

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматизації, комп'ютерно-інтегрованих технологій та систем
(повна назва)

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

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Розроблення системи управління мобільною роботизованою платформою
на основі машинного навчання
(тема)

Виконав:

здобувач 4 року навчання,
групи АКТАКІТ-22-1.
Ілля АНДРЮХІН.
(власне ім'я прізвище)

Спеціальності 151 Автоматизація та
комп'ютерно-інтегровані технології
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Автоматизація та
комп'ютерно-інтегровані технології
(повна назва освітньої програми)

Керівник ас. Роман СТРИЛЕЦЬ
(посада, власне ім'я прізвище)

Допускається до захисту
Зав. кафедри КІТАРБІ

(підпис)

Ігор НЕВЛЮДОВ.
(власне ім'я прізвище)

2026р.

Я, Андрюхін Ілля Олексійович, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволеної допомоги під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів має містити посилання на відповідне джерело.

«16» червня 2026 р.



Ілля АНДРЮХІН

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКИТС
Кафедра _____ КІТАРБІ
Рівень вищої освіти _____ перший (бакалаврський)
Спеціальність _____ 151 Автоматизація та комп'ютерно-інтегровані технології
Тип програми _____ освітньо-професійна
Освітня програма _____ Автоматизація та комп'ютерно-інтегровані технології

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАРБІ _____
(підпис)

«__» ____ 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Андрюхину Іллі Олексійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Розроблення системи управління мобільною роботизованою платформою на основі машинного навчання

затверджена наказом по університету від "04" червня 2026р. № 472 Ст

2. Термін подання студентом роботи "13" липня 2026р.

3. Вихідні дані до роботи: _____

3.1 Мова програмування Python;

3.2 Симуляційне середовище Webots;

3.3 Бібліотеки Ultralytics, Stable-Baselines3, Gymnasium;

3.4 Платформа TurtleBot3 Burger;

4. Перелік питань, що потрібно опрацювати в роботі _____

4.1 Аналіз предметної області;

4.2 Проектування архітектури системи керування;

4.3 Технічна реалізація YOLO та DQN;

4.4 Тестування системи та аналіз результатів;

4.5 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Графічний матеріал у вигляді презентації – 16 арк. ф. А4.

6. Консультанти розділів роботи

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області та огляд рішень	30.04 – 04.05.26	Виконано
2	Проектування архітектури системи	05.05 – 18.05.26	Виконано
3	Підготовка даних та навчання детектора YOLO	19.05 – 25.05.26	Виконано
4	Реалізація RL-середовища та навчання DQN	27.05 – 04.06.26	Виконано
5	Експерименти та аналіз результатів	05.06 – 09.06.26	Виконано
6	Заходи і розрахунки для забезпечення безпечних умов праці	11.06 – 15.06.26	Виконано
7	Оформлення пояснювальної записки	16.06 – 26.06.26	Виконано
8	Подання роботи на перевірку Інтернет-системою StrikePlagiarism	11.07.26	Виконано
9	Подання роботи на рецензію	11.07.26	Виконано
10	Подання роботи на підпис зав. кафедри	11.07.26	Виконано
11	Подання кваліфікаційної роботи в ЕК	12.07.26	Виконано

Дата видачі завдання 27.04.2026р.

Здобувач


(підпис)

Ілля АНДРЮХІН

Керівник роботи

(підпис)

ас. Роман СТІЛЕЦЬ
(посада, власне ім'я прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 62 с., 12 табл., 18 рис., 3 дод., 37 джерел.

МОБІЛЬНИЙ РОБОТ, МАШИННЕ НАВЧАННЯ, НЕЙРОННА МЕРЕЖА, YOLO, НАВЧАННЯ З ПІДКРІПЛЕННЯМ, DQN, WEBOTS, АВТОНОМНА НАВІГАЦІЯ, КОМП'ЮТЕРНИЙ ЗІР.

Об'єктом дослідження є процес функціонування та керування мобільною роботизованою платформою в умовах невизначеного середовища з перешкодами.

Предметом дослідження є методи, моделі та програмні засоби побудови системи керування мобільною платформою на основі згорткових нейронних мереж та навчання з підкріпленням.

Метою роботи є розробка та дослідження системи управління мобільною роботизованою платформою на основі методів машинного навчання, що поєднує модуль технічного зору для розпізнавання об'єктів та модуль навчання з підкріпленням для автономної навігації.

У роботі проведено аналіз мобільних роботизованих платформ та методів машинного навчання, розроблено модульну архітектуру системи керування, натреновано детектор пішоходів на основі архітектури YOLOv11 та DQN-агента навігації у симуляційному середовищі Webots. Проведено експериментальні дослідження у детермінованому та рандомізованому середовищах. Детектор досяг метрики mAP@0.5 близько 0,99, а агент навігації – стовідсоткової успішності доїздів у детермінованому середовищі без зіткнень. Отримані результати підтверджують працездатність запропонованого гібридного підходу.

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

ABSTRACT

Bachelor's thesis: 62 pages, 12 tables, 18 figures, 3 appendices, 37 sources.

MOBILE ROBOT, MACHINE LEARNING, NEURAL NETWORK, YOLO, REINFORCEMENT LEARNING, DQN, WEBOTS, AUTONOMOUS NAVIGATION, COMPUTER VISION.

The object of research is the operation and control of a mobile robotic platform in an uncertain environment with obstacles.

The subject of research is the methods, models and software tools for building a mobile platform control system based on convolutional neural networks and reinforcement learning.

The goal of the work is to develop and study a control system for a mobile robotic platform based on machine learning methods, combining a computer vision module for object recognition and a reinforcement learning module for autonomous navigation.

The thesis analyses mobile robotic platforms and machine learning methods, develops a modular architecture of the control system, and trains a pedestrian detector based on the YOLOv11 architecture together with a DQN navigation agent in the Webots simulation environment. Experiments were carried out in deterministic and randomized environments. The detector achieved a mAP@0.5 of about 0,99, while the navigation agent reached a 100% success rate in the deterministic environment without collisions. The obtained results confirm the feasibility of the proposed hybrid approach.

The obtained results can be attributed to Sustainable Development Goal 9 "Industry, Innovation and Infrastructure", in particular to target 9.4 "Development of high-tech mechanical engineering".

ЗМІСТ

Перелік скорочень	9
Вступ.....	10
1 Аналіз предметної області та постановка задачі	13
1.1 Мобільні роботизовані платформи та задачі їх керування	13
1.2 Класичні та інтелектуальні методи керування	14
1.3 Комп'ютерний зір та детекція об'єктів	16
1.4 Навчання з підкріпленням для задач навігації	17
1.5 Огляд середовищ симуляції робототехнічних систем	18
1.6 Постановка задачі.....	19
2 Проєктування системи управління мобільною платформою	21
2.1 Загальна архітектура системи	21
2.2 Модель мобільної платформи та симуляційне середовище	22
2.3 Модуль технічного зору	23
2.4 Формалізація задачі навігації як задачі навчання з підкріпленням	24
3 Технічна реалізація системи	27
3.1 Засоби та інструменти розробки	27
3.2 Підготовка набору даних та навчання детектора YOLO	27
3.3 Реалізація середовища навчання та навчання DQN-агента	30
3.4 Реалізація керуючого контролер.....	31
4 Аналіз отриманих результатів	32
4.1 Результати навчання детектора об'єктів	32
4.2 Результати навчання DQN-агента.....	35
4.3 Оцінка якості навігації та порівняння сценаріїв	36

4.4 Аналіз результатів дослідження гібридної системи керування.....	38
4.5 Синтез ПІ-регулятора та аналіз стійкості системи.....	38
4.6 Безпека праці під час розроблення системи.....	43
Висновки	47
Перелік джерел посилання	49
Додаток А Лістинги ключових фрагментів програмного коду	53
Додаток Б Супровідні та цифрові матеріали проєкту	60
Додаток В Демонстраційний матеріал.....	61

ПЕРЕЛІК СКОРОЧЕНЬ

MH – машинне навчання;

ПІД – пропорційно-інтегрально-диференціальний регулятор;

ШІ – штучний інтелект;

AP – Average Precision (середня точність для окремого класу);

API – Application Programming Interface (інтерфейс прикладного програмування);

BGRA – Blue-Green-Red-Alpha (формат «синій-зелений-червоний-альфа»);

CNN – Convolutional Neural Network (згорткова нейронна мережа);

DQN – Deep Q-Network (глибока Q-мережа);

FPS – Frames Per Second (кадрів за секунду);

GPS – Global Positioning System (система глобального позиціонування);

GPU – Graphics Processing Unit (графічний процесор);

IDE – Integrated Development Environment (інтегроване середовище розробки);

IoU – Intersection over Union (міра перекриття рамок);

LiDAR – Light Detection and Ranging (лазерний далекомір);

mAP – mean Average Precision (усереднена середня точність);

MDP – Markov Decision Process (марковський процес прийняття рішень);

PPO – Proximal Policy Optimization (проксимальна оптимізація стратегії);

RGB – Red-Green-Blue (колірна модель «червоний-зелений-синій»);

RL – Reinforcement Learning (навчання з підкріпленням);

ROS – Robot Operating System (операційна система для роботів);

SAC – Soft Actor-Critic («м'який» актор-критик);

SB3 – Stable-Baselines3 (бібліотека для навчання з підкріпленням);

SGD – Stochastic Gradient Descent (стохастичний градієнтний спуск);

SLAM – Simultaneous Localization and Mapping (одночасна локалізація та побудова мапи);

YOLO – You Only Look Once (сімейство одноетапних детекторів об'єктів).

ВСТУП

Сучасний етап розвитку робототехніки характеризується стрімким переходом від жорстко запрограмованих промислових маніпуляторів до автономних мобільних систем, здатних діяти у складних, заздалегідь не описаних середовищах. Мобільні роботизовані платформи вже сьогодні застосовують у логістиці, складському господарстві, сільському господарстві, медицині, а також в охоронній та військовій сферах, де здатність апарата діяти самостійно, без постійного втручання оператора, є визначальною для ефективного виконання поставлених задач.

Класичні підходи до побудови систем керування мобільними роботами, що ґрунтуються виключно на заздалегідь визначених правилах та детермінованих алгоритмах (ПД-регулятори, методи планування шляху на кшталт A*), добре працюють у стабільних умовах, проте втрачають ефективність, коли середовище змінюється, з'являються нові типи перешкод або виникає потреба розпізнавати раніше не зустрічні об'єкти. Подолати ці обмеження дозволяють методи машинного навчання, які надають системі здатність самостійно формувати модель поведінки на основі накопиченого досвіду та сенсорних даних.

Особливо перспективним є поєднання двох напрямів машинного навчання. З одного боку, згорткові нейронні мережі та сучасні детектори об'єктів сімейства YOLO забезпечують надійне розпізнавання об'єктів у реальному часі за зображенням з бортової камери. З іншого боку, методи навчання з підкріпленням, зокрема глибокі Q-мережі (DQN), дозволяють роботу самостійно навчитися стратегії руху до цілі та уникнення перешкод шляхом взаємодії із середовищем. Таке гібридне рішення об'єднує сприйняття та керування в єдиній інтелектуальній системі.

Актуальність теми зумовлена зростаючою потребою у мобільних автономних системах, здатних діяти у динамічних умовах, а також швидким розвитком технологій штучного інтелекту, що роблять такі системи технічно

реалізованими навіть на обчислювально обмежених платформах. Розробка та дослідження системи керування мобільною платформою на основі машинного навчання відповідає сучасному рівню наукових і технічних знань та має практичну цінність.

