supervised Learning) передбачає використання набору даних, що містить вхідні параметри та вихідні значення.

Модель навчається на цих даних, щоб потім прогнозувати результати для нових входів. Для керування роботами цей підхід може використовуватися для розпізнавання команд користувача або аналізу поведінкових даних.

В методі навчання без учителя (англ. Unsupervised Learning) алгоритми не мають попередньо мічених даних. Модель самостійно шукає структури або закономірності в даних. Це підходить для аналізу великих обсягів даних користувачів і адаптації інтерфейсу до різних сценаріїв використання.

Метод навчання з підкріпленням (англ. Reinforcement Learning) використовує зворотній зв'язок від середовища, щоб навчити модель досягати кращих результатів. Модель отримує винагороди або покарання за свої дії, що допомагає їй поступово вдосконалюватися [12].

Машинне навчання надає широкі можливості для покращення інтерфейсів керування роботами, роблячи їх більш адаптивними, ефективними та інтуїтивними для користувачів. Інтеграція ML дозволяє системам навчатися на основі зібраних даних, підвищувати точність виконання команд та автоматизувати рутинні завдання, що сприяє покращенню користувацького досвіду.

Підходи на основі глибоких нейронних мереж можуть забезпечувати складні аналізи поведінкових даних і прийняття рішень. Наприклад, це може включати розпізнавання образів або жестів для інтуїтивного керування роботами. Це підгалузь машинного навчання, яка використовує багат шарові нейронні мережі для аналізу складних даних та виявлення прихованих закономірностей. Глибинне навчання дозволяє моделям ШІ виконувати завдання високої складності, такі як розпізнавання зображень, мови, жестів та прийняття рішень в режимі реального часу. У контексті розробки інтерфейсів для керування роботами, глибинне навчання використовується для створення більш точних та адаптивних систем взаємодії.

Основою глибинного навчання є штучні нейронні мережі, що складаються з численних шарів нейронів. Кожен нейрон у мережі отримує інформацію, перетворює її за допомогою ваг та передає на наступний шар нейронів.

Шари мережі складається з вхідного шару, кількох прихованих шарів та вихідного шару. Кількість прихованих шарів і кількість нейронів у них визначають глибину мережі.

Глибинні нейронні мережі вчаться на основі великих обсягів даних. Процес навчання полягає в налаштуванні ваг кожного нейрона для мінімізації помилки прогнозу.

Алгоритм зворотнього поширення (англ. Backpropagation) призначений для коригування ваг у мережі. Він дозволяє моделі зменшувати похибки шляхом коригування кожної ваги на основі результату, отриманого на виході мережі.

Тип мереж CNN використовується для аналізу зображень та відео. CNN здатні розпізнавати візуальні патерни, такі як об'єкти або жести, що робить їх корисними для інтерфейсів на основі комп'ютерного зору.

RNN підходять для обробки послідовних даних, таких як голос або текст. Вони мають "пам'ять" про попередні стани, що робить їх ефективними для аналізу послідовностей даних [13].

Для коректної роботи нейронних мереж використовуються активаційні функції, які визначають, як буде перетворюватися сигнал на кожному шарі. Найпоширеніші функції: ReLU (англ. Rectified Linear Unit), сигмоїдальна функція та tanh.

Глибинне навчання відкриває широкі можливості для створення інтерфейсів керування роботами, роблячи їх більш інтуїтивними, адаптивними та ефективними. Використання глибинних нейронних мереж дозволяє створювати системи, які здатні розпізнавати складні патерни поведінки користувачів та адаптуватися до їхніх потреб. Інтеграція глибинного навчання у системи керування роботами забезпечує нові підходи до автоматизації та покращення взаємодії між користувачем і роботом.

Інтерфейси на основі природної мови (англ. Natural Language Interfaces, NLI) дозволяють користувачам взаємодіяти з системами через звичайну розмовну мову, замість традиційних способів введення, таких як кнопки або меню. Це підхід до інтерактивної взаємодії, який використовує технології обробки природної мови, штучного інтелекту та голосових асистентів для спрощення і поліпшення користувацького досвіду.

Основні компоненти інтерфейсів на основі природної мови включають обробку природної мови (англ. NLP – Natural Language Processing), голосові асистенти,

технології розпізнавання голосу (англ. Speech-to-Text), технології синтезу мовлення (англ. Text-to-Speech) [14].

NLP дозволяє системам розуміти, інтерпретувати та генерувати людську мову. Ця технологія використовується для аналізу введених голосових або текстових команд, розпізнавання смислових структур у мові та формування відповідних дій.

Голосові асистенти використовують NLP для сприйняття голосових команд та виконання завдань. Вони можуть інтегруватися з системами керування роботами для прийому голосових команд та їх виконання.

Асистенти Google Assistant, Amazon Alexa, Apple Siri можуть керувати роботами або іншими системами через природну мову.

Системи розпізнавання мови перетворюють голосові команди в текстові для подальшої обробки. Технології, такі як Google Speech API або Microsoft Azure Speech забезпечують високу точність розпізнавання мови в реальному часі.

Text-to-Speech перетворює текстові відповіді системи на аудіо. Це дає можливість інтерфейсу "розмовляти" з користувачем, що покращує інтерактивність та зручність використання.

Інтерфейси з використанням розпізнавання жестів дозволяють користувачам взаємодіяти з системами, включаючи роботи, за допомогою рухів рук і тіла. Такі інтерфейси забезпечують природну та інтуїтивну форму управління, знижуючи потребу в фізичних кнопках або пристроях керування. Розпізнавання жестів використовує технології комп'ютерного зору, машинного навчання та штучного інтелекту для аналізу й інтерпретації рухів користувача.

Технології комп'ютерного зору дозволяють системам аналізувати зображення та відео, розпізнавати об'єкти й відстежувати рухи. Для розпізнавання жестів використовуються камери та сенсори для захоплення рухів рук та тіла. Глибинне навчання дозволяє інтерфейсам автоматично розпізнавати та класифікувати жести користувачів із високою точністю. Спеціалізовані пристрої, такі як Leap Motion, Microsoft Kinect, або Intel RealSense використовуються для захоплення рухів користувача в реальному часі. Вони відстежують положення рук, пальців та інших частин тіла, дозволяючи системі коректно інтерпретувати жести [15].

Жестові інтерфейси дозволяють користувачам керувати системами інтуїтивно та без спеціальної підготовки, що підвищує комфорт і зручність взаємодії. Такі інтерфейси дозволяють керувати роботами на відстані або в умовах, де фізичний контакт з пристроями недоцільний або неможливий.

Доповнена реальність (AR) – це технологія, яка дозволяє інтегрувати віртуальні елементи в реальний світ, створюючи інтерактивне середовище для користувачів.

AR для керування роботами використовує віртуальні зображення, моделі, дані або елементи, які накладаються на реальний світ за допомогою пристроїв, таких як смартфони, планшети або спеціалізовані окуляри.

Це робить управління роботами більш інтуїтивним, зручним і ефективним.

AR дозволяє користувачам бачити віртуальні моделі роботів, дані про їх роботу, маршрути руху або стан у реальному середовищі. Це дозволяє взаємодіяти з системою на більш інтуїтивному рівні, порівняно з традиційними інтерфейсами. За допомогою AR користувачі можуть візуалізувати не лише роботу робота, а й потенційні сценарії та шляхи його дій. Це полегшує процес планування та керування, дозволяючи краще оцінити ситуацію.

Доповнена реальність для керування роботами часто використовує Computer Vision для розпізнавання та аналізу реального середовища. Технологія дозволяє системам сприймати об'єкти, позиції та жести в реальному світі, що є важливим для точного накладання віртуальних елементів на реальну сцену. Взаємодія між AR та Computer Vision дозволяє створювати інтуїтивно зрозуміле управління роботами за допомогою візуальних підказок або віртуальних елементів, що робить процес більш ефективним і реалістичним.

Розпізнавання об'єктів, відстеження рухів і жестів, позиціонування та накладання є основними аспектами використання Computer Vision у AR для визначення та взаємодії з реальними об'єктами та роботами.

За допомогою AR можна відображати маршрути, по яким рухається робот, або цілі, які він повинен досягти. Це дозволяє легко налаштовувати та коригувати завдання робота, забезпечуючи більш точне управління [16].

AR дозволяє побачити віртуальні елементи, такі як траєкторії, перешкоди, або стан робота, без необхідності звертатися до складних графічних інтерфейсів.

Це робить управління роботами більш інтуїтивним та зрозумілим. AR може забезпечити дистанційне управління роботами без необхідності фізичної взаємодії, що особливо корисно в небезпечних або важкодоступних середовищах.

Крім того, AR може надати додаткову інформацію у реальному часі, наприклад, попередження про перешкоди або рекомендації щодо оптимальних маршрутів. Візуалізація таких даних у контексті реального світу значно підвищує ефективність роботи оператора.

Розвиток підходів та технологій у комп'ютерному зорі та AR дозволяє вирішувати складні завдання, підвищуючи продуктивність та точність у різних галузях.

Впровадження цих рішень в автоматизацію, робототехніку та аналіз великих даних відкриває нові можливості для бізнесу, науки та повсякденного життя.

Класифікація інноваційних підходів та огляд основних технологій представлений на рис. 1.1.

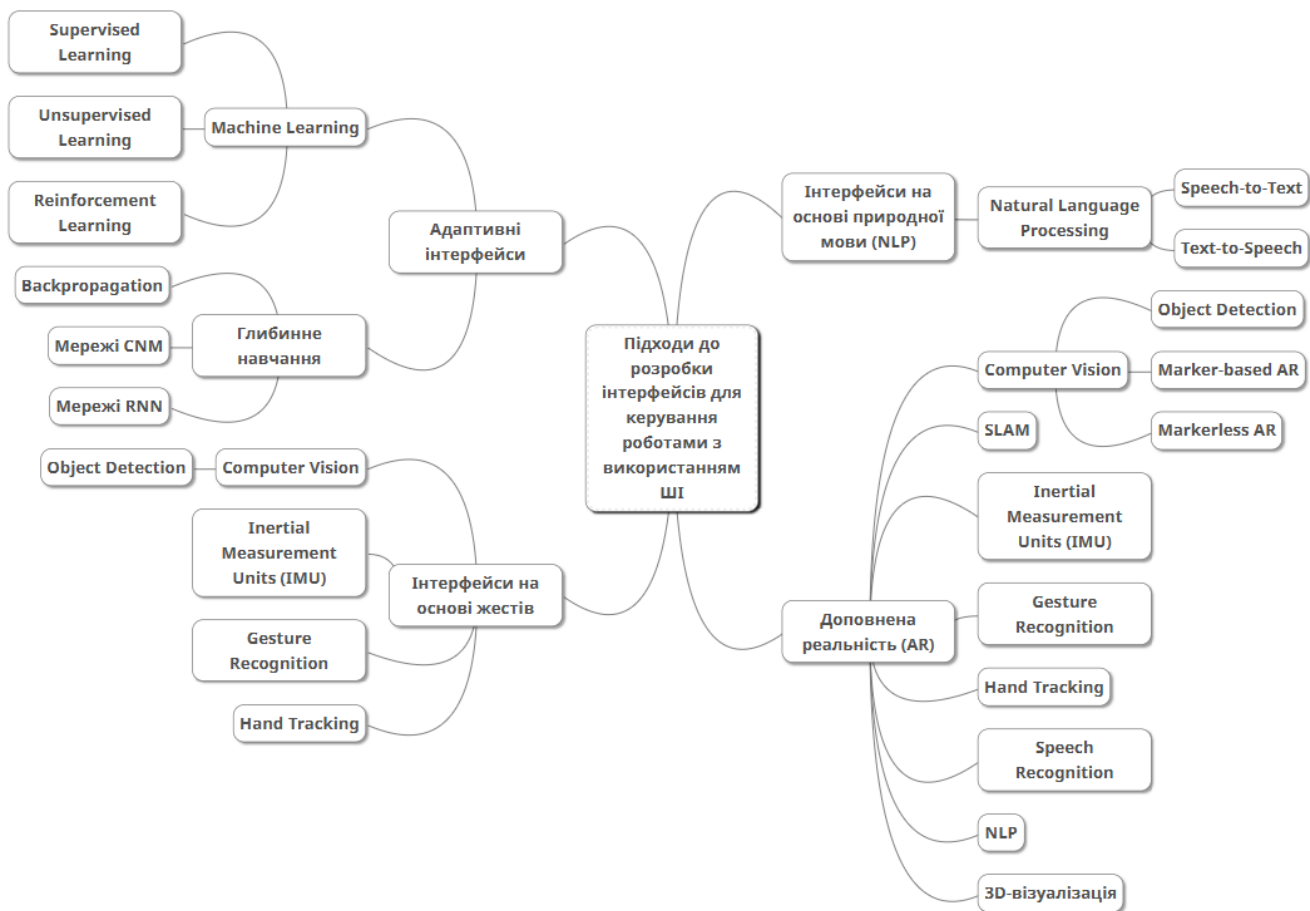


Рисунок 1.1 – Класифікація підходів та огляд основних технологій

1.3 Аналіз сучасних інтерфейсів для керування роботами

Сучасні інтерфейси для керування роботами використовують різноманітні технології, спрямовані на забезпечення інтуїтивної та ефективної взаємодії між користувачем і робототехнічними системами. Оскільки сфера робототехніки активно розвивається, інтерфейси стають все більш адаптивними, інтерактивними та гнучкими.

Графічний інтерфейс користувача (GUI) є одним із найпоширеніших способів взаємодії людини з комп'ютерами, а також з робототехнічними системами. Він включає в себе використання візуальних елементів, таких як вікна, кнопки, меню, текстові поля та іконки, для полегшення взаємодії з системою. У контексті керування роботами GUI забезпечує зручний та інтуїтивний спосіб контролю над робототехнічними системами, дозволяючи користувачам бачити стан роботи, керувати його функціями та виконувати різні завдання через графічний інтерфейс/

Кнопки є одними з найбільш основних елементів управління, які дозволяють виконувати команди, такі як "почати", "зупинити", "повернути" або "виконати завдання".

Меню дозволяють користувачам вибирати різні параметри та функції системи. Вони можуть містити налаштування для керування роботом, вибір завдань або перехід до додаткових інструментів.

Іконки та значки надають зручний спосіб запуску команд чи функцій системи через графічне представлення завдань або параметрів.

Панелі інструментів містять набір часто використовуваних команд і функцій для швидкого доступу. Це можуть бути команди для маніпуляції з роботом або перегляду його стану.

Індикатори показують поточний стан роботи, включаючи стан батареї, швидкість, місцезнаходження або інші параметри. Вони можуть використовувати кольори або графічні зображення для спрощення інтерпретації стану.

