

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод мережевого оператора для управління
"Swarm-bot" - system

(тема)

Виконав:

студент II курсу, групи КСМм-22-1
Вітренко В.С.
(прізвище, ініціали)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

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

Освітня програма Комп'ютерні системи та мережі
(повна назва освітньої програми)

Керівник: доц. Токарев В.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

(підпис)

Коваленко А.А.

(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

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

Освітня програма _____ Комп'ютерні системи та мережі _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Вітренку Віталію Сергійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Метод мережевого оператора для управління "Swarm-bot" - system _____

затверджена наказом по університету від “ 06 ” листопада 2023 р. № 1298Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 15 січня 2024 р.

3. Вхідні дані до роботи _____ 1) провести дослідження методу мережевого оператора для управління інтелектуальними мобільними «s-bots», що входять до складу однієї «Swarm-bot» - system; 2) провести огляд та аналіз планувальників маршрутів; 3) провести огляд та аналіз методів запобігання колізіям; 4) провести дослідження способу побудови адаптивної системи управління «s-bots».

4. Перелік питань, що потрібно опрацювати у роботі _____

1) огляд літератури за темою роботи;

2) аналіз предметної області;

3) вибір та обґрунтування методики дослідження;

4) проведення експериментальних досліджень;

5) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____

Слайд-презентація – 25 слайдів _____

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури за темою роботи	07.11.23 - 13.11. 23	
2	Вибір та обґрунтування методики дослідження	14.11. 23 - 20.11. 23	
3	Вибір інструментальних засобів	21.11. 23 - 23.11. 23	
4	Розробка моделей протоколів	24.11. 23 - 06.12. 23	
5	Проведення експериментів	07.12.23 - 23.12.23	
6	Оформлення матеріалів кваліфікаційної роботи	26. 12. 23 - 02.01. 24	
7	Подання кваліфікаційної роботи керівникові та її попередній захист	03. 01. 24 - 06.01. 24	
8	Подання кваліфікаційної роботи на рецензування	09.01.24 - 12.01. 24	

Дата видачі завдання 06 листопада 2023 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Токарев В.В. _____
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 63 с., 18 рис., 2 табл., 1 дод., 14 джерел.

АЛГОРИТМ УПРАВЛІННЯ, ІНТЕЛЕКТУАЛЬНІ МОБІЛЬНІ «S-BOTS», МЕРЕЖЕВИЙ ОПЕРАТОР, «S-BOT», «SWARM-BOT» - SYSTEMS.

Метою кваліфікаційної роботи є дослідження можливості застосування методу мережевого оператора для управління інтелектуальними мобільними «s-bots», що входять до складу однієї «Swarm-bot» - system.

У ході виконання кваліфікаційної роботи досліджується метод мережевого оператора для розробки алгоритму управління однієї «Swarm-bot» - system. Алгоритм управління однієї «Swarm-bot» - system розробляється для вирішення задачі переміщення інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system, з довільних початкових станів до довільних кінцевих станів, з дотриманням динамічних фазових обмежень в умовах відсутності повної інформації про середовище та обміну інформацією між ними.

ABSTRACT

Master's thesis: 63 pages, 18 figures, 2 tables, 1 appendices, 14 sources.

CONTROL ALGORITHM, INTELLIGENT MOBILE «S-BOTS»,
NETWORK OPERATOR, «S-BOT», «SWARM-BOT» - SYSTEMS.

The purpose of the qualification work is to study the possibility of using the network operator method to control intelligent mobile «s-bots» that are part of one «Swarm-bot» - system.

In the course of the qualification work, the method of the network operator for the development of the control algorithm of one «Swarm-bot» - system is investigated. The control algorithm of one «Swarm-bot» - system is developed to solve the problem of moving intelligent mobile «s-bots» that are part of one «Swarm-bot» - system from arbitrary initial states to arbitrary terminal states with observance of dynamic phase restrictions in the absence of full information about the environment and the exchange of information between them.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	7
ВСТУП	8
1 ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ УПРАВЛІННЯ ІНТЕЛЕКТУАЛЬНИМИ МОБІЛЬНИМИ «S-BOTS» В ОДНІЙ «SWARM-BOT» - SYSTEM	10
1.1 Планувальники маршрутів для інтелектуальних мобільних «s-bots» ...	10
1.2 Методи потенційних полів для управління інтелектуальними мобільними «s-bots».....	17
1.3 Реактивні підходи управління інтелектуальними мобільними «s-bots»	19
1.4 Управління інтелектуальними мобільними «s-bots» на основі прогнозуючих моделей.....	21
1.5 Методи управління інтелектуальними мобільними «s-bots» на основі навчання	23
2 НЕЙРОННА МЕРЕЖА ДЛЯ УПРАВЛІННЯ ІНТЕЛЕКТУАЛЬНИМИ МОБІЛЬНИМИ «S-BOTS», ЩО ВХОДЯТЬ ДО СКЛАДУ ОДНІЄЇ «SWARM-BOT» - SYSTEM.....	26
3 ДОСЛІДЖЕННЯ МЕТОДУ МЕРЕЖЕВОГО ОПЕРАТОРА ДЛЯ УПРАВЛІННЯ ІНТЕЛЕКТУАЛЬНИМИ МОБІЛЬНИМИ «S-BOTS», ЩО ВХОДЯТЬ ДО СКЛАДУ ОДНІЄЇ «SWARM-BOT» - SYSTEM	36
3.1 Постановка задачі.....	36
3.2 Обчислювальний експеримент	40
ВИСНОВКИ.....	46
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	47
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	50

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

БЗ – база знань

ПЗ – програмне забезпечення

DQN – глибока Q-мережа (англ., Deep Q Networks)

MATLAB – середовище моделювання (англ., Matrix Laboratory)

MPC – управління з прогнозуючими моделями (англ., Model Predictive Control)

ORCA – оптимальне взаємне уникнення колізій (англ., Optimal Reciprocal Collision Avoidance)

RVO – взаємні швидкісні перешкоди (англ., Reciprocal Velocity Obstacle)

ВСТУП

В даний час набула широкого поширення нова парадигма, орієнтована на спільне використання наукових досягнень, які надають можливості за допомогою передових алгоритмів та програмних засобів керувати інтелектуальними мобільними «s-bots». У науковій літературі показано, що в даний час активно розвиваються методи формування та побудови «Swarm-bot» - system на базі передових методів та технологій. Такі системи використовуються для управління інтелектуальними мобільними «s-bots» в одній «Swarm-bot» - system, які виконують загальне завдання або завдання з цілями, що розділяються, в умовах невизначеності.

В якості інтелектуальних мобільних об'єктів зі структурою, що перебудовується, і програмованої логікою використовуються «s-bots» різного призначення, які можуть виконувати поставлені перед ними завдання для досягнення цільових функцій у трьох середовищах:

- на землі;
- під водою;
- у повітрі.

Управління інтелектуальними мобільними «s-bots» в одній «Swarm-bot» - system представляє собою новий підхід, що вивчає можливості побудови високотехнологічних «Swarm-bot» - system із сукупності автономних інтелектуальних мобільних «s-bots», з перебудовуваною структурою і програмованою логікою для досягнення цільового функціоналу, який не може бути досягнутий окремим «s-bot» або для яких колективне виконання поставленої задачі є більш ефективним.

В основі управління інтелектуальними мобільними «s-bots» в одній «Swarm-bot» - system лежить swarm intelligence. Сутність swarm intelligence полягає в тому, що swarm, що складається з інтелектуальних мобільних «s-bots», володіє більшими можливостями порівняно з його окремими

індивідами. За допомогою встановлення комунікацій між інтелектуальними мобільними «s-bots» в одній «Swarm-bot» - system та взаємодії з physical environment можливе вирішення завдань, на виконання яких у окремого «s-bot» не вистачає ресурсів.

В даний час не існує універсального підходу до створення легко масштабованих систем управління інтелектуальними мобільними «s-bots», що входять до складу однієї «Swarm-bot» - system, який допускав би використання довільної математичної моделі об'єктів управління, і у якому пошук управлінь не виконувався б у реальному масштабі часу. Проблема є важливою і нерозв'язаною, що визначає її актуальність. У кваліфікаційній роботі досліджується можливість застосування методу мережного оператора для управління інтелектуальними мобільними «s-bots», що входять до складу однієї «Swarm-bot» - system.

1 ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ УПРАВЛІННЯ ІНТЕЛЕКТУАЛЬНИМИ МОБІЛЬНИМИ «S-BOTS» В ОДНІЙ «SWARM-BOT» - SYSTEM

1.1 Планувальники маршрутів для інтелектуальних мобільних «s-bots»

Постійні зміни physical environment є головною перешкодою при створенні систем управління інтелектуальними мобільними «s-bots», що входять до складу однієї «Swarm-bot» - system. Специфіка управління інтелектуальними мобільними «s-bots» у нестационарній physical environment полягає в тому, що траєкторію перешкод, що рухаються, розрахувати заздалегідь вкрай важко. Щоб уникнути зіткнень з перешкодами, що рухаються, необхідно знати їх стани і мати можливість передбачати траєкторію їх руху (рисунок 1.1).



Рисунок 1.1 – Приклад планування маршруту

Тоді інтелектуальні мобільні «s-bots» можуть рухатися спланованим маршрутом, відхиляючись від нього лише для ухилення від зіткнень з перешкодами.

Планувальники маршруту поділяються на:

- централізовані;
- незв'язані.

Планування руху інтелектуальних мобільних «s-bots» може бути здійснено на основі спільного конфігураційного простору об'єктів або із застосуванням непов'язаних методик, що розділяють проблеми планування та координації руху.

Спільний конфігураційний простір формується у вигляді декартового добутку конфігураційних просторів окремих інтелектуальних мобільних «s-bots», тоді як нескладні планувальники здійснюють пошук маршрутів незалежно друг від друга. З метою уникнення колізій нескладні планувальники використовують діаграми координації.

