

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Охорона (назва факультету)

Кафедра комп'ютерних систем та мереж

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр

(назва освітнього ступеня)

на тему: Математичне та програмне забезпечення комп'ютеризованих систем
автоматичного керування роботизованими механізмами
на основі нейронних мереж

Виконав: студент 6 курсу групи СІм-61

спеціальності

123 «Комп'ютерна інженерія»

(шифр і назва спеціальності (напряму підготовки))

Мудрий І.В.

(підпис)

(прізвище та ініціали)

Керівник

Баран І.О.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Тиш Є.В.

(підпис)

(прізвище та ініціали)

Завідувач

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

кафедри

Рецензент

Цуприк Г.Б.

(підпис)

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра комп'ютерних систем та мереж

(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« »

20__ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

за спеціальністю

123 Комп'ютерна інженерія

студенту

Мудрому Ігорю Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи

Математичне та програмне забезпечення комп'ютеризованих систем
автоматичного керування роботизованими механізмами
на основі нейронних мереж

Керівник роботи

Баран Ігор Олегович., к.т.н. доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом по університету від « 06 » грудня 2022 року № 4/7-986

2. Термін подання студентом роботи 20.12.2022

3. Вихідні дані до роботи

наукові літературні джерела

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

1 Аналіз предметної області дослідження. 2 Теоретична частина.

3. Практична частина. 4 Охорона праці та безпека в надзвичайних ситуаціях

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Тема, мета, задачі, об'єкт, предмет, новизна дослідження. 2. Актуальність дослідження.

3. Парадигма навчання із підкріпленням. 4. Фундаментальні методи вирішення

5. Визначення вимог до проєктованої системи.

6. ПЗ та технології, використані в розробці..

7. Задача переміщення об'єкта агентом у вказану зону. 8. Її модель та результати навчання.

9. Модель та результати навчання для зміщення маніпулятора. 10. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Осухівська Г.М., доцент	06.12.22	12.12.22
Безпека в НС		06.12.22	12.12.22

7. Дата видачі завдання _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Затвердження теми кваліфікаційної роботи	06.12.22	Виконано
2	Аналіз літературних джерел	07.12-08.12.22	Виконано
3	Обґрунтування актуальності дослідження	08.12-09.12.22	Виконано
4	Аналіз предмету дослідження та предметної області	10.12-11.12.22	Виконано
5	Проведення дослідження методів та засобів аналітичного опрацювання даних	11.12-12.12.22	Виконано
6	Оформлення розділу «Аналіз предметної області дослідження»	12.12-13.12.22	Виконано
7	Оформлення розділу «Теоретична частина»	13.12-14.12.22	Виконано
8	Оформлення розділу «Практична частина»	14.12-15.12.22	Виконано
9	Оформлення розділу «Охорона праці та безпека в надзвичайних ситуаціях»	07.12-12.12.22	Виконано
10	Нормоконтроль	11.12-14.12.22	
11	Попередній захист роботи	15.12.22	Виконано
12	Захист кваліфікаційної роботи	21.12.22	

Студент _____
(підпис)

Мудрий І.В. _____
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Баран І.О. _____
(прізвище та ініціали)

АНОТАЦІЯ

Математичне та програмне забезпечення комп'ютеризованих систем автоматичного керування роботизованими механізмами на основі нейронних мереж // Кваліфікаційна робота за освітнім рівнем «магістр» // Мудрий Ігор Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІм-61 // Тернопіль, 2022 // с. – 70, рис. – 33, табл. – 3 , аркушів А1 – 10 , бібліогр. – 36.

Ключові слова: НЕЙРОННІ МЕРЕЖІ, АВТОНОМНІ АПАРАТИ, МАНІПУЛЯТОР, НАВЧАННЯ З ПІДКРІПЛЕННЯМ, МОДЕЛЮВАННЯ

Кваліфікаційна робота присвячена дослідженню автоматичного керування роботизованими механізмами. Результатом проведених робіт є змодельоване середовище для навчання інтелектуальних агентів виконання захоплення та підняття об'єктів. Для апробування даної розробки в реальних пристроях необхідно спочатку застосувати будь-які методи або алгоритми по побудові реконструкцій оточення у віртуальному середовищі. Після калібрування та навчання, штучні нейромережі будуть здатні виконувати поставлені завдання у реальному світі.

Результати цієї роботи можуть бути використані для навчання реальних роботизованих пристроїв, здатних взаємодіяти з навколишнім середовищем у вигляді різних хапальних механізмів. Розроблена концепція навчання роботизованих механізмів може суттєво розширити спектр завдань, які вирішуються автономними апаратами. У перспективі, на основі отриманих знань можна проектувати складніші роботизовані системи, що виконують завдання з автоматичного сортування та транспортування об'єктів, які проводять асистування на виробництві, в лабораторіях чи побуті, та виконують нешаблонні маніпуляції з об'єктами.

ANNOTATION

Mathematical and software development of computerized systems for automatic control of robotic mechanisms based on neural networks // Master thesis // Mudryi Ihor // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Systems and Nets, group CIm - 61 // Ternopil, 2022 // p. – 70, fig. – 33 , table. – 3, Sheets A1 - 10 , Ref. - 36.

Keywords: NEURAL NETWORKS, AUTONOMOUS DEVICES, MANIPULATOR, REINFORCEMENT LEARNING, SIMULATION

The thesis deals with the study of automatic control of robotic mechanisms. The result of the work is a simulated environment for training intelligent agents to grasp and lift objects. To test this development in real devices, it is necessary to first apply any methods or algorithms for building reconstructions of the environment in a virtual environment. After calibration and training, artificial neural networks will be able to perform tasks in the real world.

The results of this work can be used to train real robotic devices capable of interacting with the environment in the form of various grasping mechanisms. The developed concept of training robotic mechanisms can significantly expand the range of tasks solved by autonomous devices. In the future, based on the knowledge gained, it is possible to design more complex robotic systems that perform the tasks of automatic sorting and transportation of objects, provide assistance in production, in laboratories or at home, and perform non-standard manipulations with objects.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ

ML-Agents – проект із відкритим вихідним кодом, який дозволяє іграм та симуляторам служити середовищем для навчання інтелектуальних агентів

PhysX – кросплатформенний фізичний рушій для симуляції ряду фізичних явищ, а також комплект засобів розробки на його основі

PnP (Perspective-n-Point) – проблема оцінки положення відкаліброваної камери за набором із n 3D-точок у наборі та їхніх відповідних 2D-проекцій на зображенні

PPO (Proximal Policy Optimization) – сімейство безмодельних алгоритмів навчання з підкріпленням

PyTorch – фреймворк машинного навчання для мови Python з відкритим вихідним кодом

Q-learning – метод, який застосовується у штучному інтелекті для навчання інтелектуальних агентів

RGB (Red Green Blue) – адитивна колірна модель

SAC (Soft Actor-Critic) – алгоритм, який оптимізує стохастичну політику

SIFT (Scale-Invariant Feature Transform) алгоритм із області комп'ютерного зору, який виявляє і описує локальні ознаки зображення

SURF (Speed Up Robust Feature) – запатентований детектор і дескриптор локальних функцій у комп'ютерному зорі

TD (Temporal Difference) методи – методи часових відмінностей

V-REP – гнучка і масштабована платформа для робомодельовання

IA – інтелектуальний агент

ПЗ – програмне забезпечення

ІІІ – штучний інтелект

ІІІМ – штучна нейронна мережа

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ	6
ВСТУП.....	9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ	12
1.1. Особливості системи керування автономним апаратом	12
1.2. Навчання алгоритмів у змодельованому середовищі.....	13
1.3. Рандомізація змодельованих даних.....	14
1.4. Локалізація та класифікація об'єктів.....	16
1.4.1. Локалізація та класифікація із застосуванням карт глибини	17
1.4.2. Локалізація та класифікація без застосуванням карт глибини.....	18
1.5. Архітектура гібридної системи управління.....	21
1.5. Висновки до розділу	24
РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА	25
2.1. Загальна інформація про штучні нейронні мережі.....	25
2.2. Принцип роботи штучних нейронних мереж.....	26
2.3. Парадигма навчання з учителем	28
2.4. Парадигма навчання без вчителя.....	30
2.5. Парадигма навчання із підкріпленням.....	31
2.5.1. Загальна інформація.....	31
2.5.2. Фундаментальні методи вирішення	34
2.5.3. Функція вартості.....	37
2.6. Вибір програмних засобів для реалізації моделювання роботи маніпулятора	38
2.7. Визначення вимог до проєктованої системи.....	40
2.8. Висновки до розділу	42
РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА.	43
3.1. Структура програми.....	43
3.1.1. Клас ArmHandAgent.....	43

3.1.2. Клас RobotController	44
3.1.3. Клас ArticulationJointController	45
3.2. Задача переміщення об'єкта агентом у вказану зону.....	45
3.2.1. Опис змодельованого завдання	45
3.2.2. Результати навчання інтелектуальних агентів	46
3.3. Модель маніпулятора.....	48
3.4. Результати навчання із захоплення маніпулятором об'єкта	51
3.4.1. Опис навчального середовища	51
3.4.2. Результати навчання	53
3.5. Результати навчання з підняття об'єкта маніпулятором	58
3.6. Висновки до розділу	60
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	61
4.1. Охорона праці	61
4.2. Комп'ютерне забезпечення процесу оцінки радіаційної та хімічної обстановки.....	64
4.3. Висновки до розділу	66
ВИСНОВКИ.....	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	68
ДОДАТОК А. Тези конференції	

ВСТУП

Актуальність теми. У сучасному світі безпілотні роботизовані механізми виконують величезний спектр завдань, які можуть бути рутинними, біологічно або фізично небезпечними, мати низьку продуктивність або занадто дорогі при використанні традиційних методів. Можливість застосування таких механізмів для автоматизації будь-яких процесів дає змогу використовувати безпілотні роботизовані апарати.

Багато завдань можуть виконуватися паралельно, тому природним кроком є розгортання кількох автономних пристроїв. Група маніпуляторів може ефективно виконувати широкий клас завдань, таких як сортування, транспортування, асистування на виробництві, у лабораторіях чи побуті, робота в небезпечних місцях. Однак для цього потрібна високоточна автономізація апаратів, а традиційні методи малоефективні в широкому діапазоні завдань. Для поліпшення ефективності у ряді таких завдань можна використовувати ШНМ [1], що набули упродовж останніх років широкого використання. Вони є інструментом, за допомогою якого можна ефективно вирішувати великий пласт завдань. Зокрема, це розпізнавання образів і мовлення, прогнозування часових рядів, переклад тексту, постановка медичного діагнозу, алгоритми поведінки в іграх і т.д. Цей перелік регулярно зростає так, як ШНМ є більш ефективними та витісняють традиційні алгоритми. Робототехніка у цьому розвитку не стає винятком.

Машинне навчання, зокрема ШНМ, незважаючи на давню появу, широкого розвитку отримали відносно недавно. Особливо воно поширилося у сфері розпізнавання (голосу, зображення), прогнозування часових рядів, та з недавніх пір й у робототехніці. Великі компанії на кшталт Google, Amazon, Microsoft, Tesla мають власні підрозділи та розробки у цій галузі. Меншим фірмам доводиться шукати готові рішення або замовляти власну розробку. Попит на ШНМ високий, оскільки вони дозволяють автоматизувати багато процесів, які автоматизувати з допомогою звичайних алгоритмів складно і дорого [1].

Мета дослідження: розробка нейромережевого способу управління роботом-маніпулятором у змодельованому середовищі.

В роботі поставлено та розв'язано **наступні задачі:**

- вивчити структуру системи керування роботизованими механізмами;
- вивчити існуючі моделі та методи навчання ШНМ;
- дослідити особливості навчання ШНМ для управління роботизованими механізмами;
- розробити програмні засоби для моделювання роботи маніпулятора на основі ШНМ.

Об'єкт дослідження: процес управління роботизованими механізмами.

Предмет дослідження: нейромережеві моделі, призначені для управління автономними роботами.

Методи дослідження: Методологічною основою дослідження є графічне моделювання фізичного середовища, дослідження методів навчання з підкріпленням, програмна реалізація обраного методу та проведення чисельних експериментів щодо навчання моделі роботизованих пристроїв.

Наукова новизна отриманих результатів:

- розроблено віртуальне середовище для симуляції роботи маніпулятора та проведення навчання ІА;
- проведено дослідження впливу різних гіперпараметрів функції вартості та використання різної кількості давачів сприйняття оточення на швидкість та якість навчання ІА;
- запропоновано використати алгоритм PPO, котрий показав значно кращі результати з визначення складних багаторівневих послідовностей, аніж SAC;
- розроблено спеціалізоване ПАЗ для навчання ШНМ.

Практичне значення одержаних результатів. Потенційними споживачами розробки можуть стати пошуково-рятувальні служби, так як група дронів, котра здатна досліджувати велику територію за мінімальний проміжок часу; космічна промисловість, оскільки управляючий сигнал залежно від відстані не може

дозволяти безпосередньо управляти роботизованим пристроєм через колосальні затримки; військова сфера застосування, так як управляючий безпілотним апаратом сигнал може глушитися зловмисником і т.д.

Публікації. Результати дослідження апробовано на X науково-технічній конференції «Інформаційні моделі, системи та технології» у вигляді опублікованих тез [9].

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається з вступу, 4 розділів, висновків, списку використаної літератури та додатків. Обсяг роботи: пояснювальна записка – 71 арк. формату А4, графічна частина – 10 аркушів формату А1.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1. Особливості системи керування автономним апаратом

Система управління автономним апаратом у загальному випадку виконує такі функції:

- сприйняття та розпізнавання вхідної інформації;
- взаємодія з оператором (людиною);
- планування та перепланування поведінки;
- керування виконанням дій;
- керування приводами;
- взаємодія з іншими роботами.

Які з перерахованих функція використовуються у тому чи іншому апараті – залежить від самого апарата. Використання ШНМ додає ще одну важливу функцію - здатність та необхідність її навчати.

Для використання ШНМ з метою управління автономним апаратом необхідно вирішити, як мінімум, дві природні проблеми:

- мінімізація вхідних даних
- збір навчальної вибірки.

Першою проблемою є величезний потік даних. Використання автономних механізмів завжди має на увазі, що апарат буде взаємодіяти з навколишнім світом. Найкращим способом сприйняття є візуальний, але обробка візуальної інформації за допомогою ШНМ є вкрай ресурсомістким завданням.

Другою проблемою є збір навчальної вибірки. Але для складних чи нових систем немає заздалегідь зібраної інформації, а її збір є дуже трудомістким процесом.

1.2. Навчання алгоритмів у змодельованому середовищі

Збір навчальних даних для обробки інформації з давачів сприйняття оточення або для навчання поведінкових алгоритмів у змодельованому реалістичному середовищі може значно збільшити вплив методів машинного навчання на робототехніку, забезпечуючи більш швидкий, масштабований та недорогий спосіб збору даних, ніж це можливо в реальному середовищі.

Особливо багатообіцяюче виглядають методи глибокого навчання з підкріпленням, останні результати яких дозволяють досягти рівня продуктивності, порівнянного з можливостями людини, таких завдань як Atari [2] і управління роботами [3]. При цьому потрібно близько сотні тисяч або мільйона навчальних прикладів [2], збирання яких може зайняти тисячі годин, що робить його непрактичним для багатьох пристроїв і завдань. При цьому глибоке навчання з підкріпленням у процесі збору навчальних прикладів виконує випадкові дії, які можуть вивести з ладу фізичне обладнання. Тому можливість вивчати лінії поведінки, повністю змодельовані та розроблені у віртуальних середовищах, дає перевагу у швидкості та безпеці навчання роботизованих механізмів.

На жаль, розбіжності між симуляторами фізики та реальним світом ускладнюють перенесення моделей поведінки із симуляції. Процес налаштування параметрів моделювання відповідно до поведінки фізичної системи займає багато часу і схильний до помилок. Навіть у добре опрацьованій модельованій системі відсутні деякі фізичні ефекти, такі як гнучкість, люфт, зношування, які не вловлюються сучасними симуляторами фізики. До того ж, імітуючі давачі з низькою точністю, такі як камери, часто не здатні відтворити реалістичність і шуми, присутні на реальних пристроях. Ці відмінності, відомі під загальною назвою розрив реальності (reality gap), створюють перешкоду використання змодельованих даних на реальних роботах.

Один із підходів подолання розбіжностей між змодельованими даними та реальними полягає в тому, щоб симулятор якнайбільше відповідав фізичній

реальності з використанням високоякісного рендерингу. Використання лише реалістичного рендерингу RGB було досліджено в [4] і показало хороші результати. Використання реалістичних даних про глибину з 3D -моделей шляхом всебічного моделювання таких факторів, як шум сенсорів, коефіцієнт відображення матеріалів та геометрію поверхонь дозволяло моделям, навченим на візуалізованих зображеннях, досить добре переноситися у реальний світ [5].

