

УДК [004.41+42]:681.5

DOI <https://doi.org/10.32782/IT/2025-1-29>

Владислав ХАРЧУК

аспірант кафедри інформаційних технологій та комп'ютерної інженерії, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005
ORCID: 0009-0001-1648-7541

Сергій ТКАЧЕНКО

кандидат технічних наук, доцент кафедри інформаційних технологій та комп'ютерної інженерії, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005
ORCID: 0000-0003-1156-3151
Researcher ID: AAI-7727-2020

Бібліографічний опис статті: Харчук, В., Ткаченко, С. (2025). Прогнозування руху роботизованої мобільної платформи у двовимірному просторі з урахуванням динамічних перешкод. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 218–225, doi: <https://doi.org/10.32782/IT/2025-1-29>

ПРОГНОЗУВАННЯ РУХУ РОБОТИЗОВАНОЇ МОБІЛЬНОЇ ПЛАТФОРМИ У ДВОВИМІРНОМУ ПРОСТОРІ З УРАХУВАННЯМ ДИНАМІЧНИХ ПЕРЕШКОД

Дана робота присвячена розробці методу прогнозування маршруту мобільного роботу та моделювання середовища з використанням мови Python.

Мета роботи. Розробка методу прогнозування траєкторій мобільних роботів у середовищі з динамічними перешкодами, забезпечуючи адаптивну маршрутизацію в реальному часі та уникнення зіткнень.

Методологія. Дослідження базується на математичному моделюванні руху мобільної платформи в двовимірному просторі за допомогою модифікованого алгоритму D^* Lite. Було використано підхід до моделювання, який представляє середовище як дискретну сітку, що включає як статичні, так і динамічні перешкоди. Статичні перешкоди імітують фіксовані об'єкти, такі як стіни чи інші нерухомі елементи, тоді як динамічні перешкоди визначаються їх мінливим положенням, швидкістю та напрямком руху. Для прогнозування шляхів об'єктів на основі їх поточного стану використовується векторне представлення руху. Зокрема, аналіз зосереджується на таких параметрах, як положення, швидкість і напрямок руху. Запропонований статистичний критерій передбачає вибір стабільного маршруту шляхом вивчення частоти вершин у діапазоні потенційних траєкторій.

Наукова новизна. Удосконалено метод планування шляху мобільного робота в динамічному середовищі завдяки ретельному оцінюванню як статичних, так і динамічних перешкод. Математичну структуру, яка використовується для прогнозування руху об'єктів, було розширено за допомогою інкрементального методу D^* Lite. Крім того, введено новий статистичний критерій для вибору оптимальної траєкторії, який враховує стабільність шляху.

Висновки. Розроблена методика дозволяє здійснювати навігацію мобільної платформи у двовимірному просторі з наявністю рухомих перешкод. Він пропонує адаптивне планування траєкторії, реагування на зміни положення перешкод з метою уникнення зіткнень. Програмне моделювання підтвердило здатність алгоритму обчислювати маршрути в реальному часі. Були протестовані сітки різних розмірів: 20×20 , 50×50 , 75×75 , 100×100 . Продовження досліджень має здійснюватись у напрямку обґрунтування оптимального розміру сіток відображення середовища мобільної платформи, розміру комірки сітки з точки зору необхідного часу реакції та точності позиціювання навігаційною системою.

Ключові слова: навігаційна система, векторне представлення руху, моделювання середовища, прогнозування траєкторій, пошук оптимального маршруту, мобільні роботи.

Vlad KHARCHUK

Postgraduate Student at the Department of Information Technologies and Computer Engineering, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave., Dnipro, Ukraine, 49005, vharcuk02@gmail.com
ORCID: 0009-0001-1648-7541

Serhii TKACHENKO

Candidate of Technical Sciences, Associate Professor at the Department of Information Technology and Computer Engineering, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave., Dnipro, Ukraine, 49005

ORCID: 0000-0003-1156-3151

Researcher ID: AAI-7727-2020

To cite this article: Kharchuk, V., Tkachenko, S. (2025). Prognozuvannya ruhu robotizovanoj mobilnoj platformy u dvovymirnomu prostori z urahuvannjam dinamichnykh pereshkod [Prediction of robotic mobile platform motion in two-dimensional space with consideration of dynamic obstacles]. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 218–225, doi: <https://doi.org/10.32782/IT/2025-1-29>

PREDICTION OF ROBOTIC MOBILE PLATFORM MOTION IN TWO-DIMENSIONAL SPACE WITH CONSIDERATION OF DYNAMIC OBSTACLES

This work is devoted to the development of a method for predicting the route of a mobile robot and modeling the environment using the Python language.

