

МЕТОДИ ПЛАНУВАННЯ ШЛЯХУ ТА УНИКНЕННЯ ПЕРЕШКОД ДЛЯ АВТОНОМНИХ МОБІЛЬНИХ РОБОТІВ

І. Берізка

*Львівський національний університет імені Івана Франка,
вул. Тарнавського, 107, 79005 м. Львів, Україна
ihor.berizka@lnu.edu.ua*

Планування шляху розв'язує задачу пошуку маршруту від початкової позиції робота до цільової на основі даних про робоче середовище із доступних сенсорів. Уникнення перешкод є важливою складовою таких алгоритмів, адже робот повинен дістатись цільової локації без зіткнень із іншими об'єктами. Методи виявлення та уникнення перешкод надають можливість роботам ефективно пересуватись у своєму робочому середовищі при цьому мінімізуючи ймовірність колізій. У статті наведено загальну класифікацію алгоритмів та розглянуто типових представників із оглядом переваг та недоліків кожного із методів. Більш детально розглянуто методи на основі штучних потенціальних полів. Методи на основі штучних потенціальних полів знайшли широке практичне застосування в робототехніці для виявлення та уникнення перешкод. Основними перевагами цього типу методів є ефективність, надійність, простота реалізації та реактивність. Вони базуються на ідеї, що робот може притягуватись до цільової локації чи точки та відштовхуватись від перешкод використовуючи віртуальні сили, які генеруються штучним потенціальним полем. Результируюче поле генерує регіони високого потенціалу біля перешкод та низького в цільовій точці. Основним недоліком цих методів є проблеми локального мінімуму. У роботі наведено розширений огляд як класичного варіанту методу штучних потенціальних полів так і кілька модифікацій, які показують кращі результати за швидкістю чи оптимальністю шляху.

Ключові слова: робототехніка, виявлення та уникнення перешкод, штучне потенціальне поле, автономні роботи, мобільні роботи.

Вступ.

Робототехніка пропонує доволі широкий набір задач. Під час розробки робота необхідно приділяти увагу багатьом як апаратним так і програмним аспектам. З архітектурного боку робота можна розглядати як машину, котра працює за принципом зображеним на рис. 1. Частина із збором інформації з навколишнього середовища розглянуто у статті [8]. В даній статті аналізуються методи для етапів моделювання та планування на прикладі задачі про уникнення перешкод.

Під час розробки автономних систем необхідно розв'язувати задачі планування шляху, локалізації в приміщеннях та на вулиці, уникнення перешкод, виявлення об'єктів та класифікації інших машин, людей, тварин та різних знаків. Планування шляху є однією з фундаментальних технологій у автономних системах. Розрізняють два види планування шляху: локальний та глобальний. Для глобального планування необхідна інформація із географічних інформаційних систем та глобальна локалізація (наприклад дані з GPS

давачів), в той час як для локального планування необхідно знати відносну позицію робота у просторі та техніку уникнення перешкод. Головною задачею планування шляху є обчислення безпечної траєкторії руху робота до цільової точки. Для цього необхідно враховувати механічні розміри та властивості самого робота, а також різні перешкоди. Такі методи зазвичай розглядають ідеальну модель робота та його робочого простору.



Рис. 1. SPA (sense-plan-act) парадигма – алгоритм роботи машини за принципом вимірювання – планування – дія. Адаптовано із [9].

Fig. 1. The sense-plan-act (SPA) paradigm. Adopted from [9].

Основною перевагою даного підходу є повний розв'язок задачі. Проте, як тільки додається певна невідомість чи невизначеність в робочому середовищі – такі підходи зазнають краху.

Саме тому використовують техніки уникнення перешкод. Основна ідея полягає також в русі до цільової точки, але із зміною траєкторії якщо виявлено перешкоди на поточній. Основним недоліком таких алгоритмів – є локальність. Графічне представлення алгоритму планування руху та уникнення перешкод наведено на рис. 2.

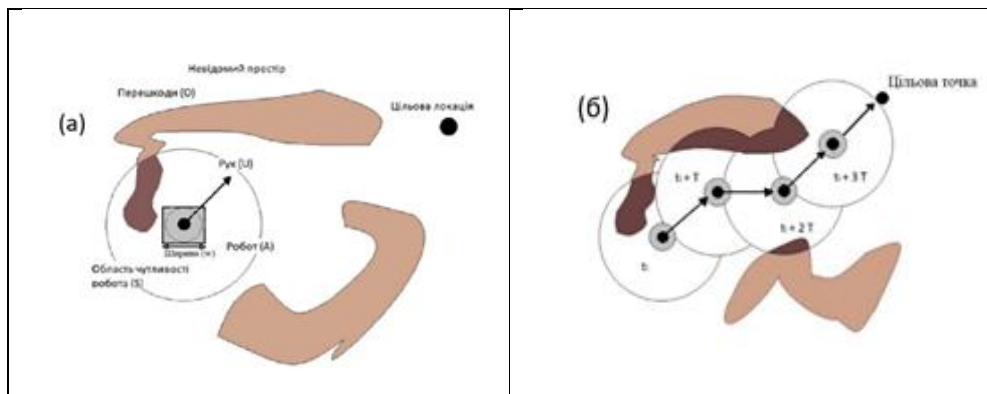


Рис. 2. Задача уникнення перешкод включає в себе дві підзадачі: обчислення руху, який уникає виявлені сенсорами перешкоди та рух робота до цільової локації (а). Результат застосування даного алгоритму на кожній ітерації – це послідовність рухів, що ведуть робота до цілі (при цьому оминають перешкоди) (б).

Fig. 2. Obstacle avoidance problem consists of two subtasks: computing motion control that voids collisions with the obstacles gathered by the sensors and driving the robot towards target location (a). The result of applying this technique at each step is a sequence of motions that drive the vehicle free of collisions to the target.