1.3. Рандомізація змодельованих даних

Інший перспективний підхід до подолання розбіжностей між змодельованими і реальними даними полягає в тому, що замість збору даних для навчання в одному середовищі, що моделюється, ми додаємо елемент випадковості в симулятор з метою представити моделі в широкому діапазоні середовищ під час навчання. Підсумком даної процедури при достатній мінливості в симуляції стане модель, яка ефективно отримує найважливіші ознаки як у симуляції, так і на реальному устаткуванні без додаткового навчання.

Мета внесення випадковості до елементів симуляції – забезпечити достатню змодельовану мінливість під час навчання, щоб під час тестування модель могла бути узагальнена для реальних даних. Приклад даної процедури представлений на рис. 1.1. Зліва представлені сцени, змодельовані за різних параметрах, основі яких навчаються ШНМ. Справа продемонстровано реальний макет для тестування навчених мереж [6].

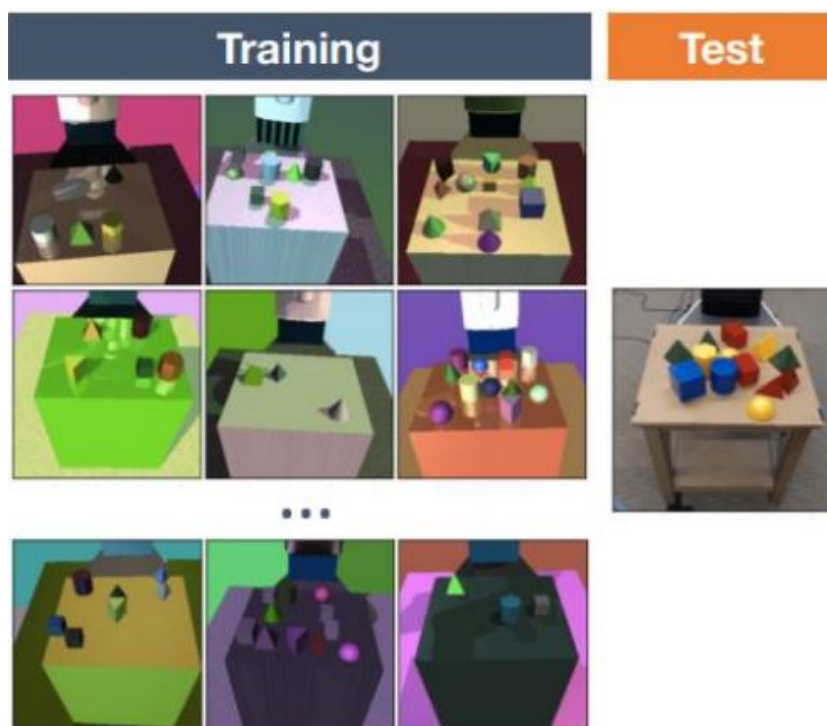


Рис. 1.1. Ілюстрація підходу до рандомізації симуляції [6]

У процесі рандомізації можуть використовуватися такі аспекти предметної області:

- форма об'єктів і їх розмір;
- число відволікаючих об'єктів;
- положення об'єктів;
- текстури об'єктів;
- текстури землі, стін, фону;
- положення, орієнтація та поле зору камери;
- число джерел світла;
- положення, орієнтація та характеристики джерел світла;
- тип та кількість випадкового шуму, доданого до дачивів.

Підсумком даної процедури при достатній мінливості в симуляції стане модель, навчена в симуляторі, котра набуде узагальненого досвіду для роботи в реальному світі без додаткового навчання.

Рандомізація предметної області - перспективний напрямок досліджень, спрямований на подолання розриву між реальністю та розробку алгоритмів поведінки роботів, що вивчаються в процесі моделювання. Глибоке навчання з підкріпленням може дозволити вивчити складніші алгоритми поведінки у моделюванні за допомогою великомасштабного дослідження та оптимізації, а рандомізація предметної галузі може стати важливим інструментом для використання таких політик на реальних роботах.

1.4. Локалізація та класифікація об'єктів

Процес розпізнавання предмету у неструктурованих сценах – складна сфера поточних досліджень комп'ютерного зору. Робототехніка є одним із важливих напрямків, де здатність чітко і швидко ідентифікувати об'єкти, що цікавлять, має вирішальне значення для універсальних роботів, що виконують завдання в неструктурованому повсякденному середовищі, такій як будинок, офіс або склад.

Специфічна проблема сприйняття для робототехніки має низку унікальних особливостей, що відрізняють її від інших проблем комп'ютерного зору. Традиційна система розпізнавання об'єктів завжди опрацьовує значне число різних об'єктів. Один із способів упоратися з цим – запровадити ієрархію, виконуючи розпізнавання на рівні категорії, а не на рівні примірника. Завдання узагальнення від кількох екземплярів до цілої категорії об'єктів залишається складним, і існують численні тести та завдання, такі як Caltech 256 [7] та PASCAL VOC [8], щоб сприяти прогресу у цій галузі. З іншого боку, для конкретного робота в конкретному середовищі кількість унікальних об'єктів відносно невелика (можливо, близько сотень). Це дозволяє розглядати завдання як проблему розпізнавання екземплярів, збираючи великий обсяг навчальних даних кожного об'єкта. Крім того, робот може використовувати дані, одержані від різних сенсорних пристроїв, таких як камери та датчики глибини [9].

Локалізація об'єктів у робототехніці також є додатковими проблемами, яких немає в тестах розпізнавання об'єктів на рівні категорій. Середовищ реального світу дуже зашарашені, містять багато перекриттів і часто містять більшу кількість різних об'єктів в одній сцені. Роботам часто доводиться взаємодіяти з одними об'єктами у своєму середовищі, оминаючи інші. Це означає, що роботизована система сприйняття повинна точно локалізувати об'єкти після виявлення. Крім того, щоб робот міг швидко реагувати на зміни у навколишньому середовищі, роботизована система сприйняття має працювати у режимі реального часу [9].

1.4.1. Локалізація та класифікація із застосуванням карт глибини

В даний час існує ряд різних підходів до розпізнавання та локалізації предметів у складних неструктурованих сценах. Першим є використання сенсорів для збирання кольорових та глибинних зображень. Прикладами даних сенсорів є різні лідари, або більш прості давачі типу Kinect [10]. У його складі є інфрачервоний проектор спільно з монохромною КМОП-матрицею, що дає змогу одержувати тривимірне зображення за довільного природного світла (рис. 1.2).

Під час навчання розпізнаванню та локалізації об'єктів на основі інформації про глибину вдається точніше виділити контури кожного об'єкта. На основі контурів, кольору та глибини будується повна хмара точок, з якої виділяються тривимірні сіткові моделі об'єктів. Потім з отриманих моделей витягуються і реєструються локальні особливості об'єктів [11].

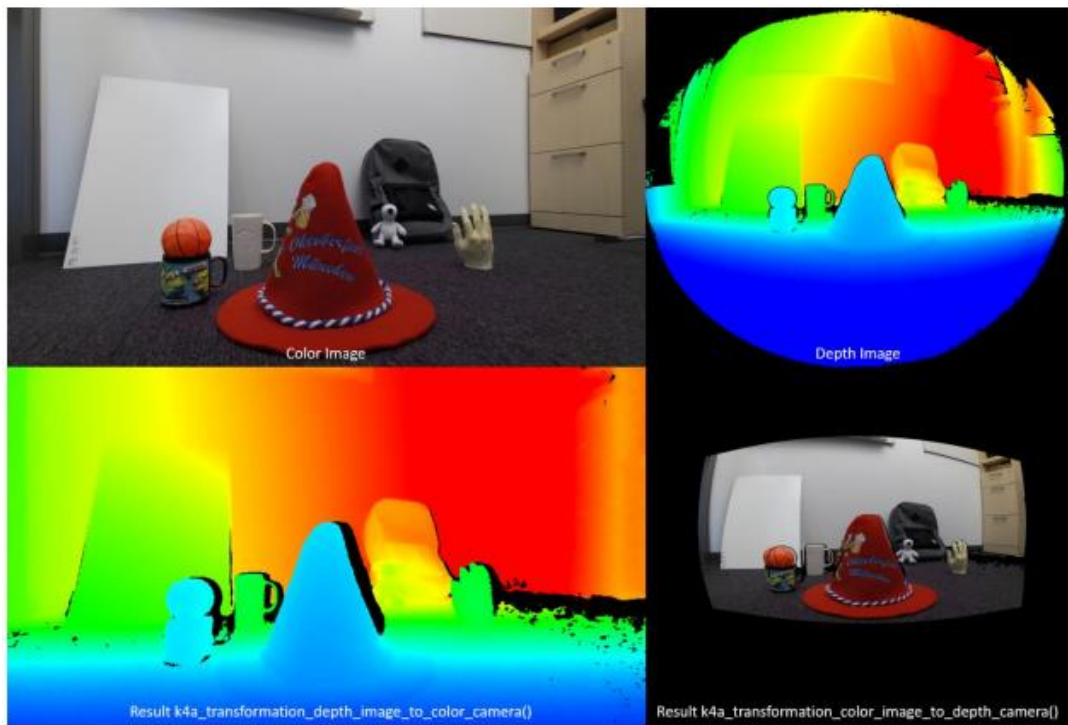


Рис. 1.2. Отримані дані з пристрою Kinect

Під час тестування з огляду на кольорове зображення та хмару точок сцена сегментується за окремими виявленими об'єктами. Для кожного сегментованого об'єкта відбувається класифікація на основі одержаних глобальних та локальних ознак, отриманих на етапі навчання. На основі інформації про глибину підбирається підтримуюча площина. Тоді відбуваються різні перевірки узгодженості сцени з виявленими об'єктами з метою унеможливлення помилок класифікації, виключаються гіпотези з геометрично несумісними об'єктами, відбувається уточнення сегментації об'єктів. Цей процес може повторюватися кілька разів, поки сегменти сцени не припинять перебудовуватися та калібруватися [11]. Даний алгоритм переміг у конкурсі з розпізнавання екземплярів Solutions in Perception Challenge, який проводився Willow Garage [12].

1.4.2. Локалізація та класифікація без застосування карт глибини

Завдання оцінки позиціонування моделі об'єкта за одним зображенням є

добре вивченою проблемою у літературі. У разі точкових функцій вона відома як PnP [13]. Припускаючи, що відомо досить точних відповідностей між функціями 2D-зображення і характеристиками 3D - моделі, потрібно використовувати обраний алгоритм рішення PnP, щоб отримати оцінку позиціонування об'єкта.

Завдання розпізнавання одного об'єкта та визначення його позиції по одному зображенню вимагає вирішення двох підзадач:

- знаходження відповідностей між характеристиками зображення та характеристиками моделі;
- оцінка пози моделі, яка найкраще узгоджується із цим набором відповідностей.

Але навіть за наявності добре помітних інваріантних ознак, таких як SIFT[14] або SURF[15], неспівпадаючі відповідності неминучі, що змушує застосовувати надійніші методи оцінки для практично усіх модернових систем розпізнавання предметів. Вичерпний огляд методів розпізнавання та відстеження тривимірних об'єктів на основі моделей доступний у [16].

Проблема розпізнавання 3D-об'єктів з урахуванням моделей з'явилася досить давно. За цією тематикою представлена широка література, але вона далека від вирішення її найбільш загальної форми. Існують дві проблеми, які сильно впливають на продуктивність будь-якого алгоритму розпізнавання.

Перша проблема – це складність сцени. Ця проблема виникає через кількість ознак, причому обидва крайні значення значно знижують швидкість розпізнавання. З цим пов'язана проблема повторюваних об'єктів: неоднозначність зіставлення, що вноситься екземплярами об'єкта, що повторюються, є величезною проблемою для надійних алгоритмів визначення поз об'єктів, оскільки зіставлені ознаки можуть належати різним екземплярам об'єкта, незважаючи на їх правильність. Часто використовуються такі моделі, як дерева рішень або кластеризація простору зображень, але помилкові спрацьовування часто виникають через те, що алгоритми не можуть упоратися з несподіваною складністю сцени.

Друга проблема пов'язана з масштабованістю та затримкою системи. У

системах, що працюють онлайн, необхідно досягти компромісу між точністю та затримкою, залежно від вимог для кожного конкретного завдання. У робототехніці час реакції роботів, що працюють у динамічному середовищі, часто обмежується затримкою їхнього сприйняття. Збільшення обсягу вхідних даних для обробки, збільшення роздільної здатності камери, або використання декількох камер, зазвичай призводить до серйозних штрафів з точки зору часу обробки.

Загалом алгоритм для класифікації об'єктів та обчислення точного їх розташування щодо камери ділиться на кілька етапів. Першим етапом є навчання, на якому створюються розріджені тривимірні моделі об'єктів. Для цього робиться набір зображень з об'єктом у різних позах. Далі відбувається вилучення надійних ознак, наприклад SIFT[14] або SURF[15], які утворюють розріджену тривимірну модель об'єкта. Далі відбувається етап калібрування: правильне вирівнювання та масштаб для кожної моделі обчислюються для відповідності реальним розмірам об'єкта та визначають відповідну систему координат.

Слід зазначити, що цей підхід сильно залежить від хорошого калібрування моделей, особливо з погляду масштабування, так як глибина оцінюється лише на основі масштабу кожної моделі. Тому під час створення моделей необхідно виявляти особливу обережність, щоб встановити належний масштаб. Наприклад, помилка моделювання 1 мм для об'єкта 10 см створює похибку визначення глибини до 3 см на відстані 1 м, що є досить великою, щоб викликати проблеми для робота-маніпулятора [17].

Розпізнавання об'єктів за локальними ознаками на зображеннях відбувається у кілька етапів. Першим є знаходження відповідностей. Цей етап відноситься до точного зіставлення характеристик зображення з характеристиками, що належать певному предмету. Далі йде процес оцінки пози об'єкта, яка буде геометрично узгодженою зі знайденими відповідностями. Після цього відповідно до масштабу предмета у базі даних і зображення знаходиться точна відстань до об'єкта.

Важливо, що вибір камери та об'єктива може сильно вплинути на точність оцінки пози. Тим не менш, камери стають якіснішими та дешевшими, а їх габарити

зменшуються. Це призводить до того, що стають легко доступними паралельне використання декількох камер для отримання виду сцени з різних ракурсів та високою роздільною здатністю. Цей підхід ефективний, тому що він забезпечує оцінку глибини, стійкість до оклюзій прямої видимості та зростання ефективного кола огляду.

Крім того, більш висока роздільна здатність може потенційно покращити розпізнавання складних предметів та точність алгоритмів оцінки пози, але часто з великими штрафними витратами. Додаткова роздільна здатність часто призводить до збільшення кількості помилкових спрацьовувань, а також до серйозного погіршення умов, затримки та зниження пропускної спроможності.

Крім всього вищесказаного, використання декількох камер дозволяє проводити реконструкцію відбитих сцен у віртуальному середовищі (рис. 1.3). Це може спростити процес навчання ІА по взаємодії в навколишніми об'єктами.



Рис. 1.3. Локалізація та класифікація об'єктів із зображення (а - в) ;
реконструкція даних сцен у віртуальному середовищі (г)

1.5. Архітектура гібридної системи управління

З метою зменшення обсягу вхідних даних на систему, яка відповідає за

виконання будь-яких дій, можна використовувати гібридний підхід до реалізації систем ШІ [18]. Даний метод заснований на розподілі загальної системи управління на логічну та асоціативну частини (рис 1.4).

У цьому представленні асоціативний рівень обробляє інформацію, отриману від довкілля. Під інформацією маються на увазі будь-які дані: зображення, давачі наближення, рівень кисню, температура, теплові давачі, інформація, отримана з інших апаратів, команди від оператора тощо. Вся отримана інформація обробляється для створення асоціацій різних об'єктів та подальшої їх класифікації, кластеризації та узагальнення. Набір інформації залежить від цілей, які має виконувати апарат. Обробка інформації може повністю або частково займатися ШНМ.