Можливість візуалізації даних та стану. Це може бути карта або діаграма, яка показує поточне місцезнаходження робота та його маршрут.

Це дозволяє користувачеві бачити, де робот знаходиться у просторі.

Діаграми та графіки можуть відображати важливі параметри роботи, такі як продуктивність, час виконання завдань, або використання ресурсів.

Інтерактивні елементи дозволяють точно задавати параметри роботи робота, такі як швидкість руху або цільові координати. Можливість вводити текстові команди або параметри, які потім інтерпретуються системою для виконання відповідних дій.

У промислових умовах GUI часто використовується для налаштування та контролю роботи роботизованих рук або інших механізмів. Оператор може використовувати інтерфейс для вибору завдань, спостереження за роботом через візуальні індикатори та виконання команд для маніпуляції об'єктами або контролю за процесом виробництва [17].

Для автономних транспортних засобів, таких як дрони або наземні роботи, GUI може забезпечувати можливість віддаленого керування, де оператор задає маршрути, переглядає стан робота та змінює параметри роботи в реальному часі. Це особливо важливо у сфері логістики, сільського господарства або військових операцій.

В освітніх програмах для навчання робототехніці часто використовують GUI для демонстрації базових операцій робота. Студенти можуть візуально налаштовувати завдання, контролювати процес та спостерігати за роботою робота, використовуючи зрозумілі візуальні елементи.

Графічний інтерфейс є інтуїтивно зрозумілим для більшості користувачів, навіть якщо вони не мають великого досвіду у сфері робототехніки це є значною перевагою у використанні. GUI робить управління роботами доступним для широкого кола користувачів завдяки зрозумілим візуальним елементам. Завдяки організованому інтерфейсу з кнопками та панелями інструментів користувачі можуть швидко виконувати потрібні дії та налаштування. Це особливо корисно у випадках, коли потрібна швидка реакція на зміни у роботі робота.

До недоліків можна віднести активну участь оператора для виконання команд, що може бути менш ефективним у випадках, коли потрібна автономність робота або автоматичне прийняття рішень.

Для складних завдань може знадобитися більше часу для налаштування всіх параметрів через GUI, порівняно з іншими інтерфейсами, такими як голосове керування або автоматизовані системи на основі ШІ.

Графічний інтерфейс користувача особливо корисний для простих і середньо складних завдань, де важливо мати візуалізацію даних та швидкий доступ до функцій. Проте для складніших систем, де потрібна автономність або адаптивність, GUI може бути менш ефективним, порівняно з інтерфейсами на основі штучного інтелекту або жестового управління.

Інтерфейси, що використовують голосові команди, дозволяють користувачам керувати роботами, використовуючи природну мову. Ці інтерфейси працюють завдяки технологіям обробки природної мови (NLP) та розпізнавання голосу.

До переваг можна віднести взаємодію з роботами без необхідності натискати кнопки або вводити текст. Сучасні системи голосового управління здатні підтримувати багато мов, що робить їх універсальними.

У шумному середовищі або при неправильному розпізнаванні голосу можуть виникати помилки та більш складні команди важко сформулювати за допомогою голосу, що можна віднести до недоліків.

Можуть використовуватися в робототехнічних навчальних середовищах для демонстрації та контролю роботи роботів, у промислових та виробничих системах, де оператори взаємодіють з роботами через панелі керування або екрани

Інтерфейси на основі розпізнавання жестів використовують камери та сенсори для розпізнавання рухів рук та тіла, що дозволяє користувачам керувати роботами за допомогою жестів. Такі інтерфейси часто використовуються у поєднанні з системами комп'ютерного зору. Користувачі можуть взаємодіяти з роботами безпосередньо через рухи, що робить інтерфейс дуже природним і зручним. Не потребує фізичного контакту з пристроями, що робить такі інтерфейси зручними в умовах, де використання сенсорних екранів або кнопок є неможливим.

Але для розпізнавання жестів необхідні камери високої роздільної здатності або спеціалізовані сенсори. Жестові інтерфейси можуть бути менш точними в умовах поганого освітлення або при надмірному навантаженні на систему.

Використовуються у медичних роботах для хірургічних операцій, де необхідне безконтактне керування інструментами також у промисловості та логістиці для безконтактного керування автоматизованими системами складування та транспортування.

Інтерфейси на основі доповненої реальності (AR). Інтегрує віртуальні об'єкти у реальне середовище через пристрої, такі як смартфони або окуляри. Це дозволяє користувачам бачити віртуальні елементи, які допомагають керувати роботами. Користувачі можуть бачити віртуальні моделі роботів або їхні маршрути, накладені на реальне середовище. Можливість взаємодіяти з роботами через віртуальні елементи, що робить управління більш наочним.

Інтерфейси на основі доповненої реальності ще не повсюдно впроваджені в індустрію через технологічні обмеження, що є недоліком та технології AR можуть бути дорогими через необхідність у спеціалізованому обладнанні.

Доповнена реальність застосовується для віддаленого керування роботами у складних середовищах (наприклад, у будівництві або при ремонті складного обладнання). У навчальних системах для відображення роботизованих моделей і демонстрації процесу їх роботи в реальному часі.

Інтерфейси з використанням віртуальної реальності (VR). Дозволяє користувачам повністю зануритися у віртуальне середовище, де вони можуть взаємодіяти з роботами або їх моделями. Взаємодія відбувається через шоломи або інші пристрої. Користувач отримує повний контроль над віртуальним середовищем, що дозволяє точно керувати роботами або симулювати їхню роботу. VR дозволяє бачити кожен елемент роботи робота, що полегшує процес керування та навчання.

До недоліків можна віднести необхідність спеціалізованих шоломів та контролерів. Тривале використання VR може викликати втому або дискомфорт у користувачів.

Інтерфейси з використанням віртуальної реальності використовуються для тренувань операторів та керування дистанційними роботами у важкодоступних або небезпечних умовах.

Інтерфейси на основі ШІ використовують алгоритми машинного навчання та

глибинного навчання для адаптації до дій користувача та автоматичного прийняття рішень.

Це можуть бути голосові або текстові асистенти, адаптивні системи керування або інші інтерфейси, що використовують ШІ для підвищення ефективності. Системи ШІ можуть навчатися на основі поведінки користувача та автоматично покращувати управління роботами.

Система керування роботом може навчатися на основі дій оператора і поступово пропонувати оптимальніші варіанти керування або автоматично налаштовувати параметри для виконання типових завдань. Інтерфейси на основі ШІ здатні аналізувати велику кількість даних у реальному часі та приймати рішення на основі цих даних. Це особливо корисно в умовах, де необхідно швидко реагувати на зміни або визначати оптимальні стратегії. Робот може використовувати алгоритми ШІ для прийняття рішень під час навігації складними середовищами або для автоматичної адаптації до змінних умов у виробничому процесі.

Інтерфейси на основі ШІ можуть використовувати технології обробки природної мови для розпізнавання голосових команд, а також технології комп'ютерного зору для розпізнавання жестів. Це дозволяє створювати інтуїтивні інтерфейси, де користувач може взаємодіяти з роботом без фізичних контролерів або кнопок. Користувач може давати роботу голосові команди, такі як "підніми об'єкт" або "рухайся вперед", а система на основі ШІ забезпечить точне розпізнавання команд і їх виконання.

ШІ дозволяє автоматизувати рутинні процеси або завдання, що виконуються роботом, знижуючи потребу в постійному контролі з боку оператора. Інтерфейс на основі ШІ може самостійно аналізувати ситуацію та виконувати завдання без участі людини. Робот може автоматично виконувати такі дії, як переміщення вантажів у складі, на основі аналізу даних про їх розташування та оптимальні шляхи переміщення [18].

Основні технології для реалізації інтерфейсів на основі ШІ.

Машинне навчання дозволяє системам аналізувати та навчатися на даних, що надходять у режимі реального часу.

Інтерфейси можуть використовувати ML для аналізу поведінки користувачів і адаптації до їхніх потреб.

Використання алгоритмів ML для оптимізації траєкторій руху робота в складних середовищах або адаптація інтерфейсу під індивідуальні переваги користувача.

Глибинні нейронні мережі використовуються для складного аналізу великих обсягів даних. Ці мережі можуть розпізнавати патерни в даних і приймати рішення на основі отриманих результатів.

Глибинне навчання також використовується для розпізнавання голосу, зображень та жестів. Інтерфейс може використовувати глибинне навчання для розпізнавання складних жестів або міміки користувача, що дозволяє керувати роботом за допомогою природних рухів.

NLP використовується для розуміння голосових команд або текстових запитів користувача. Ця технологія дозволяє системам на основі ШІ сприймати та інтерпретувати людську мову, а також реагувати на неї. Користувач може давати складні голосові команди, наприклад, "перевірити всі сенсори і звітувати про їхній стан", і система інтерпретує цю команду та виконає запит.

Комп'ютерний зір використовується для аналізу зображень і відео з камер робота. Це дозволяє інтерфейсам розпізнавати об'єкти, рухи та ситуації в реальному середовищі, що допомагає роботу орієнтуватися та взаємодіяти з об'єктами. Робот може розпізнавати перешкоди або інші об'єкти на своєму шляху, автоматично коригуючи свої дії або попереджаючи користувача.

Створення інтерфейсів на основі ШІ вимагає значних ресурсів та спеціалізованих знань. Для роботи складних моделей ШІ необхідні потужні обчислювальні системи.

Сучасні інтерфейси для керування роботами охоплюють широкий спектр технологій: від традиційних графічних інтерфейсів до інноваційних рішень на основі штучного інтелекту, доповненої та віртуальної реальності. Вибір інтерфейсу залежить від специфіки застосування робота, умов експлуатації та потреб користувача. Інтерфейси на основі ШІ, AR та жестового управління стають все більш популярними завдяки своїй адаптивності, інтуїтивності та інтерактивності.

Аналіз переваг та недоліків сучасних інтерфейсів для керування роботами наведений в таблиці 1.1

Таблиця 1.1 – Аналіз сучасних інтерфейсів для керування роботами

Тип інтерфейсу	Опис	Переваги	Недоліки
Графічний інтерфейс користувача	Інтерфейс, що дозволяє керувати роботом через графічні елементи на екрані, такі як кнопки, меню	Інтуїтивність, доступність, простота використання	Вимагає екрану, фізичного доступу до пристрою
Голосовий інтерфейс	Керування роботом через голосові команди, розпізнавання мовлення з використанням технологій NLP	Зручність, можливість безконтактної взаємодії, простота використання	Можливі проблеми з точністю розпізнавання, шумове забруднення середовища
Інтерфейс на основі жестів	Керування роботом через розпізнавання жестів або рухів користувача.	Безконтактність, природність взаємодії	Потребує спеціального обладнання для розпізнавання, обмеження точності
Інтерфейс доповненої реальності	Інтеграція віртуальних елементів у реальний світ для візуалізації та керування роботами.	Інтерактивність, покращена візуалізація процесів	Високі вимоги до обладнання, вартість впровадження

1.4 Компоненти автономних роботизованих систем

Автономні роботизовані системи складаються з декількох взаємопов'язаних підсистем, які забезпечують їхню здатність до самостійного функціонування, прийняття рішень та адаптації до змінних умов середовища.

Основні компоненти автономних роботизованих систем є сенсори та модулі збору даних, обчислювальний блок, програмне забезпечення та алгоритми ШІ, система навігації та контролю руху, система живлення та інтерфейс користувача.

Сенсори є невід'ємною частиною робототехнічних систем, оскільки вони забезпечують збирання даних про навколишнє середовище і стан самого робота. Універсальні сенсори дозволяють роботам адаптуватися до різних умов, виконувати різноманітні завдання та реагувати на зміни в реальному часі. Дослідження універсальних типів сенсорів спрямоване на розуміння їх характеристик, застосувань та можливостей інтеграції в робототехнічні системи [19].

Основні універсальні типи сенсорів:

- сенсори відстані (англ. Distance Sensors);
- сенсори руху (англ. Motion Sensors);
- сенсори дотику (англ. Tactile Sensors);
- сенсори зображень (англ. Image Sensors);
- сенсори тиску (англ. Pressure Sensors);
- сенсори температури (англ. Temperature Sensors).

Сенсори відстані (англ. Distance Sensors) вимірюють відстань до об'єктів у навколишньому середовищі.

Вони можуть використовувати різні технології, такі як ультразвук, інфрачервоне випромінювання або лазери (лідар). Ультразвукові сенсори відправляють звукові хвилі і вимірюють час їх повернення після відбиття від об'єкта. Інфрачервоні сенсори вимірюють інтенсивність відбитого інфрачервоного світла.

Лідар (англ. LiDAR) використовує лазери для вимірювання відстані з високою точністю. Застосовуються у роботах для навігації, уникнення перешкод, картографії середовища.

Система наведення лазером відображена на рис. 1.2.

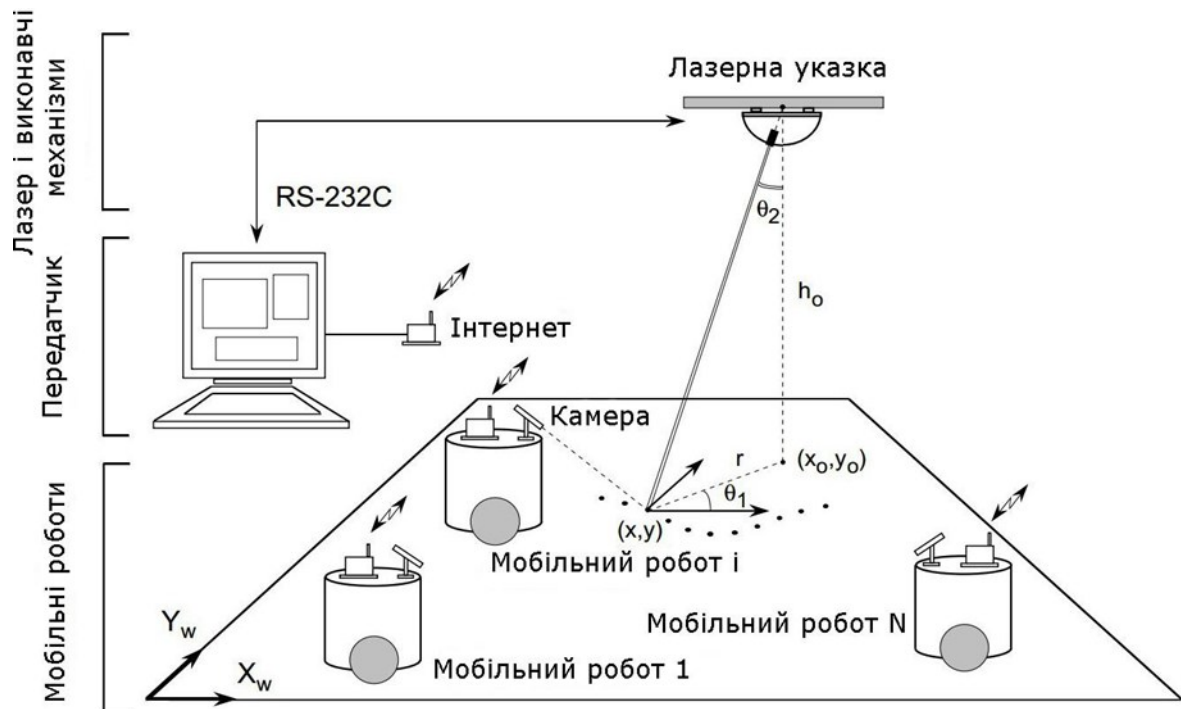


Рисунок 1.2 – Лазерна система наведення робота

Сенсори руху (англ. Motion Sensors) використовуються для виявлення або вимірювання руху робота або об'єктів у середовищі. Ці сенсори можуть визначати як власний рух робота, так і рух зовнішніх об'єктів. Акселерометри вимірюють прискорення робота в різних напрямках. Гіроскопи визначають кутове прискорення і допомагають визначати орієнтацію робота. Інклінометри вимірюють нахил відносно землі.