Метою кваліфікаційної роботи є розробка системи управління мобільною роботизованою платформою на основі методів машинного навчання, що поєднує модуль технічного зору для розпізнавання об'єктів та модуль навчання з підкріпленням для автономної навігації, з метою підвищення автономності, адаптивності до змін середовища та ефективності руху до цілі.

Об'єктом дослідження є процес функціонування та керування мобільною роботизованою платформою в умовах невизначеного середовища з перешкодами.

Предметом дослідження є методи, моделі та програмні засоби побудови системи керування мобільною платформою на основі згорткових нейронних мереж (детектор YOLO) та навчання з підкріпленням (алгоритм DQN), а також їхній вплив на якість навігації та рівень автономності робота.

Для досягнення поставленої мети в роботі необхідно вирішити такі задачі:

- провести аналіз сучасних мобільних роботизованих платформ, методів керування ними та підходів машинного навчання, придатних для інтеграції в систему автономного робота;
- обґрунтувати вибір апаратно-програмної бази та симуляційного середовища для розробки й тестування системи;
- розробити загальну архітектуру системи керування, що поєднує підсистеми сприйняття та прийняття рішень;
- підготувати набір даних та натренувати нейромережевий детектор об'єктів на основі архітектури YOLO;
- формалізувати задачу навігації як задачу навчання з підкріпленням і натренувати DQN-агента у симуляційному середовищі;
- провести експериментальні дослідження та оцінити ефективність запропонованого рішення за об'єктивними метриками.

Практичне значення роботи полягає в тому, що розроблена система та отримані результати можуть бути використані як основа для подальшого впровадження у реальні робототехнічні комплекси, а також у навчальному процесі під час вивчення дисциплін, пов'язаних з інтелектуальними системами керування та машинним навчанням.

Кваліфікаційну роботу виконано згідно з ДСТУ 3008:2015 [1], керуючись навчальним посібником з підготовки кваліфікаційної роботи [2] та відповідними методичними вказівками [3].

Отримані результати роботи можна також віднести до Цілі сталого розвитку 9 «Промисловість, інновації та інфраструктура», а саме до завдання 9.5 щодо активізації наукових досліджень, нарощування технологічних можливостей та заохочення інновацій, оскільки робота розвиває інтелектуальні технології керування автономними роботизованими системами на основі методів машинного навчання.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Мобільні роботизовані платформи та задачі їх керування

Мобільний робот – це автоматизований пристрій, здатний самостійно переміщуватися у просторі за допомогою власних рушіїв, виконувати задані функції та взаємодіяти з навколишнім середовищем. Сучасні мобільні роботи класифікують за типом пересування (колісні, гусеничні, крокуючі, літальні, водні), сферою застосування, рівнем автономності та типом системи керування. Найпоширенішими є колісні платформи завдяки простоті конструкції, високій швидкості руху та енергоефективності; саме вони найкраще підходять для роботи в приміщеннях та на рівних поверхнях.

Серед колісних конфігурацій окремо виділяють диференційний привід, за якого двома незалежно керованими колесами задаються як лінійна, так і кутова швидкості робота, а стабільність забезпечується опорним (пасивним) колесом. Диференційна схема є найпростішою у реалізації та керуванні, тому її широко застосовують у дослідницьких і навчальних платформах. Саме таку кінематичну схему використовує обрана в роботі платформа.

За рівнем автономності мобільні роботи поділяють на ручні, дистанційно керовані, напівавтономні та повністю автономні. Повністю автономні системи здатні виконувати задачі без втручання оператора, спираючись лише на дані давачів та вбудовані алгоритми. Побудова саме такої повністю автономної поведінки – здатності самостійно доїхати до заданої цілі, уникаючи перешкод, – є ключовою задачею цієї роботи.

Систему керування мобільним роботом зазвичай розглядають як багаторівневу структуру. На виконавчому рівні відбувається безпосереднє керування швидкістю коліс; на тактичному рівні розв'язуються задачі локальної навігації та уникнення перешкод; на стратегічному рівні здійснюється планування маршруту та вибір цілі. У розроблюваній системі тактичний рівень реалізовано

методом навчання з підкріпленням, що є основною науково-технічною новизною роботи.

Не менш важливою за керування рухом є задача сприйняття середовища. Мобільний робот повинен не лише вимірювати відстані до перешкод, але й розпізнавати семантично значущі об'єкти – насамперед людей (пішоходів), – щоб коригувати свою поведінку та забезпечувати безпеку.

1.2 Класичні та інтелектуальні методи керування

Історично першими для керування мобільними роботами застосовували класичні (детерміновані) методи. До них належать ПІД-регулятори для стабілізації руху, методи планування шляху на графах (алгоритми Дейкстри, A^*), методи штучних потенційних полів для уникнення перешкод, а також скінченні автомати для формування поведінки. Головні переваги цих методів – передбачуваність, обчислювальна простота та інтерпретованість: поведінку системи легко проаналізувати та формально верифікувати.

Водночас класичні методи мають суттєве обмеження: вони потребують точної математичної моделі середовища та об'єкта керування. У динамічних, заздалегідь не описаних умовах, коли з'являються нові типи перешкод або змінюється конфігурація простору, ефективність таких методів різко знижується, а їх ручне переналаштування стає трудомістким. Це стимулювало перехід до інтелектуальних методів керування на основі машинного навчання.

Машинне навчання (МН) – це розділ штучного інтелекту, який вивчає алгоритми, здатні самостійно вдосконалювати свою поведінку на основі даних, без явного програмування правил. Прийнято розрізняти три основні парадигми машинного навчання: навчання з учителем, навчання без учителя та навчання з підкріпленням (рисунок 1.1).

Класифікація методів машинного навчання



Рисунок 1.1 – Класифікація методів машинного навчання

Навчання з учителем передбачає наявність розміченого набору даних, де кожному прикладу відповідає правильний результат; алгоритм навчається мінімізувати розбіжність між прогнозом і правильною відповіддю. У мобільній робототехніці цей підхід застосовують передусім для розпізнавання та детекції об'єктів – саме до нього належать згорткові нейронні мережі та детектори YOLO. Навчання без учителя працює з нерозміченими даними та виявляє в них приховані закономірності (кластеризація, зменшення розмірності). Навчання з підкріпленням відрізняється тим, що агент навчається шляхом взаємодії із середовищем, отримуючи винагороду за корисні дії; саме воно найкраще підходить для задач керування рухом.

У розроблюваній системі свідомо застосовано гібридний підхід: детекція об'єктів виконується методом навчання з учителем (YOLO), а керування рухом – методом навчання з підкріпленням (DQN).

Рівні системи керування мобільним роботом

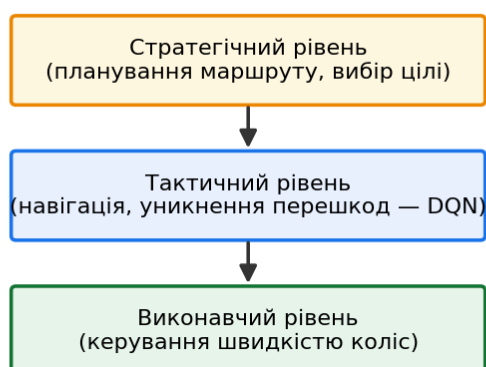


Рисунок 1.2 – Рівні системи керування мобільним роботом

1.3 Комп'ютерний зір та детекція об'єктів

Основою сучасних систем технічного зору є згорткові нейронні мережі (Convolutional Neural Networks, CNN). Вони здатні автоматично виділяти ієрархічні ознаки зображення – від простих (краї, кути) до складних (форми, об'єкти) – та використовувати їх для класифікації, детекції та сегментації. Серед відомих архітектур CNN – VGG, ResNet, MobileNet, EfficientNet, які відрізняються глибиною, точністю та обчислювальною складністю.

Задача детекції об'єктів полягає у визначенні як класу об'єкта, так і його положення на зображенні у вигляді обмежувальної рамки (bounding box). Алгоритми детекції поділяють на два класи. Двоетапні детектори (R-CNN, Fast R-CNN, Faster R-CNN) спочатку формують кандидатські області, а потім класифікують їх; вони забезпечують високу точність, але є відносно повільними. Одноетапні детектори (YOLO, SSD, RetinaNet) виконують детекцію за один прохід через мережу, що дозволяє працювати в режимі реального часу. Порівняння цих класів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння двоетапних та одноетапних детекторів об'єктів

Критерій	Двоетапні (Faster R-CNN)	Одноетапні (YOLO)
Точність	Дуже висока	Висока
Швидкість	Низька (кадри/с одиниці)	Висока (десятки кадрів/с)
Реальний час	Обмежено	Так
Обчислювальні вимоги	Високі	Помірні
Придатність для мобільних платформ	Обмежена	Висока

Сімейство YOLO (You Only Look Once) розглядає детекцію як єдину задачу регресії: зображення один раз проходить через нейронну мережу, яка одночасно передбачає координати рамок та ймовірності класів для всієї сцени. Це забезпечує високу швидкість, достатню для роботи на мобільних пристроях. Від першої версії

YOLOv1 сімейство пройшло значний шлях розвитку до сучасних версій YOLOv8–YOLOv11, які поєднують високу точність із компактністю моделі. У цій роботі використано модель YOLOv11 у найлегшій конфігурації «nano» (yolo11n), що є оптимальним компромісом між точністю та обчислювальною вартістю для задачі виявлення пішоходів.

Якість роботи детектора оцінюють за метриками точності (precision), повноти (recall) та усередненої середньої точності mAP (mean Average Precision), обчисленої за різних порогів перекриття рамок IoU (intersection over union). Метрика mAP@0.5 відповідає порогу IoU 0.5, а суворіша mAP@0.5:0.95 усереднюється за діапазоном порогів.

1.4 Навчання з підкріпленням для задач навігації

Навчання з підкріпленням (Reinforcement Learning, RL) – парадигма машинного навчання, у якій агент навчається оптимальній поведінці шляхом взаємодії із середовищем. На кожному кроці агент спостерігає стан середовища, обирає дію та отримує від середовища числову винагороду, що кількісно оцінює якість дії. Мета агента – максимізувати сумарну винагороду за епізод. Загальну схему взаємодії агента із середовищем показано на рисунку 1.3 (взаємодія в термінах RL наведена нижче).