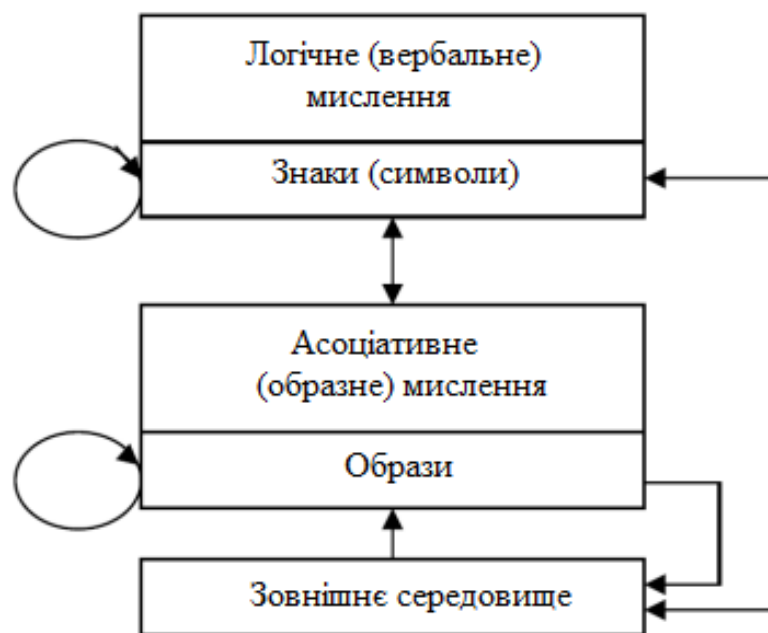


Рис. 1.4. Два рівні інтелектуальної системи

Логічний рівень обробляє інформацію, отриману від асоціативного рівня, і на її основі приймає будь-які дії у зовнішньому середовищі. На цьому рівні також можна використовувати як класичні методи інженерії, так і ШНМ.

Даний підхід дозволяє розділити завдання управління автономним апаратом

на групу підзадач. Це особливо актуально при використанні ШНМ, оскільки навчити групу невеликих нейронних мереж, кожна з яких виконуватиме своє підзавдання набагато простіше в плані збору навчальної вибірки.

Найпоширеніший приклад використання гібридної системи управління - це застосування лідарів та камер для детекції об'єктів та побудови віртуального середовища на асоціативному рівні обробки інформації, та будь-які інші алгоритми на логічному (рис. 1.5). Даний підхід сприяє не тільки поділу завдання керування автомобілем, а й дозволяє тестувати, а якщо йдеться про ШНМ, то й навчати алгоритми поведінки на моделях. Моделі, у свою чергу, повинні лише імітувати інформацію, що генерує асоціативний рівень, а не відтворювати реалістичне навколишнє середовище.



Рис. 1.5. Приклад асоціативного перетворення інформації

У цій роботі розглядатимуться можливості використання ШНМ для обробки інформації на логічному рівні. Інформація для навчання збиратиметься із застосуванням моделі, що імітує схожі процеси.

1.6. Висновки до розділу

Розглянуто особливості предметної області – специфіку функціонування систем управління автономними апаратами, способи сприйняття довкілля, методи навчання поведінковим алгоритмам із застосуванням рендерингу, рандомізації та локалізації і класифікації об'єктів.

Також описано специфіку структури гібридної системи управління.

РОЗДІЛ 2

ТЕОРЕТИЧНА ЧАСТИНА

2.1. Загальна інформація про штучні нейронні мережі

ШНМ, чия матмодель, та навіть її апаратне або програмне відображення, створена як аналог структури та роботи біологічних нейромереж, котрі існують в нервових клітинах будь-якого живого організму. Елементарною одиницею ШНС є штучний нейрон [19].

Вхідною інформацією для штучного нейрона (рис. 2.1) є добутки ознак (X_i) на відповідні їм ваги (W_i). Отримані результати складаються суматором і передаються як аргумент на функцію активації f . Значення цієї функції є вихід нейрона (Y).

У модель, показану на рис. 2.1, включений пороговий елемент, який має символічне позначення b_k і показує приріст чи зменшення вхідного сигналу, котрий поступає на функцію активації [19].

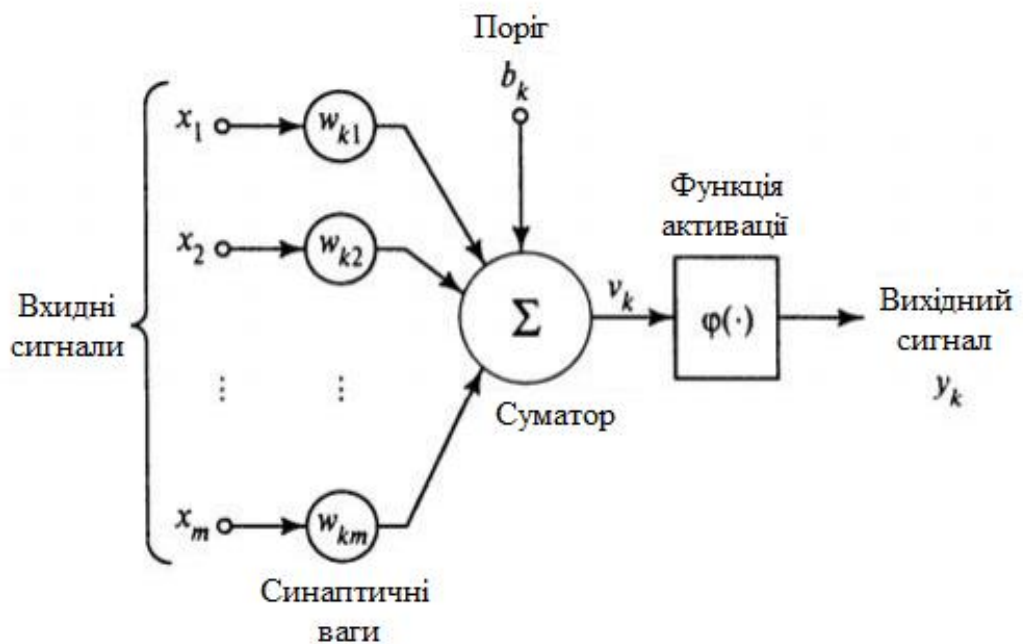


Рис. 2.1. Нелінійна модель нейрона

Існує безліч архітектур ШНМ: одношарові і багатошарові мережі прямого поширення, рекурентні, рекурсивні, мережі довгої короткострокової пам'яті і т.д. У цій роботі розглядатимуться мережі прямого поширення.

Вони характеризуються тим, що сигнал поширюється суворо від вхідного шару до вихідного. Ця архітектура складається з 3 типів шарів: вхідних (input), прихованих (hidden) та вихідних (output) (рис. 2.2.). Коректна робота ШНМ передбачає, що при отриманні будь-яких ознак на вхід, на виході буде отримано результат. Відповідно ШНМ повинна мати два і більше шарів. Як правило, кількість прихованих шарів залежить від складності завдання, і кожен прихований шар здійснює перехід з одного простору ознак до іншого.

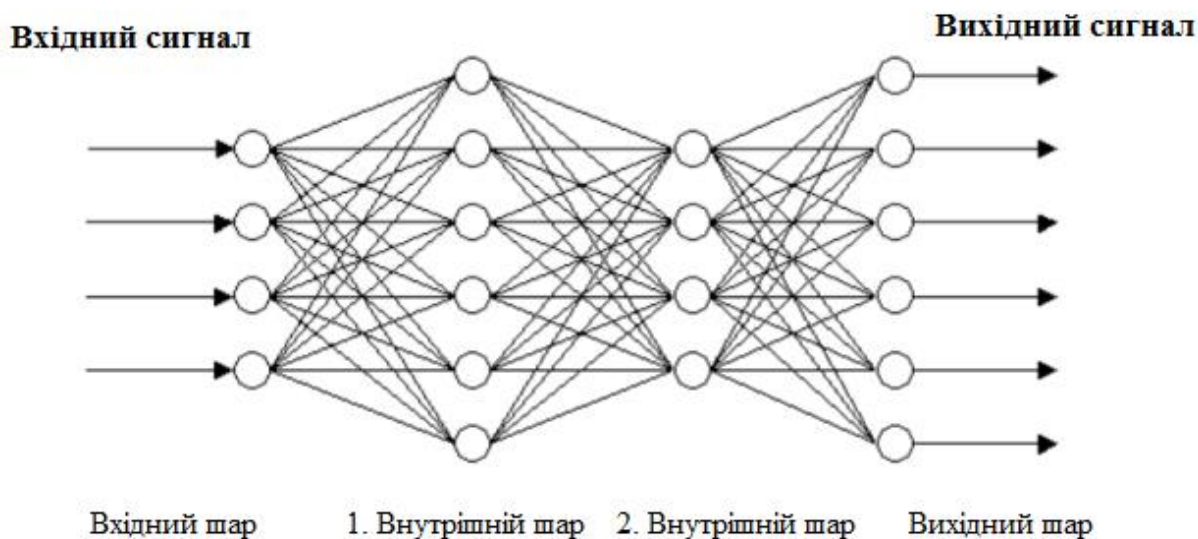


Рис. 2.2. Базова структура ШНМ прямого поширення

2.2. Принцип роботи штучних нейронних мереж

Як було сказано вище, ШНМ складається з трьох типів шарів та може мати різну кількість входів, виходів та прихованих шарів. Архітектура мережі залежить від розв'язуваного завдання.

При обробці інформації ШНМ дані подаються на вхідний шар, і з нього йдуть

на перший прихований шар і при цьому множаться на вагу зв'язку. На першому прихованому шарі кожен нейрон підсумовує твори ознак і ваг, що прийшли на нього. Потім дана сума подається як аргумент у функцію активації, і значення функції і буде вихідним значенням нейрона.

Існує безліч різних функцій активації, що мають різні області значень, і краще схильних до різноманітних завдань. Наведу кілька прикладів: на рис. 2.3 зображено сигмоїду. Її областю вихідних значень є від 0 до 1.

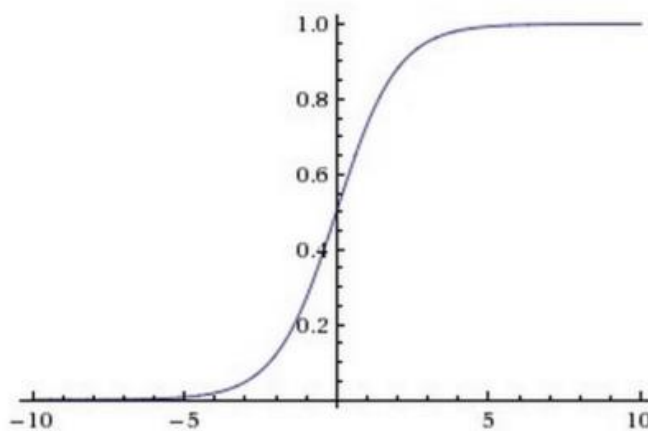


Рис. 2.3. Графік значень сигмоїди

На рис. 2.4 зображено гіперболічний тангенс із областю вихідних значень від -1 до 1.

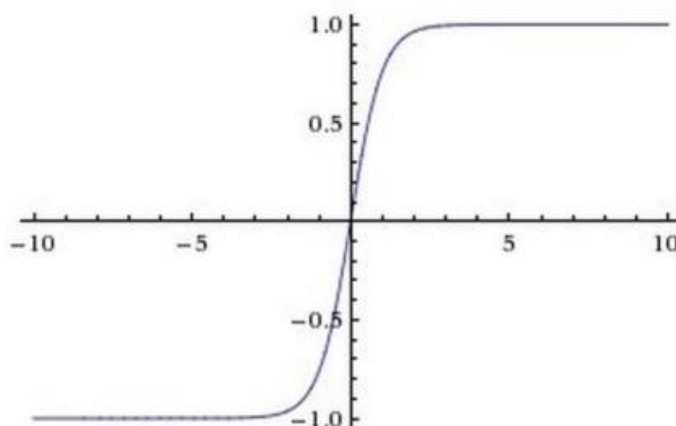


Рис. 2.4. Графік значень гіперболічного тангенсу

Видача нейроном вихідного значення називається активацією нейрона. Після того, як всі нейрони першого прихованого шару будуть активовані, вони також передадуть ознаки, але вже нові, сформовані ними, наступному шару, і так аж до вихідного.

Для коректної роботи ШНМ, її потрібно навчати. Навчання буває з учителем, без учителя та з підкріпленням.

2.3. Парадигма навчання з учителем

В зарубіжній літературі носить назву Supervised learning- один із розділів машинного навчання, присвячений вирішенню задачі знаходження залежностей між вхідним та вихідним векторами даних. Алгоритм передбачає, що з навчання відомі і вхідні, і вихідні вектори. Разом вони називаються представницькою або навчальною вибіркою, і на їхній основі мережа навчатиметься, узагальнюючи отримувану інформацію в процесі коригування синоптичних ваг [19].

Блокова діаграма, що ілюструє цю форму навчання представлена на рис. 2.5.

Поняття вчителя можна трактувати як наявність знання про довкілля. При цьому саме середовище невідоме ШНМ, що навчається. У представленій діаграмі, мережі, що навчається, і вчителю надходить інформація з-зовні. Базуючись на вбудованих знаннях вчитель може сформулювати і надіслати ШНМ, що навчається, необхідний відгук, відповідний даному вхідному вектору. Цей бажаний результат є оптимальними діями, які має зробити ШНМ. Параметри ШНМ змінюються під дією навчального вектора та сигналу помилки (error signal). Він є різницею між бажаним сигналом та поточним відгуком ШНМ. Властиво коригування параметрів проходить крок за кроком, щоб зімітувати ШНМ поведінки вчителя. Ця емуляція у певному сенсі має бути оптимальною. Отже, у процесі такого навчання вчитель передає свої максимально можливі знання до мережі. Закінчивши навчання вчитель може бути відключений, а ШНМ може працювати в подальшому із середовищем сама [19].

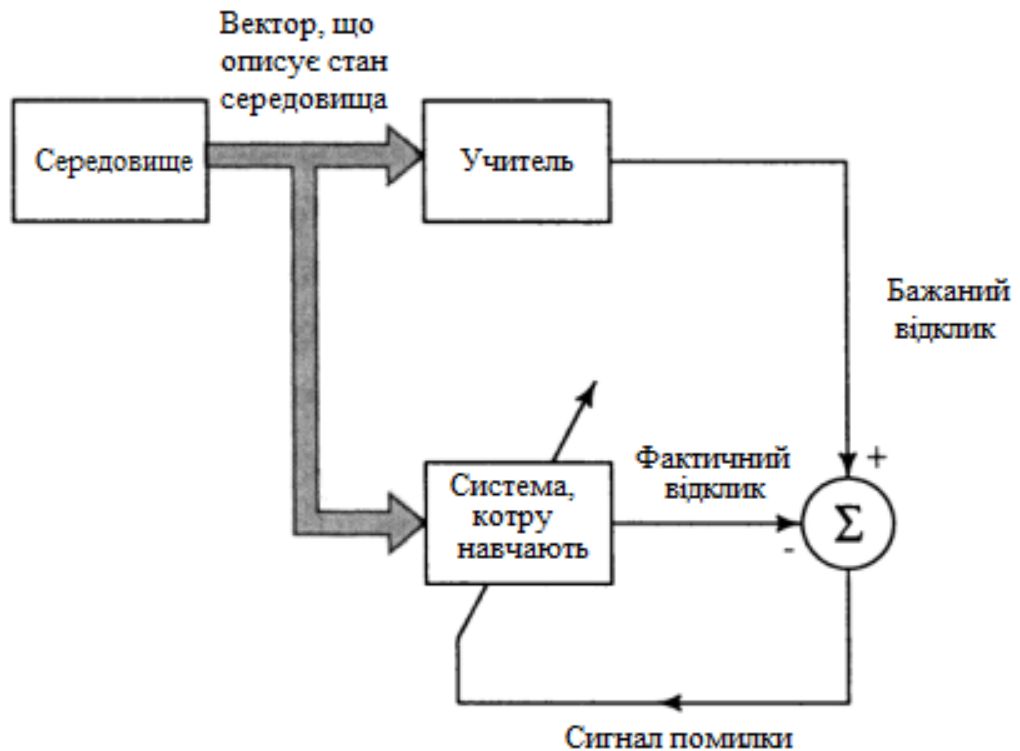


Рис. 2.5. Блокова діаграма навчання з учителем

Наведене вище навчання з учителем є формою навчанням на базі виправлення помилок. Це замкнута система зі зворотним зв'язком, яка не включає навколишнє середовище. Продуктивність власне такої системи можливо визначити у термінах середньоквадратичної помилки чи властиво суми квадратів помилок на власне навчальній вибірці, котра показана у виді функції від вільних параметрів системи. Для цієї функції ймовірно сформувати багатовимірну поверхню помилки (error surface) у координатах вільних параметрів.