Централізоване планування не застосовується на великих групах інтелектуальних мобільних «s-bots» через «прокляття» розмірності. Незв'язані планувальники здатні знаходити маршрути для великої кількості інтелектуальних мобільних «s-bots». Найчастіше пошук оптимальних маршрутів руху незв'язними планувальниками призводить до втрат оптимальності. Ці втрати можуть виражатися, наприклад, у дуже довгих об'їзних маршрутах об'єктів.

Також існують планувальники, які не можуть бути чітко віднесені до спільного чи незв'язаного планування. Такі підходи вдаються до застосування дорожніх карт, які мають покривати вільні конфігураційні простори інтелектуальних мобільних «s-bots». У цих підходах робиться акцент на швидкості обчислень чи практичній реалізованості. Планування у сполученому конфігураційному просторі є повноцінним з погляду оптимальності, але витратним з обчислювальної точки зору. Воно застосовується лише для інтелектуальних мобільних «s-bots», функціонуючих у простій physical environment. Незв'язні планувальники швидко знаходять маршрути, але вони часто не є оптимальними. Підходи на

основі дорожньої карти націлені на досягнення компромісу між продуктивністю та оптимальністю маршрутів.

Одна з груп дослідників запропонувала алгоритм управління «s-bot» в нестационарній physical environment з трасуванням рухомих перешкод. Для планування маршрутів ця група дослідників використовувала евристичний алгоритм A* (рисунок 1.2).

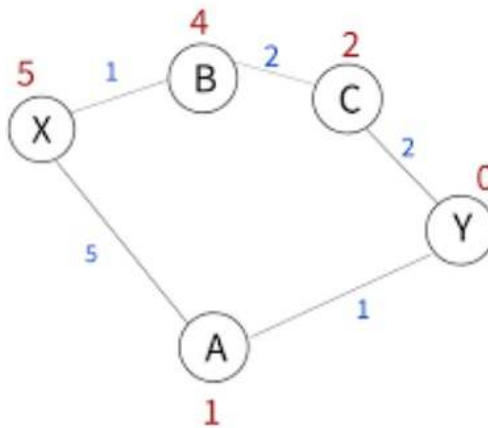


Рисунок 1.2 – Приклад евристичного алгоритму A*

Даний алгоритм здатний спланувати оптимальний маршрут з початкового стану в кінцевий в стаціонарному робочому середовищі з фіксованими перешкодами. Він може бути адаптований для безперервного пошуку інтелектуальними мобільними «s-bots» маршрутів в умовах безперервно змінного середовища.

Інша група дослідників запропонувала методи планування маршрутів для інтелектуальних мобільних «s-bots», функціонуючих у середовищах із динамічними перешкодами. Відмінною рисою цих методів є здатність прогнозування руху динамічних перешкод. Спрогнозований рух враховується при формуванні траєкторій руху «s-bots».

Одна з груп вчених проводила дослідження, пов'язані з декомпозицією physical environment на комірки, які враховують статичні та динамічні фазові

обмеження. Спочатку алгоритм здійснює розкладання вільного простору на комірки, а потім виконує пошук оптимального маршруту по результатууючому графу суміжності. Використання спільних планувальників є недоцільним, якщо в групі велика кількість об'єктів, оскільки це призводить до високої розмірності спільного конфігураційного простору пошуку. Природним шляхом усунення цієї проблеми є стиснення спільного простору пошуку. Одним із способів такого стиснення є обмеження деяких маршрутів, які можуть бути призначені для об'єктів. Одним із найпопулярніших підходів стиснення сполученого простору є підхід, заснований на дорожніх картах.

Група дослідників опублікувала звіт про один підхід створення складової дорожньої карти на основі суперграфа. Складова дорожня карта обчислюється у два етапи:

- етап № 1. Для кожного окремого «s-bots» будується дорожня карта ;
- етап № 2. Побудовані дорожні карти поєднуються в одну дорожню карту.

Єдина дорожня карта використовується для формування узгоджених маршрутів, які дають безліч можливих маршрутів для всіх «s-bots», що виключають появу колізій. Основна ідея полягає у пошуку маршрутів з урахуванням єдиної дорожньої карти, які б доставляли об'єкти в кінцеві стани без колізій.

В одній із наукових праць дослідниками представлено дорожню карту на основі суперграфа. Автори цієї наукової роботи пропонують два варіанти реалізації суперграфа:

- плоский суперграф;
- багаторівневий суперграф.

Вузли плоского суперграфа вказують на припустимі розміщення об'єктів. Дугам графа відповідають вільні від перешкод маршрути. Багаторівневий суперграф розроблено з метою скорочення плоского суперграфа. Вузли багаторівневого суперграфа містять інформацію про об'єкти, що входять до підграфу. На вузли багаторівневого суперграфа

накладено обмеження, згідно з яким підграфи не повинні перетинатися, тобто вузли одного підграфа не можуть блокувати шляхи до іншого підграфа. Головним недоліком методів дорожніх карт на основі суперграфів полягає в тому, що вони застосовуються тільки на невеликих групах через експоненційну залежність кількості вузлів графа від кількості інтелектуальних мобільних «s-bots» в одній «Swarm-bot» - system.

Також відомий підхід на основі дорожньої карти, в якому автори розробили багатофазний планувальник, який використовує сполучне дерево для планування безперешкодних маршрутів. Даний підхід передбачає створення графа, вузлами якого є початкові та кінцеві стани інтелектуальних мобільних «s-bots», а дуги представляють можливі зв'язки вузлів. Коренем сполучного дерева стає вузол, що найближче розташований до центру карти. На першому етапі підходу створюється план, згідно з яким «s-bots» переміщуються до листя сполучного дерева за маршрутами, що забезпечують відсутність зіткнень. На другому етапі об'єкти переміщуються на позиції, з яких вони можуть потрапити до термінальних станів, не перешкоджаючи іншим об'єктам. Це досягається за рахунок розгляду об'єктів по порядку відповідно до глибини розташування їх термінального стану у сполучному дереві. На третьому етапі об'єкти переміщуються в вільні термінальні стани, що залишилися. Варто зазначити, що в рамках цього підходу лише один об'єкт може здійснювати рух. Лише на заключному етапі всі об'єкти можуть рухатися одночасно, якщо це не призведе до зіткнень. Згодом метод планування дорожніх карт став найпопулярнішим підходом до планування маршрутів. На основі дорожніх карт було створено безліч різних алгоритмів планування маршрутів як для поодиноких «s-bot» так і для груп «s-bots».

Складання дорожніх карт при плануванні маршрутів для групи «s-bots» є актуальною сучасною проблемою (рисунок 1.3).

В одній із наукових праць автор «стискає» простір пошуку, розкладаючи вихідну картку на підграфи. Потім здійснюється планування маршрутів між підграфами з подальшою координацією руху усередині

підграфів. В іншій науковій роботі запроваджується поняття динамічних мереж, які формуються між об'єктами, що знаходяться в межах діапазону зв'язку. Об'єкти, що розділяють динамічну мережу, використовують спільний планувальник, що базується на ймовірнісних дорожніх картах.

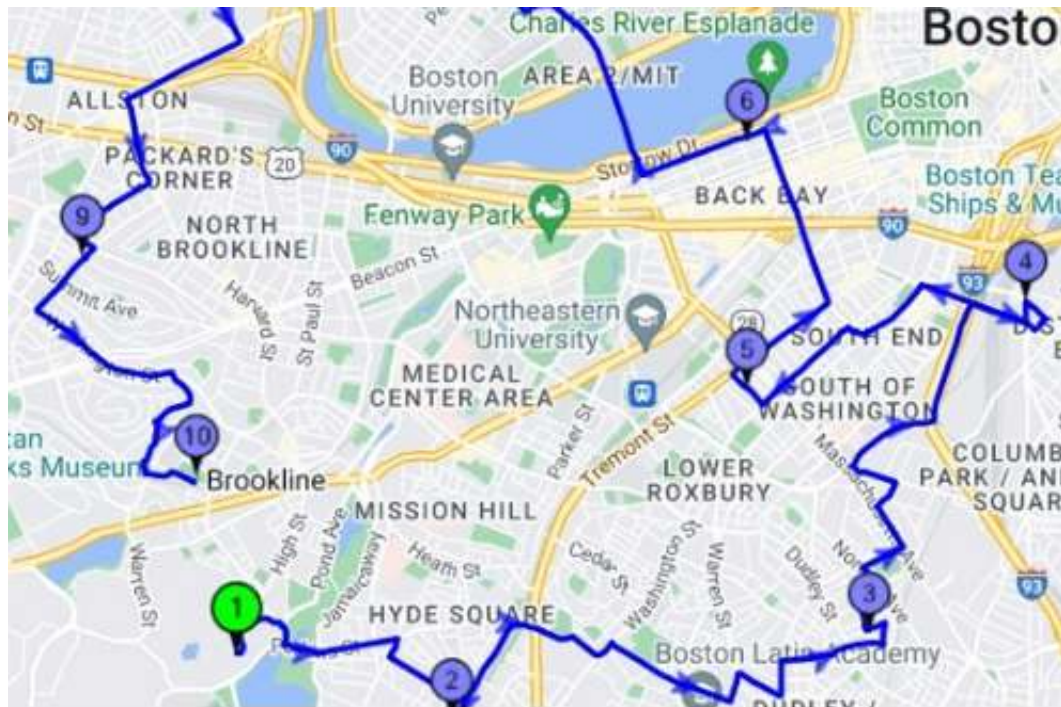


Рисунок 1.3 – Приклад дорожньої карти на основі суперграфа

В іншій науковій роботі запроваджується поняття динамічних мереж, які формуються між об'єктами, що знаходяться в межах діапазону зв'язку. Об'єкти, що розділяють динамічну мережу, використовують спільний планувальник, що базується на ймовірнісних дорожніх картах. Для виявлення змін, середовище описується мережевою картою. Щоразу, коли під час постійного оновлення мережевої карти виявляються значні зміни, ініціюється обчислення діаграми Вороного. Діаграма Вороного, що відображає схему зв'язків вільного простору, є відправною точкою в процесі вилучення дорожніх сегментів для отримання кінцевої дорожньої карти. Обчислення оптимальної конфігурації цих сегментів дозволяє визначити найкоротші маршрути та максимальну компактність мережі. Даний метод призводить до

ефективного групового планування маршрутів за невисоких обчислювальних витрат.