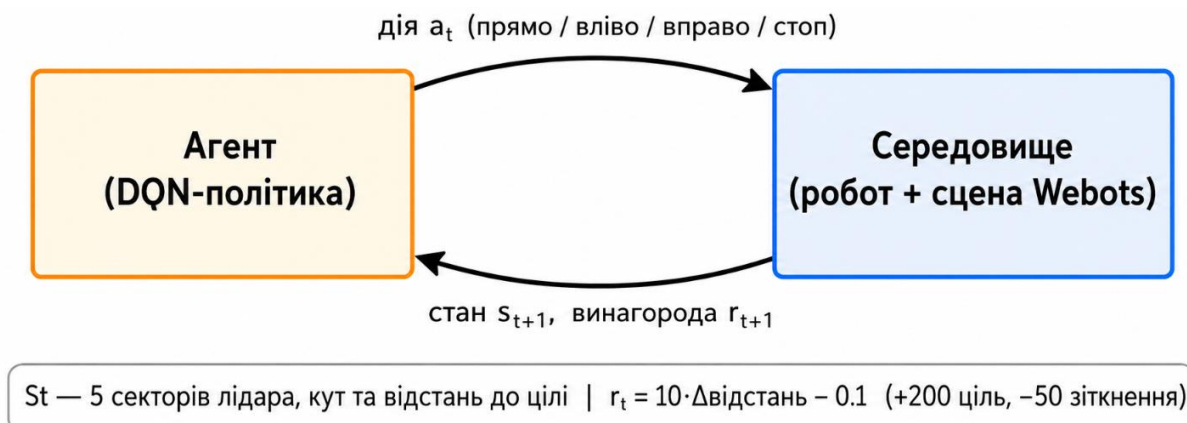


Рисунок 1.3 – Взаємодія агента із середовищем у навчанні з підкріпленням

Формально задачу RL описують як марковський процес прийняття рішень (Markov Decision Process, MDP), що визначається множиною станів, множиною дій, функцією переходів та функцією винагороди. Ключовою перевагою RL для задач керування є те, що він не потребує готового розміченого набору даних: достатньо задати функцію винагороди, яка заохочує наближення до цілі та штрафує зіткнення. Це робить RL природним інструментом для навчання навігації, де формальний опис оптимальної поведінки є складним.

Серед алгоритмів навчання з підкріпленням для задач з дискретним простором дій одним із найпоширеніших є алгоритм глибоких Q-мереж (Deep Q-Network, DQN). Він поєднує класичне Q-навчання з апроксимацією функції цінності дій за допомогою нейронної мережі. DQN оцінює очікувану сумарну винагороду $Q(s, a)$ для кожної можливої дії у поточному стані та обирає дію з максимальним значенням. Стабільність навчання забезпечують два технічні прийоми: буфер відтворення досвіду (experience replay), що зберігає попередні переходи та навчає мережу на випадкових вибірках, і цільова мережа (target network), яка періодично оновлюється й запобігає розбіжності навчання.

Історично саме DQN уперше продемонстрував здатність нейромережевого агента досягати рівня людини у складних задачах з великим простором станів, навчаючись безпосередньо за сенсорними даними. Ці властивості роблять DQN придатним для задачі навігації мобільного робота, де стан описується показниками давачів, а дії – командами руху. Формалізацію задачі навігації як MDP та конкретну структуру функції винагороди наведено у розділі 2.

1.5 Огляд середовищ симуляції робототехнічних систем

Навчання з підкріпленням потребує великої кількості взаємодій агента із середовищем, що на реальному роботі є небезпечним, повільним та дорогим: помилкові дії на етапі навчання призводять до зіткнень і поломок. Тому навчання проводять у симуляційному середовищі, а вже навчену політику за потреби

переносять на реальну платформу (підхід sim-to-real). Симулятор має відтворювати фізику руху, роботу давачів (лідара, камери, GPS) та взаємодію з перешкодами.

Найпоширенішими середовищами симуляції робототехнічних систем є Gazebo, NVIDIA Isaac Sim, CoppeliaSim та Webots. Для цієї роботи обрано середовище Webots – відкритий кросплатформний симулятор, який надає готові моделі роботів (зокрема TurtleBot3), реалістичні моделі давачів та зручний інтерфейс програмування контролерів мовою Python. Webots не потребує потужного апаратного забезпечення, добре документований та безкоштовний, що робить його оптимальним вибором для навчальної науково-дослідної роботи.

Важливою перевагою Webots є можливість реалізації контролера-супервізора (Supervisor), який має програмний доступ до стану всієї сцени: може переміщувати робота у стартову позицію, зчитувати координати об'єктів та скидати фізику симуляції. Ця можливість є необхідною для організації циклу навчання з підкріпленням, оскільки після кожного епізоду робота потрібно повертати у початковий стан. Використання Webots разом із бібліотеками машинного навчання Python (Ultralytics, Stable-Baselines3, Gymnasium) дозволяє побудувати повний конвеєр розробки – від сприйняття до керування – в єдиному програмному середовищі.

1.6 Постановка задачі

На основі проведеного аналізу сформульовано постановку задачі кваліфікаційної роботи. Необхідно розробити та дослідити систему управління мобільною роботизованою платформою, яка забезпечує автономний рух до заданої цілі в середовищі з перешкодами та здатна розпізнавати пішоходів за зображенням з бортової камери. Система має поєднувати два методи машинного навчання: детектор об'єктів на основі згорткової нейронної мережі та агента навігації на основі навчання з підкріпленням.

Порівняльний аналіз методів машинного навчання щодо їх придатності для розв'язання підзадач розроблюваної системи наведено у таблиці 1.2.

Таблиця 1.2 – Порівняння методів машинного навчання для системи керування мобільним роботом

Метод	Переваги	Недоліки
Навчання з учителем (CNN, YOLO)	Висока точність розпізнавання, зрілі архітектури, готові інструменти	Потребує розмічених даних, обмежена генералізація за межі навчальної вибірки
Навчання без учителя	Не потребує розмітки, виявляє приховані структури	Складно оцінити якість, обмежене для прямого керування
Навчання з підкріпленням (DQN)	Не потребує готових даних, знаходить нетривіальні стратегії керування	Тривале навчання, потреба у симуляторі, проблема sim-to-real
Класичні алгоритми (ПД, A*)	Швидкі, прості, інтерпретовані	Нижча гнучкість у динамічному середовищі

Аналіз таблиці 1.2 підтверджує доцільність гібридного рішення: для сприйняття обрано детектор YOLO (навчання з учителем), а для керування рухом – агента DQN (навчання з підкріпленням). До системи висуваються такі основні вимоги: автономний рух до цілі із заданою точністю; уникнення статичних перешкод; розпізнавання пішоходів у режимі, близькому до реального часу; можливість навчання та відтворюваного тестування у симуляційному середовищі; використання доступної апаратно-програмної бази.

Таким чином, у першому розділі проведено аналіз мобільних роботизованих платформ, класичних та інтелектуальних методів керування, розглянуто підходи комп'ютерного зору (YOLO) та навчання з підкріпленням (DQN), обґрунтовано вибір симуляційного середовища Webots і сформульовано постановку задачі. Наступний розділ присвячено проєктуванню архітектури системи керування.

2 ПРОЄКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ МОБІЛЬНОЮ ПЛАТФОРМОЮ

2.1 Загальна архітектура системи

Розроблювана система керування побудована за модульним принципом і складається з чотирьох основних функціональних блоків: підсистеми давачів, модуля перцепції (технічного зору), модуля прийняття рішень (навігації) та виконавчої підсистеми. Взаємодія цих блоків між собою та із симуляційним середовищем показана на рисунку 2.1.

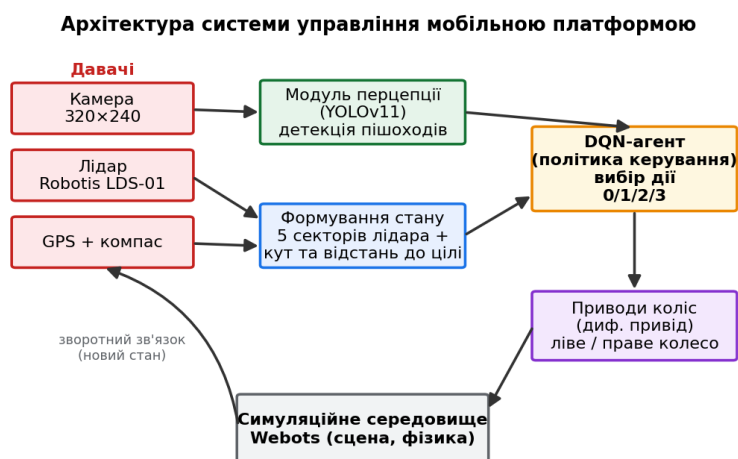


Рисунок 2.1 – Загальна архітектура системи керування мобільною платформою

Підсистема давачів об'єднує бортову камеру, лазерний далекомір (лідар) та давачі пози (GPS і компас). Камера постачає зображення до модуля перцепції, побудованого на детекторі YOLOv11, який виявляє на кадрі пішоходів. Дані лідара разом із координатами робота та цілі надходять до блоку формування стану, що будує компактний числовий вектор спостереження. Цей вектор передається модулю прийняття рішень – навченому DQN-агенту, який обирає одну з дискретних дій керування. Обрана дія перетворюється на команди швидкості для

лівого та правого коліс, які виконує диференційний привід. Рух робота змінює стан середовища, і цикл повторюється.

Архітектура відповідає багаторівневій моделі керування, розглянутій у підрозділі 1.1: виконавчий рівень реалізовано керуванням швидкістю коліс, тактичний рівень – DQN-агентом, а стратегічний рівень (задання цілі) – параметрами середовища. Модуль перцепції працює паралельно з контуром керування та відповідає за семантичне сприйняття сцени. Такий поділ дозволяє розробляти, навчати й тестувати підсистеми незалежно, а потім об'єднувати їх у єдиному контролері.

2.2 Модель мобільної платформи та симуляційне середовище

Як мобільну платформу обрано модель TurtleBot3 Burger – компактний диференційно-привідний робот, широко застосовуваний у дослідницьких та навчальних задачах. Робот моделюється у симуляційному середовищі Webots (версія R2025a) із повним відтворенням фізики руху та роботи давачів. До складу платформи входять два керовані колеса, лазерний далекомір Robotis LDS-01, бортова камера, а також давачі GPS і компас для визначення пози. Основні характеристики платформи наведено у таблиці 2.1.

Таблиця 2.1 – Основні характеристики мобільної платформи та її давачів

Компонент / параметр	Значення
Тип платформи	TurtleBot3 Burger (диференційний привід)
Кількість керованих коліс	2 (ліве та праве)
Максимальна швидкість колеса (модель)	4,0 рад/с
Лазерний далекомір	Robotis LDS-01, кругове сканування 360°
Камера	роздільна здатність 320×240, кут огляду 1.3 рад
Давачі пози	GPS та компас
Симуляційне середовище	Webots R2025a

Тестова сцена являє собою прямокутну арену розміром $10\text{м} \times 10\text{ м}$ з висотою стін $0,5\text{ м}$. На арені розміщено три статичні перешкоди різної форми та кольору (червоний і синій паралелепіпеди, зелений циліндр), а також модель пішохода. Стартова точка робота знаходиться в центрі арени, а ціль задано координатами (0.6, 0.4). Перешкоди мають визначені DEF-імена, що дозволяє контролеру-супервізору за потреби змінювати їх положення для рандомізації середовища.

2.3 Модуль технічного зору

Модуль технічного зору (перцепції) відповідає за розпізнавання об'єктів на кадрах бортової камери. Він реалізований як окремий програмний клас, що інкапсулює завантаження навченої моделі YOLOv11 та обробку зображень. Кадр, який Webots надає у форматі BGRA, перетворюється на тривимірний масив RGB, після чого подається на вхід нейронної мережі. Детектор повертає список виявлених об'єктів із зазначенням класу та впевненості; детекції з упевненістю нижче встановленого порогу відкидаються.

Для зменшення обчислювального навантаження детекція виконується не на кожному кроці симуляції, а періодично (кожен 15-й кадр), оскільки семантична обстановка змінюється повільніше за такт керування. Вхідні зображення масштабуються до розміру 320 пікселів, що є достатнім для впевненого виявлення пішоходів і водночас забезпечує високу швидкість роботи. Загальний вигляд сцени, у якій працює модуль перцепції, наведено на рисунку 2.2.