При цьому реальна поверхня помилки усереднюється (averaged) за всіма можливими прикладами, представленими як комбінації «вхід-вихід». Будь-яка конкретна дія системи з учителем буде точкою на поверхні помилок. Для збільшення продуктивності системи в часі величина помилки має зміщуватися у бік найменшої величини поверхні помилок. Цей мінімум може бути як локальним, так і глобальним. Це можна зробити, якщо система має корисну інформацію про градієнт поверхні помилок, що відповідає поточній поведінці системи. Згаданий

градієнт у будь-якій точці є вектором, котрий визначає напрямок якнайшвидшого спуску по поверхні. Для такого навчання на прикладах обчислюється моментальна оцінка (instantaneous estimate) вектора градієнта, в котрій вхідний вектор буде функцією часу. При використанні результатів оцінки рух точки поверхнею помилок зазвичай має вигляд «випадкового блукання». Проте з використанням відповідного алгоритму мінімізації функції вартості, адекватному наборі навчальних прикладів у вигляді «вхід-вихід» і достатній час навчання системи здатні вирішувати такі завдання, як класифікація образів і апроксимація функцій [19].

2.4. Парадигма навчання без вчителя

Технологія Learning without a teacher самою назвою підкреслює відсутність керівника, котрий контролював би сам процес налаштування власне ваг. З використанням такого підходу немає маркованих прикладів, котрими проводиться навчання мережі [19].

Навчання без вчителя (unsupervised) (або навчання на основі самоорганізації (self-organized)) здійснюється без дії зовнішнього вчителя або коректора, котрий контролює властиво процес навчання (рис. 2.6).

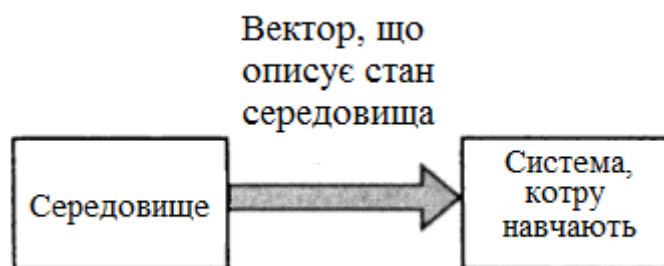


Рис. 2.6. Блокова діаграма навчання без вчителя

Можлива лише незалежна від завдання міра якості представлення, котрому повинна навчитися мережі, і вільні параметри ШНМ оптимізуються стосовно цієї міри. Після навчання мережі на статистичні закономірності вхідного сигналу ШНМ може формувати своє представлення ознак вхідних даних, що кодуються, і, саме так, формувати нові класи автоматично [19].

Таке навчання успішно застосовується для кластеризації даних, пошуку правил асоціації, скорочення розмірності даних та візуалізації даних.

2.5. Парадигма навчання із підкріпленням

2.5.1. Загальна інформація

З розвитком сучасних технологій людство ставить собі на службу системи з дедалі більшою складністю. При розробці таких систем виникає проблема оптимального контролю, яку часто неможливо вирішити реалізацією якогось певного алгоритму. Для вирішення подібних завдань на стику областей машинного навчання та оптимального управління була створена область навчання з підкріпленням, що фокусується на навчанні поведінки агента у певних оточеннях для досягнення певного результату та отримання за це винагороди (рис. 2.7).

Завдання навчання з підкріпленням передбачає наявність агента, який отримує певну інформацію про стан оточення і здатного вчиняти дії, що впливають на оточення. Мета навчання з підкріпленням полягає у навчанні агента способу отримання максимального значення числової нагороди, яка видається агенту під час здійснення певних дій.



Рис. 2.7. Схема навчання з підкріпленням

В основі навчання покладена гіпотеза про винагороду – властиво усі цілі можуть описуватися через максимізацію очікуваної (передбаченої) сумарної винагороди. Ось чому, щоб мати кращу поведінку G_t , треба максимізувати передбачувану сумарну винагороду R_t :

$$G_t = \sum_{k=0}^T R_{t+k+1} \quad (2.1)$$

У роботі агентів виділяють важливу проблему вибору між експлуатацією поточних знань агента для отримання найбільшої нагороди та дослідженням оточення з метою отримання знань про структуру оточення. Ці два варіанти вибору дій мають на увазі під собою ту саму підсумкову мету, один варіант максимізує отриману нагороду в найближчому майбутньому, а інший - у довгостроковому періоді (рис. 2.8).

Величини, які присутні на рисунку: s_t – стан; r_t – винагорода; a_t – дія.

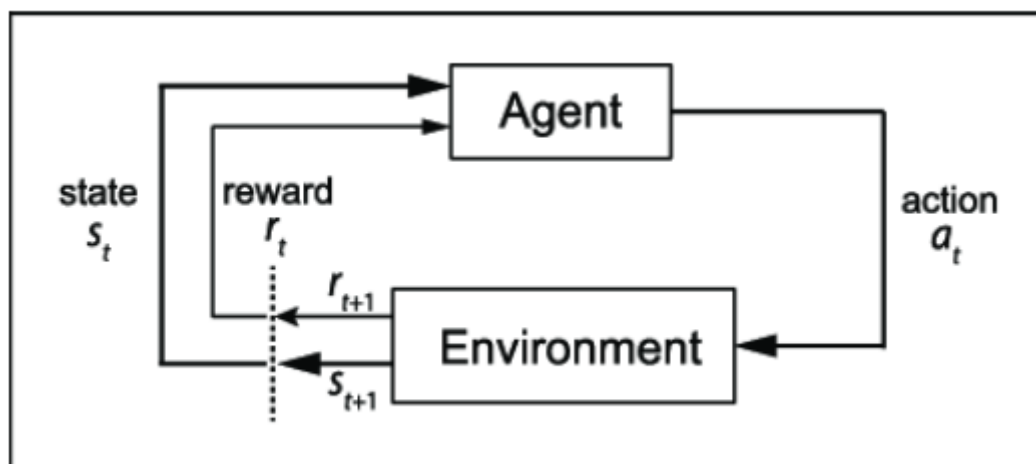


Рис. 2.8. Взаємодія середовища з агентом

Простори станів оточення можна розділити на великі групи: дискретні простору станів, у яких кожна змінна оточення є кінцевої дискретної величиною, і безперервні простору станів, у яких хоча одна змінна оточення є безперервною величиною. Потужність безлічі всіх можливих станів є кінцевою для дискретних просторів станів і нескінченною для безперервних просторів станів.

Існують стратегії дослідження оточення, які гарантовано призводять агента до оптимальної стратегії дій на просторах з невеликою потужністю безлічі всіх можливих станів. На задачах з великою або нескінченною потужністю безлічі всіх можливих станів такі методи стають непридатними через великі обчислювальні витрати. У сучасних системах навчання з підкріпленням застосовуються прості стратегії дослідження оточення, які не завжди дозволяють знайти оптимальну стратегію або знаходять її надто довго. Тому виникає потреба у розробці ефективніших алгоритмів дослідження оточень.

Задача, при якій агенту потрібно знайти стратегію π^* , котра максимізує очікувану винагороду, описується формулою (2.1):

$$\pi^* = \operatorname{argmax}_{\pi} \mathbb{E}_{\tau \sim p_{\theta}(\tau)} R(s_t, a_t) \quad (2.2)$$

Тут R – функція підкріплення, котра визначає кількість нагороди, що отримує агент, здійснюючи дію a_t , попри те, що знаходиться в стані s_t .

Одна із труднощів у вивченні стратегій дослідження оточення полягає в тому, що не існує прийнятого визначення для опису оптимальної дії з погляду дослідження оптимальної структури оточення. Оптимальна з експлуатаційної точки зору дія визначається як дія, що максимізує очікувану від цієї дії нагороду. Додаткові труднощі у формалізації поняття дослідження оточення вносить той факт, що багато сучасних систем навчання з підкріпленням не зберігають інформацію про вплив дій агента на оточення.

Приклад ШНМ як апроксиматора стратегій наведено на рис. 2.9. Тут o_t – вхідні спостереження.

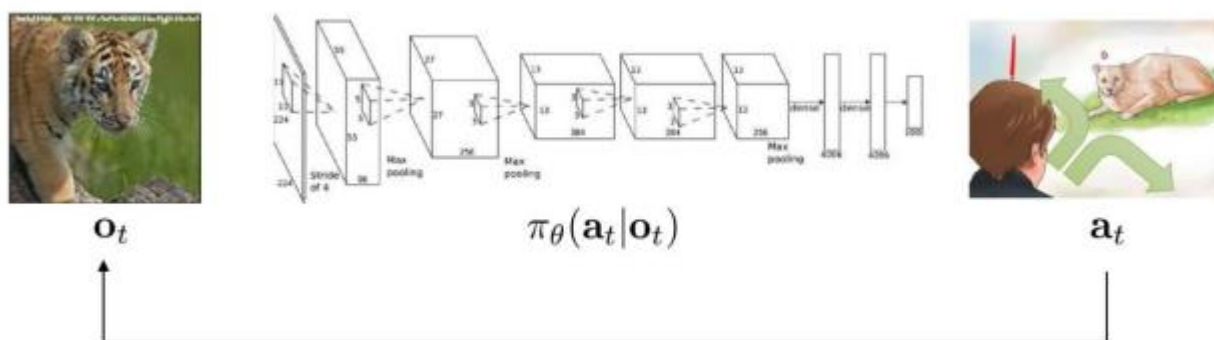


Рис. 2.9. ШНМ як апроксиматор стратегій

2.5.2. Фундаментальні методи вирішення

Існують різні підходи до навчання ШНМ методом «Навчання із підкріпленням». Нижче будуть описані основні з них.

1. Методи Монте-Карло. Не передбачають повного знання про навколишнє середовище, але вимагають наявності дослідних даних, під яким розуміється набір послідовностей станів, дій та нагород, отриманих в результаті онлайн або змодельованої взаємодії з навколишнім середовищем. Хоча для змодельованих дослідних даних і потрібна модель, даній моделі потрібно лише генерувати вибірки

переміщень, а не повний розподіл усіх можливих переміщень, як вимагає динамічне програмування (ДП) [20].

Методи Монте-Карло вирішують завдання навчання з підкріпленням, ґрунтуючись на середній цінності вибірки. Щоб переконатися, що цінність буде визначеною, методи Монте-Карло визначаються лише для епізодичних завдань. Це передбачає, що дослідні дані поділені на епізоди, і всі епізоди в результаті закінчуються, незалежно від цього, які дії обрані. Тільки після завершення епізоду відбувається оцінювання значень і стратегії змінюються.

Важливий факт про методи Монте-Карло у тому, що оцінки для кожному стану є незалежними, на відміну від методів ДП. Особливо слід зауважити, що обчислення оцінки окремого стану не залежить від кількості станів. Це робить методи Монте-Карло особливо привабливими у випадках, коли потрібно оцінити лише деяке підмножина станів.

Якщо модель недоступна, тоді корисно отримати оцінки дій, а не станів. За наявності моделі цілком достатньо оцінки станів, щоб визначити стратегію. Для цього необхідно переглянути на один крок уперед і вибрати, яка дія призводить до найкращої комбінації винагороди та наступного стану. За відсутності моделі, однак, одних оцінок стану недостатньо. Необхідно явно оцінити значення кожної дії для того, щоб у подальшому отриманні оцінки використовуватиме побудови стратегії.

Завдання оцінки стратегії полягає у обчисленні функції $Q_{\pi}(s,a)$ - очікуваної нагороди на початку в стані s , виконанні дії a , і слідує стратегії π . Це завдання є основним завданням навчання з підкріпленням.

Загальна ідея формування управління полягає у діях за схемою, яка спирається на поняття узагальненої ітерації зі стратегій (УІС). У концепції УІС використовується як наближена стратегія, і наближена функція цінності. Коригування функції цінності відбувається таким чином, щоб найкраще відповідати поточній стратегії, а стратегія постійно уточнюється залежно від поточної функції цінності (рис. 2.10).

$$V(S_t) \leftarrow V(S_t) + \alpha [G_t - V(S_t)]$$

Maximum expected future reward starting at that state
Former estimation of maximum expected future reward starting at that state
learning rate
Discounted cumulative rewards

Рис. 2.10. Метод Монте-Карло

Як правило, у методах Монте-Карло чергуються операції оцінювання та поліпшення від епізоду до епізоду. Після завершення епізоду відбувається оцінювання стратегії, використовуючи винагороду, яка була отримана в епізоді, після чого стратегія покращується. Алгоритм, що відповідає даній схемі, отримав назву ІС-алгоритм Монте-Карло з вивчаючими стартами.

Узагальнюючи сказане вище, слід сказати, що методи Монте-Карло мають 3 види переваг у порівнянні з методами ДП:

- перше, оптимальну поведінку можна побудувати в результаті безпосередньої взаємодії з довкіллям, при цьому не потрібно знання моделі динаміки довкілля;
- по-друге, ці методи можна використовувати у поєднанні з методами моделювання, що дозволяють отримати змодельовані сукупності епізодів;
- по-третє, методи Монте-Карло можуть легко та ефективно фокусуватися на невеликій підмножині станів.

2. TD, Q-learning. TD поєднують у собі ідеї методів Монте-Карло та ДП. Як і методи Монте-Карло, TD -методи можуть навчатися без моделі динаміки оточення, а не безпосередньо з досвіду. Як і методи ДП, TD -методи оновлюють розрахункові оцінки, частково ґрунтуючись на інших оцінках, при цьому не очікуючи на фінальний результат [21].

В основі алгоритмів Q-навчання (Q тут значить Quality - якість) лежить ідея, що при відомій Q-функції ймовірно вибирати наступну дію без наявності окремої моделі для стратегії:

$$\pi'(a_t|S_t) = \begin{cases} 1, \text{ якщо } a_t = \operatorname{argmax}_{a_t} Q^\pi_{S_t, a_t} \\ 0, \text{ в інших випадках} \end{cases} \quad (2.3)$$

Як і TD -методи, так і методи Монте-Карло використовують досвід, щоб вирішити завдання передбачення. З деякого даного досвіду проходження стратегії π , обидва методи оновлюють їх оцінки V функції V_π для станів S_t , які не є фінальними та присутні у даному досліді. У разі відвідування стану S_t у момент часу t обидва методи коригують свої оцінки ґрунтуючись на тому, що трапилося після відвідин даного стану (рис. 2.11).

$$\begin{array}{l} \text{Monte Carlo} \quad V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)] \\ \\ \text{TD Learning} \quad V(S_t) \leftarrow V(S_t) + \alpha[\underbrace{R_{t+1}}_{\text{Reward } t+1} + \underbrace{\gamma V(S_{t+1})}_{\text{Discounted value on the next step}} - \underbrace{V(S_t)}_{\text{Previous estimate}}] \\ \hspace{15em} \text{TD Target} \end{array}$$

Рис. 2.11. Порівняння метода TD-навчання та метода Монте-Карло

2.5.3. Функція вартості

Більшість алгоритмів навчання з підкріпленням включає поняття функції вартості – функції від стану (або від пари стану та дії), яка оцінює, наскільки вартим ним для агента перебувати в заданому стані (або наскільки вартим є застосування цієї дії в даному стані). Поняття у тому, наскільки варто якась дія чи стан виявляється у понятті майбутньої очікуваної нагороди. Нагорода, яку агент може розраховувати отримати у майбутньому, залежить від дій, що приймаються надалі, тому функції вартості визначені для кожної стратегії прийняття рішень.

Механізм оцінки дій є складною і неоднозначною процедурою, оскільки винагорода може характеризувати дії, виконані задовго до отримання винагороди,

або правильну послідовність дій, яка нараховує значну кількість кроків.

Існують різні підходи до цієї проблеми. Перший - це видавати винагороду плавно і часто. Але при цьому перед програмістом постає завдання планування дій, за які агент отримуватиме винагороди, а також їхню раціональну кількість.

Інший перспективний спосіб поліпшення навчання із сильно відстроченою винагородою - ієрархічне навчання із підкріпленням. Ієрархічне навчання з підкріпленням намагається розкласти довгострокове завдання на низку цілей та підзавдань. Розклавши проблему, ми ефективно розширюємо часові рамки, у яких приймаються рішення.

Функція вартості - повідомляє про найбільшу передбачувану майбутню винагороду, котру отримає агент у кожному стані (рис. 2.12). Значення кожного стану є загальною сумою винагороди, котру агент може отримати вподальшому, починаючи з цього стану.

$$v_{\pi}(s) = \mathbb{E}_{\pi} [R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s]$$

Expected Reward Given that state
 discounted

Рис. 2.12. Функція вартості для навчання з підкріпленням

2.6. Вибір програмних засобів для реалізації моделювання роботи маніпулятора

Для розробки програмних модулів навчання ІА управління роботизованими пристроями, необхідно змодельовати пристрій у середовищі з реалістичною фізикою, і розробити конфігурації ІА. Агенти повинні мати можливість взаємодіяти з середовищем, отримувати зворотну інформацію та навчатися в процесі. Виходячи з описаного вище, вибір засобів реалізації починається з вибору середовища розробки, яке має можливість моделювати реалістичні фізичні взаємодії об'єктів.

