

УДК 004.45:004.89

*Дейнеко А.О., здобувач**Духновська К.К., к.т.н.**Державний університет «Житомирська політехніка»*

СИСТЕМА РЕАЛЬНОГО ЧАСУ ДЛЯ КЕРУВАННЯ АВТОНОМНИМ АГЕНТОМ

Інтенсивний розвиток концепції Internet of Everything (IoE) зумовлює необхідність створення надійних та швидкодіючих програмних інтерфейсів для віддаленого управління автономними пристроями. Традиційні підходи до побудови таких систем часто спираються на архітектуру REST та класичний протокол HTTP, що створює значний мережевий оверхед при полінгу (polling) і не забезпечує достатньої швидкодії для передачі телеметрії мобільних об'єктів у режимі реального часу. Відповідно, актуальним завданням є розробка клієнт-серверних рішень на базі повнодуплексних протоколів зв'язку, інтегрованих з ефективними алгоритмами навігації.

Метою даної роботи є розробка програмного комплексу, який складається з інтерактивної веб-панелі керування та бекенд-сервера, здатного розраховувати оптимальні траєкторії руху інтелектуального агента з урахуванням динамічних перешкод та безперервно транслювати його стан.

Запропонована система базується на розподіленій клієнт-серверній архітектурі. Серверна частина реалізована мовою Python з використанням мікрофреймворку Flask [1] та бібліотеки Eventlet для асинхронної обробки з'єднань. Стан автономного агента в будь-який момент часу формалізується вектором стану S :

$$S = \{x, y, B, status\},$$

де x, y — поточні координати об'єкта на дискретній сітці, B — рівень заряду акумулятора у відсотках, $status$ — поточний стан виконання задачі. Двосторонній обмін вектором S між сервером та клієнтом здійснюється через протокол WebSocket у форматі JSON із частотою оновлення 2 Гц, що забезпечує консистентність даних [2].

Клієнтська частина побудована на базі JavaScript-фреймворку Vue.js [3]. Візуалізація робочого простору у вигляді карти зайнятості (Occupancy Grid Map) реалізована за допомогою технології HTML5 Canvas. Це дозволяє оптимізувати рендеринг двовимірної сітки без переваження DOM-дерева браузеру.

Базовим алгоритмом пошуку найкоротшого шляху обрано алгоритм A^* (A — Star). На кожній ітерації алгоритм розширює вузол сітки n , який мінімізує цільову функцію $f(n)$ [4]:

$$f(n) = g(n) + h(n),$$

де $g(n)$ — фактична вартість пройденого шляху від старту, а $h(n)$ — евристичне наближення відстані до цілі. Для забезпечення руху агента у восьми напрямках (включаючи діагональні переміщення) як евристику застосовано евклідову метрику:

$$h(n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2}.$$

Використання структури даних «бінарна купа» (binary heap) для зберігання відкритого списку вершин дозволило досягти часової складності алгоритму $O(E \log V)$, де V — кількість вільних клітинок, а E — кількість можливих переходів. Особливістю реалізованого навігаційного модуля є механізм динамічного перепланування маршруту. У разі зміни топології карти оператором безпосередньо під час руху об'єкта, система автоматично перевіряє валідність залишку шляху і, за необхідності, ініціює розрахунок нової траєкторії в обхід перешкоди. Додатково в системі імплементовано кінематичну модель енергоспоживання об'єкта. Зміна рівня заряду ΔB при переміщенні між суміжними вузлами n_i та n_{i+1} розраховується за формулою:

$$\Delta B = k * d(n_i, n_{i+1}),$$

де k — базовий коефіцієнт енергоспоживання, а d — просторова відстань між вузлами. Таким чином, діагональні кроки мають більшу вартість $\approx 1,414$ і відображаються у пропорційно більших витратах ресурсу. При досягненні критичного рівня заряду алгоритм забезпечує автоматичне повернення агента на координати базової станції для підзарядки.

Розроблений програмний комплекс успішно демонструє можливість ефективного управління автономним агентом у реальному часі через веб-інтерфейс. Подальший розвиток проєкту передбачає розширення серверної архітектури для забезпечення мультиагентної взаємодії (Swarm Robotics) та інтеграцію алгоритмів оптимізації траєкторій.

Список використаних джерел:

1. Grinberg M. Flask Web Development: Developing Web Applications with Python. – O'Reilly Media, 2018. – 316 p.
2. The WebSocket API (WebSockets) [Електронний ресурс] // MDN Web Docs. – URL: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
3. Vue.js: The Progressive JavaScript Framework [Електронний ресурс] // Офіційна документація Vue.js. – URL: <https://vuejs.org/>
4. Hart P. E., Nilsson N. J., Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths // IEEE Transactions on Systems Science and Cybernetics. – 1968. – Vol. 4, No. 2. – P. 100–107.