Застосовуються для контролю руху роботів, стабілізації під час руху, управління дронами, виявлення руху об'єктів.

Наприклад, акселерометри та гіроскопи використовуються у роботах для стабілізації під час руху по нерівних поверхнях.

Сенсори дотику (англ. Tactile Sensors) вимірюють фізичний контакт робота з об'єктами або поверхнями. Вони можуть визначати силу натискання або контакт з різними матеріалами. Резистивні сенсори дотику вимірюють зміну опору при тиску на поверхню сенсора. Ємнісні сенсори використовують зміну ємності для визначення контакту або наближення об'єкта.

П'єзоелектричні сенсори генерують електричний сигнал у відповідь на механічний тиск.

Застосовуються в маніпуляторах для точного захоплення об'єктів, контролю сили натискання, а також для забезпечення безпеки під час взаємодії з людьми. Робот, оснащений сенсорами дотику, може точно визначити силу, з якою потрібно утримувати крихкі об'єкти.

Сенсори зображень (англ. Image Sensors) захоплюють візуальні дані з навколишнього середовища, які можуть бути оброблені для отримання інформації про об'єкти, їх розташування та рух. Камери RGB захоплюють зображення в стандартних кольорових форматах, які можна аналізувати для розпізнавання об'єктів. Камери глибини (англ. Depth Cameras) захоплюють як зображення, так і дані про глибину, що дозволяє створювати 3D-моделі навколишнього середовища. Тепловізори вимірюють інфрачервоне випромінювання і можуть виявляти температурні відмінності об'єктів. Використовуються для виявлення об'єктів, навігації, розпізнавання образів, аналізу оточення. Камери глибини використовуються для побудови тривимірної карти приміщення, що дозволяє роботу орієнтуватися в просторі та уникати перешкод.

Сенсори тиску (англ. Pressure Sensors) вимірюють силу тиску на певну поверхню. Вони можуть бути як у вигляді мембран, які реагують на тиск повітря або рідин, так і у вигляді контактних сенсорів. Мембранні сенсори реагують на зміни тиску у рідинах або газах. Ємнісні сенсори тиску використовують зміну ємності при зміні тиску на поверхню. Застосовуються для контролю стану середовища (наприклад, тиску в системах гідравліки), виявлення дотику або натискання, використання в тактильних системах.

Наприклад, сенсори тиску використовуються в роботизованих маніпуляторах для точного управління процесами, що вимагають контролю сили натискання.

Сенсори температури (англ. Temperature Sensors) вимірюють температуру об'єктів або середовища. Вони можуть використовуватися як для вимірювання температури робота, так і для оцінки зовнішніх умов. Термістори вимірюють зміни опору залежно від температури. Інфрачервоні сенсори температури захоплюють теплове випромінювання від об'єктів для вимірювання температури без фізичного

контакту. Застосовуються для моніторингу стану роботів, контроль температури в середовищі, виявлення перегрівання робота або його компонентів.

Використання інфрачервоних сенсорів для моніторингу температури в умовах, де потрібен контроль за станом оточення або об'єктів.

Основна частина автономної роботизованої системи, яка відповідає за обробку інформації, отриманої від сенсорів – це обчислювальний блок. Саме він забезпечує можливість аналізу великих обсягів даних у реальному часі, здійснення складних обчислень і прийняття рішень на основі цих даних.

CPU (Central Processing Unit) – центральний процесор обробляє загальні завдання, координує роботу всієї системи та виконує обчислення середньої складності. Він забезпечує управління всіма компонентами системи та передачу даних між модулями. CPU підходить для виконання основних алгоритмів, контролю стану сенсорів, а також загальної логіки управління.

GPU (Graphics Processing Unit) – графічний процесор є потужним інструментом для обробки великих масивів даних паралельно. GPU часто використовується для обробки відео та зображень, що необхідно для задач комп'ютерного зору.

GPU підходить для обробки даних із сенсорів зображень, таких як камери, забезпечуючи розпізнавання об'єктів, аналіз навколишнього середовища та виявлення перешкод.

TPU (Tensor Processing Unit) – це спеціалізовані процесори, оптимізовані для завдань машинного навчання, особливо глибокого навчання та обробки нейронних мереж. TPU забезпечують високу продуктивність і швидкість обробки для алгоритмів штучного інтелекту.

TPU підходять для задач, що потребують високої обчислювальної потужності при виконанні моделей машинного навчання, таких як розпізнавання образів, прогнозування та самонавчання.

NPU (Neural Processing Unit) – процесор для обробки нейронних мереж, який спеціалізується на задачах ШІ. NPU інтегрується для прискорення обчислень, пов'язаних із навчанням і розпізнаванням моделей.

Перевага NPU полягає в його здатності значно знижувати енергоспоживання порівняно з GPU або TPU, що є важливим для портативних і мобільних автономних систем.

FPGA (Field Programmable Gate Array) – програмовані інтегральні схеми, які можуть бути налаштовані для виконання специфічних завдань, таких як обробка сигналів та складних алгоритмів ШІ.

FPGA забезпечують гнучкість, дозволяючи налаштовувати обробку під конкретні задачі, що особливо корисно в автономних системах, де потрібна висока швидкість і точність виконання. Програмне забезпечення та алгоритми ШІ:

Програмне забезпечення включає алгоритми, які аналізують дані з сенсорів, розпізнають об'єкти, будують маршрути, передбачають можливі перешкоди і формують команди для виконання завдань. Машинне навчання дозволяє системам покращувати ефективність на основі досвіду, що допомагає роботам адаптуватися до нових умов.

Система навігації та контролю руху. Включає елементи, які забезпечують точне позиціонування в просторі, уникнення перешкод і виконання рухових завдань. Завдяки цим модулям автономні системи можуть пересуватися в складних умовах, наприклад, у приміщеннях, зонах з перешкодами або на відкритій місцевості.

Батареї або інші джерела енергії, які забезпечують живлення для всіх компонентів робота. Автономні системи зазвичай обладнані акумуляторами з можливістю автоматичної зарядки або заміни, що дозволяє їм працювати протягом тривалого часу.

Інтерфейс користувача є важливим компонентом автономної роботизованої системи, який забезпечує можливість ефективного взаємодії між оператором і системою. Інтерфейс дозволяє оператору моніторити стан роботи, керувати його діями, а також отримувати зворотний зв'язок щодо виконання завдань.

Основні функції інтерфейсу користувача.

Моніторинг стану системи. Інтерфейс показує інформацію про поточний стан робота, включаючи рівень заряду акумулятора, робочу температуру, стан сенсорів, місцезнаходження та швидкість. Оператор може контролювати основні параметри і швидко реагувати на можливі зміни чи проблеми.

GUI може відображати картину оточення робота, створену на основі даних із сенсорів (наприклад, камер, LiDAR або ультразвукових датчиків).

Це дозволяє оператору бачити потенційні перешкоди, маршрути та ключові об'єкти, з якими робот може взаємодіяти, що особливо корисно для віддаленого управління.

Оператор може задавати команди, обирати режими роботи або маршрути для робота безпосередньо через інтерфейс. Дозволяє налаштовувати параметри роботи, такі як швидкість, чутливість сенсорів, межі дій, що допомагає адаптувати систему під конкретні умови або вимоги задачі.

GUI сповіщає оператора про будь-які збої, несправності, низький рівень заряду, перешкоди на шляху тощо. Повідомлення дозволяють швидко реагувати на помилки, запобігати можливим аварійним ситуаціям і вчасно виконувати технічне обслуговування.

Інтерфейс відображає поточне завдання робота, його статус, етап виконання та орієнтовний час завершення. Це допомагає оператору відслідковувати прогрес роботи і, за потреби, коригувати дії робота або змінювати пріоритети завдань.

GUI може зберігати історію виконаних дій, помилок, змін у параметрах, що дозволяє аналізувати роботу робота з часом. Історія подій є корисною для діагностики несправностей, аналізу продуктивності та оптимізації майбутніх завдань.

Інтерфейс користувача може бути доступний через віддалені пристрої, що дозволяє операторам контролювати систему на відстані через мобільний додаток, веб-інтерфейс або інший пристрій.

1.5 Алгоритми ШІ в автономних роботизованих системах

Алгоритми штучного інтелекту відіграють ключову роль у підвищенні адаптивності, автоматизації та інтуїтивності взаємодії користувачів з інтерфейсами.

Штучний інтелект в інтерфейсах дозволяє системам навчатися, реагувати на дії користувачів та адаптуватися до зміни умов у режимі реального часу.

Це значно полегшує роботу з робототехнічними системами, роблячи інтерфейси більш "розумними" і корисними для користувачів.

Розглянемо основні алгоритми ШІ, що використовуються в інтерфейсах для різних задач.

Алгоритми класифікації використовуються для розпізнавання і класифікації вхідних даних, таких як жести користувача, голосові команди або зображення. Це дозволяє системам приймати рішення на основі цих класифікацій [20].

Алгоритм K-Nearest Neighbors (KNN) використовується для класифікації на основі близькості до "сусідів" у просторі даних. Призначений для розпізнавання жестів або голосових команд. Коли користувач вводить нову команду, алгоритм визначає, до якої класифікаційної групи вона належить, і виконує відповідну дію.

Support Vector Machines (SVM) використовується для класифікації в багатовимірному просторі з використанням гіперплощин для поділу класів. Застосовується для класифікації жестів або образів у інтерфейсах, щоб точно визначати дії користувача. Алгоритми кластеризації використовуються для поділу даних на групи на основі їх подібностей без попереднього маркування.

Алгоритм K-Means визначає центри кластерів у наборі даних і групує подібні дані разом. Використовується для виявлення патернів поведінки користувачів в інтерфейсі. Наприклад, можна виявити, як різні користувачі взаємодіють з системою і адаптувати інтерфейс під їхні звички [21].

DBSCAN (англ. Density-Based Spatial Clustering of Applications with Noise) кластеризує дані на основі щільності точок у просторі. Відмінністю є здатність ігнорувати "шум" або викиди. Використовується для ідентифікації аномалій у поведінці користувачів або для виявлення різних типів користувачів за їхніми діями.

Алгоритми регресії використовуються для прогнозування та аналізу взаємозв'язків між змінними.

Лінійна регресія використовується для прогнозування значення залежної змінної на основі однієї або кількох незалежних змінних.

Застосовується для прогнозування поведінки користувачів, наприклад, визначення, які функції інтерфейсу будуть використовуватися найбільш активно.

Поліноміальна регресія використовується для моделювання нелінійних відносин між змінними.

Застосовується для прогнозування складних взаємодій у інтерфейсі, таких як аналіз складних жестів або сценаріїв.

Конволюційні нейронні мережі (CNN) – це нейронні мережі, спеціалізовані на обробці зображень та відео. Вони можуть автоматично виділяти особливості зображення (краї, текстури, форми), що робить їх ефективними для аналізу візуальних даних. Камери можуть захоплювати рухи користувачів, і CNN буде використовуватися для класифікації цих рухів як жестів для керування роботом. Інтерфейс може використовувати розпізнавання обличчя для автентифікації користувачів або забезпечення індивідуальних налаштувань.

Архітектура CNN проілюстрована на рис. 1.3. Мережа складається з восьми шарів. Перші п'ять шарів є згортковими (C1-5), а наступні три – повнозв'язковими (FC6-8).

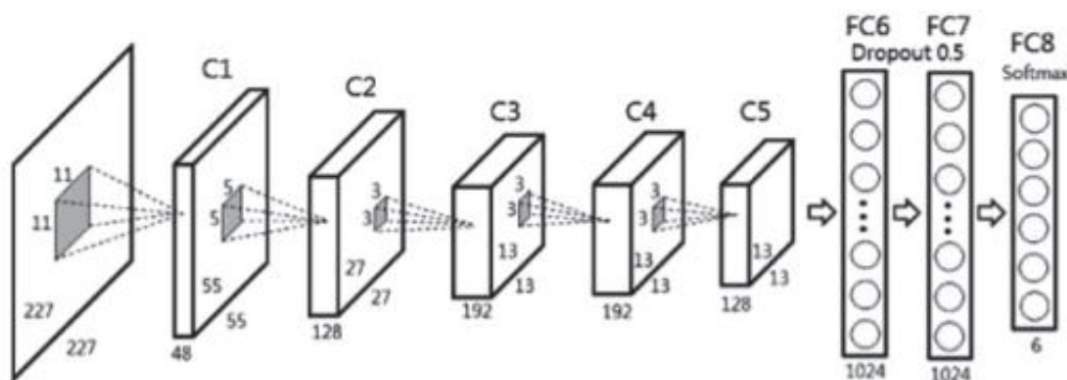


Рисунок 1.3 – Архітектура CNN

Описана модель згорткової нейронної мережі CNN найкраще підходить для розпізнавання виразів обличчя для шести емоцій. Структура запропонованого алгоритму має хорошу генералізацію та ефективність класифікації.

Рекурентні нейронні мережі (RNN) та довготривала короткочасна пам'ять (LSTM) спеціалізуються на роботі з часовими рядами та послідовними даними. Вони зберігають інформацію про попередні стани, що робить їх ефективними для аналізу послідовностей, таких як мова або рухи [22].

Використовується для обробки послідовних звукових сигналів і розуміння голосових команд користувачів та прогнозуванні дій користувачів. Система може прогнозувати наступні дії користувача на основі попередніх, що дозволяє робити інтерфейс більш інтуїтивним та адаптивним.