Проаналізувавши такі середовища розробки, як Unity, Unreal Engine, CryEngine, V-REP, а також можливість написання власного підходу до симуляції фізики, було обрано Unity. Основними причинами стали:

- додавання у 2020 році фізичної бібліотеки PhysX 4.0, яка забезпечує більш точне та продуктивне моделювання реалістичної фізики взаємодії об'єктів;
- передбачена можливість моделювання рухів за допомогою суглобів;
- велика кількість бібліотек та плагінів, у тому числі і по ШНМ;
- велика база документації;
- зручне відстеження помилок та роботи програми.

Для навчання ШНМ було обрано метод навчання з підкріпленням. Даний метод показав хороші результати у широкому колі завдань, таких як перемога чемпіона світу в Go та навчанні ігор Atari, що можна вважати окремими випадками навчання агента взаємодіяти з середовищем.

Після аналізу готових рішень вибір зводиться до двох мов програмування: C# та Python.

Python – високорівнева мова програмування, котра володіє строгою типізацією даних. Серед її переваг можна виділити високу продуктивність, хорошу читаність коду, кросплатформеність, можливість об'єктно-орієнтованого підходу, автоматичне керування пам'яттю, підтримку багатопотокових обчислень, а також велику кількість рішень, бібліотек та документації для роботи з ШНМ.

C# - об'єктно-орієнтована мова програмування, створена як мова розробки програм для платформи Microsoft .NET Framework. Серед переваг можна виділити хорошу читаність коду, автоматичне керування пам'яттю, підтримку багатопотокових обчислень, а також велику кількість бібліотек для вирішення величезного спектра завдань.

Порівнюючи мови програмування, Python є лідером для навчання ШНМ. Серед основних переваг можна виділити високу продуктивність, багато прикладів, бібліотек, документації.

Оскільки основна частина роботи - це моделювання середовища, і навчання

агента взаємодіючи з середовищем виконувати будь-які заплановані дії, виникає необхідність відстежувати дані дії візуально. Звідси випливає, що основна робота відбуватиметься мовою програмування C#, на рушію Unity. Але ця мова не підтримує можливості навчання високопродуктивних ШНМ, тому процес навчання відбуватиметься мовою програмування Python. Зв'язок між блоками різними мовами програмування буде базуватися за допомогою набору інструментів для агентів машинного навчання ML-Agents.

ML-Agents — це проект із відкритим вихідним кодом, який дозволяє іграм та симуляторам служити середовищем для навчання ІА. Даний проект надає можливість підключення сучасних алгоритмів на основі бібліотеки машинного навчання PyTorch та навчання ІА для ігор 2D, 3D та VR / AR. Також надається можливість використання Python API для навчання агентів із використанням навчання з підкріпленням, імітаційного навчання, нейроеволюції або будь-яких інших методів.

2.7. Визначення вимог до проектованої системи

Для досягнення поставленої мети роботи необхідно провести проектування програмного комплексу, що демонструє процес навчання ШНМ заданим поведінковим алгоритмам. В якості поведінкових алгоритмів були обрані завдання переміщення агентом об'єктів за допомогою штовхання, і захоплення об'єктів маніпулятором.

Виходячи зі наведеного вище можна дійти висновку, що навчання ґрунтуватиметься на інформації, одержуваної агентом від навколишнього середовища, і реакції середовища на дії агента. Розглянемо діаграму, що ілюструє цю взаємодію (рис. 2.13). При отриманні сигналу, запиту на новий стан, агент отримує інформацію про навколишнє середовище. На основі цієї інформації він робить будь-які дії. Середовище перетворюється на новий стан, і спираючись на

функцію вартості винагороджує агента. Потім відбувається візуалізація, і модель перетворюється на стан очікування запиту перехід у наступну епоху.

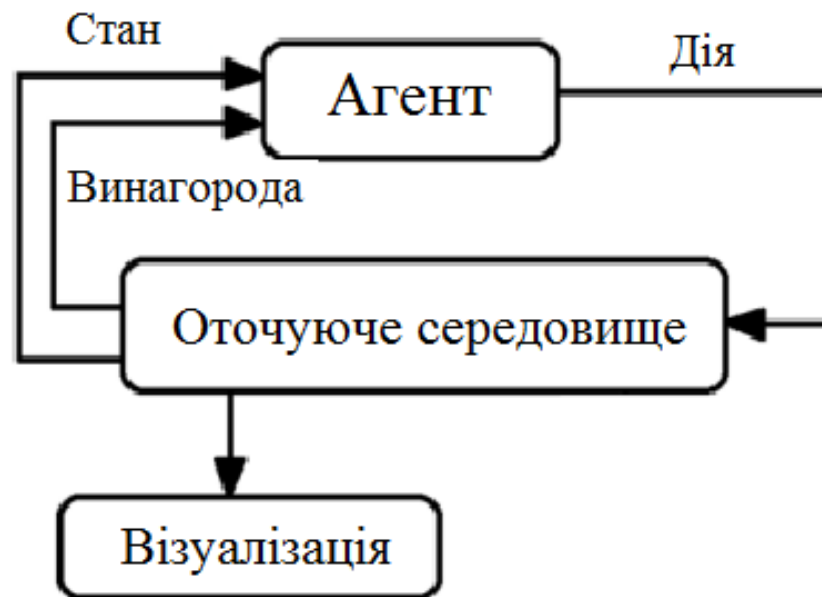


Рис. 2.13. Діаграма взаємодії елементів системи навчання

Діаграма на рис. 2.13 показує загальні принципи навчання ІА. Взаємодія елементів лише на рівні мов програмування представлена на рис. 2.14.

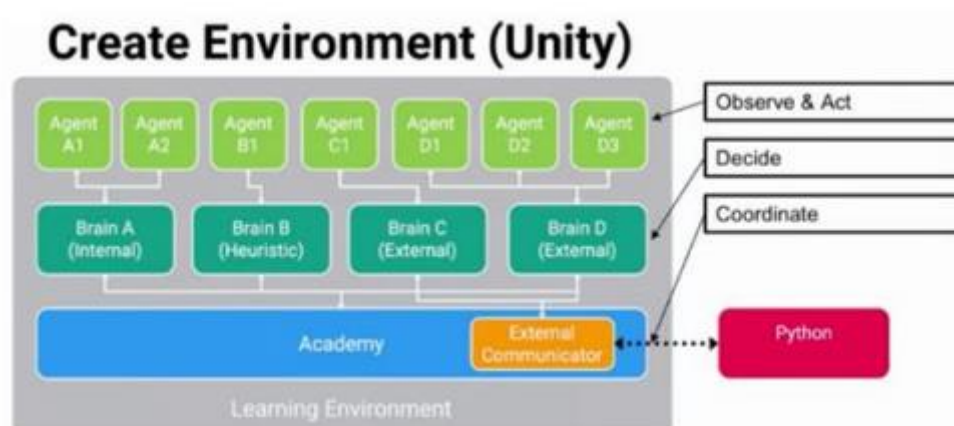


Рис. 2.14. Діаграма взаємодії елементів системи

Усі експерименти проводились на ноутбучі з характеристиками:

- процесор Intel Core i5-9300H;
- графічний процесор: NVIDIA GeForce GTX 1650;
- ОЗП 8 Гб (1 потік).

2.8. Висновки до розділу

Наведена загальна інформація про ШНМ та принцип їх роботи. Докладно описано різні парадигми навчання – з учителем, без учителя, з підкріпленням. Для третьої парадигми наведено фундаментальні методи її вирішення (Монте-Карло, TD, Q-learning). Описано особливості застосування функції вартості.

Виконано обґрунтування необхідних програмних компонент для реалізації моделювання роботи маніпулятора. Визначені проектні вимоги до пропонованої системи.

РОЗДІЛ 3

ПРАКТИЧНА ЧАСТИНА

3.1. Структура програми

Нижче наведено основні методи та класи, які відповідають за працездатність розробленого ПЗ.

3.1.1. Клас ArmHandAgent

Основне завдання цього класу – взаємозв'язок змодельованої сцени та навчального агента, запуск та перезапуск сцени, отримання додаткової інформації про навколишнє середовище, а також завдання функції винагороди за дії. Клас містить такі основні поля та методи:

- `refabArena` - містить об'єкт обмежує арени навколо агента. Використовується для випадкового розташування об'єктів за кожної симуляції;
- `goal` - об'єкт, що характеризує область, куди агент повинен доставити об'єкт;
- `block` – об'єкт, який необхідно транспортувати до `goal`;
- `armRobot` - об'єкт, керований агентом, що містить набір суглобів, з'єднаних зчленуваннями;
- `armHand` - об'єкт, розташований на кінці останнього зчленування `armRobot`, і дозволяє захоплювати об'єкти;
- `target` - не фізичний об'єкт, розташований між захватами `armHand`. Використовується для розрахунку відстані між захопленням і об'єктом;
- `captureDetect` - змінна, що характеризує положення захоплення;
- `keycodes` – список клавіш для ручного управління;
- `ScoredAGoal` - метод, що відповідає за винагороду та перезапуск сцени після виконання поставленого завдання;

- MoveAgent - метод, який відповідає за управління зчленуваннями. Може отримувати команди як від користувача, так і з ШНМ;
- OnActionReceived - метод, що відповідає за функцію винагороди агента;
- CollectObservations - метод, який відповідає за інформацію, що надходить від змодельованого середовища до агента;
- Heuristic – метод, який відповідає за ручне управління агентом;
- ResetBlock - метод, що відповідає за скидання положення та швидкості об'єкта;
- ResetRobot - метод, який відповідає за скидання положень зчленувань маніпулятора.

3.1.2. Клас RobotController

Основне завдання класу – зберігати всі об'єкти, з'єднані зчленуваннями у маніпуляторі. Є сполучним елементом між методами, що відповідають за керування, та окремими суглобами.

Нижче описані основні поля та методи цього класу:

- joints – список всіх наявних суглобів у моделі маніпулятора;
- GetCurrentJointRotations - повертає положення всіх наявних у маніпуляторі зчленувань;
- StopAllJointRotations - метод, що відповідає за зупинку руху всіх суглобів;
- RotateJoint - метод, який відповідає за рух заданого суглоба в заданому напрямку;
- ForceJointsToRotations - метод, що відповідає за паралельну зміну положень кількох зчленувань.

3.1.3. Клас ArticulationJointController

Основним завдання класу є керування окремими суглобами маніпулятора.

Варто навести опис основних його полів та методів:

- `RotationDirection` - перераховувальний тип, що визначає необхідність докладання сили до з'єднання суглобів в тому, або іншому напрямку;
- `FixedUpdate` - метод, який відповідає за прикладання сили до з'єднання суглобів протягом деякого часу;
- `CurrentPrimaryAxisRotation` - метод, що повертає поточний стан зчленування між вибраними суглобами;
- `RotateTo` - метод, що розраховує новий стан для з'єднання суглобів з урахуванням сил, що прикладаються;
- `ForceToRotation` - спосіб, що дозволяє змінити положення суглоба без застосування сил. Необхідний повернення до початкового стану.

3.2. Задача переміщення об'єкта агентом у вказану зону

З метою перевірки різних алгоритмів навчання ІА з різними варіаціями вхідних параметрів було змодельовано завдання переміщення об'єкта агентом у вказану зону.

3.2.1. Опис змодельованого завдання

На рис. 3.1 продемонстровано сцену, в якій агентом є синій квадрат. Білим квадратом є об'єкт, який потрібно перемістити в зелену зону. Переміщення об'єкта можливе лише методом його штовхання агентом, жодних функцій зчеплення елементів не передбачено. Промені, що походять від агента, є сенсорами, здатними визначати тип об'єкта, а також відстань до нього.

Кожна симуляція обмежена за кількістю пройдених епох. Це зроблено з

метою виходу симуляції з глухої ситуації, коли об'єкт розташований в кутку сцени, і агент не має можливості зрушити його. На початку кожної нової симуляції агент та об'єкт розміщуються у випадкових місцях.

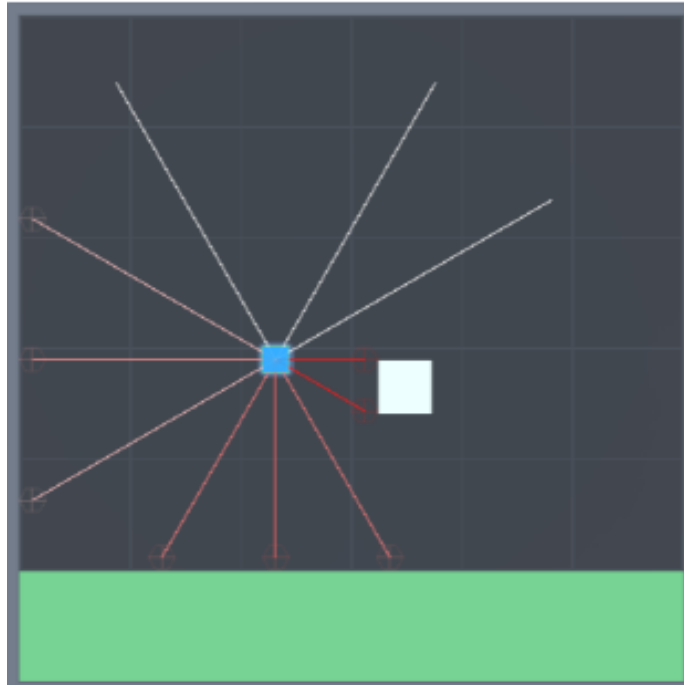


Рис. 3.1. Змодельоване середовище для навчання ІА

3.2.2. Результати навчання інтелектуальних агентів

Для досягнення розумного ступеня автономії ІА дві основні вимоги – це сприйняття та міркування. Перша забезпечується сенсорною системою, яка збирає інформацію про агента щодо навколишнього середовища. Кількість і якість сенсорів може сильно відрізнятися і підбирається індивідуально для кожного завдання. На рис. 3.2 продемонстровані різні конфігурації сенсорів, які відрізняються кутом огляду та максимальною вимірюваною відстанню. Як алгоритми навчання ІА застосовувалися SAC і PPO.

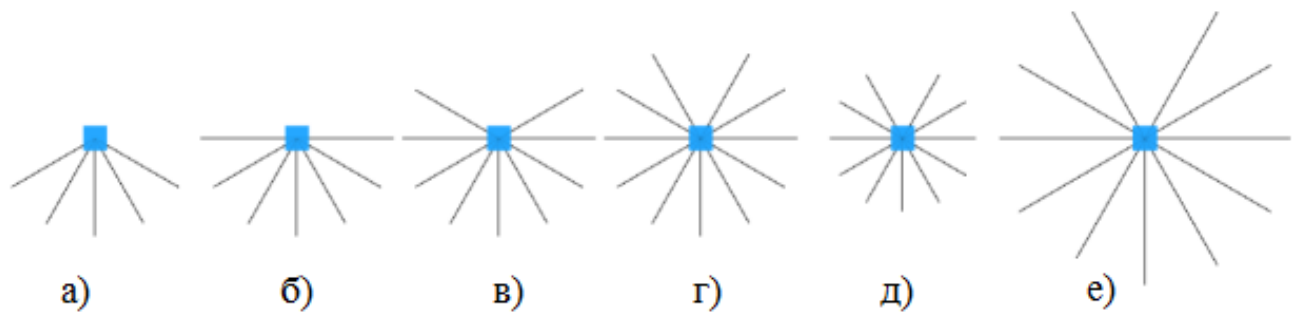


Рис. 3.2. Різні конфігурації сенсорів ІА (а - 120 градусів, б - 180 градусів, в - 240 градусів, г - 300 градусів, д - 300 градусів, максимальна довжина променя L зменшена на 25%, е - 300 градусів, довжину променя L збільшено на 50%)

Сконфігуровані ШНМ для алгоритмів SAC і PPO мають по 2 приховані повнозв'язкові шари прямого поширення, котрі мають по 256 нейронів. Період навчання агентів складає 2,5 мільйонів епох. Результати навчання ІА із різними конфігураціями сенсорів представлені у табл. 3.1.

Таблиця 3.1.