Незв'язані планувальники вирішують завдання планування для сукупності «s-bots» без урахування інших об'єктів групи, що дозволяє швидко знаходити прийнятні маршрути за рахунок втрати оптимальності. Незв'язні підходи поділяються на:

- централізовані;
- децентралізовані.

Незв'язний централізований планувальник передбачає самостійне, незалежне планування об'єктами маршрутів. Інтелектуальні мобільні «s-bots» в одній «Swarm-bot» - system повинні повідомляти координатору про свої наміри. Координатор, розпізнавши наміри «s-bots», приймає або відхиляє їх залежно від стану всієї чи частини сукупності об'єктів.

В основі незв'язного децентралізованого підходу лежить розподілена система, у якій кожен інтелектуальний мобільний «s-bot» прагне вирішити завдання управління у своїй зоні відповідальності. Спочатку планування маршрутів здійснюється окремо для кожного об'єкта. Під час проходження «s-bots» знайденими маршрутами застосовуються різні методи, спрямовані на запобігання та запобігання колізій. Ці методи можуть належати одному з двох класів:

- пріоритетному плануванню;
- координації маршрутів.

При плануванні маршрутів на основі пріоритетів інтелектуальні мобільні «s-bots» ранжуються за рівнем їх важливості в «Swarm-bot» - system. Потім вибираються об'єкти в порядку зменшення пріоритету і для кожного з них знаходиться маршрут, що дозволяє ухилитися від зіткнень зі статичними перешкодами, а також з раніше обраними об'єктами, які розглядаються як перешкоди, що рухаються. Планування маршрутів із найвищим пріоритетом можна здійснювати будь-яким методом одиночного планування. Особливість

пріоритетних підходів у тому, що дозволяють перейти від колективного планування до одиночного планування.

Алгоритм A^* відноситься до методів пріоритетного планування. На початковому етапі цей алгоритм використовується для пошуку оптимального маршруту для кожного інтелектуального мобільного «s-bot». Потім ці маршрути перевіряються глобальною схемою призначення пріоритетів на наявність колізій. Слід зазначити, що цей підхід не застосовується у великих групах, оскільки час обчислення зростає експоненційно залежно від кількості «s-bots» в одній «Swarm-bot» - system.

1.2 Методи потенційних полів для управління інтелектуальними мобільними «s-bots»

Методика управління інтелектуальними мобільними «s-bots» за допомогою потенційних полів вперше була запропонована А. Платоновим. Метод полів потенціалів має на увазі вплив на інтелектуальні мобільні «s-bots», що входять до складу однієї «Swarm-bot» - system векторів сили (рисунок 1.4). Його суть полягає в тому, що маршрут інтелектуальних мобільних «s-bots» будується на основі рішення спеціального рівняння, яке включає силу, що притягує «s-bots» до термінального стану, та силу, що відштовхує «s-bots» від перешкод.

Ці сили створюють для «s-bots» точки тяжіння та відторгнення у певних областях робочого середовища. Притягуючі сили наближають «s-bots» до кінцевих станів, а відштовхуючі – віддаляють їх від перешкод. У кожній точці простору «s-bot» рухається відповідно до сумарної дії відштовхувальних та притягуючих сил. Сума векторів, що притягають і відштовхують, обчислюється з деяким невеликим тимчасовим кроком, після чого «s-bot» здійснює рух у обчисленому напрямку.

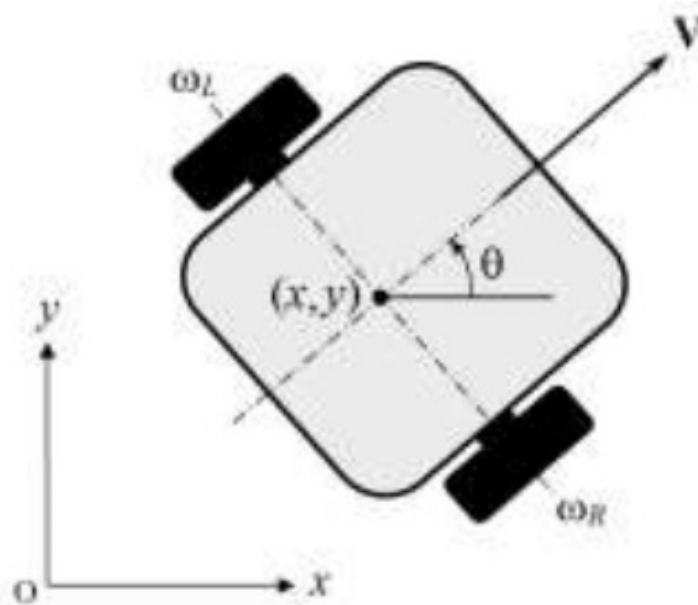


Рисунок 1.4 – Приклад впливу на «s-bots» векторів сили

Для визначення діючих на «s-bot» сил використовуються формули з галузі фізики. Вектори сили розраховуються лише за показаннями сенсорів у поточний час. Якщо перешкода знаходиться поза зоною видимості інтелектуальних мобільних «s-bot», то воно не чинить опору.

На сьогоднішній день метод полів потенціалів є одним із найпопулярніших підходів управління інтелектуальними мобільними «s-bots», що входять до складу однієї «Swarm-bot» - system, про що свідчать численні роботи, написані на цю тему. Алгоритми, засновані на методі потенціалів, можуть застосовуватися у разі, якщо контури перешкод апроксимовані опуклими багатокутниками або колами. Метод потенціалів дозволяє досить просто обчислювати напрямок руху інтелектуальних мобільних «s-bots», якого часто достатньо системі управління при плануванні переміщення «s-bots». Варто зазначити, що метод потенціалів дозволяє використати наближену інформацію про перешкоди. Імовірність влучення інтелектуальних мобільних «s-bots» у локальний мінімум є головним мінусом методів на основі полів потенціалів.

1.3 Реактивні підходи управління інтелектуальними мобільними «s-bots»

Реактивні підходи управління мають на увазі, що інтелектуальні мобільні «s-bots», що входять до складу однієї «Swarm-bot» - system, здійснюють спільні зусилля, створені задля запобігання колізій. У проведених дослідженнях вводиться достатня та необхідна умова ухилення від зіткнень «s-bots» з рухомою перешкодою, що переміщається з відомою швидкістю. Даний підхід був застосований у задачі запобігання зіткненням типу «s-bot»-to-«s-bot», що призвело до появи методу взаємних швидкісних перешкод (RVO) (рисунок 1.5).

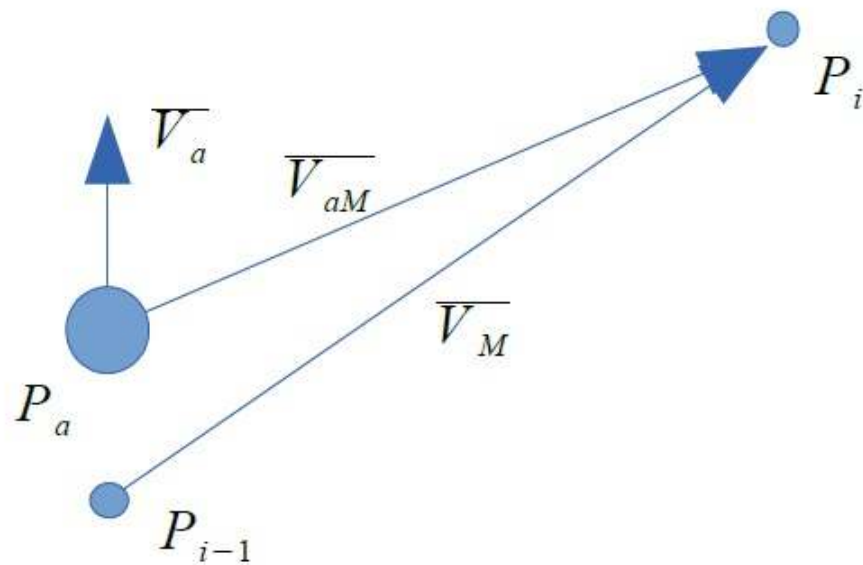


Рисунок 1.5 – Приклад роботи методу взаємних швидкісних перешкод

Метод оптимального взаємного уникнення колізій (ORCA) був розроблений з метою швидкого пошуку оптимальних швидкостей для інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system, у реальному масштабі часу. На відміну від методу взаємних швидкісних перешкод, ORCA забезпечує достатню умову уникнення

зіткнень для всіх «s-bots», що входять до складу однієї «Swarm-bot» - system (рисунок 1.6).



Рисунок 1.6 – Приклад роботи методу оптимального взаємного уникнення колізій

Для кожного «s-bot» визначаються дозволені швидкості, що забезпечують рух «s-bot» без колізій, після чого в області допустимих швидкостей знаходиться оптимальна швидкість. Варто зазначити, що алгоритм ORCA не має на увазі наявність комунікаційної мережі. Даний метод доцільно застосовувати, якщо інформація про стани, швидкості та форми інших «s-bot» має дуже високу точність. Даний метод застосовується не тільки для динамічних перешкод, що функціонують на площині (представлено тривимірну модифікацію методу ORCA для уникнення тривимірних статичних перешкод).

Група дослідників опублікувала версію алгоритму ORCA, адаптовану для реальних «s-bots», що входять до складу однієї «Swarm-bot» - system, у реальному масштабі часу. Вони пом'якшили вимоги до точності даних, що зчитуються датчиками та сенсорами. Суть цієї версії полягає в тому, що перед обчисленням оптимальних швидкостей, «s-bots» обмінюються інформацією про стани і швидкості тільки локально, не залучаючи до обміну інформацією відсторонені «s-bots». Це рішення дозволяє одночасно стримати комунікацію на деякому рівні і вирішити завдання уникнення зіткнень для двох «s-bots». Популярність ORCA була викликана двома основними причинами.

По-перше, даний метод забезпечує достатню умову уникнення зіткнення для «s-bots», що входять до складу однієї «Swarm-bot» - system.

По-друге, метод є децентралізованим і не має на увазі наявність комунікаційної мережі.