Автокодувальники (англ. Autoencoders) – це нейронні мережі, які навчаються стискати вхідні дані та потім відновлювати їх. Вони можуть бути використані для виявлення аномалій або для зменшення шуму в даних. Використовуються для виявлення нетипової поведінки користувачів або потенційних збоїв у системі. Можуть бути використані для вивчення найбільш ефективних патернів взаємодії з інтерфейсом та їхньої оптимізації.

Генеративно-змагальні мережі (GAN) складаються з двох нейронних мереж – генератора і дискримінатора. Генератор створює нові дані, тоді як дискримінатор оцінює їхню автентичність. Застосовується для створення нових сценаріїв взаємодії та поліпшення користувацького досвіду. GAN можуть створювати нові віртуальні середовища або сценарії для тестування різних дій користувачів в інтерфейсі.

Генерація нових рішень або ідей для покращення взаємодії користувачів з інтерфейсом.

Алгоритми обробки природної мови використовуються для розуміння і обробки людської мови. Це дозволяє створювати голосові інтерфейси, чат-боти та інші системи, що взаємодіють з користувачами через текст або мову.

Алгоритм Word2Vec використовується для перетворення слів у вектори, що дозволяє моделі краще розуміти зв'язки між словами.

Алгоритм BERT (англ. Bidirectional Encoder Representations from Transformers). Моделі на основі BERT можуть розуміти контекст речень, що робить їх ефективними для розпізнавання запитів користувачів у чат-ботах або голосових інтерфейсах. Використовуються для інтерактивного спілкування з користувачами через текстові або голосові повідомлення [23].

Чат-боти можуть надавати інформацію, допомагати у виконанні завдань або обробляти запити. ШІ в інтерфейсах може використовуватися для розпізнавання голосових команд та виконання завдань, що полегшує взаємодію з системами.

Алгоритми ШІ забезпечують новий рівень адаптивності та автоматизації для інтерфейсів, роблячи взаємодію з системами більш природною, інтуїтивною та ефективною.

У теперішній час безпека даних є одним з найважливіших параметрів технології, оскільки користувач може передавати конфіденційні дані.

Використання штучного інтелекту в інтерфейсах керування роботами значно підвищує адаптивність, інтуїтивність і автоматизацію взаємодії з системами. Однак інтеграція ШІ в інтерфейси також ставить нові виклики щодо забезпечення безпеки, конфіденційності та захисту від зовнішніх атак. Це робить критично важливим застосування ефективних механізмів захисту для захисту систем і даних.

Основні загрози при використанні ШІ в інтерфейсах є зловживання ШІ-алгоритмами; атаки на дані навчання ; конфіденційність даних та вразливості в інтерфейсах.

Шкідливі актори можуть маніпулювати алгоритмами ШІ для отримання небажаного доступу або введення неправдивих даних для впливу на рішення системи. Враховуючи, що моделі ШІ часто навчаються на великих наборах даних, атаки можуть бути спрямовані на маніпуляцію даними або їх підробку, що призведе до викривлення результатів або неправильних рішень. ШІ може обробляти великі обсяги конфіденційної інформації, такої як особисті дані користувачів, що створює загрозу порушення конфіденційності у разі витоку або крадіжки цих даних. Інтерфейси, засновані на ШІ, можуть мати вразливості, які зловмисники можуть використовувати для несанкціонованого доступу або маніпуляцій з системою.

Інтеграція механізмів захисту при використанні ШІ в інтерфейсах є важливою частиною забезпечення безпеки систем керування роботами. Захист даних, моделей, а також контроль доступу та моніторинг системи допомагають знизити ризики та забезпечити безпечну і надійну роботу інтерфейсів. Це сприяє підвищенню довіри до технологій ШІ та дозволяє уникнути багатьох потенційних загроз, пов'язаних з їх використанням.

Впровадження механізмів захисту при використанні ШІ в інтерфейсах допоможе захистити від несанкціонованого доступу та зберегти конфіденційність та цілісність даних.

Моніторинг системи та виявлення аномалій допомагають швидко реагувати на потенційні загрози або атаки.

Механізми захисту при використанні ШІ в інтерфейсах:

- захист даних навчання та алгоритмів;
- механізми захисту моделей ШІ;
- контроль доступу та аутентифікація;
- моніторинг та виявлення аномалій;
- захист конфіденційності.

Захист даних навчання та алгоритмів (англ. Data and Algorithm Protection).

Для захисту моделей ШІ важливо забезпечити безпеку даних, на яких навчаються алгоритми.

Якщо вхідні дані скомпрометовані, це може призвести до викривлення моделі. Використання шифрування для зберігання та передачі даних, що використовуються для навчання ШІ. Це забезпечує захист даних від несанкціонованого доступу під час їх використання в системах ШІ.

Впровадження механізмів автентифікації для перевірки достовірності даних перед їхнім використанням для навчання моделі. Використання хеш-функцій та цифрових підписів для забезпечення того, що навчальні дані не були змінені.

Механізми захисту моделей ШІ (англ. Model Protection).

Моделі ШІ самі можуть стати об'єктом атаки, де зловмисники можуть маніпулювати або копіювати моделі для власної вигоди. Моделі ШІ можна зашифрувати для захисту від несанкціонованого доступу або викрадення. Це допоможе захистити моделі під час виконання або зберігання. Захист від атаки через зворотне проектування (англ. Model Inversion Attack) дозволяє зловмисникам відновлювати початкові дані на основі відповідей моделі. Для захисту можуть використовуватися механізми диференційованої приватності (англ. Differential Privacy), які додають шум до відповідей моделі, не впливаючи на її точність.

Контроль доступу та аутентифікація (англ. Access Control and Authentication).

Оскільки інтерфейси з використанням ШІ можуть містити конфіденційні дані або критично важливі функції, необхідно забезпечити надійну систему аутентифікації для користувачів та контроль доступу до різних частин системи [24].

Двофакторна аутентифікація (2FA) забезпечує додатковий рівень безпеки, вимагаючи від користувача введення другого способу підтвердження своєї особи.

Процес аутентифікації відображено на рис. 1.4.

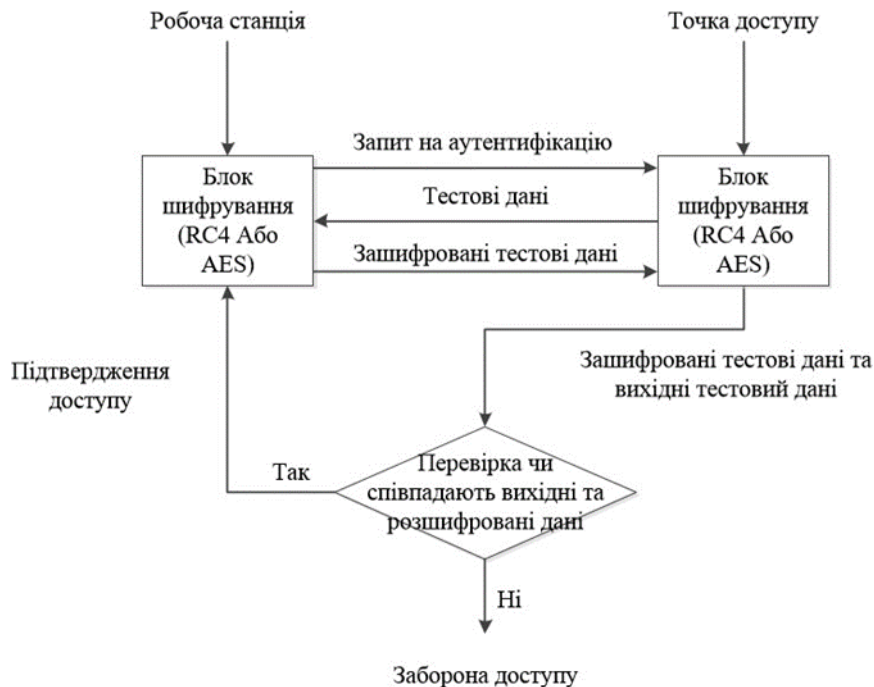


Рисунок 1.4 – Процес аутентифікації

Рольовий контроль доступу (англ. Role-Based Access Control, RBAC). Визначення ролей і прав доступу до різних функцій інтерфейсу залежно від статусу користувача. Це дозволяє обмежити доступ до критичних функцій тільки авторизованим користувачам.

Контроль доступу на основі поведінки (англ. Behavior-Based Access Control), це моніторинг поведінки користувачів і виявлення аномальної активності для запобігання потенційним загрозам.

Моніторинг та виявлення аномалій (англ. Anomaly Detection and Monitoring). Системи на основі ШІ можуть бути піддані різним типам атак, тому необхідно реалізувати постійний моніторинг поведінки системи та виявлення аномалій.

Використання ШІ для виявлення аномальних поведінкових патернів у системі. Це дозволяє виявляти підозрілі дії, які можуть вказувати на атаку або неправомірне використання системи.

Зберігання журналів подій і дій користувачів для подальшого аналізу та виявлення потенційних загроз.

Захист конфіденційності (англ. Privacy Protection). Інтерфейси на основі ШІ часто працюють з конфіденційними даними, тому важливо забезпечити захист таких даних від несанкціонованого доступу або розголошення.

Диференційована приватність (англ. Differential Privacy) – технологія, яка додає випадковий шум до результатів моделі ШІ, що дозволяє зберігати приватність даних користувачів, зберігаючи при цьому точність результатів.

Знеособлення даних (англ. Data Anonymization) – знеособлення особистої інформації у наборах даних для навчання ШІ, щоб уникнути прив'язки даних до конкретних користувачів.

Використання шифрування для захисту конфіденційної інформації під час її зберігання або передачі між різними частинами системи.

Захист від атак на інтерфейс користувача (англ. Interface Attack Protection). Інтерфейси на основі ШІ можуть піддаватися специфічним атакам, які спрямовані на отримання контролю або маніпуляції з діями користувача.

Для захисту від атак типу "людина посередині" (англ. Man-in-the-Middle, MitM) використовується шифрування трафіку між клієнтом і сервером для запобігання перехопленню або модифікації переданих даних. Також рекомендується використовувати сертифікати SSL/TLS, регулярне оновлення криптографічних алгоритмів і перевірку автентичності серверів для підвищення рівня безпеки.

Для захисту від атак на інтерфейси API необхідна реалізація механізмів аутентифікації та шифрування даних для захисту API, що використовуються інтерфейсом. Додатково рекомендується впровадження обмежень доступу на основі IP-адрес, токенів доступу, а також використання протоколів OAuth 2.0 або JWT для надання безпечного доступу до API.

Для верифікації команд користувача реалізуються механізми підтвердження критичних дій користувача перед їх виконанням для запобігання помилковим командам або зловживанням.

Система зберігає всі критичні дії для можливості їх перевірки або відновлення у разі помилки. Це досягається через журналювання подій із зазначенням часу, користувача та деталей операцій. Реалізація ролей і прав доступу дозволяє зменшити ризик виконання критичних дій некваліфікованими користувачами, а також забезпечує мінімізацію привілеїв, надаючи користувачам лише ті права, які необхідні для виконання їхніх завдань.

1.6 Висновки до першого розділу

У першому розділі проведено детальний аналіз вихідних даних та визначено ключові вимоги до інтерфейсів для керування роботами. Були розглянуті сучасні інноваційні підходи, такі як адаптивні інтерфейси, інтерфейси на основі природної мови, жестові інтерфейси та інтерфейси доповненої реальності. Кожна технологія була проаналізована з точки зору принципів її роботи. На основі цього аналізу були виявлені основні переваги та недоліки кожного підходу.

Особливу увагу було приділено алгоритмам штучного інтелекту (ШІ), які є основою для адаптивності і автоматизації роботизованих систем. Використання ШІ дозволяє системам самостійно приймати рішення і підвищує зручність для користувачів, які можуть контролювати роботу автономних роботів інтуїтивно і з мінімальним втручанням. Розглянуто механізми захисту, що забезпечують безпечно впровадження ШІ, захист даних і мінімізацію ризиків, пов'язаних із можливими загрозами.

Таким чином, отримані результати дозволяють зробити висновок, що вибір конкретного типу інтерфейсу залежить від завдань, які стоять перед системою керування роботами, і вимог до ефективності, зручності та точності керування.

2 РОЗРОБЛЕННЯ АВТОНОМНОЇ РОБОТИЗОВАНОЇ СИСТЕМИ

2.1 Структура автономної роботизованої системи

Розроблена автономна роботизована система складається з декількох основних модулів, кожен із яких відповідає за певний аспект роботи системи.

Сенсорна підсистема.

Сенсорна підсистема є джерелом інформації про оточення та стан системи.

Лідар визначає відстань до об'єктів шляхом аналізу відбитого лазерного сигналу. Забезпечує тривимірне моделювання середовища. Використовується для виявлення перешкод.

Камера збирає візуальну інформацію про середовище. Використовується для розпізнавання об'єктів (за допомогою алгоритмів комп'ютерного зору).

GPS забезпечує географічну локалізацію. Допомагає в глобальному позиціонуванні на відкритій місцевості.

Інерційний модуль (IMU) вимірює прискорення та кутову швидкість. Дозволяє відстежувати орієнтацію та рух системи.

Ультразвукові сенсори використовуються для близької навігації та уникнення зіткнень.

Модуль обробки даних.

Фільтр Калмана використовується для згладжування шуму в даних.

Алгоритми об'єднання даних (sensor fusion) інтегрують інформацію з різних джерел.

Побудова карти середовища (SLAM). Алгоритм SLAM (Simultaneous Localization and Mapping) одночасно локалізує систему та будує карту. Забезпечує актуальну інформацію про середовище в реальному часі.

Навігаційний модуль.

Модуль відповідає за визначення оптимального маршруту для руху до цільової точки з урахуванням перешкод.

Алгоритм A^* знаходить оптимальний шлях на основі карти середовища.

Dijkstra використовується для точного планування, якщо потрібно обчислити мінімальну витрату енергії. Динамічне оновлення маршруту враховує нові перешкоди, що виникають під час руху. Перебудовує маршрут у реальному часі.

Модуль ухвалення рішень.

Відповідає за вибір наступних дій на основі поточного стану системи та отриманих даних. Виконує аналіз даних від сенсорів. Приймає рішення: рух вперед, зупинка, обхід перешкоди. Впровадження нейронних мереж для передбачення руху перешкод або оптимальних траєкторій.

Контролер руху.

Модуль забезпечує точне управління виконавчими механізмами системи.

PID-контролер використовується для підтримки стабільної швидкості та позиції.

MPC (Model Predictive Control) планує траєкторію на декілька кроків уперед, враховуючи обмеження на рух.

Система управління двигунами генерує команди для двигунів, наприклад: змінює швидкість або напрямок.