Існує кілька загальноприйнятих способів класифікації алгоритмів виявлення та уникнення перешкод. У цій роботі наведемо класифікацію на дві групи за підходом: класичні та евристичні. Такий підхід до класифікації спрощує вибір алгоритмів під конкретні задачі, адже кожен із підходів має ряд переваг та недоліків. На рис.3 наведено діаграму поділу алгоритмів та типових представників кожного класу.

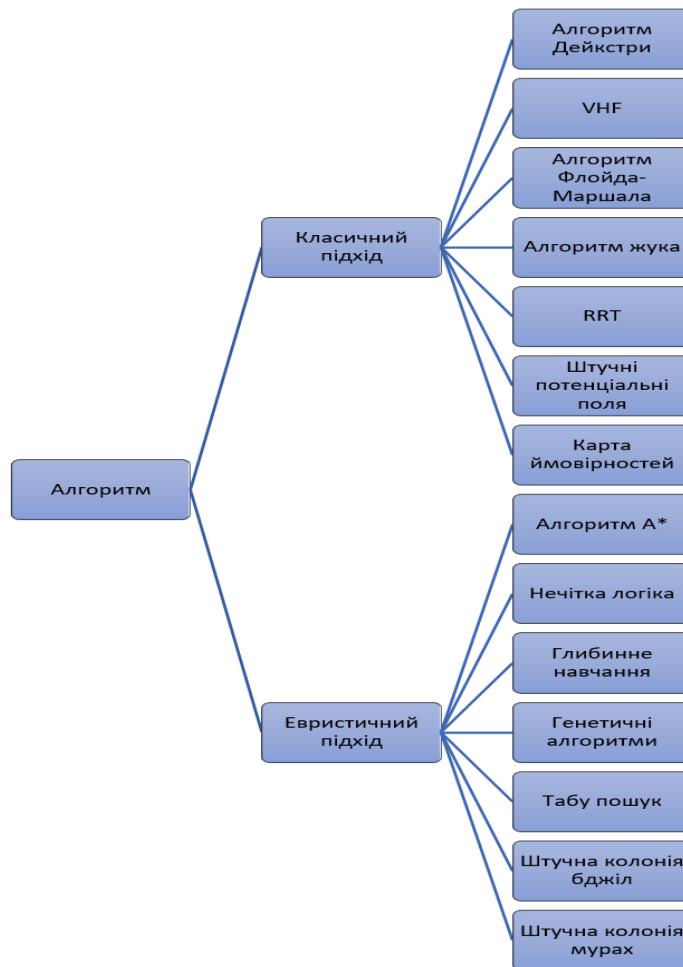


Рис. 3. Класифікація алгоритмів виявлення та уникнення перешкод.

Fig. 3. Classification of obstacle detection and avoidance algorithms.

Коротко розглянемо найбільш популярних представників. Одним із типових класичних алгоритмів – є алгоритм Дейкстри, який вперше був представлений у 1956 р. та опублікований у 1959 р. [10]. Алгоритм вирішує проблему пошуку найкоротшого шляху між двома вершинами напрямленого графа. Саме таке представлення робочого середовища робота є одним із найпопулярніших [11]. Проте варто зазначити, що такий

підхід менш ефективний коли початкова та кінцеві точки сильно віддалені. Це суттєво збільшує кількість обчислень, причому частина із них є не важливими для оптимального шляху. Також швидкодія алгоритму спадає із ростом кількості вершин графа [12]. Незважаючи на описані вище недоліки, різні модифікації алгоритму все ж широко застосовують у практичних задачах та проводять експерименти. Детальніше застосування алгоритму Дейкстри та його модифікацій наведено у роботах [13, 14, 15, 16, 17, 18].

Навіть зважаючи на наявність більш оптимальних алгоритмів для планування шляху алгоритми на основі поведінки жуків все ще широко використовуються. Базову версію алгоритму представили Люмельський та Степанов [1], в якому робот рухається вздовж виявленої перешкоди аж доки вона не закінчиться. Якщо напрямок до цілі вільний – машина починає далі їхати до цільової точки. Загалом алгоритм поділяють на три варіації, які розглянуті детальніше у [19, 20, 21].

Ще одним підходом до уникнення перешкод є VHF (vector field histogram) метод. Початкова версія алгоритму розроблена Боренштейном та Кореном [2]. Метод розв'язує задачу в два етапи: обчислення множини потенційних напрямків руху та вибір одного з них. Алгоритм працює з ймовірнісними розподілами перешкод і тому добре адаптований для роботи з нечіткими сенсорами такими як ультразвукові сонари. Також наявні дослідження показують, що мобільні роботи можуть використовувати VHF для руху у середовищі із щільно розташованими перешкодами на високих середніх швидкостях та можуть проходити через вузькі місця (такі як двері) чи вузькі коридори без коливань [2]. У [22] наведено покращену 3D-VHF версію алгоритму, яка застосовується для літаючих роботів (дронів).

У список типових та доволі широко використовуваних алгоритмів також входить RRT (Rapidly Exploring Random Tree), який вперше був представлений ЛаВалем у [23]. Основна ідея алгоритму полягає у побудові дерева, вузли якого позначають стани, а ребра відповідають за переходи між станами. Корінь дерева вважається початковим станом, а всі інші вершини вважаються досяжними вздовж шляху від кореня до відповідної кінцевої вершини (стану). Алгоритм можна описати наступними кроками:

- Ініціалізація. Згенерувати пусте дерево T із вершинами V та ребрами E . Додати початкову вершину до дерева.
- Генерація випадкової точки. Згенерувати випадкову вершину у робочому просторі q_{rand} .
- Вибір найближчої вершини. З усіх вершин дерева T вибрати найближчу до q_{rand} вершину q_{near} за певним критерієм (наприклад евклідова відстань)
- Розростання дерева. Із вершини q_{near} провести крок у напрямку до q_{rand} утворюючи нову вершину q_{new} . Відстань між q_{near} та q_{new} обмежується певним кроком Δq
- Перевірка колізій. Якщо на шляху від q_{near} до q_{new} немає перешкод додати до поточну вершину до множини вершин V , додати ребро (q_{near}, q_{new}) до E .
- Критерій закінчення пошуку. Якщо q_{new} розміщене достатньо близько до q_{goal} - завершити пошук. В іншому випадку перейти до кроку генерації нової вершини.