Можна виділити два основні недоліки цього сімейства підходів.

Перший недолік полягає в тому, що ці методи припускають, що дані, що зчитуються сенсорами і датчиками, є високоточними.

Другий недолік полягає в наявності великої кількості параметрів, що настраюються, від яких сильно залежить досягнення задовільних навігаційних характеристик.

1.4 Управління інтелектуальними мобільними «s-bots» на основі прогнозуючих моделей

Управління з прогнозуючими моделями (MPC) – методологія управління, заснована на вирішенні в реальному масштабі часу послідовності завдань оптимального управління з кінцевим тимчасовим горизонтом, що поєднує у собі механізм зворотного зв'язку і методи обмеженої опуклої оптимізації.

Завдання оптимального управління враховують поточні виміри станів об'єкта управління та обмеження на траєкторії та керування. Спрогнозоване управління подається на вхід об'єкта в поточний момент часу, доки не буде виміряно наступний стан об'єкта. MPC знаходить застосування у різних галузях, де на керуючі впливу накладаються обмеження.

Наприклад, у колективній роботі угруповання транспортних засобів або у завданні планування маршрутів для інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system у реальному масштабі часу (рисунок 1.7).

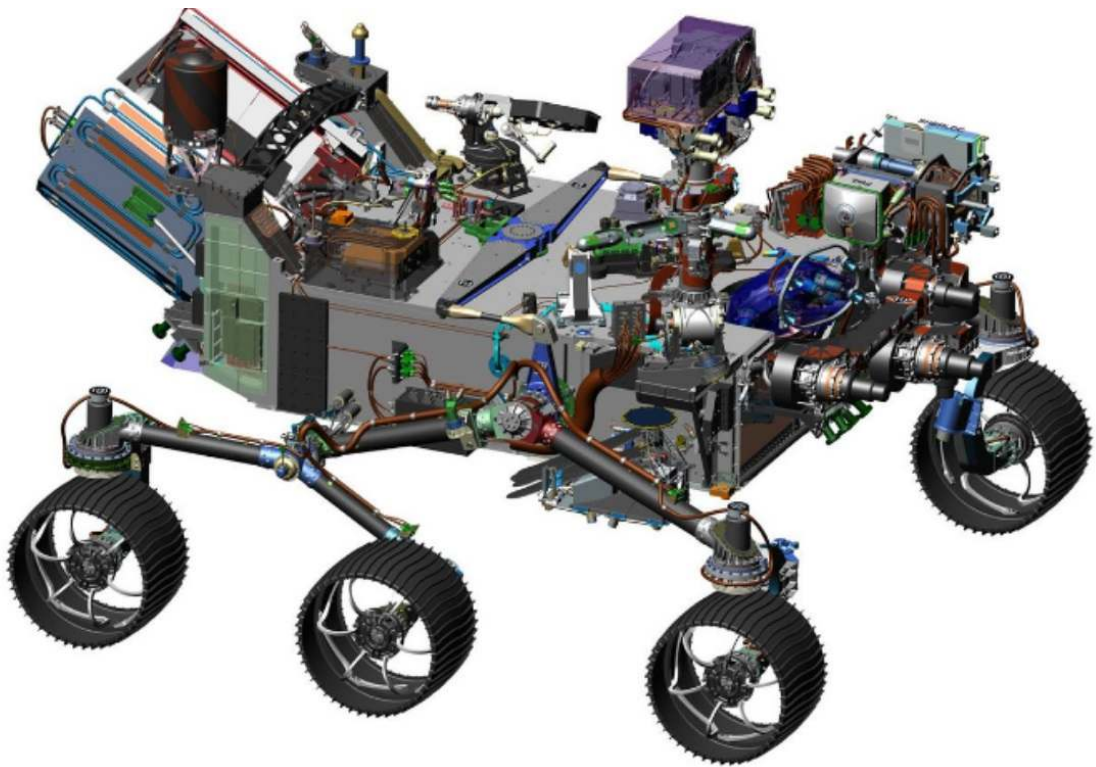


Рисунок 1.7 – Приклад інтелектуального мобільного «s-bots», яким необхідно керувати

На сьогоднішній день MPC є популярним методом пошуку управління нелінійними та багатозв'язковими системами. Можливість урахування жорстких обмежень на фазові та керуючі змінні, здатність працювати без експертного втручання протягом тривалого часу – основні переваги даної методології. Критерії якості в прогнозуємій задачі оптимального управління, дозволяють враховувати вимоги до процесу управління. Однак ці ж фактори можуть стати причиною нереалізованості управління «s-bots».

Наприклад, коли швидкість вирішення задачі оптимального управління в реальному масштабі часу не відповідає темпу оновлення інформації про стан системи. У задачі управління «Swarm-bot» - system з накладеними обмеженнями велику увагу привернула розподілена версія DMPC, яка призводить до менших обчислювальних витрат і меншого навантаження на зв'язок, ніж централізовані MPC.

Однак використання DMPC може призвести до суттєвої невідповідності між передбачуваною інформацією та фактичною, через що важко гарантувати стабільну рекурсивну роботу алгоритму. Підхід DMPC запропонований для колективного досягнення мети інтелектуальними мобільними «s-bots», що входять до складу однієї «Swarm-bot» - system, в якій накладено певні обмеження.

Науково обґрунтовано, що ітеративне вирішення проблеми сусідніх «s-bots» за один такт для кожного «s-bot» забезпечує рекурсивну реалізованість та бажану групову поведінку з гарантованою збіжністю. Цей ітеративний підхід зазвичай потребує більше часу для обчислень та накладає велике навантаження на канали зв'язку.

Передбачається, що алгоритми DMPC мають менші обчислювальні витрати і навантаження на зв'язок. Як свідчать недавні дослідження, проведені групою провідних вчених, методологія DMPC типу «лідерпослідовник» була успішно застосована в задачі керування строєм наземних та повітряних інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system.

1.5 Методи управління інтелектуальними мобільними «s-bots» на основі навчання

Колектив дослідників розробив децентралізовану методику вирішення колізій, яка не вимагає високої точності даних, що зчитуються сенсорами. Дані дослідження викликають особливий інтерес, тому що в її основі лежить проста і зрозуміла передбачаюча модель, в основі якої лежить багатосаровий персептрон.

Для навчання нейронної мережі, дослідники зібрали великий набір даних, що складається з відображення зашумлених вимірювань сенсорів у швидкості. На вхід нейронної мережі подається власний термінальний стан «s-bot» та стан, оточуючих його «s-bots».

Виходом нейронної мережі є збільшення швидкості, необхідне для безпечного запобігання зіткнень і наближення до мети. В експериментальній частині було успішно протестовано навчену нейронну мережу із зашумленими показаннями сенсорів: в ході безперервного циклу, поки «s-bot» не досяг кінцевої мети, нейронна мережа генерує збільшення швидкості і здійснює коригування руху «s-bot» (рисунок 1.8).

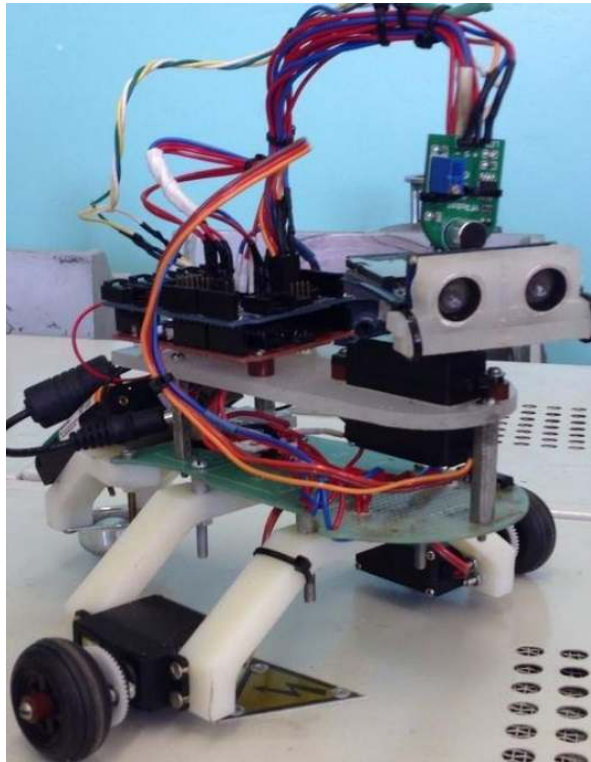


Рисунок 1.8 – Приклад інтелектуального мобільного «s-bots», який управляється нейронною мережею

«S-bots» приймають рішення самостійно, спираючись на дані, що зчитуються, незалежно від інших «s-bots». Основною проблемою даного підходу є збір точних відображень, що становлять навчальну вибірку. З метою набору навчальної вибірки розмістили «s-bots» в центр координатної площини, провели вибірку бажаної швидкості для завдання випадкового напрямку руху і генерували кілька перешкод в оточенні «s-bots» з початковими швидкостями, згенерованими випадково.

Ця процедура повторюється багато разів із різними параметрами симуляції. Варто зауважити, що збільшення швидкостей обчислюють методом оптимального взаємного уникнення колізій.

Для розширення навчальної вибірки генерують додаткові записи за допомогою накладання шуму на наявні записи, а також використовують дані сусідніх «s-bots», якщо вони успішно уникають колізії. Важливо відзначити, що пошук оптимальних швидкостей вирішується як завдання класифікації, а не регресії. Щоб визначити кількість діапазонів швидкостей як вихідних класів, дослідники вирішують завдання кластеризації методом k-середніх. Кількість результуючих кластерів дозволяє оцінити оптимальну кількість необхідних класів. У деяких випадках навчена нейронна мережа здатна уникати зіткнення із статичними перешкодами. Однак метод має низку недоліків:

- по-перше, дослідники навчили модель з точністю до 64%;
- по-друге, в навчальній вибірці не враховуються сценарії взаємодії «s-bots» зі статичними перешкодами.