Objective. Development of a method for predicting mobile robot trajectories in an environment with dynamic obstacles, ensuring adaptive real-time routing and collision avoidance.

Methodology. The research is based on mathematical modeling of mobile platform movement in two-dimensional space using a modified D* Lite algorithm. The modeling approach represents the environment as a discrete grid that includes both static and dynamic obstacles. Static obstacles simulate fixed objects such as walls or other immobile elements, while dynamic obstacles are defined by their changing position, speed, and direction of movement. Vector representation of motion is used to predict object paths based on their current state. In particular, the analysis focuses on parameters such as position, velocity, and direction of movement. The proposed statistical criterion involves selecting a stable route by examining the frequency of vertices within the range of potential trajectories.

Science novelty. The method of mobile robot path planning in a dynamic environment has been improved through careful evaluation of both static and dynamic obstacles. The mathematical structure used for predicting object movement has been expanded using the incremental D* Lite method. Additionally, a new statistical criterion for optimal trajectory selection has been introduced, which takes into account path stability.

Conclusions. The developed methodology enables navigation of a mobile platform in two-dimensional space with moving obstacles. It offers adaptive trajectory planning, responding to changes in obstacle positions to avoid collisions. Software modeling confirmed the algorithm's ability to compute routes in real-time. Different grid sizes were tested: 20×20 , 50×50 , 75×75 , 100×100 . Further research should be conducted in the direction of justifying the optimal size of environment mapping grids for the mobile platform, grid cell size in terms of required reaction time and positioning accuracy by the navigation system.

Key words: navigation system, vector representation of motion, environment modeling, trajectory prediction, optimal route search, mobile robots.

Актуальність проблеми. Розвиток порівняно простої, але разом з тим доступної елементної бази малих контролерів, мікрокомп'ютерів, мікровідеокамер та інших пристроїв, призначених для роботи у побутовому чи офісному середовищі, дозволяє інакше поглянути на проблеми обслуговування закритих громадських локацій. Так, під час презентації часто виникає потреба у рухомому столі для оперативного надання учасникам роздаткового матеріалу, а доповідачу – засобів керування інтерактивним обладнанням. Іноді відвідувач може побажати замовити собі води, кави, канцелярського знаряддя. Може виникнути потреба проведення фото чи відеозйомки зі змінною точкою спостереження, чи просто непомітно прибрати дрібне сміття, і таке інше. Різноманітність задач, що може вирішувати малий кіберфізичний об'єкт у закритому громадському просторі, досить широка, але його

можливості завжди обмежені рухомою платформою, що має переміщувати цільове обладнання в межах закритої локації з рухомими й нерухомими перешкодами. Таким чином, існує проблем безпечної навігації кіберфізичних об'єктів в умовах закритої локації з людьми.

Аналіз останніх досліджень і публікацій. Проблематика знайшла своє відображення в наукових працях вітчизняних та зарубіжних учених, які приділяли увагу вивченню проблем навігації з різних методологічних позицій. У роботі (Ганенко, 2023, с. 93) було наведено алгоритми планування шляху МР, які діляться на: класичні, евристичні та еволюційні. З них був обраний евристичний підхід для цього підходу потрібно знати наперед середовище та стартову та кінцеву точки розташування робота. Тому було вирішено повести аналіз методів у наступних роботах.

У роботі (Gupta, 2016, с. 122) було розглянуто алгоритм Дейкстри – це алгоритм пошуку, який обчислює найкоротший шлях з одного джерела для графа з невід’ємною вартістю ребер, створюючи дерево найкоротшого шляху. На жаль алгоритм працює в середовищі яке не враховує рухомі перешкоди, що не підходить цілям роботи. Тому було розглянуто генетичний алгоритм, який використовується для побудови шляху мобільного роботу та базується на принципі природного добору, оптимізуючи шлях через селекцію, схрещування та мутації (Nagib, 2005, с. 186). Наведений алгоритм наводить інший підхід по вирішенню проблеми побудови шляху але не призначений для прогнозу шляху переміщення в динамічному середовищі.