Інтерфейс користувача.

Адаптивний інтерфейс забезпечує зручну взаємодію з системою для оператора.

Відображення карти середовища, поточного маршруту, стану роботизованої системи. Забезпечує можливість задавати цільову точку, налаштовувати параметри руху. Інтерфейс підлаштовується під рівень знань оператора.

Модуль навчання та адаптації.

Використовується для вдосконалення алгоритмів ухвалення рішень на основі досвіду роботи системи. Система автоматично налаштовує параметри руху під зміну зовнішніх умов.

При взаємодії між модулями дані із сенсорів надходять у модуль обробки, де вони очищаються та об'єднуються. Модуль планування використовує оброблені дані для створення маршруту. Модуль ухвалення рішень визначає поточну поведінку системи на основі маршруту та сенсорної інформації.

Контролер руху отримує команди для виконання конкретних дій. Інтерфейс користувача дозволяє оператору переглядати стан системи та вносити корективи.

Структура розробленої автономної роботизованої системи представлена на рис. 2.1.

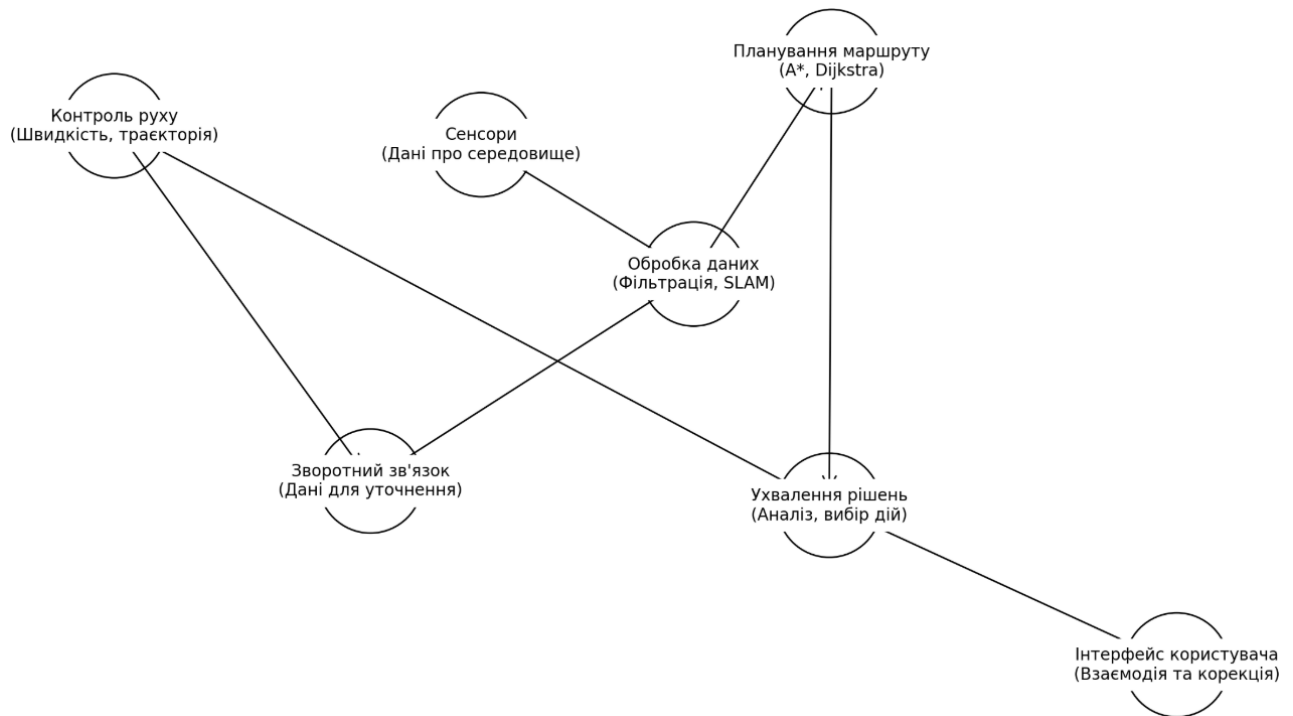


Рисунок 2.1– Структура розробленої автономної роботизованої системи

Структура дозволяє представити автономну роботизовану систему як комплекс модулів, що працюють разом для забезпечення автономності, точності, адаптивності та зручності. Ця структура може бути використана для подальшого вдосконалення системи або розробки нових функціональних можливостей.

2.2 Алгоритм роботи автономної роботизованої системи

Алгоритм роботи автономної роботизованої системи, що розробляється описує послідовність дій системи для виконання завдань у автономному режимі. Він складається з етапів, що охоплюють обробку сенсорних даних, планування, ухвалення рішень, контроль руху та зворотний зв'язок [25].

До початкового етапу входить перевірки системи, завдання цільової точки або маршруту та побудова початкової карти середовища.

До перевірки системи входить:

- тестування апаратних компонентів (сенсори, приводи, контролери);
- калібрування сенсорів (лідар, GPS, IMU тощо);
- завантаження конфігурацій (карта місцевості, цільові параметри).

Для завдання цільової точки або маршруту оператор задає кінцеву точку через інтерфейс користувача або система отримує її автоматично.

Для побудови початкової карти середовища система формує карту на основі початкових даних від сенсорів.

Основний цикл виконується до досягнення цілі або завершення задачі.

Лідар, камера, GPS, IMU передають інформацію про оточення та стан системи.

Дані включають відстань до перешкод, візуальну інформацію для розпізнавання об'єктів та точні координати та орієнтацію робота.

Далі використовується фільтр Калмана для усунення неточностей.

Наступним будується карта. Алгоритм SLAM оновлює карту середовища в реальному часі, додаючи нові перешкоди.

Наступним кроком є планування маршруту. Проходить генерація оптимального маршруту на основі карти. Якщо з'являються нові перешкоди, система перебудовує маршрут у реальному часі.

Система аналізує маршрут та поточну ситуацію. PID або MPC забезпечує точне дотримання маршруту.

Контроль швидкості та орієнтації. Система отримує дані про виконані дії. Коригує рух, враховуючи відхилення від маршруту. Повідомляє оператору через інтерфейс користувача.

Інтерфейс користувача відображає поточний стан системи та карти та маршрут руху. Оператор може вносити корективи в маршрут або параметри руху.

Розроблений алгоритм охоплює всі ключові аспекти автономної роботи системи, що забезпечує її надійність, точність і зручність.

2.3 Розрахунок стійкості автономної роботизованої системи

Для розрахунку стійкості автономної роботизованої системи скористаємося підходами з теорії автоматичного управління [26].

Для автономної роботизованої системи розглянемо основні компоненти, які зазвичай включаються в її модель:

- електромеханічна частина;
- кінематична модель (рух системи в просторі);
- сенсорна система (зворотній зв'язок для управління);
- контролер (реалізація алгоритму управління).

Модель складається з диференційних рівнянь для кожного компонента системи.

Перехід до передатних функцій для автономної роботизованої системи виконується шляхом застосування перетворення Лапласа до диференційних рівнянь, отриманих у математичній моделі, за умови нульових початкових умов.

Електромеханічна частина.

Рівняння електричної підсистеми після перетворення Лапласа:

$$LsI(s) + RI(s) = V(s) - k_e\Omega(s) \quad (2.1)$$

Передатна функція для струму $I(s)$ від напруги $V(s)$:

$$I(s) = \frac{V(s) - k_e\Omega(s)}{Ls + R} \quad (2.2)$$

Рівняння механічної підсистеми після перетворення Лапласа:

$$Js\Omega(s) + b\Omega(s) = k_tI(s) - T_{load}(s) \quad (2.3)$$

Передатна функція для кутової швидкості $\Omega(s)$ твід струму $I(s)$:

$$\Omega(s) = \frac{k_tI(s) - T_{load}(s)}{Js + b} \quad (2.4)$$

Об'єднуючи ці рівняння, отримуємо передатну функцію $\Omega(s)$ від $V(s)$:

$$\Omega(s) = \frac{k_t \left(\frac{V(s)}{Ls+R} \right) - T_{load}(s)}{Js + b} \quad (2.5)$$

Кінематична модель.

Для поступального руху після перетворення Лапласа

$$ms^2 X(s) + b_x s X(s) = F_x(s) \quad (2.6)$$

Передатна функція для $X(s)$ від $F_x(s)$:

$$X(s) = \frac{F_x(s)}{ms^2 + b_x s} \quad (2.7)$$

Для обертального руху після перетворення Лапласа

$$Js^2 \Theta(s) + b_\theta s \Theta(s) = \tau(s) \quad (2.8)$$

Передатна функція для $\Theta(s)$ від $\tau(s)$:

$$\Theta(s) = \frac{\tau(s)}{Js^2 + b_\theta s} \quad (2.9)$$

Сенсорна система

Вихід $y(t)$ пов'язаний зі станом $x(t)$ після перетворення Лапласа

$$Y(s) = CX(s) + DU(s) \quad (2.10)$$

Передатна функція для виходу $Y(s)$ від входу $U(s)$:

$$Y(s) = \left(C \cdot \frac{1}{ms^2 + b_x s} \right) U(s) \quad (2.11)$$

Контролер (ПД-алгоритм).

Рівняння ПД-контролера після перетворення Лапласа

$$U(s) = \left(K_p + \frac{K_i}{s} + K_d s \right) E(s) \quad (2.12)$$

Передатна функція для $U(s)$ від похибки $E(s)$:

$$U(s) = \frac{K_d s^2 + K_p s + K_i}{s} E(s) \quad (2.13)$$

Об'єднавши всі компоненти, отримуємо повну передатну функцію системи:

- вхід $V(s)$ -> вихід $\Omega(s)$ для електромеханіки;
- вхід $\tau(s)$ -> вихід $\Theta(s)$ для обертального руху;
- вхід $F_x(s)$ -> вихід $X(s)$ для поступального руху;
- контролер впливає на похибку $E(s) = R(s) - Y(s)$, де $R(s)$ – бажане значення.

Зазначені передатні функції дозволяють аналізувати стійкість, динамічні властивості та робити подальші синтез чи оптимізацію системи.

Застосування критерію Гурвіца.

Критерій Гурвіца визначає стійкість системи на основі характеристичного рівняння:

$$P(s) = a_0 s^n + a_1 s^{n-1} + a_2 s^{n-2} + \dots + a_{n-1} s + a_n = 0 \quad (2.14)$$

де, a_i – коефіцієнти характеристичного полінома.

Для системи бути стійкою (всі корені характеристичного рівняння повинні мати від'ємні дійсні частини), визначники Гурвіца мають бути додатними.

$$\Delta_1 > 0, \quad \Delta_2 > 0, \quad \dots, \quad \Delta_n > 0$$

Для характеристичного рівняння:

$$P(s) = s^4 + 3s^3 + 5s^2 + 7s + 2 = 0$$

Коефіцієнти: $a_0=1, a_1=3, a_2=5, a_3=7, a_4=2$.

$$H = \begin{bmatrix} 3 & 7 & 0 & 0 \\ 1 & 5 & 2 & 0 \\ 0 & 3 & 7 & 0 \\ 0 & 1 & 5 & 2 \end{bmatrix}$$

$$\Delta_1 = 3 > 0,$$

$$\Delta_2 = 8 > 0,$$

$$\Delta_3 = 38 > 0,$$

$$\Delta_4 = 174 > 0.$$

Система є стійкою, оскільки всі визначники Гурвіца додатні.

Діаграми Найквіста (рис. 2.2) відображає залежність уявної частини $\text{Im}(W(j\omega))$ передатної функції від її дійсної частини $\text{Re}(W(j\omega))$.

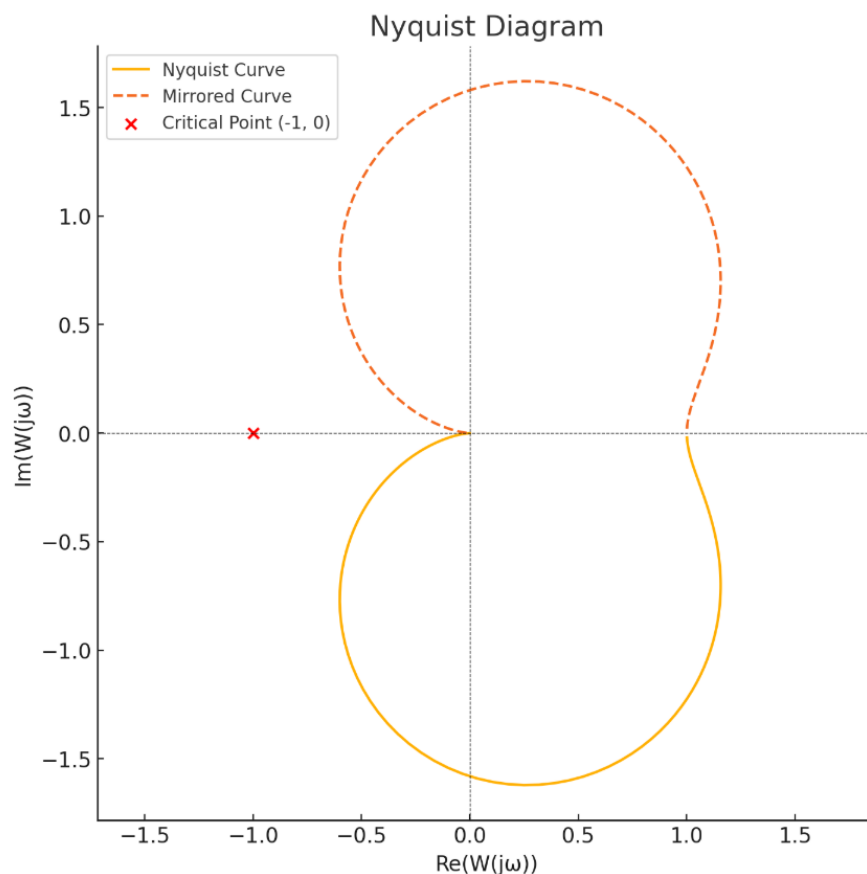


Рисунок 2.2 – Діаграми Найквіста

Для стійкості системи замкнутого контуру діаграма Найквіста не повинна охоплювати точку $(-1, 0)$, якщо система є стійкою у розімкнутому контурі.

У графіку жовта суцільна лінія наближається до критичної точки $(-1, 0)$, але не охоплює її. Це свідчить про те, що замкнена система може бути стабільною, якщо немає полюсів у правій півплощині.

Діаграма Михайлова (рис. 2.3) зображає залежність уявної частини характеристичного полінома $\text{Im}(P(j\omega))$ від дійсної частини $\text{Re}(P(j\omega))$, коли частота ω змінюється від 0 до нескінченності.