Навчання з підкріпленням представляє клас алгоритмів машинного навчання, який застосовується у завданнях управління. Навчання з підкріпленням дозволяє «s-bots» отримувати винагороди під час успішної взаємодії із середовищем. Завдання координації інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system, у межах даної парадигми має формулювання марковського процесу. Кінцева мета алгоритму – навчити об'єкт оптимально реагувати на стани, щоб послідовність дій призводила до найбільшої сукупної винагороди. Один з основних принципів навчання з підкріпленням полягає в дослідженні середовища, яке чергується із застосуванням набутого досвіду. Іншими словами, тестування функції, що шукається, відбувається під час її експлуатації.

2 НЕЙРОННА МЕРЕЖА ДЛЯ УПРАВЛІННЯ ІНТЕЛЕКТУАЛЬНИМИ МОБІЛЬНИМИ «S-BOTS», ЩО ВХОДЯТЬ ДО СКЛАДУ ОДНІЄЇ «SWARM-BOT» - SYSTEM

Вперше нейронні мережі були описані у роботі А. Г. Івахненка, який запропонував метод групового обліку аргументів. Нейронні мережі – це обчислювальні моделі, що складаються з безлічі простих процесорних блоків, які називаються нейронами, які працюють паралельно і розташовані у взаємопов'язаних шарах (рисунок 2.1).

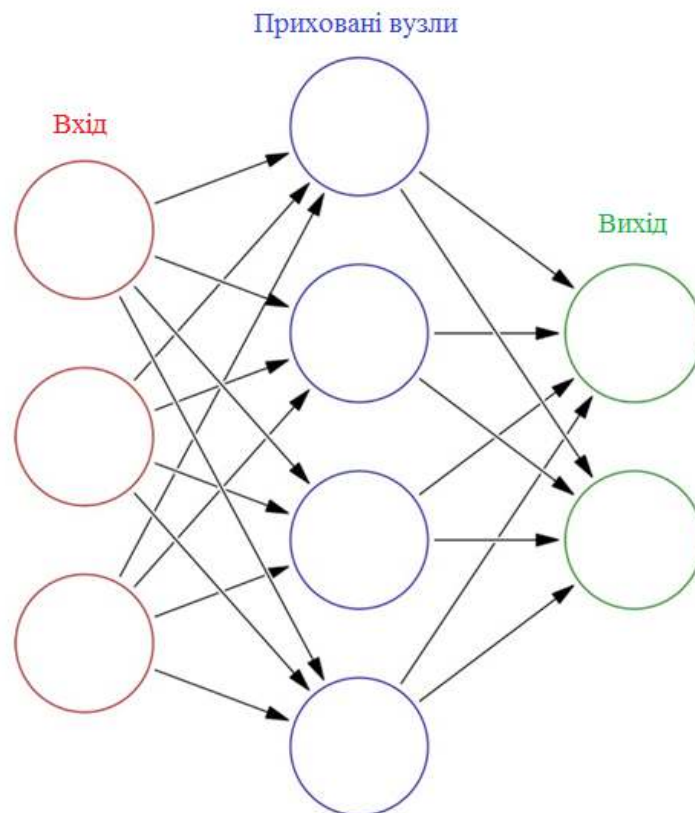


Рисунок 2.1 – Приклад архітектури нейронної мережі

Нейронні мережі навчаються виконувати певні завдання у вигляді навчання, під час якого перебуває оптимальна сила зв'язків між нейронами. В

даний час нейронні мережі успішно використовуються для вирішення прикладних завдань:

- обробки відео та аудіо сигналів;
- оптимального перемикання режимів регуляторів;
- розпізнавання тимчасових послідовностей;
- моделювання процесів пам'яті;
- кодування та декодування;
- стиснення та фільтрації інформації.

Нейронні мережі вже активно використовуються у завданнях управління.

Коли з'явилися:

- нові архітектури нейронних мереж;
- парадигми навчання;
- функції активації;
- методи оптимізації;
- регуляризації і т. д. – це призвело до появи нових програмних пакетів, що дозволяють швидко та ефективно вирішувати завдання у різних галузях.

У тому числі, з розвитком парадигми навчання з підкріпленням стало активніше досліджуватись завдання автономного управління інтелектуальними мобільними «s-bots» (рисунок 2.2).

Методи, засновані на нейронних мережах, використовувалися в задачах генерації маршрутів інтелектуальними мобільними «s-bots» в реальному часі, що забезпечують їх просування до мети без зіткнень (рисунок 2.3). Розглянемо нейромережевий метод планування та управління траєкторією руху інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system. З метою спрощення навчання запропоновано модель нейронної мережі Хопфілда для генерації траєкторії в реальному часі у нестационарному physical environment. На базі нейронної мережі група дослідників розробила систему планування маршрутів з уникненням зіткнень

між інтелектуальними мобільними «s-bots». Вони модифікували нейронну мережу Хопфілда, завдяки чому вдалося підвищити ефективність системи планування маршрутів.



Рисунок 2.2 – Приклад інтелектуального мобільного «s-bots», керованого нейронною мережею



Рисунок 2.3 – Приклад руху інтелектуального мобільного «s-bots» у реальному часі

У навчанні з підкріпленням ставиться завдання пошуку оптимальної політики, яка максимізує цільову функцію винагород. Здійснюючи дії, «s-bot» взаємодіє з *working environment*. Залежно від того, як «s-bot» взаємодіє з *working environment*, він винагороджується чи штрафується. При взаємодії з *working environment* «s-bot» повинен навчитися відтворювати оптимальні дії залежно від власного стану та стану *working environment*, щоб максимізувати сукупну винагороду.

Навчання з підкріпленням демонструє хороші результати у задачах пошуку оптимальних рішень із довгостроковим горизонтом. Підтвердженням цьому можуть бути результати алгоритмів у стратегічних іграх, у завданні управління маніпулятором і навігації.

Одна група проводила дослідження із застосуванням семантичної інформації, витягнутої із зображення за допомогою глибокої нейронної мережі. На її основі було обрано оптимальну поведінку інтелектуального мобільного «s-bot». Паралельно, інша група дослідників використовувала методи навчання з підкріпленням для навчання «s-bots» взаємодії з людьми.

На практиці *working environment* найчастіше є складним і непередбачуваним. Тому планувальник маршруту має вміти адаптуватися до змін у *working environment*. Використовуючи можливості навчання з підкріпленням, можна спроектувати планувальник таким чином, щоб «s-bot» змінював політику прийняття рішень під час взаємодії з ним. Багато дослідників звертають увагу на те, що традиційні підходи планування маршруту на основі глибоких нейронних мереж сильно залежать від ознак, які необхідно задавати вручну. Це не є ефективним підходом при обробці вхідних сигналів, що мають більшу розмірність, якими є зображення.

Деякі дослідники використовують DQN - Deep Q Networks – глибоку Q-мережу, яка апроксимує Q-функцію на основі отриманого зображення. Отримане «s-bot» RGB зображення розглядається як його поточний стан і подається на вхід DQN (рисунок 2.4).



Рисунок 2.4 – Приклад зображення, одержаного з камери інтелектуального мобільного «s-bots»

Виходом глибокої мережі є значення Q-функції, що відповідають кожній можливій дії «s-bot» (рисунок 2.5).

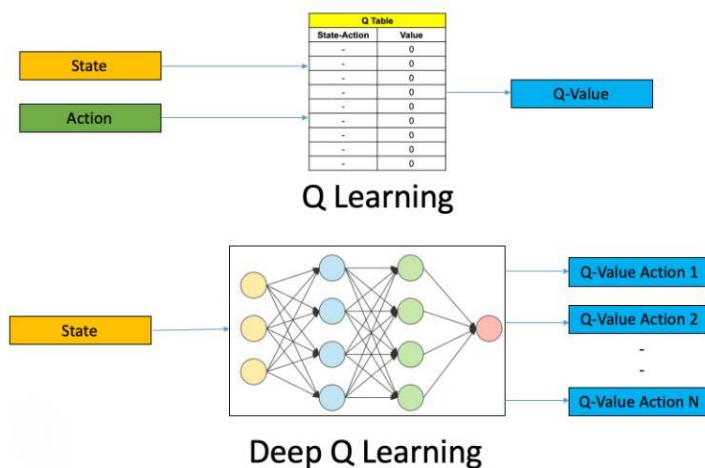


Рисунок 2.5 – Приклад глибокої Q-мережі, яка апроксимує Q-функцію на основі отриманого зображення

«S-bot» вибирає управління відповідно до стратегії вибору оптимального управління і здійснює рух у напрямку термінального стану, обминаючи перешкоди на своєму шляху.

Управління інтелектуальними мобільними «s-bots», що входять до складу однієї «Swarm-bot» - system, здійснюється у три етапи:

- етап_№1 – отримання та обробка зображення ;
- етап_№2 – отримання функції значення станів ;
- етап_№3 – вибір оптимального керування.

Етап №_1. На цьому етапі здійснюється стиск вхідного сигналу, скорочується розмірність вхідного зображення, а потім кольорове зображення переводиться в чорнобіле.

Етап №_2. На цьому етапі апроксимується Q-функція, яка проводить оцінку кожної можливої дії «s-bot» з конкретного стану і проводиться коригування вагових коефіцієнтів DQN. Виходом мережі є оцінка кожної можливої дії «s-bot».

Етап №_3. На цьому етапі вибирається оптимальна дія «s-bot».