У наступній роботі досліджено поєднання алгоритмів GA та A*, для знаходження оптимального шляху. В якому генетичний алгоритм використовується для генерації проміжних вузлів, а A* для відстеження шляху (Thangamuthu, 2023, с. 186). Розглянутий метод пришвидшує розрахунок ключових для пошуку шляху, але немає реалізації з врахуванням динамічних об’єктів. Наступною була розглянута модифікація алгоритму A* для пошуку оптимального маршруту такі як Basic Theta*, Phi*, Rectangular Symmetry Reduction (RSR), Jump Point Search (JPS) (Duchon, 2014, с. 60). У наведеній роботі описано модифікацію для пошуку оптимального маршруту. Однак метод обмежується застосуванням в середовищі зі статичними об’єктами, що не дозволяє його використання в прогнозуванні шляхів з рухомими перешкодами.

Алгоритми слідування по шляху, такі як алгоритми на основі A-star або алгоритм Дейкстри, представляють ще один клас традиційних методів, що направляють роботів вздовж заздалегідь визначеними шляхами (Kretzschmar, 2016). Ці алгоритми слідування за маршрутом ефективні у фіксованому середовищі, але можуть стикатися з проблемами при адаптації до динамічних перешкод і мінливих умов (Roberts, 2016).

Недоліком описаних методів є те що вони працює у середовищі де немає рухомих перешкод, що не цілком відповідає меті дослідження.

Більш перспективним з точки зору використання в умовах рухомих перешкод алгоритм D* Lite (Koenig, 2002, с. 479), який дозволяє урахувати рухомі та статичні перешкоди, але може використовуватись багаторазово. Ця властивість може бути застосована для прогнозування траєкторії руху мобільної платформи, якщо його використати в навігаційному алгоритмі. Навігаційний алгоритм мобільної платформи за рахунок властивості прогнозування траєкторії руху

здатен продемонструвати кращу адаптивність в основних сценаріях.

Мета дослідження. Розробка методу прогнозування можливих траєкторій руху мобільного роботу в середовищі з статичними та динамічними перешкодами з ціллю подальшого вибору оптимального маршруту, що забезпечить безпечну адаптивну навігацію у закритих громадських локаціях.

Виклад основного матеріалу. Математична модель D* Lite лежить в основі планування шляху та базується на інкрементальному пошуку найкоротшого маршруту. Використання евристичної функції для оцінки відстані дозволяє швидко знаходити оптимальні рішення, навіть у складних умовах. Алгоритм підтримує динамічне оновлення вартості шляху, при виявленні нових перешкод, що є ключовим для адаптації до змін у середовищі (Koenig, 2002, с. 481).

Алгоритм D* Lite для динамічного планування шляху використовує кілька ключових формул для обчислення вартості шляху та корекції маршруту в реальному часі. Основні формули, що застосовуються в алгоритмі, включають:

1. Обчислення ключа вузла:

Ключ для кожного вузла S визначається як k :

$$k = \min(g(n), h(n)), \quad (1)$$

де $g(n)$ – це вартість шляху до вузла n , а $h(n)$ – евристична оцінка вартості до цільового вузла.

2. Оновлення значення g :

Значення g для вузла оновлюється за формулою:

$$g(n) = \min_{n' \in S} (g(n') + c(n', n)), \quad (2)$$

де S – сусідні вузли, а $c(n', n)$ – вартість переходу від вузла n' до вузла.

3. Евристична функція:

Для обчислення евристики використовується манхеттенська відстань:

$$h(n) = |x_n - x_{goal}| + |y_n - y_{goal}|, \quad (3)$$

де (x_n, y_n) – координати поточного вузла, (x_{goal}, y_{goal}) – координати цільового вузла.