Результати навчання ІА

Конфігурації сенсорів	SAC		PPO	
	Середнє	Функція	Середнє	Функція
а - 120 градусів	2.630	9.95e-4	4.908	0.389
б - 180 градусів	3.012	2.81e-3	4.918	0.307
в - 240 градусів	3.917	3.39e-3	4.981	0.019
г - 300 градусів	4.742	3.97e-3	4.986	0.015
д - 300 градусів, L -25%	1.278	1.29e-3	4.872	0.345
е - 300 градусів, L +50%	4.979	3.39e-4	4.985	0.014

Середня винагорода та функція втрат вираховуються з останніх 50 000 епох симуляції. Середня винагорода характеризує те, наскільки добре виконується поставлене завдання, і може змінюватись від -1 до 5. З табл. 3.1 видно, що зі збільшенням кута огляду сенсорів агенти виконують завдання швидше, і

припускаються менше помилок, проте алгоритм навчання PPO здатний знайти складну послідовність рішень, яка призводить до хороших результатів навіть в умовах неповноти інформації, що надається.

Функція втрат характеризує те, як добре агент здатний пророкувати наслідки своїх дій. В алгоритмі PPO помітна залежність між інформацією, що надається, і функцією втрат, на основі якої можна підбирати типи сенсорів для різних завдань. В алгоритмі SAC на основі функції втрат можливий аналіз результатів навчання лише за достатньої повноти інформації, що надається. З усіх проведених дослідів лише в останньому функція втрат алгоритму SAC у процесі навчання мала характерну тенденцію зростання, а потім зменшення. Всі інші зміни призводили до хаотичних стрибків показника кожному етапі навчання.

Варто зазначити, що обидва алгоритми значно втрачають ефективність при зменшенні максимальної відстані, що вимірюється променем, а при збільшенні даного показника алгоритм SAC отримує достатньо інформації для ефективного просторового сприйняття сцени і пошуку ефективного вирішення задачі.

Проаналізувавши отримані результати було зроблено такі висновки:

- в умовах неповноти або спеціальної мінімізації вхідних даних алгоритм PPO показує кращі результати за більш короткий час навчання, ніж SAC;
- алгоритм SAC в умовах неповноти інформації, що надається, не здатний виявити складні, багаторівневі рішення задач, знайдених алгоритмом PPO;
- на основі функції втрат алгоритму PPO можна судити про повноту інформації, що надається. Алгоритм SAC такої можливості не надає.

3.3. Модель маніпулятора

Спроектowana модель для управління агентом є маніпулятором з 7 ступенями свободи (рис. 3.3). Враховуючи функцію захоплення та бездіяльність, це дає вибір із 17 варіантів дій у кожний момент часу

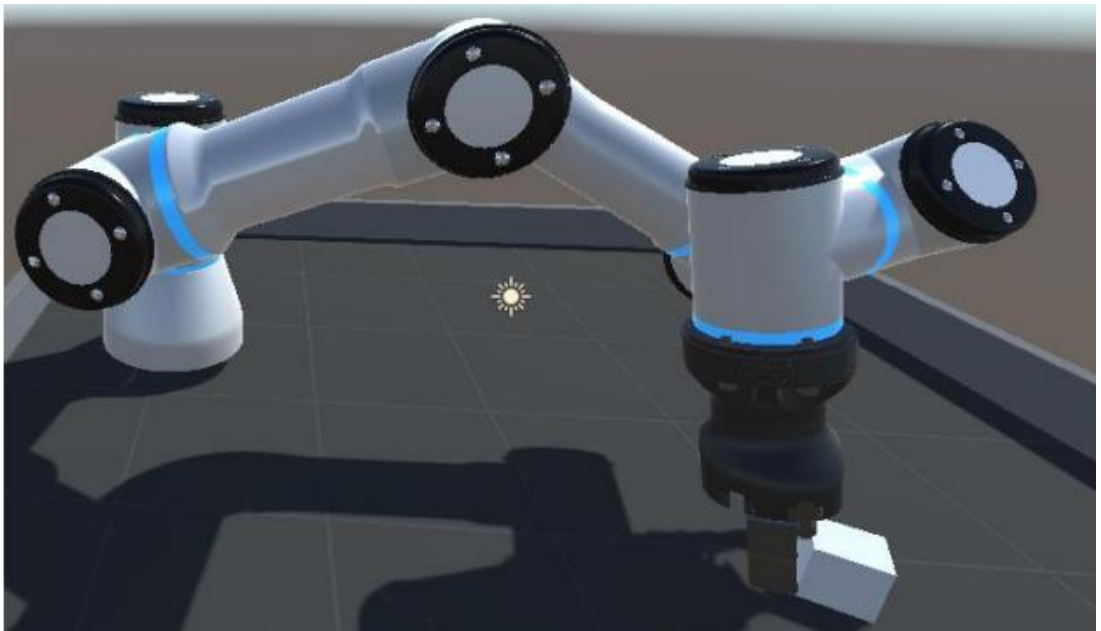


Рис. 3.3. Модель маніпулятора

Дана модель є аналогом реального маніпулятора від компанії Universal Robotics (рис. 3.4). Ця компанія займається виробництвом та продажем роботів для малого та середнього бізнесу, і позиціонує свою продукцію як універсальних помічників у автоматизації різних виробничих процесів: від складання до фарбування, від закручування гвинтів до наклеювання етикеток, від упаковки до полірування, від лиття під тиском до зварювання [22].



Рис. 3.4. Роботи- маніпулятори від компанії Universal Robots [22]

Конфігурація маніпулятора від компанії Universal Robotics використовується командою AVATRINA на змаганні \$10M ANA Avatar XPRIZE в розробці свого робота [23]. Ця команда виявилася однією з 38 півфіналістів цього змагання, розробивши аватар під назвою TRINA (рис. 3.5).

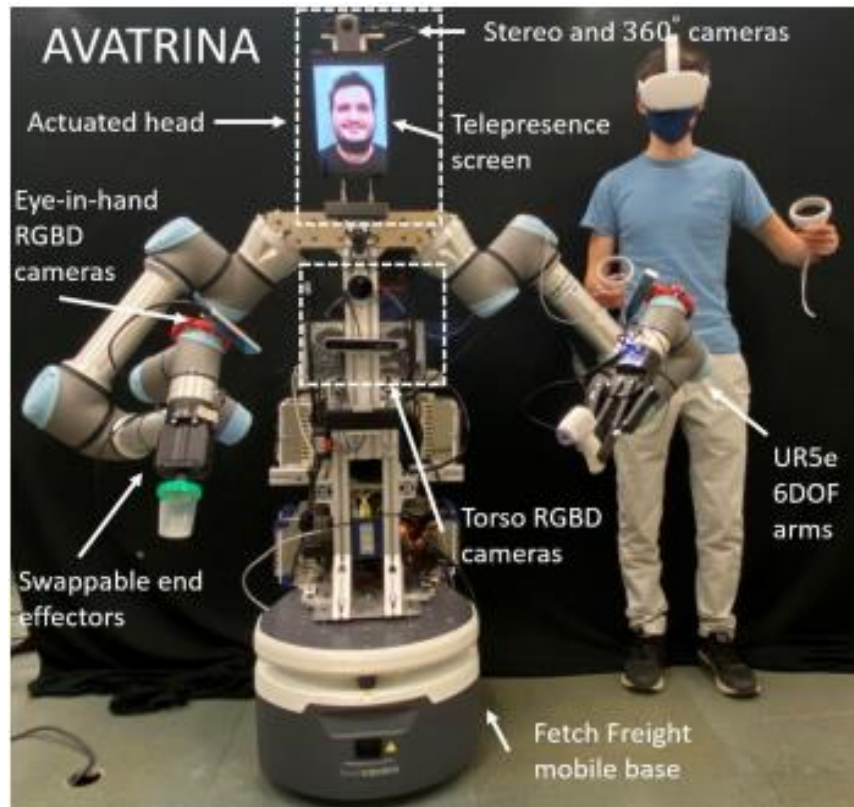


Рис. 3.5. Робот TRINA від команди TRINA [23]

Основна мета робота: передавати присутність людини у віддалене місце у реальному часі. Для цього він оснащений безліччю камер, давачами глибини та кількома маніпуляторами. Однак керованість роботизованими руками таких аватарів утруднена через більшу кількість суглобів, що видно на демонстраціях роботи на момент написання дослідження.

Вирішити цю проблему можна застосувавши ШНМ як посередника між оператором і аватаром.

3.4. Результати навчання із захоплення маніпулятором об'єкта

3.4.1. Опис навчального середовища

Змодельоване середовище зображено на рис. 3.6. Агент управляє роботизованим маніпулятором, розташованим край виділеної зони симуляції. Основна мета агента - це захоплення об'єкта, що є білим кубом. По краях зони симуляції розташовані борти для виключення випадків скидання об'єкта або виведення за краї досяжності маніпулятора у процесі навчання.

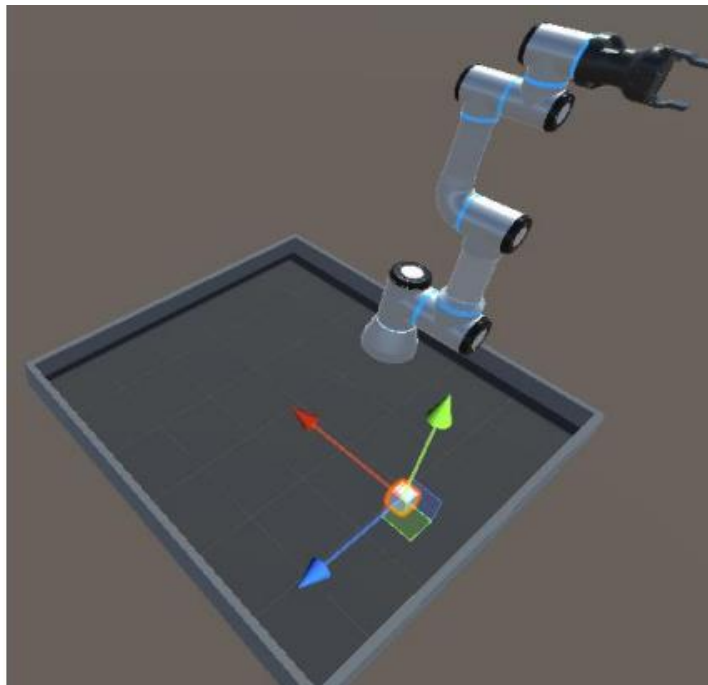


Рис. 3.6.Змодельоване середовище для захоплення об'єкта

Тривалість симуляції обмежена 10 000 епохами. У випадку, якщо агент не виконує поставлене завдання за відведений період, зона, що моделюється, перезапускається. При кожному перезапуску маніпулятор повертається у вихідний стан, об'єкт захоплення обнуляє параметри швидкості і переміщається у випадкову точку зони симуляції.

Агент виконує дії на основі інформації про його поточний стан:

- кут повороту між усіма суглобами;
- позиції зчленування щодо основи маніпулятора;
- вектори спрямування окремих суглобів до осі X у світовому просторі координат;

- вектори напрямку окремих суглобів до осі Z у просторі координат.

Для взаємодії маніпулятора з об'єктом агент отримує додаткову інформацію:

- вектор відстані від основи маніпулятора до об'єкта;
- вектор відстані від основи маніпулятора до хапального механізму;
- вектор віддалення від хапального механізму до об'єкта.

Для того, щоб агент краще прораховував складні дії та їх наслідки, у кожний момент симуляції, крім актуальної інформації на ШНМ, подаються попередні дані за останні 3 епохи. У результаті мінімальний розмір вхідного шару ШНМ налічує 237 нейронів.

Крім описаних вище даних додатково використовувалися давачі відстані, котрі розташовані на хапальному механізмі (рис. 3.7). Давачі випускають віяло з семи променів під кутом в 60 градусів з обмеженою максимальною відстанню. При попаданні променя перешкода класифікується. Відстань до перешкоди повертається на нейрон, що відповідає за об'єкт, що класифікується

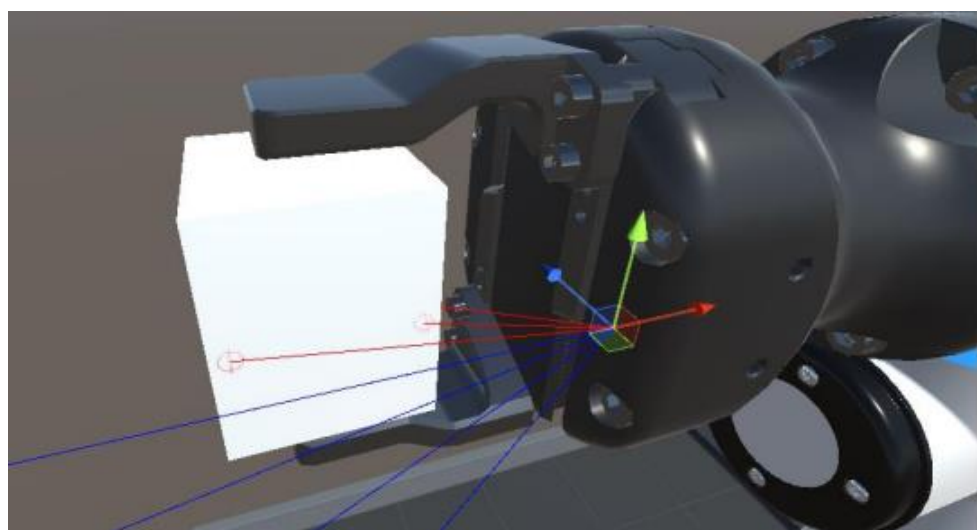


Рис. 3.7. Давачі відстані

Завдання підняття предмета маніпулятором на навчання ШНМ є досить тривіальним з погляду зміни функції вартості. На відміну від завдання, розглянутого в розділі 3.3, нагороди за підсумковий результат буде недостатньо через необхідність виконання великої послідовності дій. Це змушує додавати проміжні нагороди та штрафи, між якими необхідно дотримуватися балансу для досягнення якісного та швидкого навчання.

У ході виконання роботи було налаштовано та навчено більше сотні ШНМ. Метою даних експериментів - пошук оптимальних гіперпараметрів для функції вартості, аналіз їх впливу на процес навчання та побудова підсумкової моделі щодо підняття об'єкта.

У всіх проведених експериментах використовувалася однакова архітектура ШНМ, що складається з двох прихованих шарів, по 256 нейронів у кожному. Вихідний шар складається із 17 нейронів, що характеризують різні дії маніпулятора. Розмір вхідного шару залежить від кількості використовуваних датчиків і змінюється в різних експериментах.

3.4.2. Результати навчання

Одним із найважливіших факторів навчання з підкріпленням є функція винагороди. Від функції винагороди залежить не тільки підсумковий результат, а й швидкість збіжності алгоритму навчання ШНМ. Чим складніше завдання, якому необхідно навчити агентів, тим заплутанішим буде підсумкова функція винагород.

У задачі захоплення об'єкта маніпулятором використовувалися такі характеристики для складання функції винагороди:

- нагорода за захоплення об'єкта;
- штраф за час виконання завдання;
- штраф за дистанцію від захоплення до об'єкта;
- нагорода за підтримку маніпулятором вертикального становища.

Між усіма перерахованими характеристиками необхідно дотримуватися

балансу. Нагорода за захоплення та штраф за час виконання є константними характеристиками і не змінюються в процесі симуляції. Для пошуку правильних коефіцієнтів між штрафом за дистанцію від захоплення до об'єкта та винагородою за підтримку маніпулятором вертикального положення було проведено серію тестів. Значення використовуваних коефіцієнтів представлено табл. 3.2.

Таблиця 3.2

Значення коефіцієнта функції винагороди різних ШНМ

Назва ШНМ	Коефіцієнт для підтримки маніпулятором вертикального положення	Коефіцієнт до штрафу за дистанцію від захоплення до об'єкта	Нагорода за захоплення об'єкту	Штраф за час виконання завдання
ppo_roboARM_27	1/50	1/10	100	-50
ppo_roboARM_28	1/70			
ppo_roboARM_29	1/90			
ppo_roboARM_30	1/110			

На рис. 3.8 продемонстровано графік одержуваної винагороди у процесі навчання ШНМ, конфігурації яких описані у табл 3.2.