На базі DQN були проведені дослідження:

- дослідження № 1. Було представлено підхід до створення системи навігації «s-bots», в основі якого лежить DQN, що навчається на зображеннях. У першій версії алгоритму DQN передбачає дискретну множину дій, а в другій версії алгоритму представили DQN, здатну відображати зображення середовища в дії «s-bot», що виражені безперервними величинами (рисунок 2.6);

- дослідження № 2. Були застосовані глибокі детерміновані градієнти політики, де глибокі нейронні мережі в процесі навчання виступають акторами і критиками. Було розроблено безперервну версію DQN і запроваджено нормалізовані функції переваг. Пізніше було застосовано нормалізовану функцію переваг, що успішно навчила руку «s-bot» відчиняти двері;

- дослідження № 3. Було проведено навчання з підкріпленням, що використовуються для реалізації системи навігації «s-bot». Планувальник був навчений знаходити маршрути на основі вхідних зображень physical

environment. В експериментальній частині було продемонстровано, як навчений «s-bot» переміщується до місця, вказаного на зображенні. На відміну від багатьох попередніх методів, цей підхід не вимагає 3D-моделювання physical environment. Щоб продемонструвати можливості навченого планувальника у тестовому середовищі, були побудовані дві критичні віртуальні physical environment із перешкодами різної форми. Навчання планувальника «з нуля» на одному графічному процесорі Nvidia GeForce GTX 1080 зайняло майже 20 годин. Протестували навчену модель у реальних умовах. Планувальник успішно виконав завдання проходження «s-bot» маршруту, що складається з 10 термінальних станів, без зіткнення з перешкодами. Планувальник продемонстрував високу пристосованість до невідомих physical environment. Він може бути застосований у невідомих реальних physical environment без додаткового налаштування;

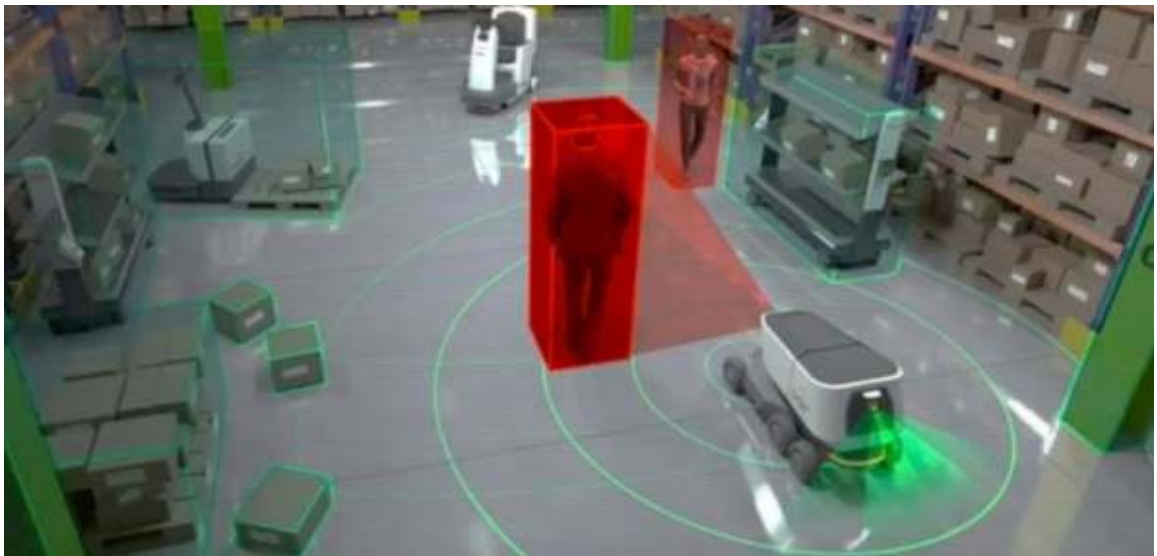


Рисунок 2.6 – Приклад системи навігації «s-bots», в основі якого лежить DQN

- дослідження № 4. Було розроблено децентралізовану систему планувальника маршрутів, що частково спостерігається, на основі методу еволюційного навчання з підкріпленням для вивчення ефективної політики

локального планування в змішаних динамічних середовищах. Даний підхід моделює поведінку динамічних перешкод за допомогою представлення на основі зображень та навчає «s-bot» оптимальної політики у змішаних динамічних physical environment. Щоб забезпечити стабільність та продуктивність навчання «s-bot», був застосований еволюційний підхід, який можна легко масштабувати для великих та складних physical environment;

- дослідження № 5. Була навчена система навігації «s-bot» у парадигмі навчання з підкріпленням. В експериментальній частині роботи продемонстровано безколізійний рух «s-bot» з початкового стану в кінцевий, без попередньої побудови карти physical environment. На вхід «s-bot» отримує візуальне відображення навколишнього physical environment у вигляді зображення. Були відзначені проблеми отримання адекватного підкріплення при взаємодії з physical environment. Показано, що навіть якщо підкріплення вдасться витягти з physical environment, то проблема запізнення сигналу, що підкріплює, і наявність шумів залишається невирішеною. З цієї причини більшість існуючих систем, заснованих на навчанні з підкріпленням, навчаються в симуляційних environment;

- дослідження № 6. Для навчання DQN на RGB- зображеннях, що надходять у реальному часі, спочатку застосовуються спеціалізовані симулятори, а потім DQN переносять на physical «s-bot». На виході DQN видає значення Q-функції, відповідні можливим управлінням. Для оптимізації алгоритму навчання переводять кольорові кадри physical environment в чорно-білі, потім скорочують розмірність, після чого останні 4 оброблені кадри об'єднуються в поточний стан physical environment для подачі на вхід DQN.

Для навігації «s-bot» з використанням навченої DQN необхідно виконати такі кроки:

- крок № 1. Отримати поточний стан physical environment (s) та позначити його за ($s = s_0$);
- крок № 2. Встановити навчену DQN для відображення станів

оптимальні дії з найбільшим значенням Q-функції, щоб оптимальна дія (a) відповідала:

$$Q(s,a) = \max_{a'} \cdot Q(s,a').$$

- крок № 3. Виконати дію (a) та отримати наступний стан s' ;
- крок № 4. Оновити поточний стан ($s = s'$) та визначити, чи є на відстані d_1 від «s-bot» що перешкоджає руху об'єкт. Якщо такий об'єкт є, припинити дії, інакше – повторити кроки № 2 та № 3.

На підставі проведених експериментів та досліджень, було розроблено наступний алгоритм навчання DQN.

Алгоритм навчання DQN:

- введення: кількість епох E, коефіцієнти дисконтування (γ), початкові вагові коефіцієнти DQN;
- виведення: оптимальні вагові коефіцієнти DQN;
- крок № 1. for $e \leftarrow 1$ to E do;
- крок № 2. Провести передобробку зображень, що описують початковий стан;
- крок № 3. Здійснити дію, вибрану за допомогою ϵ - «жадібній» стратегії;
- крок № 4. Оновити інформацію про поточний стан та, отриманий після вчинення дії підкріплення;
- крок № 5. Зберегти безліч (s_t, a_t, r_t, s_{t+1}) і випадково вибрати навчальну підвибірку. Тут s_t – поточний стан, a_t – дія, r_t – підкріплення, s_{t+1} – стан, що відповідає наступному моменту часу;
- крок № 6. Оновити ваги мережі методом стохастичного градієнтного спуску;
- крок № 7. end for.

Експерименти, проведені у симуляційному середовищі, продемонстрували здатність навченого алгоритму вибирати стратегію

управління «s-bot», яка призводить «s-bot» у заданий кінцевий стан без зіткнень зі статичними перешкодами. Після вдалого тестування навченої моделі у симуляційному середовищі було успішно проведено експерименти у реальних умовах.

Невирішеними проблемами навчання з підкріпленням є:

- досягнення «s-bots» довільних термінальних станів;
- неефективність навчальних даних (нейромережна модель часто вимагає кількох витратних, з точки зору обчислень, епізодів для закріплення в нейромережевій моделі оптимальної політики).

- одним із факторів, що ускладнюють навчання з підкріпленням, є наявність у системи ненадійної та неповної сенсорної інформації. Наприклад, «s-bot», оснащений відеокамерою, володіє неповною інформацією і тому в будь-який момент часу перебуває в частково-спостережливому стані і повинен приймати рішення з урахуванням цієї невизначеності.

3 ДОСЛІДЖЕННЯ МЕТОДУ МЕРЕЖЕВОГО ОПЕРАТОРА ДЛЯ УПРАВЛІННЯ ІНТЕЛЕКТУАЛЬНИМИ МОБІЛЬНИМИ «S-BOTS», ЩО ВХОДЯТЬ ДО СКЛАДУ ОДНІЄЇ «SWARM-BOT» - SYSTEM

3.1 Постановка задачі

Метод мережевого оператора був розроблений групою дослідників для вирішення задачі синтезу системи управління інтелектуальними мобільними «s-bots», що входять до складу однієї «Swarm-bot» - system. Цей метод належить сімейству методів символної регресії. Для отримання закодованого математичного виразу, на вхід мережевого оператора необхідно подати орієнтований ациклічний граф, в якому в якості вузлів джерел виступають аргументи та параметри функції, що шукається, а вузли строки представляють значення функції.

У проміжних вузлах графа розміщені функції двох аргументів. Дуги графа асоціюються з унарними операціями (впливають на одне значення або вираз). Для представлення графа мережевого оператора у пам'яті комп'ютера передбачено матрицю мережевого оператора. Так як граф мережевого оператора є спрямованим, а номери вузлів графа, в які входять дуги, повинні бути більшими за номери вузлів, з яких дуги виходять, то матриця мережевого оператора має верхньотрикутний вигляд. Всі елементи під головною діагоналлю дорівнюють нулю і не беруть участь у обчисленнях.

Таким чином, у матричного способу кодування математичного виразу є суттєвий недолік, який полягає в наявності великої кількості елементів, що не беруть участь у обчисленнях. При збільшенні розмірності матриці мережевого оператора кількість нульових елементів зростає нелінійно, через що робота алгоритму може сповільнитися. Для пошуку більш складних залежностей було розроблено метод вкладення матриць мережевого оператора, згідно з яким як елемент матриці виступає результат обчислення

іншої матриці мережевого оператора. Аналогічно нейронним мережам, здатним збільшувати нелінійність за допомогою додавання нових шарів, метод вкладення матриць дозволяє мережевому оператору різко «поглиблювати» структуру для пошуку складніших нелінійних зв'язків. На відміну від нейронних мереж, мережевий оператор здатний самостійно конфігурувати структуру та глибину матричних вкладень. У багатошаровому мережевому операторі різке збільшення бездіяльних елементів під час роботи еволюційного алгоритму стає особливо гострою проблемою. Були проведені дослідження з метою спробувати сформуванню нове семантичне уявлення графа математичного виразу, яке дозволяє зберігати у пам'яті комп'ютера лише необхідні елементи графа.