4. Оновлення значення rhs :

Значення rhs для цільового стану ініціалізується як нуль:

$$rhs(goal) = 0.$$

Для всіх інших вузлів:

$$rhs(n) = g(n)$$

Для прогнозування руху динамічних об’єктів в середовищі, система використовує

розвинений математичний апарат. В основі лежить векторне представлення положення об'єктів

Особливе значення в розвитку динаміки точкових частинок (і систем частинок) має представлення руху цих частинок векторами. Для такого підходу нам потрібні вектори для відображення положення, швидкості та прискорення даної частинки (Stephen T, 2004, с. 30).

Вектор положення $r(t)$ описує координати об'єкта в просторі в залежності від часу t . У двовимірному просторі він може бути представлений як:

$$r(t) = [x(t), y(t)], \quad (4)$$

де: $x(t)$ – координата по осі X ; $y(t)$ – координата по осі Y .

Вектор швидкості $v(t)$ визначає, як швидко змінюється положення об'єкта з часом. Він є похідною від вектора положення:

$$\begin{aligned} v(t) &= \frac{d}{dt} \cdot r(t) = [v_x(t), v_y(t)]; \\ v_x(t) &= \frac{dx(t)}{dt}; \\ v_y(t) &= \frac{dy(t)}{dt}, \end{aligned} \quad (5)$$

де: $v_x(t)$ – швидкість по осі X ; $v_y(t)$ швидкість по осі Y .

Прогнозування траєкторій руху динамічних перешкод відбувається на основі поточного стану об'єктів та законів їх руху з урахуванням фізичних обмежень: максимальної швидкості, можливих напрямків руху, інерційності об'єктів.

$$\begin{aligned} x(t) &= x_0 + v_x(t) \cdot t; \\ y(t) &= y_0 + v_y(t) \cdot t, \end{aligned} \quad (6)$$

При досягненні перешкодами меж робочого простору відбувається автоматичне коригування їх траєкторій для забезпечення продовження руху в межах сітки.

Продовжуючи вивчати способи розв'язання задачі планування траєкторії руху робота, було вирішено впровадити додаткові механізми прогнозування траєкторій руху, що враховують фізичні параметри об'єкта і можливі сценарії руху.

Для опису запропонованої модифікації алгоритму були використані такі методи та формули.

GetShortestPath() – метод обраховує найкоротший шлях за допомогою D* lite алгоритму.

Для отримання множини всіх шляхів

$$P = \{GetShortestPath(i), i \in N, 1 < i < n\}, \quad (7)$$

FindOptimalPath() – метод прогнозує оптимальний шлях на основі множини шляхів.

Запропонований опис алгоритму:

- нехай $P = \{p_1, p_2, \dots, p_n\}$ множина всіх шляхів;
- кожен шлях $p_i = (v_1, v_2, \dots, v_k)$, де v позначає вершини;
- $L = \min\{|p| : p \in P\}$ – довжина найкоротшого шляху;
- для кожної позиції i від 0 до $L - 1$ створюємо словник частот повторень вершин D_i , де:

$$D_i(v) = |\{p \in P : p[i] = v\}|, \quad (8)$$

Це означає, що $D_i(v)$ підраховує, скільки разів вершина v зустрічається в i -тій позиції серед усіх шляхів. Для кожної позиції i -тої знаходимо найчастішу вершину:

$$m = \arg \max\{D_i(v) : v \in V\}, \quad (9)$$

де V – множина всіх вершин на i -тій позиції.

Для кожного шляху $p \in P$ обчислюємо його оцінку співпадінь $M(p)$:

$$M(p) = |\{v \in p : v \in \{m_1, m_2, \dots, m_k\}\}|, \quad (10)$$

Підрахунок (10) визначає кількості вершин у шляху p , які співпадають з найчастішими вершинами на відповідних позиціях.

Зрештою, виберемо шлях з максимальною кількістю співпадінь:

$$\text{path_with_most_repeats} = \arg \max\{M(p) : p \in P\}.$$

Якщо кілька шляхів мають однакову максимальну оцінку співпадінь, алгоритм вибирає перший з них. Схема алгоритму наведена на рисунках 1, 2.

Фундаментальною основою системи навігації є створення структурованого дискретного простору у вигляді сітки відображення середовища розміром 20×20 клітинок зображено на рисунку 3.