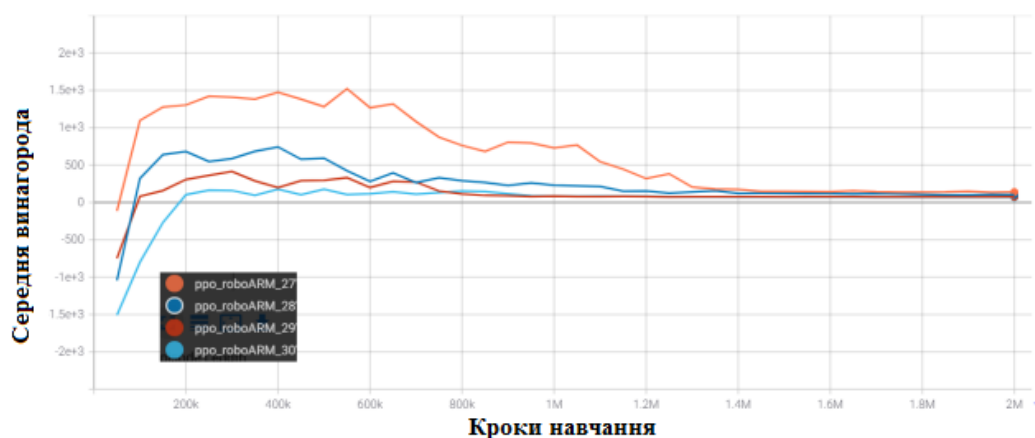


Рис. 3.8. Отримані винагороди

З графіків видно, що всі мережі успішно навчилися захоплювати об'єкт. Однак швидкість збіжності функцій у всіх різна. На рис. 3.8 видно, що винагорода за симуляція у трьох ШНМ на початку навчання вища, ніж наприкінці. Це пов'язано з тим, що агент досить швидко навчився скорочувати відстань між хапальним механізмом та об'єктом, що значно знизило штраф за відстань. Однак винагорода за підтримку маніпулятором вертикального положення залишилася, і її значення в сумі вище, ніж нагорода за захоплення. Цю проблему можна вирішити, збільшивши нагороду за виконання завдання, або зменшивши всі інші винагороди.

На рис. 3.9 продемонстровано графік тривалості симуляцій у процесі навчання ШНМ, конфігурації яких описані в табл. 3.2. Чим менший час симуляції, тим швидше агент виконує поставлене завдання - захоплює об'єкт. На графіці в період між 600 000 і 1 200 000 кроком навчання видно, що зі зменшенням значущості характеристики підтримки маніпулятором вертикального положення, швидкість збіжності алгоритму збільшується, і агент швидше навчається захоплювати об'єкт. Однак в ШНМ «ppo_roboARM_30» швидкість збіжності знову зростає. Це говорить про те, що в ШНМ «ppo_roboARM_29» досягнуто балансу між гіперпараметрами функції винагороди.

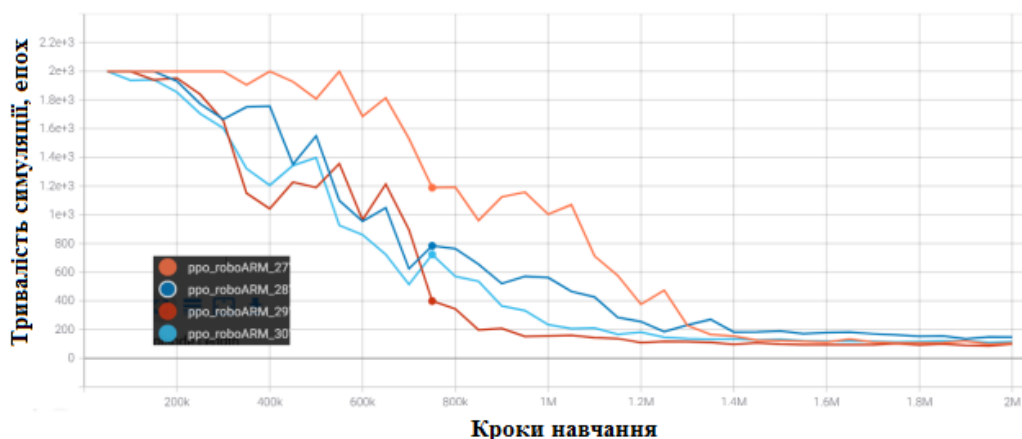


Рис. 3.9. Час роботи симуляції

У ході виконання даних експериментів було виявлено декілька проблем:

– при повороті початкового положення маніпулятора на 180 градусів агент втрачає здатність захоплювати об'єкти. Це пов'язано з тим, що всі отримані навчальні дані були змодельовані, коли об'єкти генерувалися в заданій області щодо маніпулятора. Функціонально, маніпулятор здатний повертатися і піднімати об'єкти навколо себе, але через відсутність подібного досвіду агент не здатний цього зробити;

– досить часто в процесі навчання, навіть із коректними гіперпараметрами функції винагороди, виникає ситуація, при якій агент вважає за краще закривати хапальний механізм на початку симуляції. Це призводить до нездатності захопити об'єкт.

Для вирішення першої проблеми було перероблено моделюване середовище. Початкове положення маніпулятора зміщене до центру симуляції зони (рис. 3.10). Друга проблема вирішена за допомогою введення штрафу за хибне захоплення, і додаванням до вхідної інформації для ШНМ даних про стан захоплення - відкрито або закрито. Ці зміни знизили швидкість навчання, оскільки простір можливих положень об'єкта щодо маніпулятора сильно зріс.

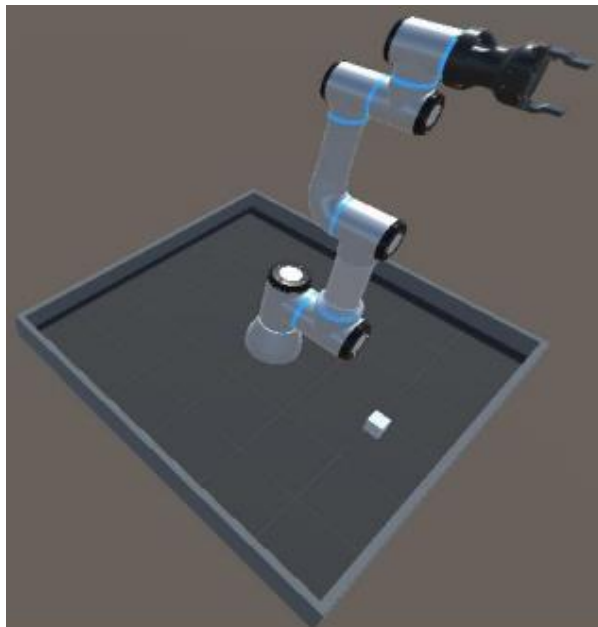


Рис. 3.10 Зміщення маніпулятора до центру зони симуляції

Було проведено серію тестів щодо навчання ІА алгоритмами SAC та PPO. Найкращі з отриманих результатів навчання представлені на рис. 3.11.



Рис. 3.11. Результати навчання ШНМ алгоритмами SAC та PPO

На рис. 3.11 продемонстровано графік тривалості симуляцій у процесі навчання ШНМ алгоритмами SAC та PPO.

На основі отриманих даних було зроблено висновок, що алгоритм SAC не здатний вирішити поставлене завдання на основі інформації, що йому надається. Основною причиною може бути брак вихідних даних для прогнозування кращої лінії поведінки, що змушує агентів шукати складніші залежності. Подальші дослідження проводилися лише з алгоритмом PPO.

Залежність якості навчання від кількості використовуваних датчиків відстані показано на рис. 3.12. На даному графіку показані найкращі з отриманих результатів для кожної конфігурації. Кількість датчиків у кожній ШНМ зазначено у табл. 3.3.

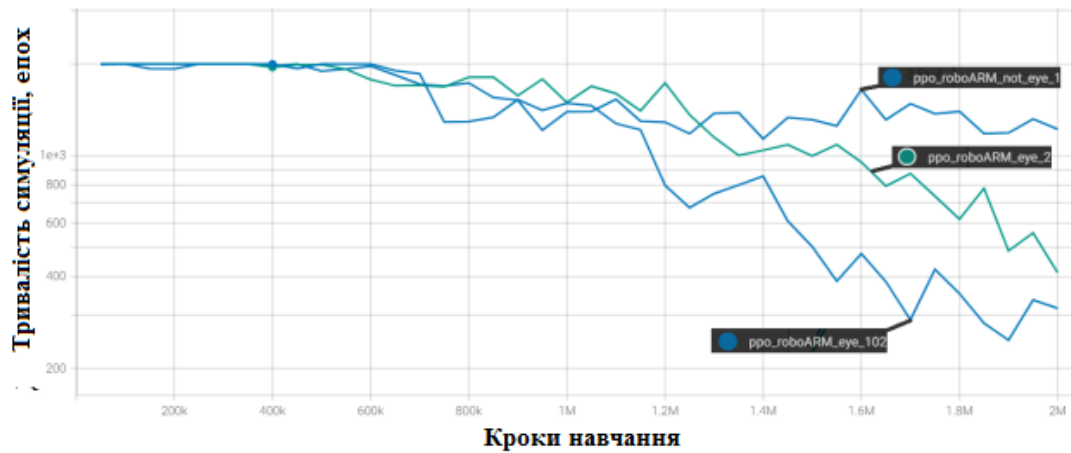


Рис. 3.12. Залежність тривалості симуляції від кількості використовуваних датчиків

Як видно з рис. 3.12, агент, який отримує інформацію з 4х датчиків, навчається швидше за інших. На другому місці стоїть агент із двома датчиками.

Таблиця 3.3

Зміни ШНМ за кількістю датчиків

ШНМ	Кількість датчиків
ppo_robARM_not_eye_1	0
ppo_robARM_eye_2	2
ppo_robARM_eye_102	4

Проаналізувавши результати, можна зробити висновок, що чим більшою наданою корисною інформацією, оперує ІА, тим швидше він навчається виконувати поставлене завдання.

3.5. Результати навчання з підняття об'єкта маніпулятором

У процесі розробки завдання із захоплення об'єкта було доповнено

необхідністю його підняття на певну висоту. У функцію винагороди було додано гіперпараметр, відповідальний за нарахування очок за висоту піднятого об'єкта. Новий ІА має назву «ppo_roboARM_eye_132», і використовує передбачену ШНМ «ppo_roboARM_eye_102», конфігурація та завдання якої розкрито в попередньому підрозділі. Графіки одержуваної винагороди у процесі навчання представлені на рис. 3.13.



Рис. 3.13. Отримана винагорода за підняття об'єкта

На рис. 3.13 видно, що агент «ppo_roboARM_eye_132» на перших кроках навчання отримав високу нагороду за виконання завдання, але згодом величина винагороди впала нижче, ніж ШНМ «ppo_roboARM_eye_102» отримувала наприкінці свого навчання. Основною причиною такого падіння стало невчасне виконання завдання. Агент отримав велику винагороду, але не мав досвіду дослідження навколишнього середовища — лише рефлексивні щодо захоплення об'єктів, передані з коефіцієнтами міжнейронних зв'язків агента «ppo_roboARM_eye_102». Підсумком стали тривалі дослідження змодельованого середовища з метою повторного отримання великої винагороди.

Розглядаючи тенденцію винагороди, видно, що в період приблизно з 50 000 до мільйона кроків навчання агент «ppo_roboARM_eye_132» досліджував навколишнє середовище і рідко захоплював об'єкт. У період із мільйона до двох із

половиною мільйонів агент захоплював об'єкт і піднімав на невелику висоту. У період із двох з половиною до трьох мільйонів кроків відбувається якісний стрибок, агент починає регулярно та швидко піднімати об'єкт на задану висоту.

Підбиваючи підсумки експерименту, можна дійти висновку, що використання попередньо навчених агентів дозволяє перенавчити ШНМ до виконання нових, складніших і багатоступінчастих завдань, але на перебудову наявних міжнейронних зв'язків, враховуючи необхідність збереження наявних можливостей, йде деякий час.

3.6. Висновки до розділу

У цьому розділі було реалізовано та протестовано алгоритми навчання ІА РРО та SAC з різними конфігураціями отриманої інформації від навколишнього середовища. В результаті експериментів було виявлено, що алгоритм РРО краще справляється з виявленням складних багаторівневих послідовностей для вирішення поставленого завдання.

Час навчання ІА залежить від CPU та GPU ПК, складності змодельованого простору, складності поставленого завдання та конфігурації ШНМ. Для навчання агента задачі захоплення об'єктів при використанні даного обладнання йде в середньому 2,5 години. Час навчання задачі з підйому об'єктів становило в середньому 6 годин.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Метою кваліфікаційної роботи магістра є дослідження математичного та ПЗ комп'ютеризованих систем автоматичного керування роботизованими механізмами на основі нейронних мереж. Оскільки, проведення робіт з розробки та використання системи передбачає використання комп'ютерної техніки, зокрема ПК та периферійних пристроїв, то обов'язковим є дотримання вимог з охорони праці і техніки безпеки.

Для ефективної і безпечної роботи колективу працівників з розробки ПЗ комп'ютерних систем, в тому числі і фахівців з підвищення ефективності контролю доступу в приміщення, необхідно організувати безпечні умови праці. При цьому керівник організації несе безпосередню відповідальність за порушення нормативно-правових актів з охорони праці [34]. Окрім цього, на робочих місцях працівників необхідно забезпечити дотримання вимог, затверджених Наказом Мінсоцполітики від 14.02.2018 за № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Згідно Вимог приміщення, де розміщені робочі місця операторів, крім приміщень, у яких розміщені робочі місця операторів великих ЕОМ загального призначення (сервер), мають бути оснащені системою автоматичної пожежної сигналізації відповідно до цих вимог;

– переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації, затвердженого наказом Міністерства України з питань надзвичайних ситуацій та у справах захисту населення від наслідків Чорнобильської катастрофи від 22.08.2005 N 161, зареєстрованого в Міністерстві юстиції України 05.09.2005 за N 990/11270 (НАПБ Б.06.004-2005);

– Державних будівельних норм "Інженерне обладнання будинків і споруд. Пожежна автоматика будинків і споруд", затверджених наказом Держбуду України від 28.10.98 N 247 (далі - ДБН В.2.5-56:2014, з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками.

В інших приміщеннях допускається встановлювати теплові пожежні сповіщувачі. Приміщення, де розміщені робочі місця операторів, мають бути оснащені вогнегасниками, кількість яких визначається згідно з вимогами ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників». Загальні технічні вимоги і з урахуванням граничнодопустимих концентрацій вогнегасної рідини відповідно до вимог НАПБ А.01.001-2014. Приміщення, в яких розміщуються робочі місця операторів сервера загального призначення, обладнуються системою автоматичної пожежної сигналізації та засобами пожежогасіння відповідно до вимог ДБН В.2.5-56:2014, ДБН В.2.5-56:2010, НАПБ А.01.001-2014 і вимог нормативно-технічної та експлуатаційної документації виробника. Проходи до засобів пожежогасіння мають бути вільними.

Лінія електромережі для живлення комп'ютера та периферійних пристроїв повинні бути виконаними як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Не допускається використовувати нульовий робочий провідник як нульовий захисний провідник. Нульовий захисний провідник прокладається від стійки групового розподільного щита, розподільного пункту до розеток електроживлення. Не допускається підключати на щиті до одного контактного затискача нульовий робочий та нульовий захисний провідники.

Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі має бути не менше площі перерізу фазового провідника. Усі провідники мають відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників,

температурному режиму та типам апаратури захисту, вимогам НПАОП 40.1-1.01-97.

У приміщенні, де одночасно експлуатуються понад п'ять комп'ютерів, на помітному, доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення. Комп'ютери повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення.

У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Не допускається підключати комп'ютери до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв. Електромережі штепсельних з'єднань та електророзеток для живлення комп'ютерної техніки повинні бути виконаними за магістральною схемою, по 3-6 з'єднань або електророзеток в одному колі. Штепсельні з'єднання та електророзетки для напруги 12 В та 42 В за своєю конструкцією мають відрізнятися від штепсельних з'єднань для напруги 127 В та 220 В. Штепсельні з'єднання та електророзетки, розраховані на напругу 12 В та 42 В, мають візуально (за кольором) відрізнятися від кольору штепсельних з'єднань, розрахованих на напругу 127 В та 220 В.

При підвищенні ефективності контролю доступу в приміщення, де для забезпечення безпеки мешканців, співробітників і збереження майна використовуються ДС, важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення, які визначені у НПАОП 0.00-7.15-18. Організація робочого місця фахівця із дослідження методів та програмно-апаратних засобів оптимізаційних процесів на основі ГА повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування

ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги». Відстань від екрана до ока фахівців, які працюють за комп'ютером визначається згідно з вимогами ДСанПіН 3.3.2.007-98.

Розміщення принтера або іншого пристрою введення-виведення інформації на робочому місці має забезпечувати добру видимість екрана комп'ютера, зручність ручного керування пристроєм введення-виведення інформації в зоні досяжності моторного поля згідно з вимогами ДСанПіН 3.3.2.007-98.

Таким чином, у результаті аналізу вимог щодо охорони праці користувачів комп'ютерів, визначено особливості організації робочих місць, вимог з електробезпеки, природного та штучного освітлення для ефективної і безпечної роботи фахівців з дослідження автоматичного керування роботизованими механізмами.