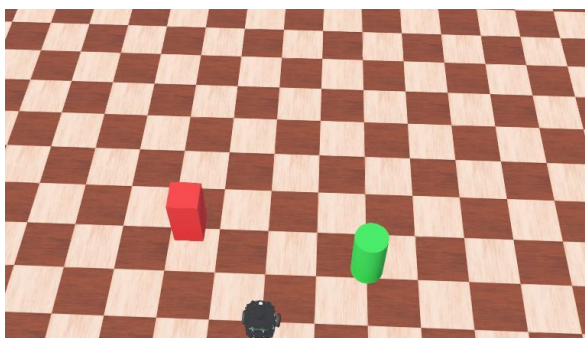


Рисунок 2.2 – Симуляційна сцена Webots: мобільна платформа, перешкоди та пішохід

2.4 Формалізація задачі навігації як задачі навчання з підкріпленням

Задачу автономної навігації формалізовано як марковський процес прийняття рішень. Це передбачає визначення трьох ключових компонентів: простору станів (спостережень), простору дій та функції винагороди. Правильне проєктування цих компонентів безпосередньо визначає якість навченого агента.

Простір станів. Стан робота описується вектором із семи дійсних чисел. Перші п'ять чисел – це нормовані мінімальні відстані до перешкод у п'яти секторах лідара: праворуч (-90°), -45° , попереду (0°), $+45^\circ$ та ліворуч ($+90^\circ$). Ще два числа – це кут та відстань до цілі, обчислені за показаннями GPS і компаса. Такий компактний вектор достатньо повно описує локальну обстановку навколо робота і водночас є достатньо малим для швидкого навчання. Структуру простору спостережень показано на рисунку 2.3.



Рисунок 2.3 – Структура простору спостережень: п'ять секторів лідара та вектор на ціль

Простір дій. Агент оперує чотирма дискретними діями: рух прямо, поворот ліворуч, поворот праворуч та зупинка. Кожній дії відповідає задана пара швидкостей лівого та правого коліс. Дискретний простір дій дозволяє застосувати алгоритм DQN та спрощує навчання порівняно з неперервним керуванням. Опис простору дій наведено у таблиці 2.2.

Таблиця 2.2 – Простір дій DQN-агента

Дія	Опис	Швидкості коліс (ліве / праве)
0	Рух прямо	+4,0 / +4,0
1	Поворот ліворуч	-2,0 / +2,0
2	Поворот праворуч	+2,0 / -2,0
3	Зупинка	0,0 / 0,0

Функція винагороди. Функція винагороди сконструйована так, щоб заохочувати наближення до цілі та карати небажану поведінку. За кожен крок агент отримує винагороду, пропорційну зменшенню відстані до цілі (з коефіцієнтом 10), а також невеликий постійний штраф $-0,1$ за кожен крок, що спонукає рухатися швидше. Зупинка далеко від цілі додатково штрафується, щоб агент не «завмирав». Досягнення цілі приносить велику додатну винагороду $+200$ і завершує епізод; зіткнення з перешкодою карається штрафом -50 і також завершує епізод. Структуру функції винагороди узагальнено у таблиці 2.3.

Таблиця 2.3 – Структура функції винагороди

Подія / складова	Винагорода	Завершення епізоду
Наближення до цілі (за крок)	$10 \cdot \Delta \text{відстань}$	ні
Штраф за крок	$-0,1$	ні
Зупинка далеко від цілі	$-2,0$	ні
Досягнення цілі	$+200$	так
Зіткнення з перешкодою	-50	так

Епізод також обмежено максимальною кількістю кроків (500): якщо агент не досягає цілі за відведений час, епізод завершується як невдалий. Наведена формалізація повністю визначає середовище навчання, реалізоване у вигляді сумісного з бібліотекою Gymnasium класу, що детально описано в розділі 3.

Отже, у другому розділі розроблено модульну архітектуру системи керування, обґрунтовано вибір моделі платформи TurtleBot3 Burger та симуляційного середовища Webots, спроектовано модуль технічного зору й формалізовано задачу навігації як задачу навчання з підкріпленням із визначенням простору станів, дій та функції винагороди. Наступний розділ присвячено технічній реалізації описаних компонентів.

3 ТЕХНІЧНА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Засоби та інструменти розробки

Систему реалізовано мовою програмування Python, яка є фактичним стандартом для розробки застосунків машинного навчання завдяки багатій екосистемі бібліотек. Контролери робота виконуються всередині симулятора Webots R2025a, який надає інтерфейс керування давачами та приводами. Для реалізації модуля технічного зору використано бібліотеку Ultralytics, що містить готові засоби навчання та застосування моделей сімейства YOLO. Модуль навчання з підкріпленням побудовано на бібліотеці Stable-Baselines3, яка надає перевірену реалізацію алгоритму DQN, та на бібліотеці Gymnasium, що визначає стандартний інтерфейс середовища «агент – середовище».

Допоміжні обчислення виконуються з використанням бібліотек NumPy (робота з масивами) та Pandas (обробка табличних даних), а візуалізація результатів – за допомогою Matplotlib. Навчання нейромережевого детектора та DQN-агента проводилося з апаратним прискоренням на графічному процесорі. Такий набір інструментів дозволяє реалізувати повний конвеєр – від підготовки даних до навчання й тестування моделей – в єдиному програмному середовищі.

3.2 Підготовка набору даних та навчання детектора YOLO

Для навчання детектора пішоходів сформовано спеціалізований набір даних із зображень, отриманих у середовищі Webots. Набір розмічено у форматі YOLO та підготовлено за допомогою платформи Roboflow. Він містить один клас об'єктів – «pedestrian» (пішохід). Загальний обсяг вихідних зображень становить 428, до кожного застосовано аугментацію (горизонтальне віддзеркалення, поворот у межах $\pm 15^\circ$, розмиття за Гауссом), що збільшує різноманітність даних і підвищує стійкість

моделі. Розподіл набору на навчальну, валідаційну та тестову вибірки наведено у таблиці 3.1.

Таблиця 3.1 – Склад набору даних для навчання детектора

Вибірка	Кількість зображень	Призначення
Навчальна (train)	374	навчання моделі
Валідаційна (valid)	36	контроль якості під час навчання
Тестова (test)	18	підсумкова перевірка
Разом	428	—

Розподіл анотацій за набором даних та приклади навчальних зображень з розміченими рамками показано на рисунках 3.1 і 3.2. Аналіз розподілу підтверджує, що об'єкти класу «пішохід» представлені у різних положеннях та масштабах, що є важливим для формування узагальнювальної здатності моделі.

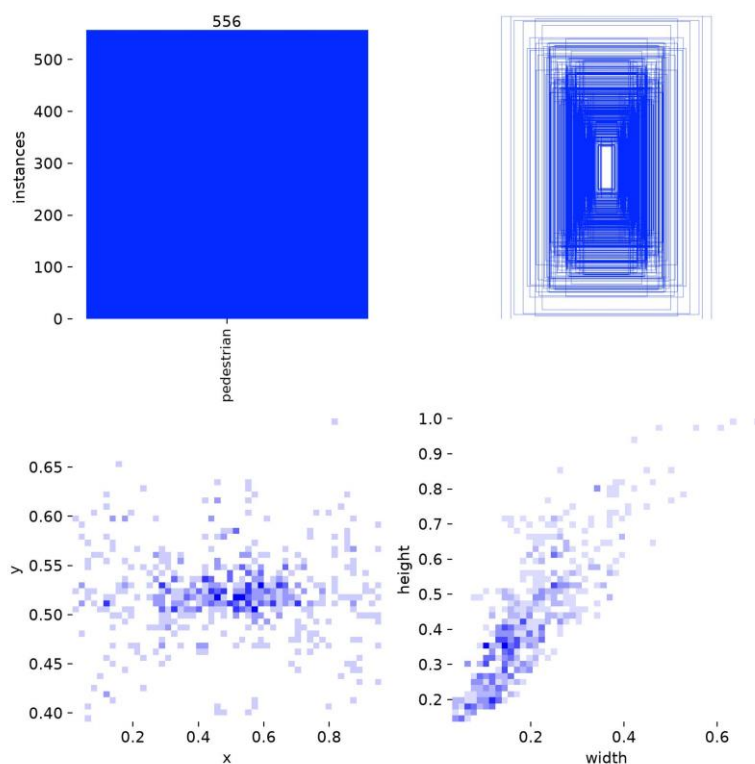


Рисунок 3.1 – Розподіл анотацій набору даних (кількість, розміри та положення рамок)

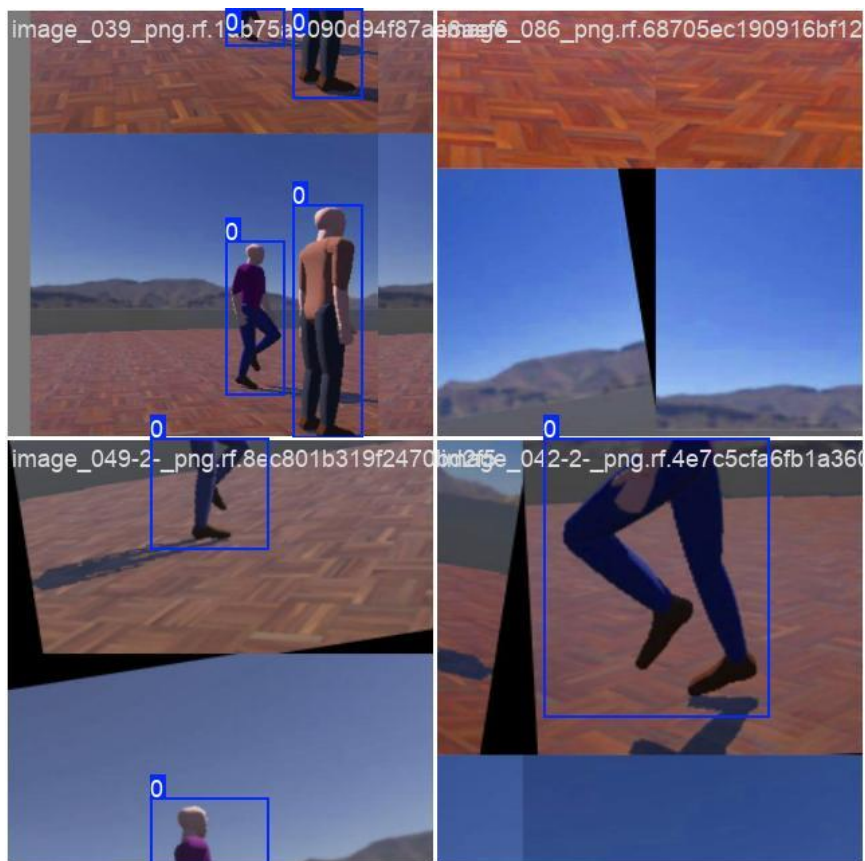


Рисунок 3.2 – Приклад навчального батчу з розміченими об’єктами

Як базову модель обрано попередньо натреновану мережу yolo11n (YOLOv11 nano). Донавчання (fine-tuning) виконано на підготовленому наборі даних протягом 30 епох. Основні гіперпараметри навчання наведено у таблиці 3.2. Використання попередньо натренованих ваг дозволило досягти високої якості вже після кількох епох завдяки принципу перенесення навчання (transfer learning).