Кожна комірка сітки відображення середовища виступає як окремий його елемент, що може характеризуватися різними станами: вільний простір, зайнятий статичною перешкодою, динамічна перешкода, початкова позиція робота чи цільова точка. Така дискретизація простору дозволяє ефективно моделювати реальне середовище та спрощує процес планування маршруту. Початкова та цільова позиції робота позначена червоною та зеленою точкою на сітці відповідно. Нерухомі перешкоди позначені коричневими комірками, що представляють незмінні об'єкти. В системі представлені різноманітними геометричними формами: прямокутниками, лініями та квадратами. Для кожної форми визначаються специфічні параметри розташування через координати характерних точок, що використовується для точного визначення зайнятих перешкодами клітинок сітки.

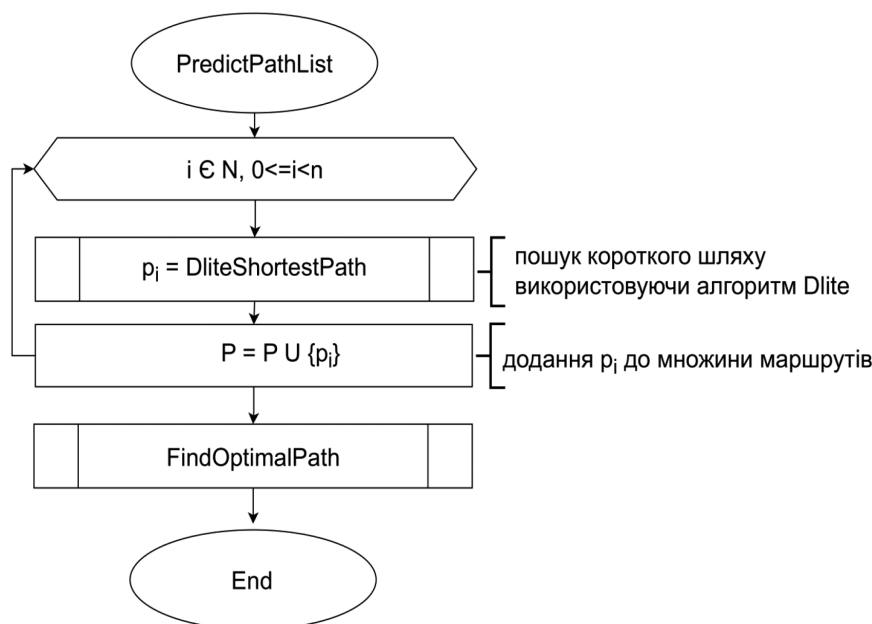


Рис. 1. Схема алгоритму виклик методу D* lite для отримання маршрутів

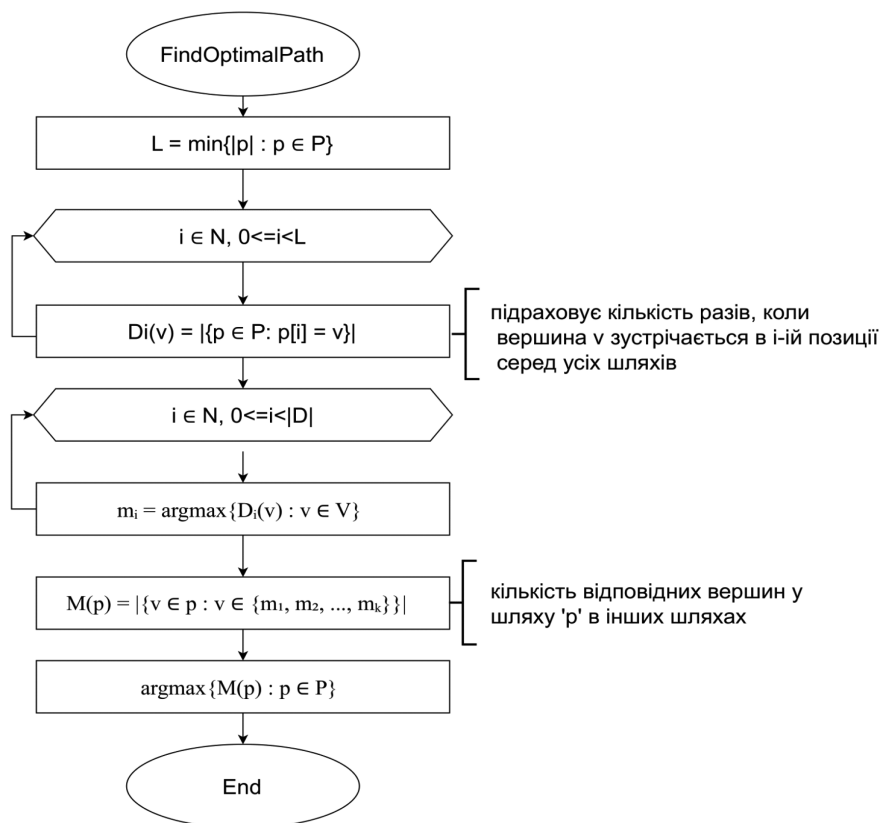


Рис. 2. Схема алгоритму пошуку оптимального маршруту

Динамічні перешкоди зображені фіолетовими точками, які вказують на об'єкти, здатні змінювати своє положення. Для них визначаються не лише початкові координати, але й вектор швидкості, напрямок руху та область можливого переміщення, що є критично важливим для подальшого прогнозування руху та планування безпечного маршруту роботи.