Серед переваг алгоритму варто зазначити стійкість до проблеми із локальними мінімумами [24]. Також цей підхід доволі добре справляється із проблемами неголономного та кінодинамічного планування шляху. Згідно із практичними результатами – алгоритм навіть враховує повороти на будь який градус враховуючи кінематичні та динамічні обмеження робота [25]. До особливостей алгоритму варто віднести те, що дерево будується випадковим чином і результируючий шлях може мати певну хаотичність в русі. Також присутність випадкового процесу – впливає на повільну збіжність та ефективність пошуку [26]. На сьогодні запропоновано доволі багато модифікацій та покращень базового алгоритму. Серед модифікацій варто відзначити RRT-Connect [27] та S-RRT [28]. Приклади застосування модифікацій алгоритму до різних типів роботів наведено у роботах [29, 30, 31].

Метод обмеження перешкод розв'язує задачу в три етапи, де результат перших двох етапів є множина потенційних напрямків для руху. На першому кроці за потреби обчислюють поточну проміжну цільову локацію. На другому – асоціюють обмеження руху із кожною перешкодою та об'єднують їх для обчислення множини бажаних напрямків. Останнім етапом є стратегія руху до цільової точки [4]. Основною перевагою методу є те, що він продемонстрував ефективний рух в обмеженому просторі.

Розглянемо кілька алгоритмів із евристичним підходом. Одним із базових алгоритмів цього класу є A^* , який був розроблений Хартом у 1968р [34] з метою покращення алгоритму Дейкстри. Покращення полягає у введенні евристичної вагової функції, яка обчислює відстань від поточної точки до цільової. Для роботи алгоритму потрібно мати граф, який буде представляти робоче середовище робота. У цьому алгоритмі площа пошуку зазвичай поділена на малі квадратні регіони, де кожному регіону відповідає вершина графа. A^* шукає шлях із найменшою вагою (наприклад мінімальним часом) серед усіх шляхів, які ведуть до цільової вершини. Алгоритм використовує функцію (1) для оцінки результатів.

$$f(n) = g(n) + h(n) \quad (1)$$

Функція $f(n)$ являє собою сумарну вагу від початкової вершини до поточної поширюючись до цільової вершини. $g(n)$ описує найменшу вагу від початкової вершини до поточної, $h(n)$ прогнозує оптимальну вагу від поточної вершини n до цільової та зазвичай обчислюється з використанням мангетенської метрики [35]. Практичні застосування для різних типів роботів наведено у працях [36, 37, 38]. Варто зазначити, що алгоритм не завжди знаходить оптимальний маршрут так як не враховує всі можливі маршрути. Також метод працює доволі повільно для великих графів через обчислення функції $f(n)$ на кожній ітерації.

Ще одним популярним підходом серед евристичних методів є застосування генетичних алгоритмів. Вперше такий алгоритм розглянув Бремерман у 1958р [39]. Підхід бере ідею зі еволюційної теорії Дарвіна та імітує концепцію виживання індивідів, які найбільш адаптовані до свого середовища. Найбільш життєздатні члени популяції виживають, в той час як слабші ні. Члени, які вижили далі передають свої гени наступним поколінням через схрещування, мутації та вибірку. Така випадкова структура інформації була використана Бремерманом для створення алгоритму пошуку можливих маршрутів.

Генетичні алгоритми випадковим чином ініціалізують популяцію використовуючи випадкову стрічку символів (аналог хромосоми) та функцію допасованості (оціночну). Далі генерується початкова популяція на основі трьох генетичних операцій: мутація, схрещення та селекція. Нові популяції генеруються доки не буде виконано певну кінцеву

умову [40]. Прикладами таких умов є обмеження по часу виконання, очікуване значення оцінки (допасованості) чи максимальна кількість поколінь. На етапі мутації елементи вибраної стрічки мутують із певною ймовірністю мутацій. На етапі схрещування елементи двох стрічок перемішуються згідно із певним правилом та утворюють дві нові стрічки. Останнім етапом є селекція на якому дві стрічки вибираються на основі ймовірності отриманої від функції допасованості та порівнюються по придатності. Хромосома, яка отримала вищу оцінку по придатності вибирається для генерації наступного покоління. Якщо вибрана хромосома задовільняє кінцеву умову – алгоритм завершується. В іншому випадку переходить до етапу генерації нового покоління. Дослідження застосування варіацій генетичних алгоритмів для планування шляху та уникнення перешкод для різних типів роботів наведено у працях [41, 42, 43, 44].

Протягом останніх декад алгоритми на основі нейронних мереж та глибинного навчання почали масово застосовуватись у різних технічних напрямках. Не стала винятком і задача виявлення перешкод та планування шляху. Алгоритми такого типу також відносять до евристичних. Огляд алгоритму із застосуванням штучних нейронних мереж у поєднанні із ієрархічним навчанням із підкріпленням (Hierarchical Reinforcement Learning) наведено у [48]. Автори порівнюють цей алгоритм із схожими підходами на основі нейронних мереж та глибинного навчання.