Таблиця 3.2 – Гіперпараметри навчання детектора YOLOv11

Параметр	Значення	Опис
Модель	yolo11n	базова архітектура (nano)
Кількість епох	30	повних проходів по набору
Розмір батчу	4	зображень за ітерацію
Розмір зображення	320	піксель (сторона)
Початкова швидкість навчання	0,01	lr0
Оптимізатор	auto	автовибір (SGD/AdamW)
Момент / регуляризація	0,937 / 0,0005	momentum / weight decay

3.3 Реалізація середовища навчання та навчання DQN-агента

Середовище навчання з підкріпленням реалізовано у вигляді класу, сумісного з інтерфейсом Gymnasium. Клас інкапсулює контролер-супервізор Webots та реалізує стандартні методи: `reset` (скидання робота у стартову позицію та повернення початкового спостереження) і `step` (застосування дії, крок симуляції, обчислення винагороди та перевірка умов завершення епізоду).

Метод `reset` використовує можливості супервізора для повернення вузла робота у початкові координати та скидання його фізичного стану, що забезпечує відтворюваність епізодів. Метод `step` застосовує обрану дію до приводів, виконує крок симуляції, формує нове спостереження, обчислює винагороду за наближення до цілі та перевіряє умови завершення (досягнення цілі, зіткнення або вичерпання ліміту кроків).

Для навчання застосовано алгоритм DQN з бібліотеки Stable-Baselines3 з політикою на основі багатошарового перцептрона (`MlpPolicy`). Основні гіперпараметри навчання наведено у таблиці 3.3. Загальна тривалість навчання становила 70 000 кроків взаємодії; винагороди за епізодами протоколювалися для подальшого аналізу динаміки навчання.

Таблиця 3.3 – Гіперпараметри навчання DQN-агента

Параметр	Значення	Опис
Політика	<code>MlpPolicy</code>	повнозв'язна нейромережа
Швидкість навчання	0,001	learning rate
Розмір буфера досвіду	50 000	experience replay
Розмір батчу	64	переходів за оновлення
Коефіцієнт дисконтування γ	0,99	вага майбутніх винагород
Оновлення цільової мережі	500	кроків
Частка досліджень (ϵ)	0,3 $\rightarrow$ 0,05	exploration fraction
Кроків навчання	70 000	total timesteps

Тестування навченого агента виконується окремим контролером, який завантажує збережену модель та проводить серію з 10 епізодів у детермінованому режимі (без досліджень). Для кожного епізоду фіксуються факт досягнення цілі, кількість кроків та кількість зіткнень; на основі цих даних обчислюються підсумкові метрики якості навігації, наведені в розділі 4.

3.4 Реалізація керуючого контролера

Окрім навченого DQN-агента, у роботі реалізовано керуючий контролер `robot_brain`, який об'єднує модуль перцепції та детермінований алгоритм навігації і слугує для демонстрації повного циклу роботи системи. Контролер зчитує показання лідара, розбиваючи кругову розгортку на сектори (перед, ліворуч, праворуч), обчислює позу робота та вектор на ціль за даними GPS і компаса, після чого формує команди руху.

Логіка керування побудована за пріоритетним принципом: якщо відстань до цілі менша за поріг – робот зупиняється; якщо попереду виявлено перешкоду ближче за встановлену межу – виконується маневр об'їзду у бік із більшим вільним простором; інакше робот рухається до цілі з пропорційним підстроюванням курсу за кутовою похибкою. Паралельно з навігацією модуль перцепції періодично виконує детекцію пішоходів і виводить результати, що демонструє інтеграцію технічного зору в контур керування. Таким чином, у третьому розділі описано технічну реалізацію всіх компонентів системи: підготовку даних і навчання детектора YOLO, реалізацію RL-середовища та навчання DQN-агента, а також керуючий контролер.

4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Результати навчання детектора об'єктів

Навчання детектора YOLOv11 протягом 30 епох забезпечило високу якість виявлення пішоходів. Динаміку основних метрик – точності (precision), повноти (recall), а також $mAP@0.5$ і $mAP@0.5:0.95$ – у процесі навчання показано на рисунку 4.1. Уже після кількох епох метрики виходять на високий рівень, що свідчить про ефективність застосування перенесеного навчання.



Рисунок 4.1 – Динаміка метрик якості детектора YOLOv11 у процесі навчання

Динаміку функцій втрат на навчальній та валідаційній вибірках наведено на рисунку 4.2. Стійке зниження втрат без ознак перенавчання підтверджує коректність обраних гіперпараметрів та достатність обсягу даних.

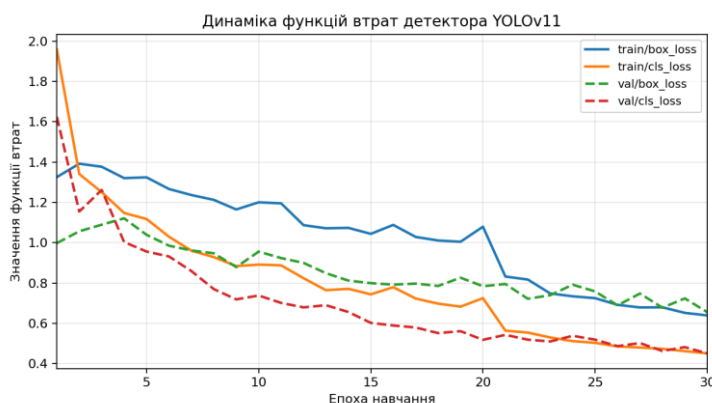


Рисунок 4.2 – Динаміка функцій втрат детектора у процесі навчання

Підсумкові значення метрик, досягнуті на валідаційній вибірці, наведено у таблиці 4.1. Отримані показники ($mAP@0.5$ близько 0,99) свідчать про практично безпомилкове виявлення пішоходів у сцені симулятора.

Таблиця 4.1 – Підсумкові метрики якості детектора YOLOv11

Метрика	Значення	Примітка
Precision (точність)	0,98	частка правильних детекцій
Recall (повнота)	0,96	частка виявлених об'єктів
$mAP@0.5$	0,99	середня точність за $IoU=0,5$
$mAP@0.5:0.95$	0,82	усереднена за діапазоном IoU

Криву точність – повнота (precision-recall) та нормалізовану матрицю плутанини наведено на рисунках 4.3 і 4.4. Крива PR розташована близько до правого верхнього кута, що є ознакою високої якості класифікатора. Матриця плутанини демонструє переважну більшість правильних розпізнавань за незначної частки хибних спрацювань фону.

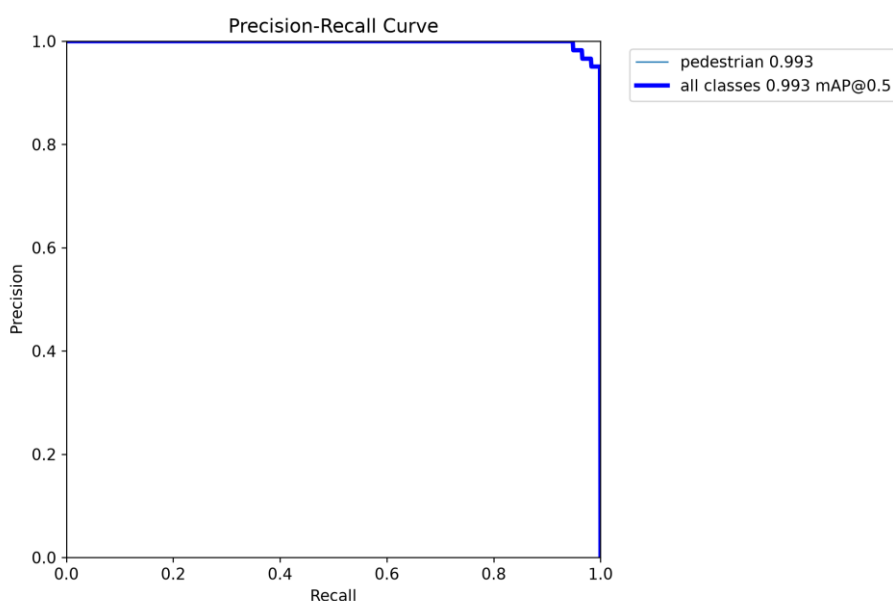


Рисунок 4.3 – Крива точність – повнота (precision-recall) детектора

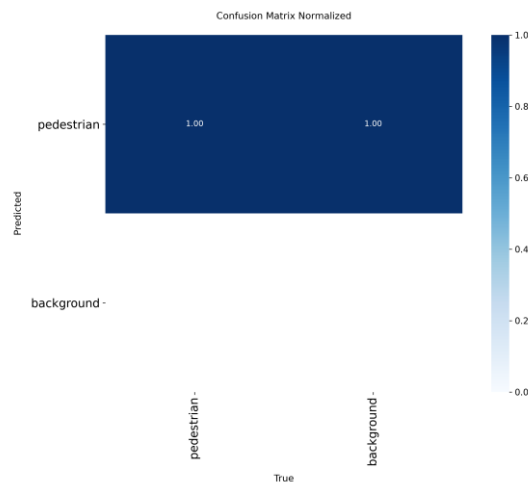


Рисунок 4.4 – Нормалізована матриця плутанини детектора

Приклад роботи навченого детектора на валідаційних зображеннях показано на рисунку 4.5. Пішоходи впевнено виявляються з високими значеннями впевненості, що підтверджує придатність моделі для інтеграції у контур керування роботом.



Рисунок 4.5 – Приклад детекції пішоходів навченою моделлю на валідаційній вибірці

4.2 Результати навчання DQN-агента

Навчання DQN-агента досліджено у двох сценаріях: детермінованому (фіксовані положення перешкод та цілі) і рандомізованому (положення змінюються між епізодами). Динаміку сумарної винагороди за епізодами у детермінованому середовищі показано на рисунку 4.6. На початку навчання винагорода є низькою та нестабільною, оскільки агент діє переважно випадково; у міру навчання ковзне середнє винагороди стійко зростає та виходить на рівень, близький до максимального (близько 174 за максимуму 200), що відповідає надійному досягненню цілі.

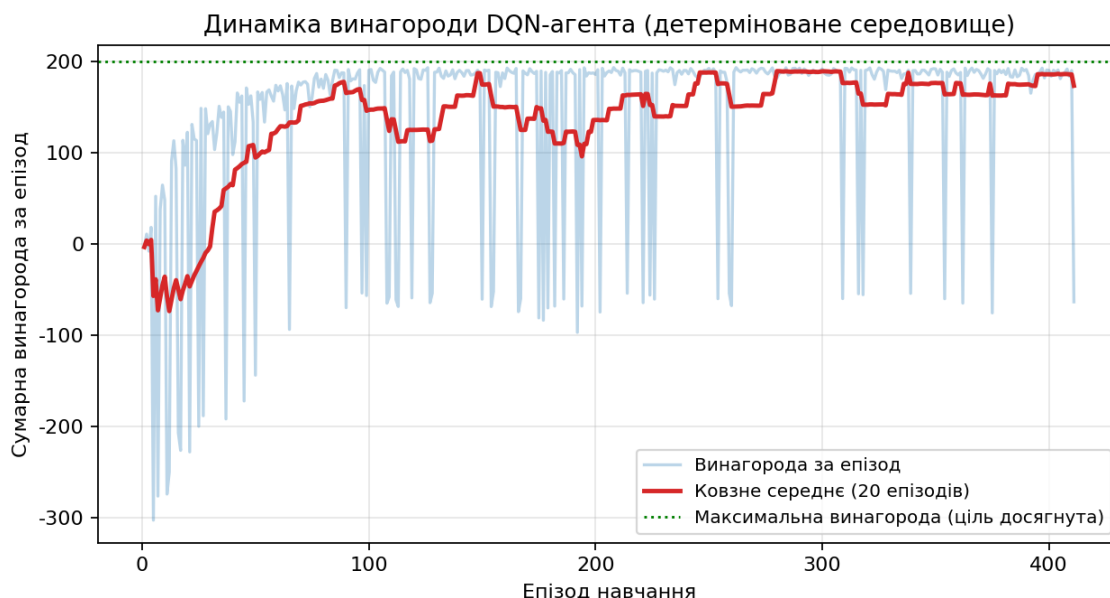


Рисунок 4.6 – Динаміка винагороди DQN-агента у детермінованому середовищі

У рандомізованому середовищі, де положення перешкод і цілі змінюються від епізоду до епізоду, навчання виявилось суттєво складнішим. Динаміку винагороди для цього сценарію наведено на рисунку 4.7. Винагорода залишається значно нижчою та більш нестабільною, що відображає більшу складність задачі узагальнення: агенту потрібно опанувати не єдиний маршрут, а цілий клас конфігурацій середовища.