4.2. Комп'ютерне забезпечення процесу оцінки радіаційної та хімічної обстановки

Екологічне співтовариство розробило сімейство інструментів комплексної екологічної оцінки. Програмне забезпечення і послуги (ESS), комерційна група ПАСА, включаючи AirWare (для повітряних проблеми якості), WaterWare (для якості води), CityWare (якість повітря і води в контексті великих міст) і EIAxpert (для надання допомоги із загальним впливом на навколишнє середовище). Функціональність в цілому схожа на RAISON, хоча з великим акцентом на моделювання і меншим акцентом на керування даними. Знову ж таки, інструменти ESS розроблені як модульні набори інструментів (доступні спеціальні системи для вирішення конкретних завдань). Компоненти включають стандартні імітаційні моделі, включаючи моделі ISC і PBM Агентства з охорони навколишнього середовища США, управління даними, в тому числі ГІС, аналіз даних (наприклад, аналіз часових рядів даних спостережень), візуалізація, а також оптимізація [35].

Іноді немає готових моделей, придатних для конкретного застосування, але тягар розробки нової програми на Фортрані або С / С ++ є надмірним. Розробка моделі оточення може відносно легко реалізувати власні моделі комп'ютерів і не турбуватися про включення процедур для вирішення рівнянь, візуалізації і т. д. Як правило, за допомогою цих інструментів користувач просто повинен вказати свою модель, використовуючи або математичні рівняння, або спеціальні графічні символи або значки, які безпосередньо представляють поведінку системи.

На даний момент є розроблені моделі комп'ютерного забезпечення процесу для оцінки радіаційної та хімічної обстановки.

GEMS – це система на основі моделей, яка підтримує оцінки схильності і ризику, надаючи доступ до одиночних і мультимедійних моделям експозиції, фізико-хімічні властивості методи оцінки, статистичний аналіз, графічні та картографічні програми з відповідними даними на навколишнє середовище, джерела, рецептори і популяції. У розробці з 1981 року, GEMS надає аналітикам 84 84 інтерактивний, легко досліджуваний інтерфейс для різних моделей, програм і даних, які необхідні для оцінки хімічного впливу і ризику [36].

HSPF – це комплексний пакет для моделювання кількості і якості стоків з багатоцільових водозборів і процесів радіації, що відбуваються в потоках або повністю змішаних озерах. Це дозволяє інтегроване моделювання землі і ґрунту, процесів забруднення при гідравлічній і осадово-хімічній взаємодії. Результатом моделювання є тимчасові дані витрати стоку, концентрація поживних речовин і пестицидів, а також дані кількості і якості води в будь-якій точці водозбору. Алгоритми якості води включають динаміку BOD / DO, вуглець, азот і фосфор. Процеси трансформації, які включені в модель це: гідроліз, фотоліз, окислення, випаровування, сорбція і біодеградація. Вторинні або «дочірні» хімічні речовини також моделюються.

Вимоги до даних для моделі можуть бути досить широкими в залежності від конкретного застосування.

Модель MMSOILS – це методологія оцінки впливу на людину і ризику для здоров'я, пов'язаних з викидами забруднень з небезпечних відходів. Мультимедійна модель, що стосується перенесення хімічної речовини в ґрунтові води, поверхневі води, атмосферу і накопичення в їжі. Шляхи впливу на людину, які розглянуті в методології включають: потрапляння в ґрунт, вдихання летких речовин в повітря і тверді частинки, шкірний контакт, прийом питної води і т.д. Ризик, пов'язаний із загальною дозою опромінення, розраховується на основі хімічної токсичності [36].

4.3. Висновки до розділу

В цьому розділі проаналізовано важливі питання охорони праці та безпеки в надзвичайних ситуаціях, висвітлено питання комп'ютерного забезпечення процесу оцінки радіаційної та хімічної обстановки.

ВИСНОВКИ

У кваліфікаційній роботі досліджено можливості використання ШНМ для управління автономними апаратами. Розглянуто особливості систем керування автономними апаратами, вивчено різні способи сприйняття навколишнього середовища. Розглянуто архітектуру гібридної системи управління, яка дозволяє використовувати навчених у змодельованих середовищах ІА для управління роботизованими механізмами у реальному світі.

Було спроектовано та розроблено віртуальне середовище з моделювання роботи маніпулятора, та навчання ІА управлінню спроектованим пристроєм. Проведено поетапне навчання ІА виконання завдання із захоплення та підняття об'єктів. Проведено дослідження впливу різних гіперпараметрів функції вартості та застосування різної кількості давачів сприйняття оточення на швидкість та якість навчання ІА. В результаті отриманих даних була навчена ШНМ для управління приводами маніпулятора у змодельованому середовищі. Навчений агент успішно виконує завдання швидкого захоплення і підняття цільового об'єкта в 100% проведених симуляцій.

На основі отриманих знань можна проектувати складніші системи, що моделюють реальні роботизовані пристрої з метою навчання, подальшим калібруванням та використанням у зовнішньому світі.

Перспективами розвитку розробки є заміна програмованих поведінкових алгоритмів роботизованих пристроїв на ті, що можуть навчатися. Ця концепція може значно знизити вартість розробки ШІ, і багаторазово розширити спектр завдань для автономних апаратів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Researchers ran a simulator to teach this robot dog to roll over. URL: <https://techcrunch.com/2019/01/17/researchers-ran-asimulator-to-teach-this-robot-dog-to-roll-over/> (дата звертання 04.12.2022).
2. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
3. Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016
4. Stephen James and Edward Johns. 3d simulation for robot arm control with deep q-learning. arXiv preprint arXiv:1609.03759, 2016.
5. Benjamin Planche, Ziyang Wu, Kai Ma, Shanhui Sun, Stefan Kluckner, Terrence Chen, Andreas Hutter, Sergey Zakharov, Harald Kosch, and Jan Ernst. Depthsynth: Real-time realistic synthetic data generation from cad models for 2.5 d recognition. arXiv preprint arXiv:1702.08558, 2017.
6. J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World,” arxiv:1703.06907, 2017.
7. G. Griffin, A. Holub, and P. Perona. The Caltech-256. Technical report, California Institute of Technology, 2007.
8. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. <http://www.pascalnetwork.org/challenges/VOC/voc2010/workshop/index.html>
9. Мудрий І.Б. Локалізація та класифікація об'єктів на зображенні. *Інформаційні моделі, системи та технології: Праці X наук.-техн. конф.* (Тернопіль, 07-08 грудня 2022 р.), Тернопіль, 2022. С. 87.

10. Microsoft Kinect. URL:<http://www.xbox.com/en-us/kinect> (дата звертання 08.12.2022).
11. Jie Tang, Stephen Miller, Arjun Singh, and Pieter Abbeel. A textured object recognition pipeline for color and depth image data. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3467– 3474. IEEE, 2012
12. Alvaro Collet, Dmitry Berenson, Siddhartha S Srinivasa, and Dave Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 48–55. IEEE, 2009.
13. Fischler, M.A. and Bolles, R.C., "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Communications of the ACM*, 1981, 24, (6), pp. 381--395.
14. Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, 60, 2, pp. 91-110, 2004
15. H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, Vol.110, No. 3, pp. 346–359, 2008.
16. Lepetit, V., Fua, P.: Monocular model-based 3D tracking of rigid objects. *Foundations and Trends in Computer Graphics and Vision* 1(1), 1–89 (Jan 2005)
17. Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The moped framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research*, 30(10):1284– 1306, 2011.
18. Ferguson I.A. Integrated Control and Coordinated Behaviour: A case for Agent Models. In: *Intelligent Agents. ECAI-94 Workshop on Agent Theories, Architecture and Languages*
19. Новотарський М.А., Нестеренко Б.Б. Штучні нейронні мережі: обчислення // *Праці Інституту математики НАН України*. – Т50. – Київ: Ін-т математики НАН України, 2004. – 408 с.

20. Машинне навчання простими словами. Частина 2. Навчання з підкріпленням. URL: <http://www.mmf.lnu.edu.ua/ar/1743> (дата звертання 07.12.2022).
21. Вступ до машинного навчання. URL: <http://specials.kunsht.com.ua/machinelearning2> (дата звертання 07.12.2022).
22. Басюк Т.М. та ін. Машинне навчання: Навчальний посібник Львів: Видавництво «Новий Світ - 2000», 2021. 315 с.
23. Кононова К. Ю. Машинне навчання: методи та моделі. Харків: ХНУ імені В. Н. Каразіна, 2020. 301 с.
24. Навчання з підкріпленням у машинному навчанні. URL: <https://evergreens.com.ua/ua/articles/reinforcement-learning.html> (дата звертання 10.12.2022).
25. Machine Learning with Python. URL: <https://www.freecodecamp.org/learn/machine-learning-with-python/> (дата звертання 12.12.2022).
26. Sutton R. S., Barto A. G. Reinforcement learning: An introduction. – Cambridge Нейронные сети: MITpress, 1998. – Т. 1. – №. 1.
27. Wiering M., Van Otterlo M. Reinforcement learning //Adaptation, Learning, and Optimization. – 2012. – Т. 12.
28. Universal robots. URL: <https://www.universal-robots.com/> (дата звертання 07.12.2022).
29. MEET THE WINNERS \$10M ANA Avatar XPRIZE. URL: <https://www.xprize.org/prizes/avatar/> (дата звертання 07.12.2022).
30. Субботін С.О. Нейронні мережі : теорія та практика: навч. посіб. Житомир : Вид. О. О. Євенок, 2020. 184 с
31. Schmidhuber, J. Deep Learning in Neural Networks: An Overview // Neural Networks. 2015. Vol. 61. P. 85–117.
32. Bengio, Y. Deep Learning / Y. Bengio, Y. LeCun, G. Hinton // Nature. 2015. Vol. 521. P. 436–444.

33. Clements P. Evaluating software architectures. / , Kazman R. Klein M. Boston, MA: Addison-Wesley; 2002. pp.33-69.
34. Барський, А. Б. Нейронні мережі: розпізнавання, управління, прийняття рішень - М .: Фінанси і статистика, 2004. - 176 с.
35. Толок А.О. Крюковська О.А. Безпека життєдіяльності: Навч. посібник. 2011. 215 с.
36. Зеркалов Д.В. Охорона праці в галузі: Загальні вимоги. Навчальний посібник. К.: Основа. 2011. 551 с.

ДОДАТОК А

Тези конференції

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ

X НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



7–8 грудня 2022 року

ТЕРНОПІЛЬ
2022

В. Ліщина, Р. Жаронський МЕТОДИ ПІДВИЩЕННЯ ПРОПУСКНОЇ ЗДАТНОСТІ В МЕРЕЖАХ LTE	
V. Lishchyna, R. Zharovskyi METHODS OF INCREASE BANDWIDTH IN LTE NETWORKS	86
О. Марчук МЕТОД ІДЕНТИФІКАЦІЇ ДОРОЖНІХ ЗНАКІВ НА ОСНОВІ ЗГОРТКОВОЇ НЕЙРОМЕРЕЖІ	
O. Marchuk ROAD SIGN IDENTIFICATION METHOD BASED ON A CONVULSIONAL NEURAL NETWORK	87
І. Мудрий ЛОКАЛІЗАЦІЯ ТА КЛАСИФІКАЦІЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННІ	
I. Mudryi LOCATION AND CLASSIFICATION OF IMAGE OBJECTS	88
Т. Патриський ЗБЕРІГАННЯ ТА ТРАНСФОРМАЦІЯ ДАНИХ У ХМАРНОМУ СЕРЕДОВИЩІ GOOGLE CLOUD BIGQUERY	
T. Patralskyi DATA STORAGE AND TRANSFORMATION IN THE CLOUD ENVIRONMENT GOOGLE CLOUD BIGQUERY	89
В. Савчук, Н. Луцук АНАЛІЗ ІСНУЮЧИХ СИСТЕМ КЛІМАТ-КОНТРОЛЮ	
V. Savchuk, N. Lutsyk ANALYSIS OF EXISTING CLIMATE CONTROL SYSTEMS	90
В. Савчук, Н. Луцук РОЗРОБКА СИСТЕМИ КЛІМАТ-КОНТРОЛЮ НА БАЗІ МІКРОКОНТРОЛЕРА ТА СЕНСОРІВ	
V. Savchuk, N. Lutsyk DEVELOPMENT OF THE CLIMATE CONTROL SYSTEM BASED ON THE MICROCONTROLLER AND SENSORS	91
С. Свергун, Р. Жаронський ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПОБУДОВАНОГО НА МІКРОСЕРВІСНІЙ АРХІТЕКТУРІ	
S. Svergun, R. Zharovskyi TESTING OF SOFTWARE BUILT ON MICROSERVICE ARCHITECTURE	92
С. Свергун, Р. Жаронський ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ, ПОБУДОВАНОГО НА МІКРОСЕРВІСНІЙ АРХІТЕКТУРІ НА ОСНОВІ BDD	
S. Svergun, R. Zharovskyi TESTING OF SOFTWARE PRODUCT BUILT ON MICROSERVICE ARCHITECTURE BASED ON BDD	93
І. Слюз, Р. Жаронський ПРИНЦИПИ ТА ОСНОВНІ ЕТАПИ КОМПЛЕКСНОГО ТЕСТУВАННЯ КОМП'ЮТЕРНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ	
I. Slyuz, R. Zharovskyi PRINCIPLES AND MAIN STAGES OF COMPLEX TESTING OF A COMPUTER INFORMATION SYSTEM	94

УДК 004.93

I. Мудрий

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ЛОКАЛІЗАЦІЯ ТА КЛАСИФІКАЦІЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННІ

UDC 004.93

I. Mudryi

LOCATION AND CLASSIFICATION OF IMAGE OBJECTS

Процес розпізнавання предмету у неструктурованих сценах – складна сфера поточних досліджень комп'ютерного зору. Робототехніка є одним із важливих напрямків, де здатність чітко і швидко ідентифікувати об'єкти, що цікавлять, має вирішальне значення для універсальних роботів, що виконують завдання в неструктурованому повсякденному середовищі, таких як будинок, офіс або склад.

Традиційна система розпізнавання об'єктів завжди опрацює значне число різних об'єктів. Один із способів упоратися з цим – запровадити ієрархію, виконуючи розпізнавання на різних категоріях, а не на рівні примірника. Завдання узагальнення від кількох екземплярів до цілої категорії об'єктів залишається складним, існують численні тести та завдання, такі як Caltech 256 та PASCAL VOC, щоб сприяти прогресу у цій галузі. З іншого боку, для конкретного робота в конкретному середовищі кількість унікальних об'єктів відносно невелика (можливо, близько сотень). Це дозволяє розглядати завдання як проблему розпізнавання екземплярів, збираючи великий обсяг навчальних даних кожного об'єкта. Крім того, робот може використовувати дані, одержані від різних сенсорних пристроїв, таких як камери та датчики глибини.

Локалізація об'єктів у робототехніці також є додатковими проблемами, яких немає в тестах розпізнавання об'єктів на рівні категорій. Середовище реального світу дуже зашарашене, містить багато перекриттів і часто містить більшу кількість різних об'єктів в одній сцені. Роботам часто доводиться взаємодіяти з одними об'єктами у своєму середовищі, омиваючи інші. Це означає, що роботизована система сприйняття повинна точно локалізувати об'єкти після виявлення. Крім того, щоб робот міг швидко реагувати на зміни у навколишньому середовищі, роботизована система сприйняття має працювати у режимі реального часу. В даний час існує ряд різних підходів до розпізнавання та локалізації предметів у складних неструктурованих сценах. Першим є використання сенсорів для збирання кольорових та глибинних зображень. Прикладами даних сенсорів є рівні лідари, або більш прості датчі типу Kinect. Під час навчання розпізнаванню та локалізації об'єктів на основі інформації про глибину вдається точніше виділити контури кожного об'єкта. На основі контурів, кольору та глибини будується повна хмара точок, з якої виділяються тривимірні сіткові моделі об'єктів. Потім з отриманих моделей витягуються і реєструються локальні особливості об'єктів.

Розпізнавання об'єктів за локальними ознаками на зображеннях відбувається у кілька етапів. Спершу знаходяться відповідності. Цей етап відноситься до точного зіставлення характеристик зображення з характеристиками, що належать певному предмету. Далі йде процес оцінки пози об'єкта, яка буде геометрично узгодженою зі знайденими відповідностями. Після цього відповідно до масштабу предмета у базі даних і зображення знаходиться точна відстань до об'єкта. Важливо, що вибір камери та об'єктивна може сильно вплинути на точність оцінки пози. Тим не менш, камери стають якіснішими та дешевшими, а їх габарити зменшуються. Це призводить до того, що стають легко доступними паралельне використання декількох камер для отримання виду сцени з різних ракурсів та високою роздільною здатністю. Цей підхід ефективний, тому що він забезпечує оцінку глибини, стійкість до оклюзій прямої видимості та зростання ефективного кола огляду.