Метод динамічних вікон, який враховує динаміку робота був запропонований у статті [3]. Алгоритм розв'язує задачу в просторі керування, використовуючи інформацію про динаміку робота. Тому метод добре адаптований для роботів із повільними динамічними властивостями чи для роботи на високих швидкостях. До недоліків відносять складність вибору шляху у середовищі із щільно розташованими перешкодами [32]. Серед основних напрямків досліджень у покращенні алгоритму є модифікації із використанням навчання із підкріпленням (Reinforcement learning) [33].

Методи на основі потенціальних полів.

1. Класичний метод штучних потенціальних полів.

Метод використовує фізичну аналогію в якій робот розглядається як частинка, що рухається в потенціальному полі. Робот та перешкоди являються позитивно зарядженими частинками у цьому полі, в той час як цільова локація виступає частинкою із негативним зарядом. Таким чином робот відштовхується від перешкод під дією F_{rep} (репульсивної сили), та притягується до цільової точки атрактивною силою F_{att} . Результируюча сила становить векторну суму всіх репульсивних сил та атрактивної. Важливо зазначити, що відстань до перешкоди суттєво впливає на модуль сили: чим ближче робот до перешкоди – тим більший модуль сили генерує перешкода та відповідно більше впливає на результируючу силу. Фактично протилежний ефект спостерігається із атрактивною силою – чим далі робот від цільової точки – тим сильніше він притягується, тобто атрактивна сила набуває більшого значення по модулю. Математичну модель кожної із сил наведено у рівняннях (2) – (4).

$$F_{tot}(q_{t_i}) = F_{att}(q_{t_i}) + F_{rep}(q_{t_i}) \quad (2)$$

$$F_{att}(q_{t_i}) = k_{att} * n_{q_{target}} \quad (3)$$

$$F_{rep}(q_{t_i}) = \begin{cases} k_{rep} \cdot \sum_j \left(\frac{1}{d \cdot (q_{t_i}, p_j)} - \frac{1}{d_0} \right) \cdot n_{p_j}, & d(q_{t_i}, p_j) < d_0 \\ 0, & \text{в інших випадках} \end{cases} \quad (4)$$

В кожен момент часу t_i рух обчислюється таким чином, щоб він збігався з напрямком результуючої сили (2). Схематично це наведено на рис. 4.

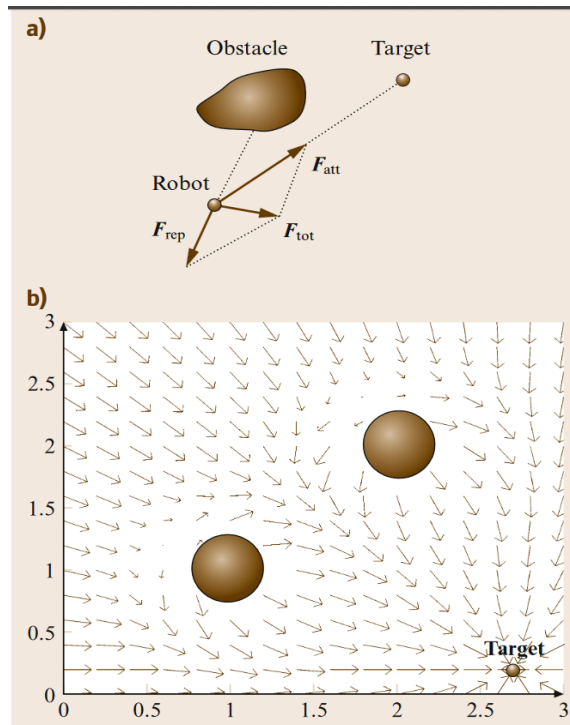


Рис. 4. (а) обчислення напрямку руху згідно з методом потенційного поля. Цільова точка діє на робота із силою F_{att} , в той час як перешкода генерує репульсивну силу F_{rep} . Результуюча сила F_{tot} визначає найбільш ймовірний напрямок руху; (б) напрямки руху обчислені класичними методами на кожній ітерації руху. Адаптовано з [9].

Fig. 4. (a) Computation of the motion direction with a potential field method. The target attracts the particle F_{att} while the obstacle exerts a repulsive force F_{rep} . The resulting force F_{tot} is the most promising motion direction; (b) Motion directions computed in each point of the space with classic method. Adopted from [9].

Також варто зазначити одну із типових проблем цього методу, а саме проблеми із локальними мінімумами. Найчастіше це спостерігається із перешкодами, які розташовані з обох боків від робота та паралельно до траєкторії його руху (рис. 5.а). Іншою типовою конфігурацією робочого середовища, яка призводить до проблеми локального мінімуму у класичному методі є перешкоди у формі підкови(рис. 5.б). для вирішення цієї проблеми використовують кілька стандартних підходів.