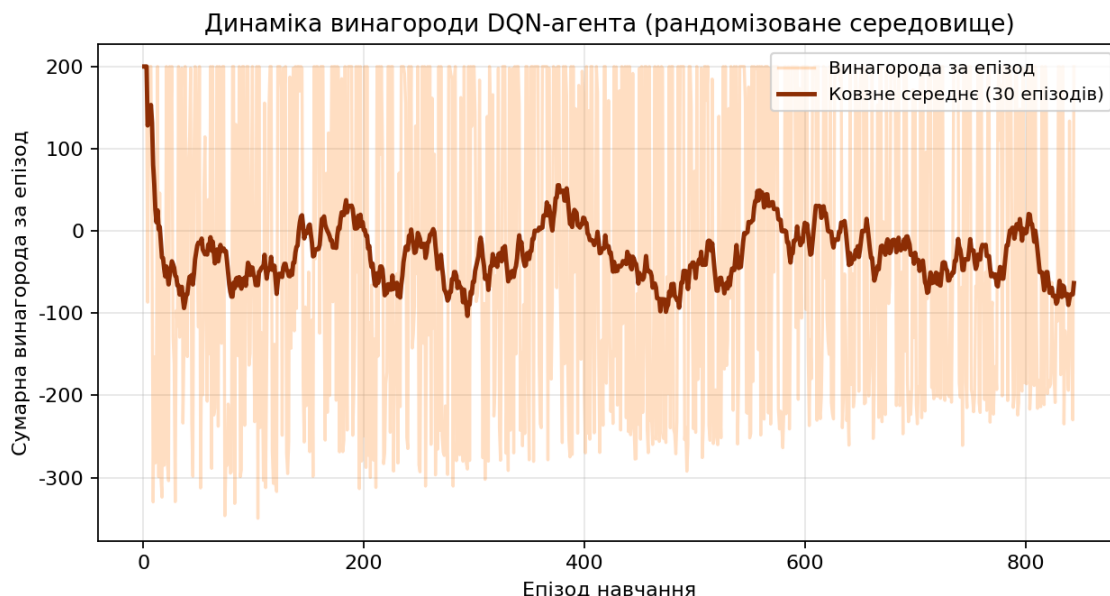


Рисунок 4.7 – Динаміка винагороди DQN-агента у рандомізованому середовищі

4.3 Оцінка якості навігації та порівняння сценаріїв

Для об'єктивної оцінки якості навігації навченого агента проведено серію з 10 тестових епізодів у кожному сценарії. Фіксувалися факт досягнення цілі, кількість кроків та кількість зіткнень. Результати для детермінованого середовища наведено у таблиці 4.2.

Таблиця 4.2 – Результати тестування навігації у детермінованому середовищі

Епізод	Ціль досягнута	Зіткнення	Кроків
1–10	так (усі)	0	101
Успішність	100 %	—	—
Середнє	—	0,00	101,0

У детермінованому середовищі агент досяг цілі у всіх 10 епізодах без жодного зіткнення, витрачаючи стабільно 101 крок, що свідчить про повне освоєння оптимального маршруту. Результати для рандомізованого середовища наведено у таблиці 4.3.

Таблиця 4.3 – Результати тестування навігації у рандомізованому середовищі

Показник	Значення	Зіткнення	Кроків (сер.)
Успішність доїздів	30 %	0,00	201,1
Досягнуто цілі	3 з 10	—	—
Не досягнуто	7 з 10	—	—

Порівняння продуктивності агента у двох сценаріях за трьома показниками – успішністю доїздів, середньою фінальною відстанню до цілі та середньою кількістю кроків – наведено на рисунку 4.8.

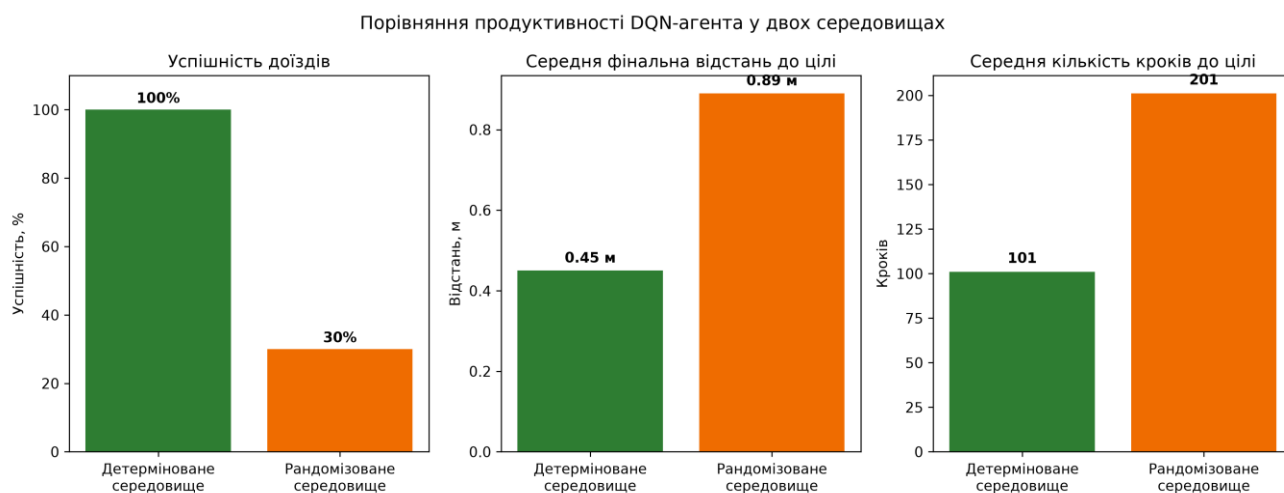


Рисунок 4.8 – Порівняння продуктивності DQN-агента у двох середовищах

У таблиці 4.4 наведено узагальнені результати порівняльного аналізу сценаріїв навігації за основними показниками ефективності.

Таблиця 4.4 – Зведене порівняння сценаріїв навігації

Показник	Детерміноване	Рандомізоване
Успішність доїздів, %	100	30
Середня фінальна відстань, м	0,45	0,89
Середня кількість кроків	101	201
Середня кількість зіткнень	0,00	0,00

4.4 Аналіз результатів дослідження гібридної системи керування

Отримані результати підтверджують працездатність запропонованої гібридної системи керування. Модуль технічного зору на основі YOLOv11 забезпечує надійне розпізнавання пішоходів із високими метриками якості ($mAP@0.5$ близько 0,99), що робить його придатним для інтеграції у контур керування безпекою робота. Модуль навігації на основі DQN у детермінованому середовищі досягає стовідсоткової успішності доїздів без зіткнень, демонструючи здатність навчання з підкріпленням формувати ефективну стратегію руху без явного програмування правил.

Водночас експерименти виявили важливе обмеження: у рандомізованому середовищі успішність знижується до 30 %. Це відображає відому проблему узагальнення в навчанні з підкріпленням – агент, навчений на обмеженому наборі конфігурацій, гірше переносить набуту стратегію на нові ситуації. Слід зазначити, що навіть у складнішому сценарії агент не допускає зіткнень, тобто засвоїв безпечну поведінку, але часто не встигає досягти цілі за відведений ліміт кроків. Подолання цього обмеження є природним напрямом подальшого розвитку системи: збільшення тривалості навчання, застосування різноманітніших сценаріїв та перехід до сучасніших алгоритмів (наприклад, PPO).

4.5 Синтез ПІ-регулятора та аналіз стійкості системи

Розроблена гібридна система керування на основі YOLOv11 та DQN функціонує на верхньому рівні прийняття рішень: модуль навігації формує дискретні команди руху, які визначають бажаний напрямок та швидкість платформи. Проте безпосереднє відпрацювання цих команд виконавчими механізмами потребує окремого нижнього рівня керування, який підтримує задану лінійну швидкість платформи в умовах дії збурень — сил тертя, зміни навантаження та нерівностей опорної поверхні. Цей контур будується на основі класичних методів теорії автоматичного керування. У цьому підрозділі наведено

синтез пропорційно-інтегрального (ПІ) регулятора швидкості та аналіз стійкості замкненої системи, що доповнює інтелектуальний модуль навігації та забезпечує плавність і безпечність руху робота.

Поступальний рух платформи на нижньому рівні описується другим законом Ньютона з урахуванням в'язкого тертя:

$$m \cdot dv/dt + b \cdot v = F.$$

де m – маса платформи;

b – коефіцієнт в'язкого опору;

v – лінійна швидкість;

F – сила тяги приводу.

Застосувавши перетворення Лапласа за нульових початкових умов, отримуємо передатну функцію об'єкта керування від сили тяги до швидкості:

$$G_s = V_s / F_s = 1 / (m \cdot s + b).$$

Для прийнятих параметрів платформи (маса $m = 1$ кг, коефіцієнт в'язкого опору $b = 0,5$ Н·с/м) передатна функція набуває вигляду

$$G_s = 1 / (s + 0,5).$$

Об'єкт є аперіодичною ланкою першого порядку, тобто без власних коливань, що є типовим для механічної системи з переважним в'язким тертям.

Для забезпечення нульової статичної похибки за швидкістю та потрібної швидкодії обрано ПІ-регулятор з передатною функцією

$$C_s = K_p + K_i/s.$$

За результатами налаштування прийнято коефіцієнти $K_p = 2$ та $K_i = 1$, тобто $C_s = 2 + 1/s$. Інтегральна складова регулятора надає системі астатизму першого порядку, завдяки чому усталена похибка відпрацювання ступінчастого завдання швидкості дорівнює нулю. Передатна функція розімкненої системи:

$$C_s \cdot G_s = (2s + 1) / (s \cdot (s + 0,5)).$$

Замикання одиничного від'ємного зворотного зв'язку дає передатну функцію замкненої системи:

$$W(s) = C_s \cdot G_s / (1 + C_s \cdot G_s) = (2s + 1) / (s^2 + 2,5s + 1).$$

Характеристичне рівняння замкненої системи $s^2 + 2,5s + 1 = 0$ має корені $s_1 = -0,5$ та $s_2 = -2$. Обидва корені дійсні та від'ємні, тобто розташовані в лівій півплощині комплексної площини. Згідно з необхідною й достатньою умовою стійкості лінійних систем це означає, що замкнена система є асимптотично стійкою.