Прогнозований рух цих об'єктів показаний синіми лініями, що відображають очікувану траєкторію їх переміщення.

Для аналізу ефективності прогнозування в програмі був використаний набір значень для розміру сітки відображення середовища: 20×20 , 50×50 , 75×75 , 100×100 . Швидкість виконання вимірювалась в секундах. Результати

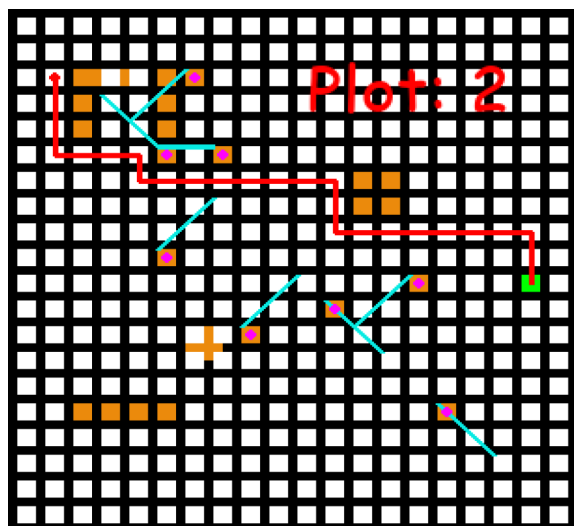


Рис. 3. Змодельована дискретна карта навігаційного простору

аналізу представлені на графіку, зображеному на рисунку 4.

В роботі було використано мову Python. Це просунута мова, і вона дуже проста у використанні. Але, поряд з простотою і коротким кодом, вона вимагає більше часу на виконання у порівнянні з C++. C++ перетворює весь код одразу в машинний, а потім виконує його. Цей метод займає менше часу у порівнянні з Python, який перетворює код рядок за рядком у байтовий код, а потім перетворює його в машинний код. Після перетворення він виконує програму рядок за рядком. Цей метод займає більше часу і робить Python повільним. Крім того, Python використовує більше пам'яті, ніж C++, що також робить його повільним. (Farzeen, 2020)

На рисунку 5 зображено шлях побудований без перешкод. Рисунки 6, 7 демонструють



Рис. 4. Витрати часу на прогнозування

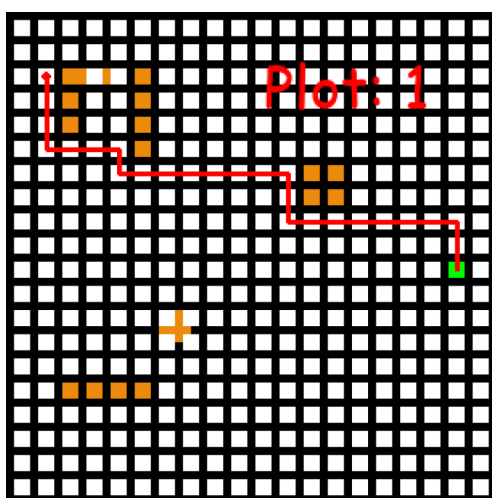


Рис. 5. Прогнозований шлях без динамічних перешкод (Прохід 1)