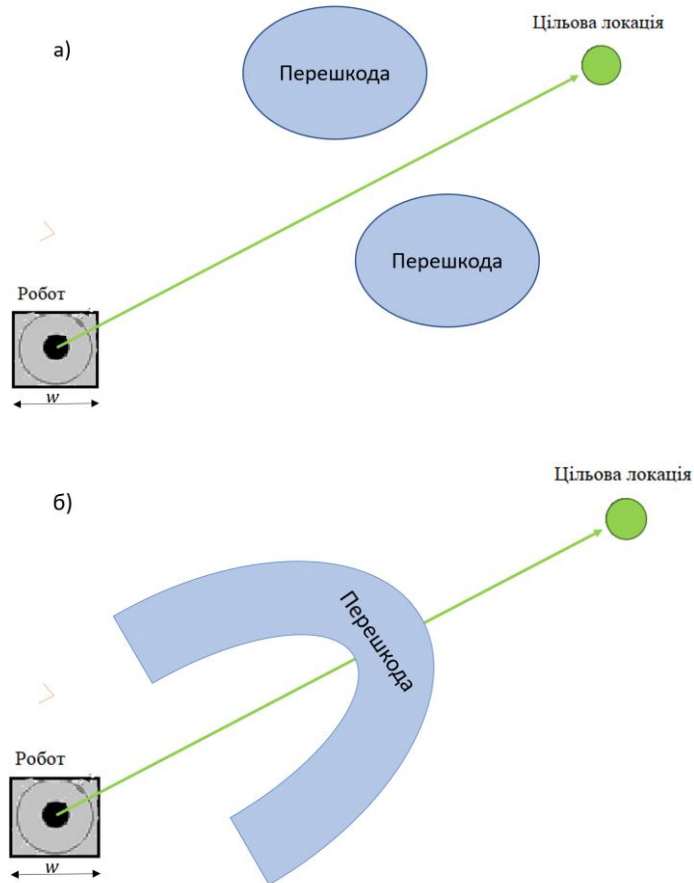


Рис. 5. Приклад конфігурацій робочого середовища, які призводять до проблеми локальних мінімумів: (а) перешкоди обабіч робота та паралельно напрямку його руху; б) підковоподібна перешкода.

Fig. 5. Example of the workplace configuration which leads to the local minima problem: (a) obstacle are located in parallel with robot motion direction and robot is between them; (b) U-shaped obstacle.

Наприклад тимчасово додають велику випадкову величину до атрактивної сили щоб «втягнути» робота із стану рівноваги між атрактивною та репульсивною силами і таким чином вийти із локального мінімуму. Детальніший огляд такого підходу розглянуто у [45].

Підсумовуючи, робимо висновок що метод штучних потенціальних полів пропонує прямий шлях від початкової позиції до цільової локації, а схильність до локальних мінімумів призводить до ускладнення методу. Саме тому розроблено доволі багато модифікацій для покращення ефективності цього методу, адже навіть із модифікаціями

він залишається ефективним та здатним працювати у режимі реального часу на системах із обмеженими обчислювальними ресурсами.

2. Метод штучних потенціальних полів з використанням нечіткої логіки.

Як і у класичному методі штучних потенціальних полів, у цій модифікації репульсивна сила є сумою репульсивних сил від кожної із перешкод. Також вихідні дані із контролера нечіткої логіки складаються із комбінацій кількох правил нечіткої логіки. Автори пропонують підхід у якому кожній репульсивній силі від перешкоди ставиться у відповідність одне правило із контролера нечіткої логіки [6]. Таким чином комбінація n правил нечіткої логіки здатна описати суму репульсивних сил кожної із n перешкод. Тоді якщо робот оточений n перешкодами, а контролер робота є побудований на основі нечіткої логіки із n правилами – такий контролер може повністю змоделювати класичний метод штучних потенціальних полів. Із таким підходом необхідно врахувати наступні три фактори: кількість правил має відповідати кількості репульсивних сил, одне правило може описувати репульсивну силу лише однієї перешкоди та комбінація всіх правил є числовою сумою репульсивних сил, які генеруються кожною перешкодою.

Із цих принципів слідує те, що субрепульсивні сили можна зімітувати використовуючи наступні правила нечіткої логіки. Контролер має n скалярних входів (наприклад d_1, \dots, d_n), один вихід у вигляді вектора \vec{y} , та n правил. У цій роботі використовується метод Такагі-Сугено [46], який описується набором правил «Якщо-Тоді» наступним чином:

$$\begin{aligned}
 & p_{th} \text{ Rule } R_p : \\
 & \text{If } d_p \text{ is } \mu_{d_p} \\
 & \text{Then } \vec{y}_p = \left(\sum_{i=1}^n \frac{\varepsilon + d_i}{d_{\max}} \right) \frac{d_{\max}}{\varepsilon + d_p} k_{rep} \left(\frac{1}{d_p} - \frac{1}{d_{\max}} \right) \vec{s}_p,
 \end{aligned} \tag{5}$$

де n - загальна кількість правил, $p = 1, \dots, n$, $\varepsilon > 0$, μ_{d_p} функція від d_p та \vec{s}_p вектор, який вказує напрямок p_{th} сенсора.

Таким чином рівняння дефазифікації контролера (6) є зваженим середнім значенням всіх виходів.

$$\vec{y} = \frac{\sum_{p=1}^n w_p \vec{y}_p}{\sum_{p=1}^n w_p}, \tag{6}$$

де p_{th} значення вагового коефіцієнта обчислюється наступним чином: $w_p = \mu_{d_p}(d_p)$.

Із всіх n виходів у (5) та рівняння дефазифікації (6) отримуємо вихід контролера нечіткої логіки (7).

$$\vec{y} = k_{rep} \left(\sum_{i=1}^n \frac{\varepsilon + d_i}{d_{\max}} \right) * \frac{\sum_{p=1}^n w_p \left(\frac{d_{\max}}{\varepsilon + d_p} \right) \left(\frac{1}{d_p} - \frac{1}{d_{\max}} \right) \vec{s}_p}{\sum_{p=1}^n w_p}. \tag{7}$$

Значення вагових коефіцієнтів обчислюють за формулою (8).

$$w_p = \mu_{d_p}(d_p) = \frac{\varepsilon + d_p}{d_{\max}} \quad (8)$$

Таким чином вихідні дані із правил нечіткої логіки в (5) зважуються згідно із (8) та усереднюються у (6). У результаті загальний вихід контролера описує сумарне репульсивне поле, а кожен окремий вихід із p_{th} правила представляє репульсивне поле p_{th} перешкоди. Із ростом цього параметра нечіткої логіки зростає репульсивне поле і відштовхує робота від перешкоди. Блок схему контролера нечіткої логіки наведено на рис. 6.