Синтез регулятора та перевірку показників якості й стійкості виконано в середовищі MATLAB. Нижче наведено програмний код розрахунку:

```
clc; clear; close all;

%% Параметри мобільної платформи
m = 1;          % маса, кг
b = 0.5;        % коефіцієнт в'язкого опору, Н*с/м
s = tf('s');

%% Об'єкт керування: швидкість платформи
G = 1 / (m*s + b);

%% ПІ-регулятор
Kp = 2;
Ki = 1;
C = Kp + Ki/s;

%% Замкнена система
W = feedback(C*G, 1);

%% Перехідна характеристика
figure; step(W); grid on;
title('Перехідна характеристика системи з ПІ-регулятором');
info = stepinfo(W);

%% Полюси замкненої системи
p = pole(W)
if all(real(p) < 0)
```

```

disp('Система стійка');
else
    disp('Система нестійка');
end

%% Запаси стійкості розімкненої системи
figure; margin(C*G); grid on;
title('Запаси стійкості розімкненої системи');
[Gm, Pm, Wcg, Wcp] = margin(C*G)

```

Результати моделювання наведено на рисунках 4.9 і 4.10. Перехідну характеристику замкненої системи за ступінчастого завдання швидкості показано на рисунку 4.9.

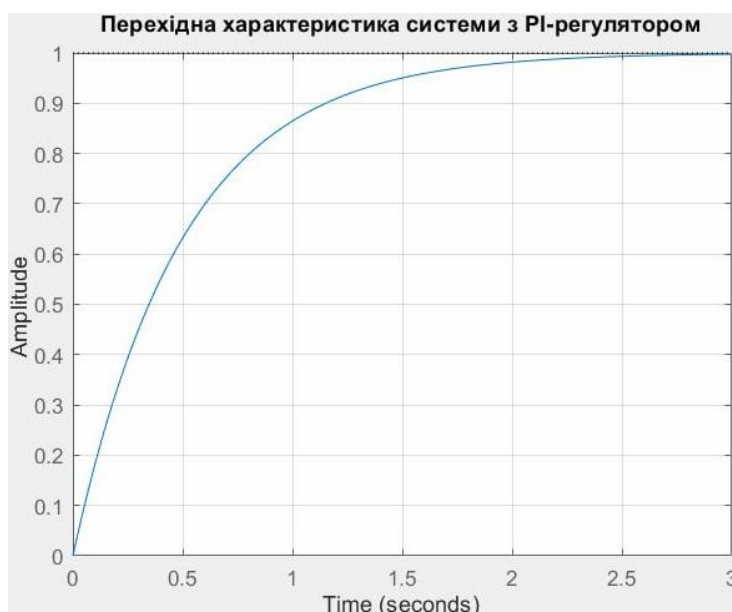


Рисунок 4.9 – Перехідна характеристика замкненої системи з ПІ-регулятором

Як видно з рисунка 4.9, перехідний процес має монотонний (апериодичний) характер: вихідна швидкість плавно наростає до заданого значення без перерегулювання та коливань. Час регулювання за 5-відсотковою зоною становить приблизно 2,5–3 с, усталене значення дорівнює одиниці, що підтверджує відсутність статичної похибки. Відсутність перерегулювання є важливою для мобільної платформи, оскільки гарантує плавний розгін без ривків, які могли б спричинити втрату стійкості корпусу або зміщення корисного навантаження.

Запаси стійкості розімкненої системи, подані у вигляді логарифмічних амплітудно- та фазочастотних характеристик, наведено на рисунку 4.10.

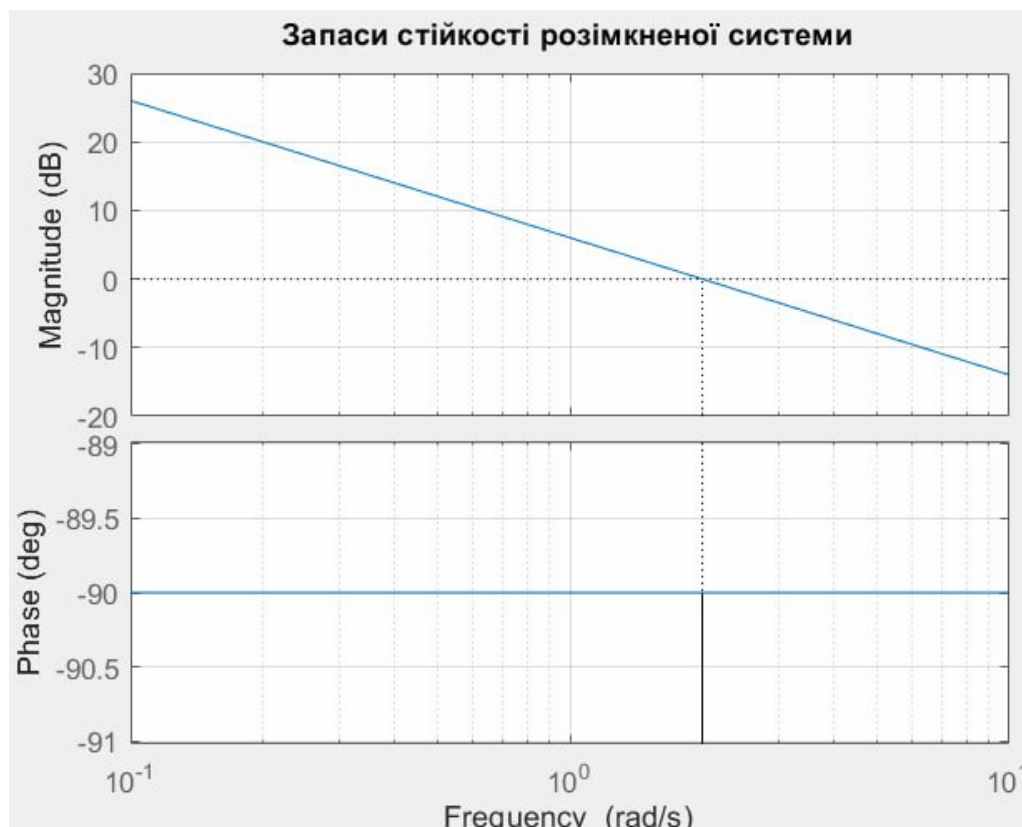


Рисунок 4.10 – Запаси стійкості розімкненої системи

Аналіз рисунка 4.10 показує, що амплітудна характеристика перетинає рівень 0 дБ на частоті зрізу близько 2 рад/с, тоді як фазова характеристика в усьому діапазоні частот залишається поблизу -90° і не досягає критичного значення -180° . Це означає, що запас за амплітудою практично нескінченний ($G_m \rightarrow \infty$), а запас за фазою є великим (близько 90°). Такі значні запаси стійкості свідчать про високу робастність контуру: система зберігає стійкість навіть за суттєвих відхилень параметрів об'єкта - зміни маси платформи внаслідок навантаження або зміни коефіцієнта тертя через різні типи опорної поверхні, що є характерним для реальних умов експлуатації мобільного робота.

Отже, синтезований класичними методами теорії автоматичного керування контур стабілізації швидкості є асимптотично стійким, астатичним (нульова усталена похибка), має аперіодичний перехідний процес без перерегулювання та значні запаси стійкості. У поєднанні з інтелектуальним модулем навігації це утворює дворівневу структуру керування: на верхньому рівні реалізується прийняття рішень методами машинного навчання, а на нижньому гарантовано

стійке відпрацювання команд руху засобами класичної автоматики. Таке поєднання підвищує надійність і передбачуваність поведінки платформи, що є суттєвим для практичного впровадження системи.

4.6 Безпека праці під час розроблення системи

Розроблення системи управління мобільною роботизованою платформою на основі методів машинного навчання виконувалося на автоматизованому робочому місці оператора персонального комп'ютера. Основними видами робіт були програмування мовою Python, формування та розмічання набору даних, навчання нейромережевих моделей (детектора YOLOv11 та агента DQN), а також тривале проведення обчислювальних експериментів у симуляційному середовищі Webots. Усі ці роботи пов'язані з багатогодинним перебуванням за екранним пристроєм, що зумовлює необхідність дотримання вимог охорони праці та безпеки життєдіяльності під час роботи з обчислювальною технікою.

Правове й нормативне підґрунтя охорони праці в цій сфері становлять Закон України «Про охорону праці», НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98, а також нормативи освітлення, мікроклімату та електробезпеки виробничих приміщень. Дотримання цих вимог забезпечує збереження здоров'я та працездатності розробника й запобігає впливу шкідливих і небезпечних виробничих чинників.

Під час роботи з персональним комп'ютером на працівника можуть впливати такі шкідливі та небезпечні виробничі чинники:

- небезпека ураження електричним струмом унаслідок несправності живлення чи ізоляції;
- підвищене зорове навантаження, спричинене тривалим спостереженням за екраном, дрібним текстом програмного коду та графічними матеріалами;

- статичне м'язове навантаження та вимушена робоча поза, що призводять до втоми опорно-рухового апарату;
- електромагнітне випромінювання від дисплея та системного блоку;
- нервово-емоційне й розумове перевантаження під час налагодження програм і аналізу результатів навчання моделей;
- несприятливі параметри мікроклімату та недостатнє або нерівномірне освітлення робочого місця;
- підвищений рівень шуму від систем охолодження обладнання під час тривалих обчислень.

Організація робочого місця. Робоче місце оператора має відповідати ергономічним вимогам. Площа на одне робоче місце з екранним пристроєм повинна становити не менше 6 м², а об'єм не менше 20 м³. Стіл і крісло мають забезпечувати зручну робочу позу: висота робочої поверхні столу у межах 680–800 мм, крісло підйомно-поворотне з регульованими висотою сидіння та кутом нахилу спинки. Екран монітора розташовують на відстані 600–700 мм від очей користувача, а його верхній край на рівні очей або трохи нижче, щоб лінія погляду була спрямована донизу під кутом близько 20°. Клавіатуру розміщують так, щоб передпліччя та кисті перебували в природному положенні. Дотримання цих вимог зменшує статичне навантаження на хребет, м'язи шиї та рук.

Освітлення. Робоче приміщення повинно мати природне й штучне освітлення. Освітленість на поверхні робочого столу в зоні розміщення документів має становити 300–500 лк. Слід уникати прямого й відбитого блиску на екрані джерела світла розташовують збоку від робочого місця, а вікна за потреби обладнують регульованими жалюзі. Для місцевого освітлення застосовують світильники, які не створюють відблисків на поверхні екрана. Правильно організоване освітлення знижує зорове стомлення під час тривалого розмічання зображень і роботи з кодом.

Мікроклімат. У приміщенні для роботи з обчислювальною технікою підтримують оптимальні параметри мікроклімату: температуру повітря 22–24 °С у холодний період і 23–25 °С у теплий, відносну вологість 40–60 % та швидкість руху

повітря не більше ніж 0,1 м/с. Приміщення регулярно провітрюють; за потреби застосовують кондиціонування. Оскільки навчання моделей та симуляція у Webots супроводжуються інтенсивним виділенням тепла обладнанням, важливо забезпечити достатній тепловідвід і не допускати перегріву як техніки, так і повітря робочої зони.