У вузлах графа мережевого оператора розташовані номери бінарних функцій, а номери над дугами позначають унарні операції. Відповідно до правил побудови графа з урахуванням математичного висловлювання, аргументи шуканої функції мають бути вузлами джерелами графа. Для представлення графа мережевого оператора у пам'яті комп'ютера використовується матриця мережевого оператора:

$$\Psi = [\Psi_{i,j}], i, j = \overline{1, L}, \quad (3.1)$$

де L – кількість вузлів у графі.

Діагональний елемент матриці (3.1) дорівнює нулю, якщо номер рядка дорівнює номеру одного з вихідних вузлів або він посилається на бінарну функцію. Недіагональні елементи посилаються на унарні функції. Всім елементам під головною діагоналлю надаються нулі. Загалом матриця мережевого оператора має $\frac{1}{2}(L^2 - L)$ нульових елементів. Щоб зробити кодування більш компактним, використовувалися впорядковані множини, які легко реалізуються за допомогою списків на мові програмування Python.

Завдання визначення схеми кодування математичного виразу є загальною для методів символної регресії і залежить від продуктивності алгоритму. Розглянемо таку функцію:

$$\begin{bmatrix} y_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} q_1 \sin(q_2 x_1 + q_2) \\ -\exp(q_4 x_1) + x_2^2 \end{bmatrix}. \quad (3.2)$$

Задано безліч параметрів:

$$Q = (q_1 \dots q_4), \quad (3.3)$$

безліч аргументів:

$$A = (a_1 = x_1, a_2 = x_2, a_3 = q_1, \dots, a_6 = q_4), \quad (3.4)$$

унарні функції:

$$F_1 = (\rho_1(z) = z, \rho_2(z) = z^2, \rho_3(z) = -z, \rho_4(z) = \sin(z), \rho_5(z) = \exp(z)), \quad (3.5)$$

та бінарні функції:

$$F_2 = (x_1(z_1, z_2) = z_1 + z_2, x_2(z_1, z_2) = z_1 z_2). \quad (3.6)$$

Для представлення математичного виразу (3.2) у вигляді закодованого мережевого оператора введемо тотожні операції:

$$\begin{aligned} x_1(z_1, 0) &= z_1 \\ x_2(z_1, 1) &= z_1 \end{aligned} \quad (3.7)$$

Номери, що розташовані праворуч над вузлами, позначають їх порядкові номери. Перші шість вузлів – початкові. Вони поділяються на 2 аргументи та 4 параметри. Останні два – вихідні. Всі вузли, за винятком вузлів джерел, пов'язані з бінарними функціями (3.6). Номери над дугами графа позначають унарні операції (3.5). Вихідні вузли посилаються на аргументи (3.4) функції (3.2). Матриця мережевого оператора для виразу (3.2) має такий вигляд:

$$\Psi = \begin{bmatrix} 0000110000 \\ 0000000002 \\ 0000000010 \\ 0000100000 \\ 0000001000 \\ 0000010000 \\ 0000201000 \\ 0000020300 \\ 0000001040 \\ 0000000105 \end{bmatrix}, \quad (3.10)$$

3.2 Обчислювальний експеримент

Обчислювальний експеримент проводився за допомогою імітаційного моделювання в середовищі моделювання MATLAB, що обумовлено його функціональністю. Дана «Swarm-bot» - system, що складається з інтелектуальних мобільних «s-bots» - $N = 17$, динаміка яких описується системою, формула (3.11). На управління гусеницями інтелектуальних мобільних «s-bots» накладено обмеження: $u_{i,1}, u_{i,2} \in [-350, 350]$. Задано початкові стани інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system, при яких «s-bots» розташовані у формі літери «А» (рисунок 3.2). В таблиці 3.1 представлені початкові координати «s-bots».

$$\begin{cases} \dot{x} = \frac{(u_{i,1} + u_{i,2}) \times \cos(\theta_i)}{2} \\ \dot{y} = \frac{(u_{i,1} + u_{i,2}) \times \sin(\theta_i)}{2}, \\ \dot{\theta}_i = \frac{(u_{i,3} + u_{i,4}) \times \cos(\theta_i)}{2} \end{cases}, \quad (3.11)$$

де x_i, y_i – координати центру мас «s-bot»;

θ_i – кут між віссю симетрії «s-bot» та віссю абсцис інерційної системи координат Oxy .

Таблиця 3.1 – Початкові координати «s-bots»

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
X_i	4	9	12	16	20	24	29	31	36	40	44	48	51	57	40	31	24
Y_i	4	19	24	40	58	62	79	89	78	61	53	40	28	19	30	30	30
θ_i	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

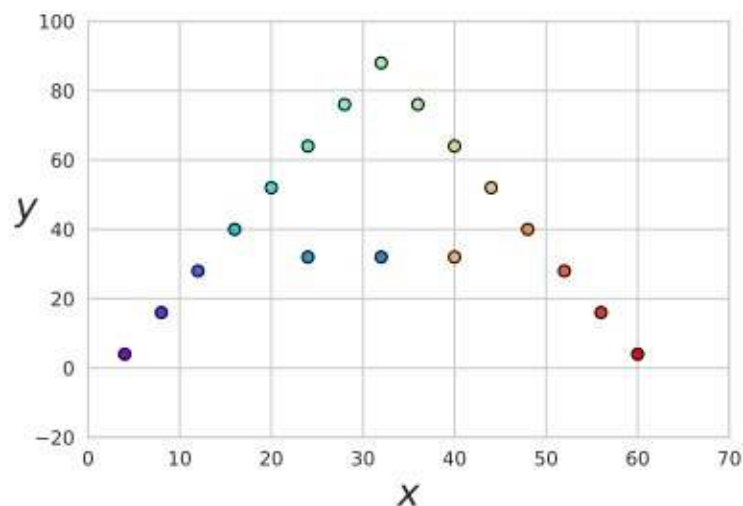


Рисунок 3.2 – Початковий стан інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system

Аналітично початкові стани інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system, описуються формулою:

$$s^i(0)=s_0^i, \quad (3.12)$$

де $i = \overline{1, N}$.

Задано кінцеві стани інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system, при яких «s-bots» розташовані у формі літери «D» (рисунок 3.3). В таблиці 3.2 представлені кінцеві координати «s-bots».

Таблиця 3.2 – Кінцеві координати «s-bots»

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
X_1	20	20	20	20	20	20	20	20	20	20	31	40	45	48	48	42	40
Y_1	10	20	28	39	45	52	60	69	78	84	80	70	64	52	44	30	21
θ_i	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

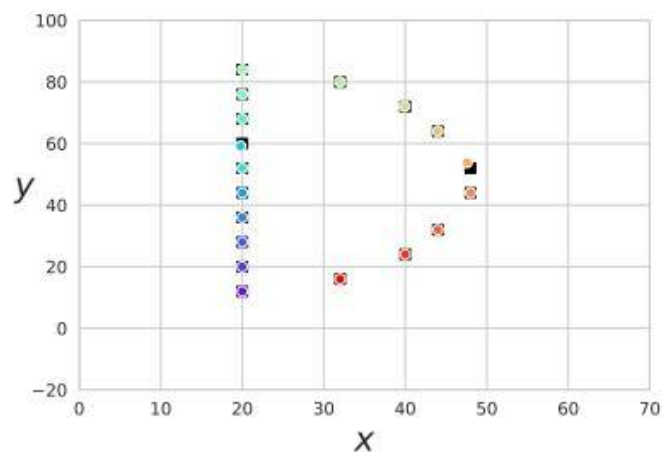


Рисунок 3.3 – Кінцевий стан інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system

Аналітично кінцеві стани інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system, описуються формулою

$$s^i(t_f) = s_f^i. \quad (3.13)$$

Вектор:

$$\Delta s^i = s^i(t) - s^i(t_f) = [x_i(t) - x_{f,i} \quad y_i(t) - y_{f,i} \quad \theta_i(t) - \theta_{f,i}]^T, \quad (3.14)$$

визначає відхилення поточного стану інтелектуального мобільного «s-bots», що входить до складу однієї «Swarm-bot» - system, від початкового стану (3.12) у момент часу $t \in [0, t_f]$ і подається на вхід системи управління, якщо не виконується умова:

$$R_0 - d_i(t) \geq 0, \quad (3.15)$$

де R_0 – радіус сфери, що задає розмір локального довкілля «s-bot»;

$d_i(t)$ – відстань між «s-bot» та перешкодою в момент часу t .

В іншому випадку на вхід системи управління подається додатковий вектор, що визначає відхилення стану «s-bot», що розглядається, від стану найближчого об'єкта перешкоди, де $s_\gamma(t)$ – положення інтелектуального мобільного «s-bots», що входить до складу однієї «Swarm-bot» - system, у момент часу t . Умова дотримання динамічних фазових обмежень задано у вигляді нерівності:

$$R^2 - \|\Delta s^i\|_2^2 < 0, \quad (3.16)$$

де R – радіус, що задає габарити «s-bots».

Необхідно виконати дії, що забезпечують переміщення інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-

bot» - system, з початкових станів (3.12) в кінцеві стани (3.13) при мінімальному значенні функціоналу та дотриманні обмежень (3.16). Початковий стан кожного інтелектуального мобільного «s-bots», що входять до складу однієї «Swarm-bot» - system фіксований і задається перед виконанням пошуку оптимальної функції управління.

Після виконання необхідних дій та обчислень відповідно до вищеописаних формул, на рисунку 3.2 представлено початковий стан групи інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system, зображених зафарбованими колами. Бажані початкові стани «s-bots» відзначені на рисунку 3.3 чорними квадратами. На цьому ж рисунку представлено фактичний кінцевий стан «s-bots». На рисунку 3.2 представлені пройдені «s-bots» маршрути.

Інтелектуальні мобільні «s-bots», що входять до складу однієї «Swarm-bot» - system, які спочатку розташовувалися у формі літери «A», успішно перемістилися та утворили форму літери «D» у межах заданої точності. Стан групи в сусідні моменти часу зображено на рисунку 3.4. Інтелектуальні мобільні «s-bots», що входять до складу однієї «Swarm-bot» - system, які досягли кінцевих станів, припиняють рух, навіть якщо в їх локальному довкіллі знаходяться інші «s-bots».