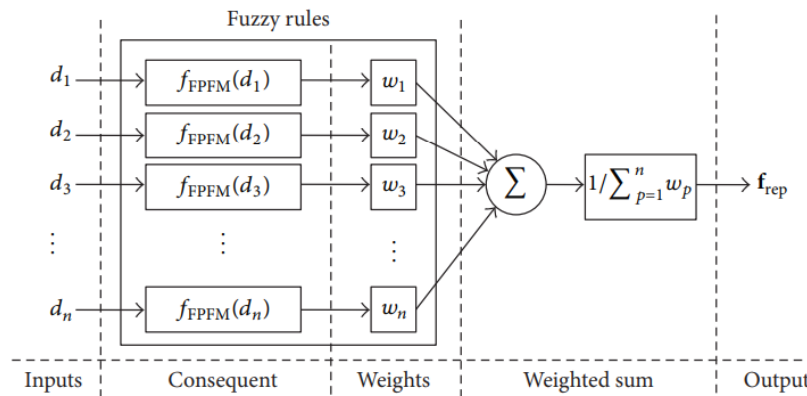


Рис. 6. Блок схема контролера нечіткої логіки для класичного методу штучних потенціальних полів. Адаптовано з [6].

Fig. 6. Block diagram of a fuzzy logic controller for the classical artificial potential field method. Adopted from [6].

У ході дослідження автори пройшли шлях від простого контролера схему якого наведено на рис. 6 до складного контролера із розподіленою структурою та додатковими входами для керування, яку зображено на рис. 8. Важливим покращенням є додавання керуючих входів. Ці додаткові входи збільшують кількість правил нечіткої логіки та враховують стратегії керування роботом розроблені експертами. Удосконалений метод штучних потенціальних полів із використанням нечіткої логіки (AFPFM) враховує не лише відстані до перешкод, але ще додаткову інформацію про робота, перешкоди та цільову локацію. Таким чином вводяться додаткові параметри, котрі враховуються у математичній моделі. Графічне пояснення цих параметрів наведено на рис. 7. Додаткові стратегії керування роботом у AFPFM описуються наступним чином:

1. Якщо робот напрямлений у бік перешкоди (тобто кут ψ_i між векторами \vec{s}_i та \vec{v}_{robot} є малим) – модуль репульсивної сили f_{rep} має бути великим.
2. Якщо напрямок атрактивної сили \vec{f}_{attr} вказує на перешкоду (кут ϕ_i між векторами \vec{s}_i та \vec{f}_{attr} малий) - модуль репульсивної сили f_{rep} має бути великим.

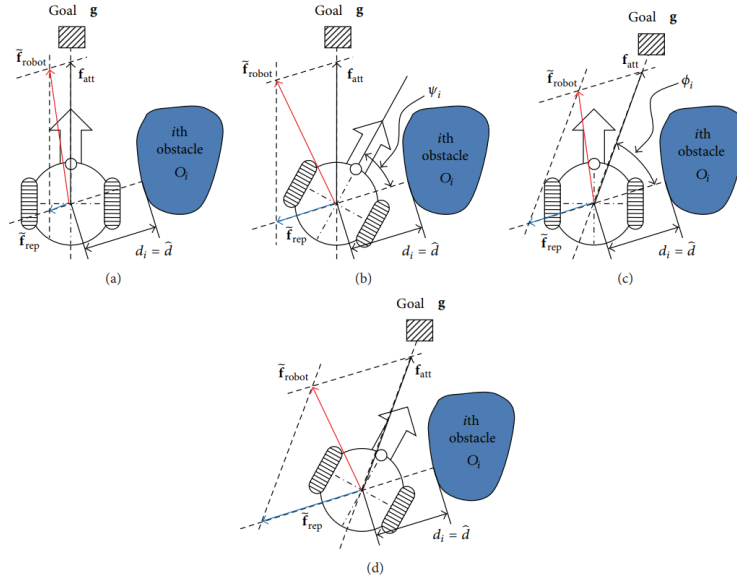


Рис. 7. Приклади обчислення сил потенціального поля із врахуванням додаткових параметрів. Адаптовано з [6].

Fig. 7. Examples of calculating potential field forces using additional parameters. Adopted from [6].

Враховуючи всі додаткові параметри опис математичної моделі наведено у рівняннях (9) - (11).

$$\begin{aligned}
 & (p, q, r, s)_{th} \text{ Rule } R_{p,q,r,s} : \\
 & \text{If } d_p \text{ is } \mu_{d_p,q}, \psi_p \text{ is } \mu_{\psi_p,r}, \phi_p \text{ is } \mu_{\phi_p,s} \\
 & \text{Then } \overrightarrow{y_{p,q,r,s}} = f_{AFPFM_{p,q,r,s}}(d_p)
 \end{aligned} \tag{9}$$

де $p = 1, \dots, n$ та $q, r, s = 1, \dots, m$, m - кількість функцій приналежності. Таким чином рівняння, яке описує репульсивну силу у AFPFM записують у вигляді (10).

$$\begin{aligned}
 f_{rep} &= - \sum_{p=1}^n \frac{(\sum_{q=1}^m \sum_{r=1}^m \sum_{s=1}^m w_{p,q,r,s} \overrightarrow{y_{p,q,r,s}} / \sum_{q=1}^m \sum_{r=1}^m \sum_{s=1}^m w_{p,q,r,s})}{n} \\
 f_{AFPFM_{p,q,r,s}}(d_p) &= k_{rep_{p,q,r,s}} \left(\sum_{i=1}^n \frac{\varepsilon + d_i}{d_{max}} \right) \frac{d_{max}}{\varepsilon + d_p} \left(\frac{1}{d_p} - \frac{1}{d_{max}} \right) s_p \\
 w_{p,q,r,s} &= \mu_{d_p,q}(d_p) \wedge \mu_{\psi_p,r}(\psi_p) \wedge \mu_{\phi_p,s}(\phi_p)
 \end{aligned} \tag{10-11}$$