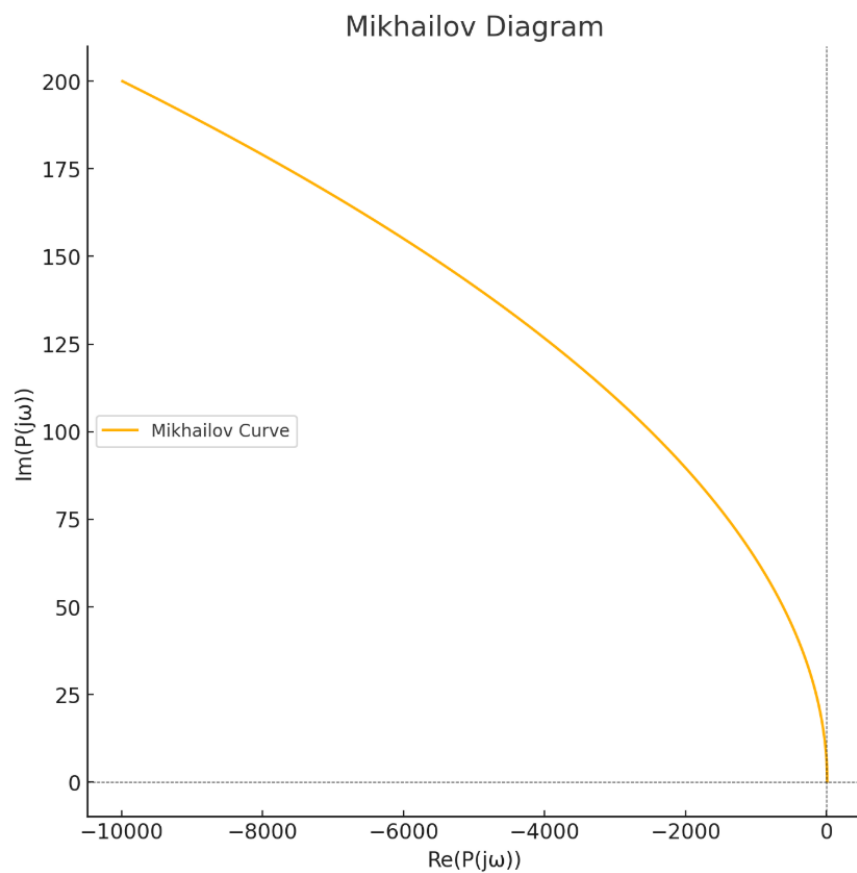


Рисунок 2.3 – Діаграма Михайлова

Крива починається в дійсній лівій півплощині $\text{Re}(P(j\omega))$. Уявна частина $\text{Im}(P(j\omega))$ поступово зменшується до нуля, що вказує на правильний напрямок руху. Крива не перетинає уявну вісь і залишається в лівій півплощині.

Система є стійкою, оскільки її крива відповідає умовам критерію Михайлова.

2.4 Висновки до другого розділу

У другому розділі роботи здійснено проектування та розробка автономної роботизованої системи. Створено функціональну структуру системи, яка включає сенсори, модуль обробки даних, модуль планування маршруту, блок ухвалення рішень, модуль контролю руху, зворотний зв'язок та інтерфейс користувача. Розроблено алгоритми, що охоплюють основні етапи функціонування автономної роботизованої системи.

Проведено аналіз стійкості автономної роботизованої системи за трьома критеріями. Критерій Гурвіца підтвердив, що всі корені характеристичного рівняння знаходяться у лівій півплощині, що гарантує стійкість. Діаграма Михайлова показала, що система відповідає всім умовам стійкості. Діаграма Найквіста підтвердила відсутність охоплення критичної точки $(-1, 0)$, що також свідчить про стійкість.

3 РОЗРОБЛЕННЯ ІНТЕРФЕЙСУ КЕРУВАННЯ РОБОТАМИ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

3.1 Програмні засоби для створення інтерфейсу керування роботами

Програмні засоби для створення інтерфейсу автономної роботизованої системи з використанням ШІ охоплює аналіз та розробку систем, що забезпечують автономне управління об'єктами за допомогою штучного інтелекту та алгоритмів комп'ютерного зору.

Використання Python та NumPy для створення автономного управління у симуляторі гри Mafia 2 є демонстрацією можливостей програмних рішень для автоматизації керування роботами у віртуальних середовищах.

Розробка інтерфейсу керування роботами вимагає використання низки інструментів і технологій, що забезпечують інтеграцію комп'ютерного зору, числових обчислень і управління периферійними пристроями.

Для реалізації автономного управління у симуляторі гри Mafia 2 використані наступні програмні засоби.

Python – високорівнева мова програмування, яка дозволила швидко розробляти та тестувати алгоритми завдяки великій кількості бібліотек.

NumPy – бібліотека для роботи з багатовимірними масивами та числовими обчисленнями, що дозволила ефективно маніпулювати даними зображень, перетворюючи їх в масиви для подальшої обробки.

OpenCV – бібліотека для комп'ютерного зору, що надала широкий спектр інструментів для обробки зображень, таких як фільтри Sobel та Canny, що дозволила виявляти контури об'єктів на зображеннях.

CTypes – бібліотека Python, яка забезпечила доступ до функцій Windows API для симуляції натискань клавіш, що є ключовим елементом в управлінні роботом у симуляції.

Для тестування та реалізації програмного забезпечення використані апаратні засоби, які дозволили обробляти візуальні дані в реальному часі.

Комп'ютер з ОС Windows для роботи з симулятором в грі Mafia 2, що потребує специфічних функцій Windows API, які використалися для симуляції натискань клавіш та керування грою.

Графічний процесор використався для пришвидшення обробки зображень, оскільки алгоритми комп'ютерного зору потребують значної обчислювальної потужності.

Модуль вводу (клавіатура) для взаємодії з грою та перевірки роботи автономного управління, симулюючи поведінку реального керування роботом.

3.2 Моделювання взаємодії користувачів з інтерфейсом

Моделювання взаємодії користувачів з інтерфейсом є важливою складовою розробки програмного забезпечення, особливо коли йдеться про системи керування транспортом, що використовують штучний інтелект.

Взаємодія з інтерфейсом має бути інтуїтивною, простою і, що найголовніше, ефективною для досягнення поставлених цілей.

В симуляційних середовищах моделювання поведінки користувачів дозволяє створити і протестувати різноманітні сценарії використання автономного робота та його можливості для адаптації до змінних умов.

Вибір симуляційного середовища для моделювання обрано гра Mafia 2, яка забезпечує реалістичне міське середовище з різними дорожніми умовами та типовими для середини двадцятого століття. Це середовище дозволяє перевірити, як користувачі можуть взаємодіяти з автономним роботом в умовах, максимально наближених до реальних.

Інтерфейс автономного роботу розроблений таким чином, щоб користувач міг легко його активувати, відслідковувати дії автопілоту та за необхідності втрутитися в управління.

Для перевірки роботи було розроблено кілька сценаріїв взаємодії, що включають активацію автопілоту, ручне втручання користувача та навчання автопілоту на основі взаємодії користувача.

Після розробки інтерфейсу та програмного забезпечення було проведено тестування з участю користувачів. Виявлено, що користувачі змогли легко розібратися з основними функціями автопілоту, а також мали можливість швидко переходити між автоматичним і ручним режимами управління.

Інтерфейс надає користувачу можливість легко активувати та деактивувати автопілот, що підвищує гнучкість взаємодії та дає можливість безперешкодно контролювати систему.

Система своєчасно повідомляє користувача про необхідність втручання у випадку непередбачених обставин, що підвищує безпеку використання.

Взаємодія користувача з системою дозволяє отримувати дані про його дії, що, у свою чергу, використовується для вдосконалення алгоритмів автопілоту, адаптуючи їх до нових умов або поведінки.

Результати дослідження свідчать про те, що розроблений інтерфейс є зручним та інтуїтивним для користувачів, що сприяє ефективній взаємодії з автопілотом. Моделювання сценаріїв використання дозволило виявити слабкі місця в роботі системи та вдосконалити алгоритми.

Перспективи розвитку включають додавання функціоналу для налаштування інтерфейсу під потреби різних користувачів, зокрема вибір мови інтерфейсу або стилю відображення інформації.

Інтеграція системи з датчиками, що дозволить моделювати взаємодію не лише в симуляторі, але й у фізичному середовищі, використовуючи реальні транспортні засоби.

Використання технологій машинного навчання для покращення адаптивності системи на основі аналізу взаємодії користувачів з інтерфейсом.

Такий підхід до моделювання взаємодії користувачів з інтерфейсом забезпечує інтерактивність та адаптивність системи, що є важливим фактором у розробці сучасних рішень з автоматизації транспортних систем.

3.3 Розроблення програмного забезпечення

Розробка програмного забезпечення для автономного робота у віртуальному симуляторі є важливим етапом у створенні автономних систем керування транспортом.

Симуляційне середовище дозволяє відпрацьовувати та тестувати алгоритми без ризику для реальних об'єктів і з мінімальними витратами.

У проєкті використовується гра Mafia 2, яка забезпечує реалістичне міське середовище для тестування поведінки автономного робота у складних умовах.

Розробка інтерфейсу автономного робота була реалізована на мові Python з використанням таких бібліотек, як NumPy для числових обчислень та OpenCV для комп'ютерного зору.

Архітектура програми складається з кількох ключових модулів.

Захоплення зображень з екрану гри.

Використовується бібліотека OpenCV для захоплення потоку зображень з вікна гри. Програма захоплює зображення у реальному часі та конвертує їх у зручний для обробки формат (BGR або RGB).

Розмір вікна гри та область захоплення налаштовуються таким чином, щоб максимально охопити дорожню частину, не захоплюючи зайвих елементів, що можуть заважати аналізу.

Програма використовує OpenCV для захоплення фреймів з гри. Зображення потрапляють у буфер, де обробляються бібліотекою NumPy.

Фрейми конвертуються в колірний простір BGR або інший, зручний для подальшої обробки.

Зображення, отримані OpenCV, передаються у буфер для тимчасового зберігання. Це дозволяє виконувати обробку в асинхронному режимі або черговий доступ до обробки фреймів.

NumPy швидко матрично обробляє зміни яскравості, контрастності, застосування фільтрів або аналізу окремих пікселів.

Зображення може бути перетворено у чорно-білий формат для полегшення аналізу контурів. Чорно-білий формат відображено на рис. 3.1.

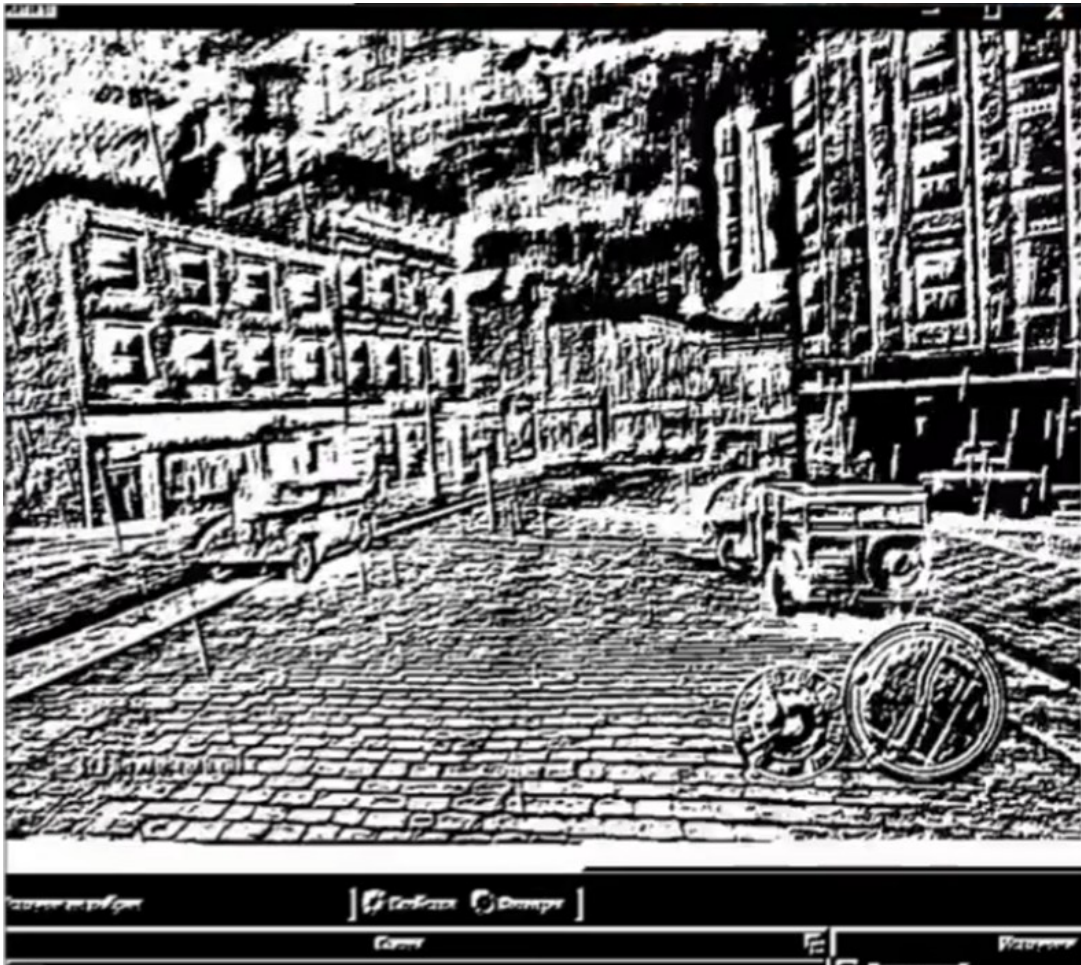


Рисунок 3.1 – Перетворення чорно-білого зображення

Лістинг 3.1 Приклад перетворення чорно-білого зображення:

```
import cv2
import numpy as np
# Захоплення зображення
frame = cv2.imread('screenshot.png')
gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

Обробка зображень та аналіз дорожньої ситуації.

Алгоритми обробки зображень включають перетворення кольорового зображення у чорно-біле (грейскейл) для полегшення виявлення контурів.

Для визначення меж дороги та інших об'єктів використовуються оператори Sobel та фільтр Canny, які дозволяють виявляти контури об'єктів (рис.3.2).

Це важливо для визначення меж дороги, розпізнавання транспортних засобів та інших перешкод.

Для виявлення меж дороги та інших об'єктів використовується алгоритм Canny, який визначає границі об'єктів за допомогою градієнтів.



Рисунок 3.2 – Обробка зображення для полегшення виявлення контурів

Лістинг 3.2 – Приклад обробки зображень