У момент часу t світло зелений «s-bots» з координатами $x = 40$, $y \approx 62$ потрапляє в локальне довкілля бірюзового «s-bots», який має координати $x \approx 35$, $y \approx 78$. З рисунку 3.4 видно, як світлозелений «s-bots» різко змінює напрямок руху. Після ухилення від зіткнення світлозелений «s-bots» продовжує рух у напрямку свого кінцевого стану.

Метод уникнення колізій із ситуативним включенням вектора стану найближчого «s-bot» у функцію управління розглянутого «s-bot», успішно досліджено у зв'язці з методом мережного оператора в задачі управління інтелектуальними мобільними «s-bots», що входять до складу однієї «Swarm-bot» - system.

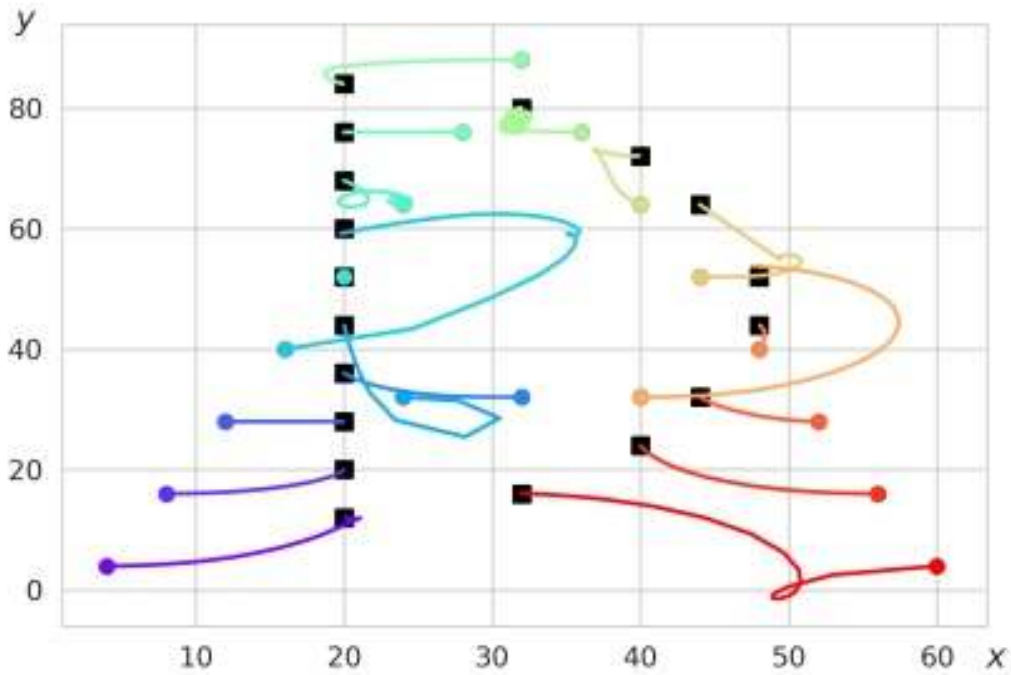


Рисунок 3.4 – Траєкторії руху «s-bots»

Під час проведення досліджень було підтверджено, що з перекладу «s-bots» у кінцеві стани з дотриманням накладених обмежень можна здійснювати пошук адаптивних функцій управління, які враховують стан перешкод у реальному масштабі часу. Тому метод мережевого оператора може успішно застосовуватися в якості планувальника, здатного здійснювати коригування маршрутів інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system при загрозі зіткнень з іншими «s-bots».

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було досліджено можливість застосування методу мережевого оператора для управління інтелектуальними мобільними «s-bots», що входять до складу однієї «Swarm-bot» - system.

Було успішно досліджено метод мережевого оператора в задачі управління інтелектуальними мобільними «s-bots», що входять до складу однієї «Swarm-bot» - system. При проведенні досліджень було підтверджено, що для досягнення «s-bots» кінцевого стану з дотриманням накладених обмежень можна здійснювати пошук адаптивних функцій управління, що враховують стан перешкод у реальному масштабі часу. Тому метод мережевого оператора можна успішно застосовувати як планувальник, який здатний здійснювати коригування маршрутів інтелектуальних мобільних «s-bots», що входять до складу однієї «Swarm-bot» - system при загрозі зіткнень з іншими «s-bots».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ткачов В.М., Токарев В.В., Радченко В.О., Лебедев В.О. Проблема передачі даних типу BIG DATA у мобільній системі «Мультикоптер-сенсорна мережа» / В.М. Ткачов, В.В. Токарев, В.О. Радченко, В.О. Лебедев // Системи управління, навігації та зв'язку. - 2017. №2(42). - С.154-157
2. Пат. 118921 Україна, МПК H04W 64/00. Спосіб передачі цифрових даних мультикоптерною системою між сегментами розподіленої сенсорної мережі та базовою станцією / В.М. Ткачов, В.В. Токарев - № u201704085; заявл. 24.04.2017; опубл. 28.08.2017. Бюл. № 16. 5с.
3. Радченко В.А., Токарев В.В., Ткачев В.М. Мобільна система передачі даних на базі динамічно реконфігурованих мультикоптерних пристроїв / В.О. Радченко, В.В. Токарев, В.М. Ткачов // Проблеми інформатизації: тези доповідей V - наук. - техн. конф., 13 - 15 лист. 2017р. - Харків, 2017. - С.36.
4. Створення науково-методичних основ забезпечення живучості мережевих систем обміну інформацією в умовах зовнішнього впливу потужного НВЧ випромінювання: звіт про НДР (заключ.) № держреєстрації 0117U003916.: Ф76/109-2017 / Харків. нац. ун-т радіоелектроніки; керівник Г. И. Чурюмов. – Харків, 2017. – 116 с.
5. Ruban I.V., Churyumov G.I., Tokarev V.V., Tkachov V.M. Provision of Survivability of Reconfigurable Mobile System on Exposure to High-Power Electromagnetic Radiation / I.V. Ruban, G.I. Churyumov, V.V. Tokarev, V.M. Tkachov // Selected Papers of the XVII International Scientific and Practical Conference on Information Technologies and Security: (ITS 2017). CEUR Workshop Processing., 30 nov. 2017 y. - Kyiv, 2017. - P. 105-111.
6. Serkov A., Kravets V., Yakovenko I., Churyumov G., Tokariev V., Nannan W. Ultra Wideband Signals in Control Systems of Unmanned Aerial Vehicles / A. Serkov, V. Kravets, I. Yakovenko, G. Churyumov, V. Tokariev, W.

Nannan // The 10h IEEE International Conference on Dependable Systems, Services and Technologies: (DESSERT'2019)., 5-7 june, 2019 y. - Leeds, 2019. - P.26 - 29.

7. Серков О. А., Пустовойтов П. Є., Яковенко І. В., Лазуренко Б. О., Чурюмов Г. І., Токарев В. В., Наннан Ванг. Надширокосмугові технології в системах управління мобільними об'єктами. / О. А. Серков, П. Є. Пустовойтов, І. В. Яковенко, Б. О. Лазуренко, Г. І. Чурюмов, В. В. Токарев, Ванг Наннан // Сучасні інформаційні системи. - 2019. - Т.3, №2. - С.22-27.

8. Серков О.А., Князев В.В., Лазуренко Б.О., Яковенко І.В., Чурюмов Г.І., Токарев В.В. Надширокосмугові технології в задачах забезпечення електромагнітної сумісності рухомих об'єктів / О.А. Серков, В.В. Князев, Б.О. Лазуренко, І.В. Яковенко, Г.І. Чурюмов, В.В. Токарев // Проблеми електромагнітної сумісності перспективних бездротових мереж зв'язку (ЕМС-2019):збірник наукових робіт четвертої міжн. наук.-техн. конф., 24 жовт. 2019 р. - Харків, 2019. - С. 55-57.

9. Krivoulya G., Tokariev V., Ilina I, Shcherbak V. Mathematical Model for Finding Probability of Detecting Victims of Man-Made Disasters Using Distributed Computer System with Reconfigurable Structure and Programmable Logic / G. Krivoulya, V. Tokariev, I. Ilina, V. Shcherbak // IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology: (PIC S&T), 06-09 oct. 2020y. - Kharkiv, 2020. - P.197 - 201.

10. Krivoulya G., Tokariev V., Ilina I., Lebediev O., Shcherbak V. Algorithm of Iterations of Distribution of Subtasks Between «S-Bot» in One «Swarm-Bot» System // Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Systems: (COLINS 2022). CEUR Workshop Proceedings., 12-13 may. 2022 y. - Gliwice, Poland. - P. 1531-1541.

11. Koshevoy N., Ilina I., Tokariev V., Malkova A., Muratov V. Implementation Of The Gravity Search Method For Optimization By Cost Expenses Of Plans For Multifactorial Experiments // Radioelectronic and Computer Systems. – 2023. Vol. 1(105). - P. 23-32. Doi: 10.32620/reks.2023.1.02

(Scopus).

12. Кривуля Г.Ф., Токарев В.В., Ільїна І.В., Кравець В.Є. Взаємодія між «s-bots» однієї «Swarm-bot» system у фізичному неорганізованому середовищі. // Системи управління, навігації та зв'язку. - 2023. №1(71). - С.108-111. Doi: 10.26906/SUNZ.

13. Krivoulya G., Koshevoy N., Tokariev V., Ilna I., Dubinsky D. Solving the Task of Topological Formation Intelligent Mobile «S-bots» for One «Swarm-bot» System // Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems: (COLINS 2023). CEUR Workshop Proceedings., 20-21 april. 2023 y. - Kharkiv, Ukraine. - pp. 273-282.

14. Токарев В.В., Вітренко В.С. Метод мережевого оператора для управління «Swarm-bot» - system / В.В.Токарев, В.С. Вітренко // Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення: збірник тез доповідей, 20 вер. 2023р. - Тернопіль: 2023. - випуск 80. - С.82 - 83.