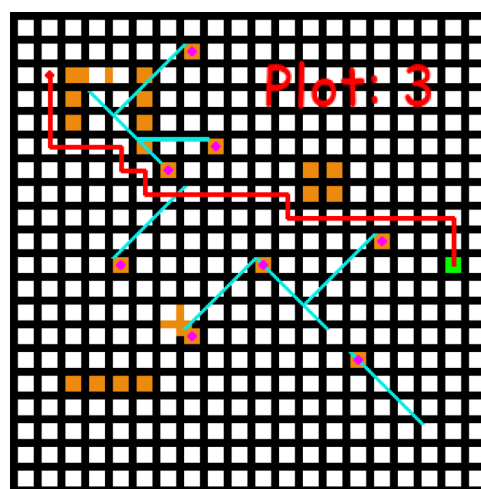


Рис. 6. Прогнозований шлях з динамічними перешкодами (Прохід 3)

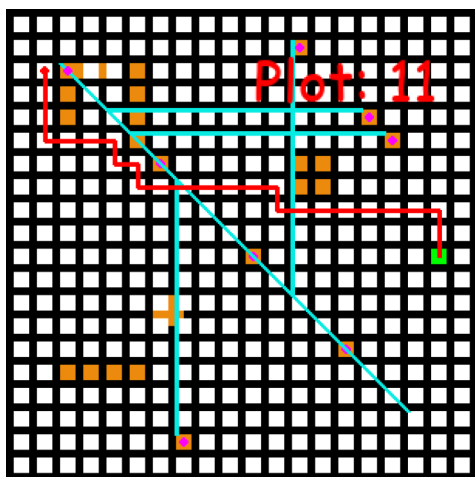


Рис. 7. Прогнозований шлях з динамічними перешкодами (Прохід 11)

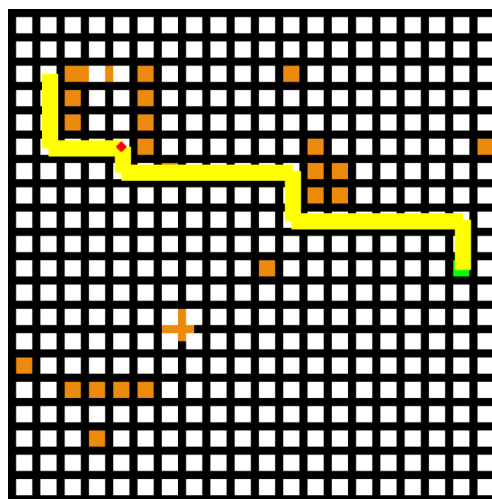


Рис. 8. Отриманий кінцевий шлях

перебудову маршруту при наявності перешкоди на шляху.

Інші ілюстрації були виключені з подання, оскільки вони повторюють маршрут, наведений на рисунку 5.

Шлях робота представлений червоною лінією, яка демонструє обчислений маршрут для досягнення кінцевої точки, враховуючи як статичні, так і динамічні перешкоди.

Кінцевий маршрут зображений на рисунку 8.

Висновки та перспективи. Представлена методика реалізації алгоритму дозволяє вирішити задачу навігації в динамічному середовищі. Комбінація точного математичного моделювання, адаптивного планування та постійного

контролю забезпечує надійне досягнення цілі навіть у складних умовах з множинними рухомими перешкодами.

Подальший розвиток може включати:

- вдосконалення алгоритмів прогнозування руху, вирішити проблему ступору в системі планування роботи.
- оптимізувати обчислювальні складності та забезпечити виконання системи в реальному часі.
- модернізація підходу до прогнозування маршруту перешкод з використання формул нелінійного руху для динамічних об'єктів.
- підвищення точності навігації в умовах невизначеності.