```
edges = cv2.Canny(gray_frame, threshold1=50, threshold2=150)
```

Алгоритм Canny є популярним методом виявлення контурів, який застосовує градієнти для виявлення змін інтенсивності у зображенні. Це дозволяє виявити межі дороги та інші важливі елементи на зображенні.

Перед тим як застосовувати алгоритми для виявлення контурів, важливо зменшити шум у зображенні, щоб уникнути хибних спрацьовувань. Для цього використовується гаусівський фільтр.

Лістинг 3.3 – Приклад використання гаусівського фільтру

```
blurred_frame = cv2.GaussianBlur(gray_frame, (5, 5), 0)
```

Щоб фокусуватися лише на тій частині зображення, де знаходиться дорога, можна визначити область інтересу (Region of Interest, ROI), наприклад, нижню частину кадру, яка містить дорогу.

Лістинг 3.4 – Приклад використання Region of Interest

```
import numpy as np
height, width = edges.shape
roi = np.array([[
    (0, height),
    (width // 2, height // 2),
    (width, height)
]], dtype=np.int32)
# Створення маски для виділення ROI
mask = np.zeros_like(edges)
cv2.fillPoly(mask, roi, 255)
# Застосування маски до зображення з контурами
masked_edges = cv2.bitwise_and(edges, mask)
```

Після того, як контури виділено, можна застосувати метод Хафа (рис. 3.3) для виявлення ліній на зображенні. Цей метод ефективно розпізнає лінії, що можуть представляти межі дороги.

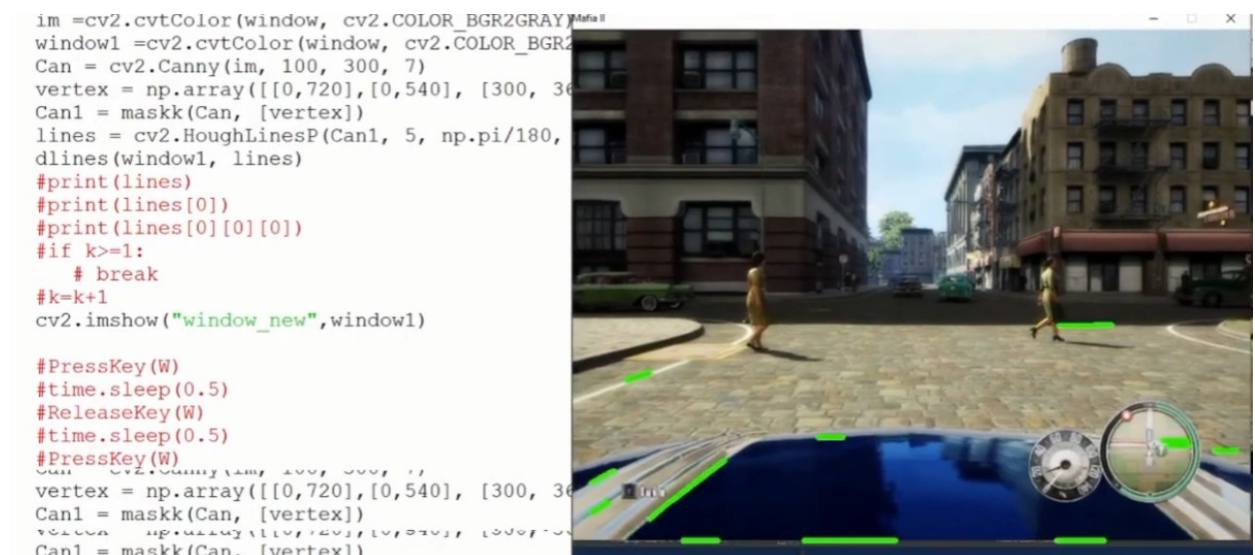


Рисунок 3.3 – Обробка зображення з використанням методу Хафа

Лістинг 3.5 – Приклад використання методу Хафа

```
lines = cv2.HoughLinesP(masked_edges, 1, np.pi / 180, 50, minLineLength=100,
maxLineGap=50)
# Відображення ліній на зображенні
line_frame = frame.copy()
if lines is not None:
    for line in lines:
        x1, y1, x2, y2 = line[0]
        cv2.line(line_frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
```

Метод Хафа (`cv2.HoughLinesP`) знаходить лінії, які відповідають контурам дороги. `minLineLength` та `maxLineGap` задають мінімальну довжину лінії та максимальний розрив між лініями, які можуть бути з'єднані.

Після виявлення меж дороги, система аналізує отримані лінії, щоб визначити напрямок руху. Наприклад, якщо лінії відхиляються вліво, автоматичним управлінням можна симулювати натискання клавіші для повороту вліво.

Лістинг 3.6 – Приклад визначення напрямку руху

```
import cv2
import numpy as np
def detect_road_boundaries(frame):
    # Перетворення зображення у чорно-білий формат
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    # Розмивання для зменшення шумів
    blurred_frame = cv2.GaussianBlur(gray_frame, (5, 5), 0)
    # Виявлення контурів
    edges = cv2.Canny(blurred_frame, 50, 150)
    # Виділення регіону інтересу (ROI)
    height, width = edges.shape
```

```

roi = np.array([[
    (0, height),
    (width // 2, height // 2),
    (width, height)
]], dtype=np.int32)
mask = np.zeros_like(edges)
cv2.fillPoly(mask, roi, 255)
masked_edges = cv2.bitwise_and(edges, mask)
# Виявлення ліній за допомогою Хафа
lines = cv2.HoughLinesP(masked_edges, 1, np.pi / 180, 50, minLineLength=100,
maxLineGap=50)
line_frame = frame.copy()
if lines is not None:
    for line in lines:
        x1, y1, x2, y2 = line[0]
        cv2.line(line_frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
    return line_frame
# Завантаження зображення або захоплення з екрану гри
frame = cv2.imread('screenshot.png')
result_frame = detect_road_boundaries(frame)
# Відображення результату
cv2.imshow('Road Boundaries', result_frame)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Алгоритм аналізує виявлені контури та визначає напрямок руху автомобіля. Наприклад, якщо алгоритм виявляє, що дорога повертає ліворуч, автопілот надсилає команду для повороту у відповідному напрямку.

Для симуляції натискань клавіш використовується бібліотека `STypes`, яка дозволяє працювати з `API Windows`.

Програма симулює натискання клавіш, таких як "вліво", "вправо" чи "гальмування", що дозволяє управляти роботом у грі.

За допомогою ctypes програма відправляє команди на натискання клавіш. Наприклад, якщо контури дороги зміщуються вліво, програма симулює натискання клавіші, яка відповідає за поворот вліво.



Рисунок 3.4 – Використання ctypes

Лістинг 3.7 – Симуляція натискання клавіш за допомогою ctypes

```
import ctypes
import time
SendInput = ctypes.windll.user32.SendInput
Q = 0x10
W = 0x11
A = 0x1E
S = 0x1F
D = 0x20
E = 0x12
R = 0x13
T = 0x14
Y = 0x15
```

```

C = 0x2E
RShift = 0x36
LControl = 0x1D
Space = 0x39
Left = 0xCB
Right = 0xCD
Up = 0xC8
Down = 0xD0
Escape = 0x01
Tab = 0x0F
Alt = 0x38
Enter = 0x1C
# C struct redefinitions
PUL = ctypes.POINTER(ctypes.c_ulong)
class KeyBdInput(ctypes.Structure):
    _fields_ = [("wVk", ctypes.c_ushort),
               ("wScan", ctypes.c_ushort),
               ("dwFlags", ctypes.c_ulong),
               ("time", ctypes.c_ulong),
               ("dwExtraInfo", PUL)]
class HardwareInput(ctypes.Structure):
    _fields_ = [("uMsg", ctypes.c_ulong),
               ("wParamL", ctypes.c_short),
               ("wParamH", ctypes.c_ushort)]
class MouseInput(ctypes.Structure):
    _fields_ = [("dx", ctypes.c_long),
               ("dy", ctypes.c_long),
               ("mouseData", ctypes.c_ulong),
               ("dwFlags", ctypes.c_ulong),
               ("time", ctypes.c_ulong),

```

```

        ("dwExtraInfo", PUL)]
class Input_I(ctypes.Union):
    _fields_ = [("ki", KeyBdInput),
                ("mi", MouseInput),
                ("hi", HardwareInput)]
class Input(ctypes.Structure):
    _fields_ = [("type", ctypes.c_ulong),
                ("ii", Input_I)]
# Actuals Functions
def PressKey(hexKeyCode):
    extra = ctypes.c_ulong(0)
    ii_ = Input_I()
    ii_.ki = KeyBdInput(0, hexKeyCode, 0x0008, 0, ctypes.pointer(extra) )
    x = Input( ctypes.c_ulong(1), ii_ )
    ctypes.windll.user32.SendInput(1, ctypes.pointer(x), ctypes.sizeof(x))
def ReleaseKey(hexKeyCode):
    extra = ctypes.c_ulong(0)
    ii_ = Input_I()
    ii_.ki = KeyBdInput(0, hexKeyCode, 0x0008|0x0002, 0, ctypes.pointer(extra) )
    x = Input( ctypes.c_ulong(1), ii_ )
    ctypes.windll.user32.SendInput(1, ctypes.pointer(x), ctypes.sizeof(x))

```

Програма працює у безперервному циклі, де кожне нове зображення обробляється, аналізується та на основі результатів аналізу приймається рішення щодо керування автомобілем. Цей підхід забезпечує реалістичну взаємодію у режимі реального часу.

Вирішення завдань захоплення та обробки зображень з використанням бібліотек OpenCV та NumPy забезпечує стабільне розпізнавання контурів дороги та перешкод. Використання алгоритмів, таких як фільтр Кенні та метод Хафа, дозволило створити простий, але ефективний підхід для аналізу зображень і прийняття рішень на їх основі.

Реалізація модулів симуляції натискання клавіш через CTypes забезпечила автономне управління роботом у симуляторі, що дозволяє відпрацьовувати алгоритми без ризику для реальних об'єктів.

Ця робота є перспективною для подальшого розвитку, оскільки закладає основи для створення більш складних систем автономного керування транспортом, які можуть бути адаптовані для використання в реальних умовах.

3.4 Дослідження трудомісткості розробки програмного продукту

Для підрахунку трудомісткості програмного продукту в людино-годинах [27] використовують формулу:

$$T = T_o + T_{\text{досл}} + T_a + T_{\text{п}} + T_{\text{нал}} + T_{\text{док}}, \quad (3.1)$$

- де,
- T – загальні витрати праці в людино-годинах;
 - T_o – витрати праці на опис завдання;
 - $T_{\text{досл}}$ – витрати на дослідження предметної області;
 - T_a – витрати на розробку алгоритму рішення задачі;
 - $T_{\text{п}}$ – витрати на програмування;
 - $T_{\text{нал}}$ – витрати на налагодження програми;
 - $T_{\text{док}}$ – витрати на підготовку документації.

Складові витрати праці можна визначити через умовну кількість команд у програмному забезпеченні, яке розробляється. До цієї кількості входять команди, що програміст повинен написати під час виконання завдання з урахуванням можливих уточнень і покращень програми. Для розробки інтерфейсу з використанням ШІ оцінюється кількість умовних операторів у коді. Умовна кількість команд Q в програмі для цього завдання може бути розрахована за такою формулою:

$$Q = q \times c \times (1 + p), \quad (3.2)$$

де, q – число операторів;

c – коефіцієнт складності завдання (1,25...2);

r – коефіцієнт корекції програми, що враховує новизну проекту (для повністю нової програми він дорівнює 0,1).

Коефіцієнт складності програми відображає відносну складність програмних завдань у порівнянні з так званим типовим завданням, складність якого приймається за одиницю. Типовим завданням вважаються задачі обліку, результат яких не виводиться на друк, а після оновлення зберігається в пам'яті комп'ютера. Для задач оперативного управління коефіцієнт складності програми становить 1,25 – 2. У розрахунках використовуємо значення 1,5.

Кількість операторів у коді, приблизно дорівнює 1000. Таким чином, умовна кількість команд буде дорівнювати:

$$Q = 1000 \times 1,5 \times (1 + 0,1) = 1650 \text{ (люд./год.)}$$

На основі умовної кількості команд Q , можна розрахувати загальні витрати праці в людино-годинах. Для цього потрібно знайти: T_o , $T_{\text{досл}}$, T_a , $T_{\text{п}}$, $T_{\text{нал}}$, $T_{\text{док}}$.

T_o приймається в діапазоні від 30 до 40 людино-годин, в залежності від складності продукту. В даному випадку, витрати на опис праці зайняли 35 години.

$T_{\text{досл}}$ розраховується за формулою:

$$T_{\text{досл}} = \frac{Q \times B}{S_{\text{досл}} \times k}, \quad (3.3)$$

де, B – коефіцієнт збільшення витрат характеризує збільшення витрат праці унаслідок недостатнього опису завдання, уточнень і деякого доопрацювання. Цей коефіцієнт може приймати значення від 1,2 до 5. Візьмемо його середнє значення $B = 3,1$;

$S_{\text{досл}}$ – кількість операторів, що припадають на 1 людино-годину (80);

k – коефіцієнт кваліфікації працівника він залежить від досвіду роботи працівника.

Залежність наведена в таблиці 3.1.

Таблиця 3.1 – Залежність коефіцієнта кваліфікації працівника від досвіду

Стаж роботи	Коефіцієнт кваліфікації k
до двох років	0,8
від двох до трьох років	1– 1,2
від трьох до семи років	1,3 – 1,4
понад семи років	1,5 – 1,6

Розрахуємо $T_{\text{досл}}$:

$$T_{\text{досл}} = \frac{(1650 \times 3,1)}{(80 \times 1,2)} = 53,28 \text{ (люд./год.)}$$

Розрахуємо витрати праці на розробку алгоритму рішення задачі в людино-годинах за формулою:

$$T_a = \frac{Q}{(S_a k)} \quad (3.4)$$

Де, T_a – витрати праці в людино-годинах;

Q – обсяг роботи (наприклад, кількість завдань чи модулів);

S_a – середня продуктивність працівника;

k – коригуючий коефіцієнт.

Розрахуємо T_a :

$$T_a = \frac{1650}{(50 \times 1,2)} = 27,5 \text{ (люд./год.)}$$

Витрати часу на програмування визначаються методом самофотографії і складають приблизно 20-30% від загальних витрат на працю. Розрахуємо їх за формулою:

$$T_{\pi} = \frac{Q}{(S_{\pi} \times k)}, \quad (3.5)$$

де, Q – кількість умовних операторів;

K – коефіцієнт кваліфікації розробника;

S_{π} – кількість операторів, зайнятих програмуванням, що припадає на одну людино-годину. В даній роботі дорівнює 25.