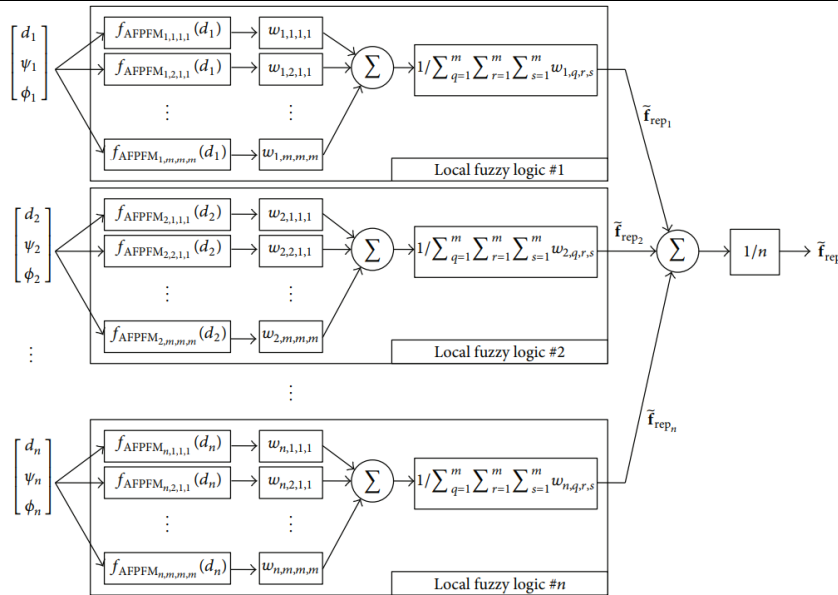


Рис. 8. Блок схема удосконаленого контролера нечіткої логіки із розподіленою структурою та додатковими входами керування. Адаптовано з [6].
 Fig. 8. Block diagram of an enhanced fuzzy logic controller with a distributed structure and additional control inputs. Adopted from [6].

Згідно із експериментами авторів [6] - запропонована модифікація методу штучних потенціальних полів генерує оптимальніший по часу та відстані шлях, а також більш плавну траєкторію руху без різких поворотів коли цього не вимагає ситуація у робочому середовищі робота. Також ця модифікація алгоритму справляється із проблемою класичного методу показаного на рис. 5а.

3. Метод штучних потенціальних полів на основі функції Гауса.

Ще одним перспективним напрямком модифікацій класичного методу штучних потенціальних полів є перехід до роботи із ймовірностями та використання різноманітних функцій для опису ймовірності зіткнення із перешкодами. Тобто ці функції фактично описують репульсивне поле, яке генерує перешкода. У роботі [7] автори дотримуються цієї методології та пропонують використовувати функцію Гауса для опису репульсивної сили. Також дещо іншим способом описують робоче середовище робота та виявлення перешкод.

Під час побудови математичної моделі автори вводять пару чисел (θ, d) , де θ позначає кут вимірювання в градусах, а d - відстань до об'єкта в метрах. Щоб отримати дані у такому форматі автори пропонують використовувати рухомий ультразвуковий сенсор відстані. Також можна запропонувати використовувати одновимірний LIDAR для сканування робочого середовища.

На початковому етапі необхідно задати порогову відстань d_{max} . Всі об'єкти, які знаходяться ближче до робота за цю відстань розглядаються як перешкоди. Таким чином це один із параметрів як може суттєво впливати на ефективність роботи алгоритму та потребує емпіричного підбору значення. Також автори розглядають робота як квадрат із стороною w . Схематично робоче середовище із додатковими параметрами наведено на рис. 9.

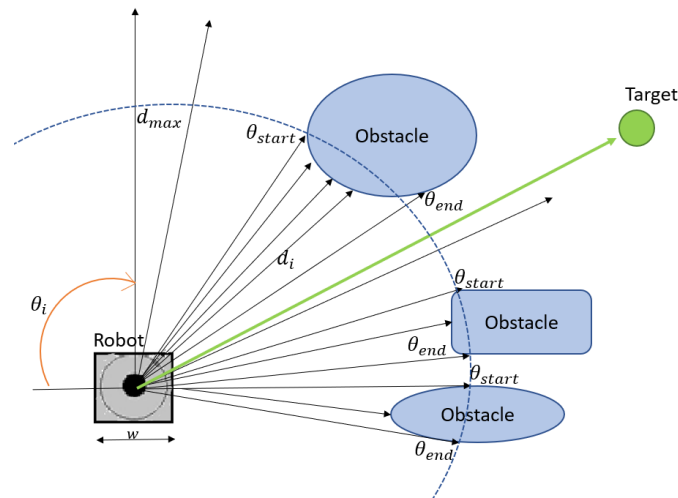


Рис. 9. Схема сканування робочого середовища робота. Адаптовано з [47].
 Fig. 9. Diagram of the robot's workspace scanning process. Adopted from [47].

На рис. 9 представлено робота, робоче середовище із трьома перешкодами різної геометричної форми та цільову локацію. Коло із штриховою лінією описує порогову відстань. Тобто радіус цього кола дорівнює d_{max} , θ_i позначає поточний кут вимірювання, а d_i - відстань до об'єкта на θ_i куті. На початковому етапі для виявлених перешкод вводять два кути $(\theta_{start}, \theta_{end})$, де θ_{start} позначає кут на якому вперше було помічено перешкоду, а θ_{end} - кут на якому перешкода закінчилась. Таким чином цих два параметра виділяють кожну перешкоду у вхідних даних із сенсорів. На подальших етапах для опису перешкод вводять ще два додаткові параметри: d_k - середня відстань до k -ої перешкоди та $\phi_k = \theta_{end} - \theta_{start}$, тобто кут, який охоплює перешкода.