ЛІТЕРАТУРА:

1. Ганенко Л.Д., Жебка В.В. Аналітичний огляд питань навігації мобільних роботів в закритих приміщеннях. *Телекомунікаційні та інформаційні технології*. 2023. № 3. С. 85–98
2. Nitin Gupta, Kapil Manga, Anand Kumar Jha, Md. Umar. Applying Dijkstra's Algorithm in Routing Process. *International Journal of New Technology and Research (IJNTR)*. 2016. Vol. 2. № 5. 122–124 p.
3. Nagib G., Charieb W. Path planning for a mobile robot using Genetic Algorithms. *Proceedings of the International Conference on Electrical, Electronic and Computer Engineering (ICEECE)*. Cairo, Egypt, 5–7 September 2004. 2005. 185–189 p.
4. Thangamuthu M., et al. Mobile robot path planning and obstacle avoidance using hybrid algorithm. *International Journal of Information Technology*. 2023. Vol. 15. 4481–4490 p. DOI: 10.1007/s41870-023-01475-5
5. Duchoň F., Babinec A., Kajan M., Beňo P., Florek M., Fico T., Jurišica L. Path planning with modified A star algorithm for a mobile robot. *Procedia Engineering*. 2014. Vol. 96. 59–69 p. DOI: 10.1016/j.proeng.2014.12.098
6. Sven Koenig, Maxim Likhachev. D* lite. The Eighteenth National Conference on Artificial Intelligence, Association for the Advancement of Artificial Intelligence (AAAI). Alberta, Canada, 28 July – 1 August 2002. 2002.
7. Kretzschmar H., Spies M., Sprunk C., Burgard W. Socially compliant mobile robot navigation via inverse reinforcement learning. *International Journal of Robotics Research*. 2016. Vol. 35. № 11. 1352–1370 p.
8. Roberts M., Smith D. Path Following Algorithms in Autonomous Mobile Robots. *IEEE Transactions on Robotics*. 2016. Vol. 30. № 5. 1123–1140 p.
9. Stephen T., Jerry B. Classical dynamics of particles and systems. *Julie M. DeSilva*. Вид. 5. 2004. 30 p.
10. Farzeen Zehra, Darakhshan, Maha Javed, Maria Pasha. Comparative Analysis of C++ and Python in Terms of Memory and Time. Preprints. 2020. DOI: 10.20944/preprints202012.0516.v1

REFERENCES:

1. Ganenko, L.D., & Zhebka, V.V. (2023). Analitichnyi ohliad pytan navihatsii mobilnykh robotiv v zakrytykh prymishchenniakh [Analytical review of navigation issues of mobile robots in indoor environment]. *Telecommunication and Information Technologies*, 3, 85–98.
2. Gupta, N., Manga, K., Jha, A.K., & Umar, M. (2016). Applying Dijkstra's algorithm in routing process. *International Journal of New Technology and Research (IJNTR)*, 2(5), 122–124.
3. Nagib, G., & Charieb, W. (2005). Path planning for a mobile robot using Genetic Algorithms. *In Proceedings of the International Conference on Electrical, Electronic and Computer Engineering (ICEECE)*. Cairo, Egypt, 5–7 September 2004. pp. 185–189.
4. Thangamuthu, M., et al. (2023). Mobile robot path planning and obstacle avoidance using hybrid algorithm. *International Journal of Information Technology*, 15, 4481–4490. <https://doi.org/10.1007/s41870-023-01475-5>
5. Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T., & Jurišica, L. (2014). Path planning with modified A* algorithm for a mobile robot. *Procedia Engineering*, 96, 59–69. <https://doi.org/10.1016/j.proeng.2014.12.098>
6. Koenig, S., & Likhachev, M. (2002). D* lite. The Eighteenth National Conference on Artificial Intelligence (pp. 476–483). Association for the Advancement of Artificial Intelligence (AAAI), Alberta, Canada, 28 July – 1 August.
7. Kretzschmar, H., Spies, M., Sprunk, C., & Burgard, W. (2016). Socially compliant mobile robot navigation via inverse reinforcement learning. *International Journal of Robotics Research*, 35(11), 1352–1370.
8. Roberts, M., & Smith, D. (2016). Path following algorithms in autonomous mobile robots. *IEEE Transactions on Robotics*, 30(5), 1123–1140.
9. Stephen, T., Jerry, B. (2004). Classical dynamics of particles and systems (5th ed., p. 30). DeSilva, J.M. (Ed.).
10. Zehra, F., Darakhshan, M., Javed, M., & Pasha, M. (2020). Comparative analysis of C++ and Python in terms of memory and time. Preprints. <https://doi.org/10.20944/preprints202012.0516.v1>