Підставляючи $Q = 1650$, $k = 1$ і приймаючи S_{π} рівним 25, визначимо витрати часу на програмування:

$$T_{\pi} = \frac{1650}{(25 \times 1,2)} = 55 \text{ (люд./год.)}$$

Витрати на налагодження програми на ПК при автономному налагодженні одного завдання визначається за формулою:

$$T_{\text{нал}} = \frac{Q}{(S_{\text{нал}} \times k)}, \quad (3.6)$$

де, Q – кількість умовних операторів;

K – коефіцієнт кваліфікації розробника;

$S_{\text{нал}}$ – кількість операторів, зайнятих налагодженням програми, що припадає на одну людино-годину.

Підставляючи $Q = 1650$, $K = 1,2$, і приймаючи $S_{\text{нал}}$ рівним 40, розрахуємо витрати на налагодження програми.

$$T_{\text{нал}} = \frac{1650}{48} \approx 34,38 \text{ (люд./год.)}$$

Витрати на підготовку документації в даній роботі не використовувалися. Тому, щоб підрахувати загальні витрати праці в людино-годинах потрібно знайти суму всіх витрат:

$$T = 35 + 53,28 + 27,5 + 55 + 34,38 = 205,16 \text{ (люд/год)}. \quad (3.7)$$

Щоб знайти кількість робочих днів D , яка потрібна для створення додатка розділимо кількість людино-годин на 8 – тривалість робочого дня в годинах. Проведемо розрахунки.

$$D = \frac{205,16}{8} = 25,65 \text{ (днів)}.$$

Таким чином, розробка інтерфейсу керування роботами з використанням штучного інтелекту займе приблизно 26 робочих днів для однієї людини за умови стандартного 8-годинного робочого дня.

3.5 Висновки до третього розділу

У третій частині була реалізована розробка інтерфейсу керування роботами з використанням штучного інтелекту ШІ, та проведений науковий експеримент.

Розробка інтерфейсу керування роботами з використанням ШІ показала, що сучасні алгоритми комп'ютерного зору та числових обчислень є потужними інструментами для створення автономних систем керування. Однак, для забезпечення більшої надійності та стабільності системи необхідна подальша оптимізація програмного забезпечення та використання методів машинного навчання. Інтерактивність інтерфейсу та модульний підхід забезпечують хорошу базу для подальшого розвитку автономних рішень, як у симуляціях, так і у фізичному світі.

4 ОХОРОНА ПРАЦІ

4.1 Аналіз умов праці на робочому місці

На робочому місці оператора ПК можуть виникати небезпечні та шкідливі фактори, такі як високий рівень шуму, несприятливі мікрокліматичні умови, недостатнє освітлення, присутність шкідливих речовин, підвищені електромагнітні випромінювання радіочастот, висока напруга електромережі, статична електрика тощо. Робота з ПК також характеризується високим рівнем психічної та фізичної напруги. Тривалий вплив несприятливих виробничих факторів, які не відповідають нормативам, може призвести до зростання захворюваності, зокрема ураження органів зору, опорно-рухового апарату та нервової системи. Тому дослідження умов праці програміста є важливим кроком для запобігання негативним наслідкам дії цих факторів.

Організація робочого місця. Приміщення, в якому знаходиться робоче місце програміста, має бути не меншою за 9 м² на працівника. Для розробки інтерфейсів із використанням ШІ на базі ПК слід забезпечити кожного розробника достатнім простором для роботи. Природне, так і штучне освітлення має бути оптимальним для зменшення навантаження на очі. Робоче місце обладнане робочим столом, стільцем та персональним комп'ютером, що складається з монітора, системного блоку, клавіатури та миші.

4.2 Промислова безпека на робочому місці

Електроживлення ПК здійснюється від трифазної чотирипровідної мережі змінного струму з напругою 220 В, частотою 50 Гц і заземленою нейтраллю. Відповідно до НПАОП 40.1-1.21-98, приміщення класифікується як таке, що не належить до категорії з підвищеною безпекою, оскільки відсутні фактори, які б створювали підвищену або особливу небезпеку.

Для забезпечення безпечних умов праці необхідно вжити низку організаційних і технічних заходів. Відповідно до НПАОП 40.1-1.32-01, для захисту людей від ураження електричним струмом у приміщенні повинна бути використана система занулення.

4.3 Виробнича санітарія у приміщенні

Робота оператора ПК за енерговитратами відноситься до категорії легких робіт. В таблиці 4.1 наведені оптимальні параметри мікроклімату в приміщеннях, де виконуються роботи операторського типу [28].

Таблиця 4.1 – Параметри мікроклімату для приміщень з ПК

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні; відносна вологість; швидкість руху повітря	22 – 24 °С; 40 – 60 %; до 0,1 м/с
Теплий	Температура повітря в приміщенні; відносна вологість; швидкість руху повітря	23 – 25 °С; 40 – 60 %; 0,1 – 0,2 м/с

Виміряні за допомогою приладів температура та вологість у лабораторії відповідають вказаним у таблиці для теплого періоду року. Слід зазначити, що для нормалізації параметрів мікроклімату слід використовувати у приміщеннях кондиціонування повітря, або забезпечити подачу свіжого повітря системами вентиляції.

Лабораторія, де виконується робота, має наступні характеристики:

- площа приміщення – 9 м² (3м × 3 м);
- обладнання – стіл з ПК і периферією – 1 шт.

Приміщення, відповідно до ДНАОП 0.00-1.31-99, має забезпечувати 6 м² площі та 20 м³ об'єму на одне окреме робоче місце з ПК [28]. Площа приміщення 9 м² та об'єм 22,5 м³, на кожне робоче місце приходиться 9 м² площі і об'єм 22,5 м³, тобто вимога виконана.

Приміщення з ПК повинні мати природне і штучне освітлення відповідно до ДБН В.25-28-2006 «Природне і штучне освітлення». Природне світло повинно проникати через бічні світлові прорізи, зорієнтовані, як правило, на північ або північний схід, і забезпечувати коефіцієнт природної освітленості (КПО) не нижче 1.5 %.

Рівень загального штучного освітлення приміщення можна перевірити за допомогою методу питомої потужності, викладеної в [28].

Розрахункова формула методу:

$$W = \frac{W_{\Sigma}}{S}, \quad (4.1)$$

де W – питома потужність, Вт/м²;

S – площа приміщення, м²;

W_{Σ} – загальна потужність освітлювальної установки Вт, яка розраховується за формулою:

$$W_{\Sigma} = W_{ce} \cdot n_{ce}, \quad (4.2)$$

де W_{ce} – потужність одного світильника, Вт;

n_{ce} – кількість світильників в приміщенні.

$$W_{\Sigma} = 100 \cdot 4 = 400 \text{ Вт}, \quad (4.3)$$

$$W = \frac{400}{48} = 8,33 \text{ Вт/м}^2. \quad (4.4)$$

Питомої потужності 8.33 Вт/м^2 по таблиці Б.3 із [28] відповідає освітленість в 250 лк при мінімальній допустимій освітленості 300 лк.

Отже, для створення сприятливих зорових умов в лабораторії необхідно збільшити кількість світильників або замінити лампи в світильниках на більш потужні.

4.4 Пожежна безпека приміщення

Пожежна безпека – це стан об'єкта, за якого мінімізується ймовірність виникнення пожежі, а в разі її виникнення забезпечується захист людей від небезпечних факторів пожежі та збереження матеріальних цінностей. Забезпечення пожежної безпеки здійснюється через систему запобігання пожежам та систему пожежного захисту. У всіх службових приміщеннях обов'язково має бути наявний «План евакуації при пожежі», який регламентує дії персоналу у разі виникнення загоряння та вказує місця розташування пожежного обладнання.

До горючих матеріалів у виробничому приміщенні належать перегородки, двері, підлоги, ізоляція кабелів тощо.

Протипожежний захист включає організаційні та технічні заходи, спрямовані на забезпечення безпеки людей, запобігання поширенню пожежі та створення умов для її успішного гасіння. Джерелами займання у виробничому середовищі можуть бути електронні компоненти ПК, прилади для технічного обслуговування, системи живлення та кондиціонування, де через різні несправності виникають перегріті елементи, іскри чи дуги, що можуть спричинити займання легкозаймистих матеріалів.

Сучасні ПК мають високу щільність розташування електронних компонентів, що сприяє виділенню великої кількості тепла під час проходження електричного струму через проводи та кабелі. Це може призвести до оплавлення ізоляції. Для відведення тепла використовуються системи вентиляції та кондиціонування повітря, які, при постійному використанні, самі можуть стати джерелом підвищеної пожежної небезпеки.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи була проведена розробка автономного управління роботом у симуляторі.

У першому розділі проведено детальний аналіз вихідних даних та визначено ключові вимоги до інтерфейсів для керування роботами. Розглянуті сучасні інноваційні підходи, такі як адаптивні інтерфейси, інтерфейси на основі природної мови, жестові інтерфейси та інтерфейси доповненої реальності. Кожна технологія проаналізована з точки зору принципів її роботи. На основі цього аналізу були виявлені основні переваги та недоліки кожного підходу. Приділено увагу алгоритмам штучного інтелекту, які є основою для адаптивності і автоматизації роботизованих систем.

У другому розділі роботи здійснено розробка автономної роботизованої системи. Створено функціональну структуру системи. Розроблено алгоритми, що охоплюють основні етапи функціонування автономної роботизованої системи. Проведено аналіз стійкості автономної роботизованої системи за трьома критеріями.

У третій частині була реалізована розробка інтерфейсу керування роботами з використанням штучного інтелекту. Після чого були змодельовані взаємодії користувачів з інтерфейсом. Описане програмне забезпечення та бібліотеки які були використані для створення інтерфейсу.

При виконанні четвертого розділу були визначені небезпечні і шкідливі виробничі фактори. Розроблені заходи і технічні засоби щодо забезпечення безпеки праці.

Кваліфікаційна робота спрямована на підвищення ефективності взаємодії людини та роботизованих систем, використання інноваційних технологій та автоматизацію, що відповідає завданню 9.4 Цілі сталого розвитку 9 "Промисловість, інновації та інфраструктура", відповідно Указу президента України №722/2019 від 30.09.2019 року "Про Цілі сталого розвитку (ЦСР) України на період до 2030 року".

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Штучний інтелект та робототехніка: сучасні рішення та перспективи / О.М. Тищенко, В.В. Мартинюк. – К.: Видавництво КПІ ім. Ігоря Сікорського, 2020. – 308 с.

2. ДСТУ 3008–2015. Звіти у сфері науки і техніки. Структура і правила оформлення. Документація. Введ. 2015. К.: Держстандарт України, 2015. – 31 с.

3. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація та 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. – 56 с.

4. Положення про академічну доброчесність [Електронний ресурс]: Наказ ХНУРЕ від 02 лютого 2021 р. № 50. Режим доступу: https://nure.ua/wp-content/uploads/Main_Docs_NURE/polozhennja-pro-akademi-chnu-dobrochesnist.pdf.

5. Указ президента України №722/2019 "Про Цілі сталого розвитку (ЦСР) України на період до 2030 року " від 30.09.2019 р. Режим доступу: <https://www.president.gov.ua/documents/7222019-29825>.

6. Автоматизація та Приладобудування ("Automation and Development of Electronic Devices" ADED-2024) [Електронний ресурс] : збірник студентських наукових статей / Харківський національний університет радіоелектроніки /редкол.: І.Ш. Невлюдов та ін. – Харків : ХНУРЕ, 2024. – Вип. 2. – с. 235-240.

7. Human-Robot Interaction: Design, Methods, and Impacts / М. Goodrich, А. Schultz. – Cambridge University Press 2021. – 320 с.

8. Development of AI-Controlled Robotic Interfaces: A Technical Overview / Smith, J., Miller, D., 2021. – с. 89-105.
9. Intelligent Robotics: Methods and Applications / Edited by D. H. Wilson. – Springer, 2020. – 411 с.
10. Robot Learning for Autonomous Robots: Deep Learning Approaches and Applications / Edited by P. Corke, R. Mahony. – MIT Press, 2021. – 312 с.
11. Advances in Robotics and AI: Modern Applications and Challenges / Edited by A. Ramachandran, B. Shaw. – Academic Press, 2021. – 426 с.
12. AI-Powered Interfaces in Robotics: Exploring the Future of Human-Robot Collaboration / P. Sanchez, D. Muller. – Elsevier, 2022. – 255 с.
13. Технології штучного інтелекту в інтерфейсах користувача [Електронний ресурс]: Режим доступу: <https://aiukraine.com/interfaces-and-ai/>.
14. Machine Learning and AI for Robotics [Електронний ресурс]: Режим доступу: <https://www.ml-for-robotics.com/>.
15. ISO 10218-1: Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots. – Geneva, 2018. – 54 с.
16. IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements. – Geneva, 2018. – 132 с.
17. McAfee, A. Artificial Intelligence in Industry 4.0: Key Concepts and Application / A. McAfee. – Boston: MIT Press, 2022. – 285 с.
18. The Impact of AI on Human-Robot Interaction / Jones, P., Smith, M. // AI & Society. – 2021. – Vol. 35. – с. 174-188.
19. The AI Revolution in Robotics: Key Concepts and Use Cases / M. L. Moore. – Springer, 2022. – 324 с.
20. Застосування штучного інтелекту в системах керування роботами / А.О. Кравченко. – Харків: Видавництво ХНУРЕ, 2021. – 260 с.
21. Механізми штучного інтелекту в робототехнічних системах [Електронний ресурс]: Режим доступу: <https://roboticsai.com/systems-and-ai/>.
22. Understanding Human-Robot Interaction through AI Systems / Rogers, S., Clarke, L. – Springer, 2020. – 399 с.

23. Інформаційно-комунікаційні технології [Електронний ресурс]: Режим доступу: <https://log.logcluster.org/ru/terminal-s-ochen-maloy-aperturoy-vsot>.
24. AI-Driven Interfaces for Robotics / Edited by L. Andrews. – Taylor & Francis, 2021. – 337 с.
25. Технічні засоби автоматизації. Підручник / І.Ш. Невлюдов, А.О. Андрусевич, О.І. Филипенко, Н.П. Демська, С.П. Новоселов. – Кривий Ріг : Криворізький коледж НАУ, 2019. – 366 с.
26. Теорія автоматичного управління. Навчальний посібник / І.Ш. Невлюдов, О.В. Токарева. – Харків: Видавництво ХНУРЕ, 2020. – 243 с.
27. Техніко-економічне обґрунтування інженерних рішень. Підручник / І.Ш. Невлюдов. – Кривий Ріг : Криворізький коледж НАУ, 2019. – 448 с.
28. 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. Стиценко, Г. В. Пронюк, Н. М. Сердюк. – Харків, 2017. – 108 с.