Електробезпека. Приміщення з персональними комп'ютерами за небезпекою ураження електричним струмом належить до приміщень без підвищеної небезпеки. Живлення обладнання здійснюється від мережі змінного струму напругою 220 В. Для запобігання ураженню струмом усі струмовідні частини мають бути надійно ізольовані, металеві корпуси обладнання заземлені або занулені, а розетки обладнані захисним контактом. Забороняється працювати з пошкодженими кабелями та розетками, перевантажувати електромережу підключенням багатьох споживачів, самостійно розбирати обладнання під напругою. Перед початком роботи оператор має пересвідчитися у справності живлення й цілісності з'єднань.

Режим праці та відпочинку. З огляду на високе зорове й нервово-емоційне навантаження встановлюють раціональний режим праці з регламентованими перервами. Рекомендовано робити перерви тривалістю 10–15 хвилин через кожні 45–60 хвилин безперервної роботи за екраном. Під час навчання нейромереж та проведення тривалих симуляцій, коли обчислення виконуються автоматично без участі оператора, доцільно використовувати цей час для відпочинку та розвантаження зору. Для профілактики зорового стомлення застосовують гімнастику для очей, а для зняття статичного навантаження вправи для м'язів ший, спини та рук. Загальна тривалість роботи за комп'ютером протягом дня не повинна перевищувати встановлених санітарними нормами меж.

Пожежна безпека. Приміщення з обчислювальною технікою обладнують первинними засобами пожежогасіння. Для гасіння електрообладнання, що перебуває під напругою, застосовують вуглекислотні або порошкові вогнегасники, використання води та пінних вогнегасників у таких випадках заборонено. Слід дотримуватися правил експлуатації електроприладів, не залишати ввімкнене обладнання без нагляду на тривалий час, забезпечувати вільні шляхи евакуації та

знати розташування засобів пожежогасіння. Оскільки навчання моделей може тривати багато годин, потрібно передбачити контроль за станом обладнання, щоб уникнути його перегріву та займання.

Дотримання наведених вимог охорони праці, ергономічної організації робочого місця, електро- та пожежної безпеки, а також раціонального режиму праці й відпочинку забезпечує безпечні та комфортні умови для розроблення й експлуатації системи керування мобільною роботизованою платформою, зберігає здоров'я та підвищує працездатність розробника.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи розроблено та досліджено систему управління мобільною роботизованою платформою на основі методів машинного навчання, що поєднує модуль технічного зору та модуль автономної навігації. Досягнуто поставленої мети та вирішено всі сформульовані задачі:

- проведено аналіз сучасних мобільних роботизованих платформ, класичних та інтелектуальних методів керування, підходів комп'ютерного зору та навчання з підкріпленням. Обґрунтовано доцільність гібридного рішення, що поєднує детектор об'єктів YOLO (навчання з учителем) та агента навігації DQN (навчання з підкріпленням);

- обґрунтовано вибір апаратно-програмної бази: мобільної платформи TurtleBot3 Burger, симуляційного середовища Webots та бібліотек машинного навчання Python (Ultralytics, Stable-Baselines3, Gymnasium). Розроблено модульну архітектуру системи керування, що об'єднує підсистеми давачів, перцепції, прийняття рішень та виконання;

- сформовано та розмічено спеціалізований набір даних із 428 зображень і натреновано детектор пішоходів YOLOv11. Досягнуто високих метрик якості: точність 0,98, повнота 0,96, mAP@0.5 близько 0,99, що підтверджує придатність моделі для роботи у реальному часі;

- формалізовано задачу навігації як марковський процес прийняття рішень із визначенням простору станів (сім числових ознак), чотирьох дискретних дій та функції винагороди. Реалізовано RL-середовище на базі Webots і Gymnasium та натреновано DQN-агента;

- проведено експериментальні дослідження у двох сценаріях. У детермінованому середовищі агент досяг стовідсоткової успішності доїздів без зіткнень (у середньому 101 крок до цілі). У рандомізованому середовищі успішність становила 30 % за відсутності зіткнень, що вказує на потребу подальшого вдосконалення узагальнювальної здатності агента.

Отримані результати мають практичну цінність і можуть бути використані як основа для подальшого впровадження у реальні робототехнічні комплекси.

Перспективними напрямками розвитку є збільшення обсягу та різноманітності навчання, застосування сучасніших алгоритмів навчання з підкріпленням (PPO, SAC), а також перенесення навченої політики на реальну апаратну платформу за методологією sim-to-real.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Введ. 2017-07-01. Київ : ДП «УкрНДНЦ», 2016. 26 с.
2. Навчальний посібник з підготовки кваліфікаційної роботи бакалавра для здобувачів вищої освіти спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / І. Ш. Невлюдов та ін. Харків : Видавництво Іванченка І. С., 2022. 151 с.
3. Методичні вказівки з підготовки кваліфікаційної роботи бакалавра спеціальності 151 / упоряд. І. Ш. Невлюдов та ін. Харків : ХНУРЕ, 2022. 66 с.
4. Невлюдов І. Ш., Андрусевич А. О., Євсєєв В. В. та ін. Проєктування мобільних маніпуляційних роботів : монографія. Харків : ХНУРЕ, 2022. 427 с.
5. Про охорону праці : Закон України від 14.10.1992 № 2694-XII ; режим доступу (<https://zakon.rada.gov.ua/laws/show/2694-12>) дата використання [10.07.26].
6. НПАОП 0.00-7.15-18. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями : затв. наказом Міністерства соціальної політики України від 14.02.2018 № 207 ; режим доступу (<https://zakon.rada.gov.ua/laws/show/z0508-18>) дата використання [10.07.26].
7. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. ДСанПіН 3.3.2.007-98 : затв. постановою Головного державного санітарного лікаря України від 10.12.1998 № 7. Київ, 1998.
8. Комплекс навчально-методичного забезпечення навчальної дисципліни "Безпека праці в індустрії ІТ- технологій" підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету [Електронний ресурс] / ХНУРЕ; розроб.: Т. Є. Стиценко, Г. В. Пронюк, І. І. Хондак. – Харків, 2024. – 171 с.

9. Siciliano B., Khatib O. (Eds.) Springer Handbook of Robotics. 2nd ed. Cham : Springer, 2016. 2227 p. DOI: 10.1007/978-3-319-32552-1.
10. Thrun S., Burgard W., Fox D. Probabilistic Robotics. Cambridge, MA : MIT Press, 2005. 647 p. (<https://probabilistic-robotics.org/>) дата використання [30.04.2026].
11. Patle B. K. et al. A review: On path planning strategies for navigation of mobile robot. Defence Technology. 2019. Vol. 15, No. 4; режим доступу (<https://doi.org/10.1016/j.dt.2019.04.011>) дата використання [30.04.26].
12. LeCun Y., Bengio Y., Hinton G. Deep learning. Nature. 2015. Vol. 521; режим доступу (<https://doi.org/10.1038/nature14539>) дата використання [04.05.26].
13. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge, MA : MIT Press, 2016; режим доступу (<https://www.deeplearningbook.org/>) дата використання [05.05.26].
14. He K. et al. Deep Residual Learning for Image Recognition. CVPR. 2016; режим доступу (<https://arxiv.org/abs/1512.03385>) дата використання [06.05.26].
15. Redmon J. et al. You Only Look Once: Unified, Real-Time Object Detection. CVPR. 2016; режим доступу (<https://arxiv.org/abs/1506.02640>) дата використання [07.05.26].
16. Jocher G., Chaurasia A., Qiu J. Ultralytics YOLO [Electronic resource]. 2023; режим доступу (<https://github.com/ultralytics/ultralytics>) дата використання [07.05.26].
17. Ultralytics. YOLO11 Documentation [Electronic resource]; режим доступу (<https://docs.ultralytics.com/>) дата використання [08.05.26].
18. Howard A. G. et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861. 2017; режим доступу (<https://arxiv.org/abs/1704.04861>) дата використання [08.05.26].
19. Sutton R. S., Barto A. G. Reinforcement Learning: An Introduction. 2nd ed. Cambridge, MA : MIT Press, 2018; режим доступу (<http://incompleteideas.net/book/the-book-2nd.html>) дата використання [12.05.26].

20. Mnih V. et al. Human-level control through deep reinforcement learning. Nature. 2015. Vol. 518; режим доступу (<https://doi.org/10.1038/nature14236>) дата використання [12.05.26].

21. Mnih V. et al. Playing Atari with Deep Reinforcement Learning. arXiv:1312.5602. 2013; режим доступу (<https://arxiv.org/abs/1312.5602>) дата використання [13.05.26].

22. Schulman J. et al. Proximal Policy Optimization Algorithms. arXiv:1707.06347. 2017; режим доступу (<https://arxiv.org/abs/1707.06347>) дата використання [13.05.26].

23. Kober J., Bagnell J. A., Peters J. Reinforcement learning in robotics: A survey. The International Journal of Robotics Research. 2013. Vol. 32, No. 11; режим доступу (<https://doi.org/10.1177/0278364913495721>) дата використання [14.05.26].

24. Raffin A. et al. Stable-Baselines3: Reliable Reinforcement Learning Implementations. Journal of Machine Learning Research. 2021. Vol. 22, No. 268; режим доступу (<https://jmlr.org/papers/v22/20-1364.html>) дата використання [14.05.26].

25. Towers M. et al. Gymnasium: A Standard Interface for Reinforcement Learning Environments. arXiv:2407.17032. 2024; режим доступу (<https://arxiv.org/abs/2407.17032>) дата використання [15.05.26].

26. Michel O. Webots: Professional Mobile Robot Simulation. International Journal of Advanced Robotic Systems. 2004. Vol. 1, No. 1; режим доступу (<https://cyberbotics.com/>) дата використання [15.05.26].

27. Cyberbotics Ltd. Webots User Guide [Electronic resource]; режим доступу (<https://cyberbotics.com/doc/guide/index>) дата використання [15.05.26].

28. ROBOTIS. TurtleBot3 e-Manual [Electronic resource]; режим доступу (<https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/>) дата використання [16.05.26].

29. Roboflow. Roboflow: Computer Vision Platform [Electronic resource]; режим доступу (<https://roboflow.com/>) дата використання [16.05.26].

30. Cadena C. et al. Past, Present, and Future of Simultaneous Localization and Mapping. IEEE Transactions on Robotics. 2016. Vol. 32, No. 6; режим доступу (<https://doi.org/10.1109/TRO.2016.2624754>) дата використання [16.05.26].

31. Zhu Y. et al. Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning. ICRA. 2017; режим доступу (<https://arxiv.org/abs/1609.05143>) дата використання [18.05.26].

32. Tobin J. et al. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. IROS. 2017; режим доступу (<https://arxiv.org/abs/1703.06907>) дата використання [18.05.26].

33. Paszke A. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. NeurIPS. 2019; режим доступу (<https://arxiv.org/abs/1912.01703>) дата використання [18.05.26].

34. Harris C. R. et al. Array programming with NumPy. Nature. 2020. Vol. 585; режим доступу (<https://doi.org/10.1038/s41586-020-2649-2>) дата використання [18.05.26].

35. Webots_Pedestrian Dataset (Version 2) [Electronic resource] / Drones (Roboflow Universe). Ліцензія CC BY 4.0; режим доступу (https://universe.roboflow.com/drones-stduo/webots_pedestrian/dataset/2) дата використання [19.05.26].

36. Hunter J. D. Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering. 2007. Vol. 9, No. 3; режим доступу (<https://doi.org/10.1109/MCSE.2007.55>) дата використання [05.06.26].

37. The pandas development team. pandas-dev/pandas: Pandas [Electronic resource]; режим доступу (<https://pandas.pydata.org/>) дата використання [05.06.26].