Хоча у багатьох експериментах або комп'ютерних симуляціях розміри робота не враховуються, тобто він вважається матеріальною точкою для спрощення симуляцій та математичних моделей. Для покращення практичності та наближення математичної моделі до реальних умов все ж необхідно враховувати геометричні розміри робота. Таким чином частково ускладнюється модель, але результати приносять більше практичної цінності. Автори пропонують гібридну модель – розміри робота враховуються, але за певними формулами вони впливають на розмір оточуючих перешкод. Таким чином перешкоди проходять процес збільшення у розмірі з точки зору робота, а сам робот вважається матеріальною точкою. Графічно цей процес наведено на рис. 10.

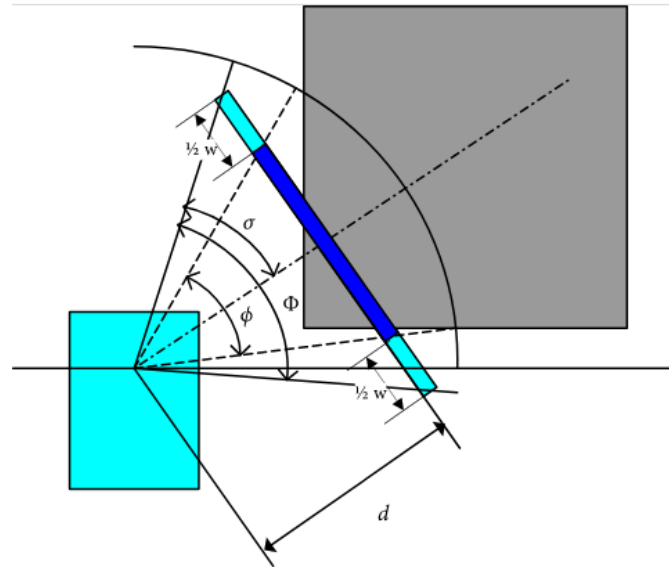


Рис. 10. Процес збільшення геометричних розмірів перешкод із врахуванням розмірів робота.
Адаптовано з [7].

Fig. 10. The process of enlarging obstacle geometrical dimensions considering the robot's size.
Adapted from [7].

Для врахування розмірів робота необхідно обчислити заново кут ϕ_k згідно із рівнянням (12).

$$\phi_k = 2\sigma_k = 2 \operatorname{atan}\left(d_k * \tan\left(\frac{\phi_k}{2}\right) + \frac{w_{robot}}{2}, d_k\right) \quad (12)$$

На рис. 10 темно-синя лінія показує, як робот “бачить” перешкоду та який кут вона покриває. Перешкода після процесу збільшення, враховуючи розміри робота, позначена світло-синім кольором. Також з цим значенням можна виконувати різні маніпуляції, щоб досягти різних результатів. Наприклад, до ширини робота можна додати кілька сантиметрів для більшої впевненості, що не буде зіткнень. Це призведе до зниження оптимальності шляху до цілі, але збільшить ймовірність уникнення зіткнень.

Наступний етап полягає у обчислення репульсивної сили кожної перешкоди згідно із рівнянням (13).

$$f_k(\theta_i) = A_k \exp\left(-\frac{(\theta_k - \theta_i)^2}{2\sigma_k^2}\right), \quad (13),$$

де θ_k позначає центральний кут перешкоди, σ_k - половина кута, який охоплює перешкода.

Коефіцієнт A_k підбирають таким чином щоб гаусіан повністю охоплював перешкоду та обчислюється згідно із рівнянням (14).

$$A_k = d_k \exp\left(\frac{1}{2}\right), \quad (14),$$

де $d_k = d_{\max} - d_k$.

Як і у решти методів на основі штучних потенціальних полів репульсивна сила f_k описує поле генероване k -ою перешкодою. Тобто загальне репульсивне поле обчислюється як сума всіх репульсивних сил від кожної перешкоди. Також варто зазначити, що автори пропонують описувати репульсивну силу як функцію від кута вимірювань, що спрощує обчислення на кінцевому етапі алгоритму.

$$f_{rep}(\theta_i) = \sum_{k=1}^n A_k \exp\left(-\frac{(\theta_k - \theta_i)^2}{2\sigma_k^2}\right) \quad (15),$$

Наступним кроком є обчислення атрактивного поля (16). Це поле описує силу, яка притягує робота у заданому напрямку θ_{goal} . Таким чином результуюче поле обчислюється згідно із рівнянням (17) та визначає безпечний напрямок руху робота на поточній ітерації (18).

$$f_{attr}(\theta_i) = \gamma |\theta_{goal} - \theta_i| \quad (16)$$

$$f_{total}(\theta_i) = f_{attr}(\theta_i) + f_{rep}(\theta_i) \quad (17)$$

$$\theta_{dir} = \arg \min(f_{total}) \quad (18)$$

У рівнянні (16) варто звернути увагу на коефіцієнт γ , який задає силу притягання до цілі та не є універсальним. Тобто в існуючих методах його потрібно підбирати під робоче середовище. Якщо значення параметра занадто мале – робот об'їде перешкоду, але шлях буде не оптимальним. А при збільшенні цього значення – шлях стає більш оптимальним, проте зростає ймовірність колізії із перешкодами. Безпечний напрямок руху обчислюється як значення аргументу за якого функція сумарного поля набуває мінімуму. Це також доволі важлива модифікація, адже спрощує обчислення у порівнянні із класичним методом. У класичному методі напрямок руху визначається як арктангенс між двома векторами, що ускладнює обчислення для систем із обмеженими ресурсами. Ще однією перевагою такого підходу є робота суто із даними від ультразвукових сенсорів відстані чи одновимірних LIDAR сенсорів. Такий підхід до вибору формату даних суттєво спрощує обчислення особливо у порівнянні з алгоритмами, які визначають та виділять перешкоди із зображень (комп'ютерний зір та обробка зображень). Також варто зазначити, що автори проводили експерименти для статичного та динамічного робочого середовища і стверджують, що алгоритм не потребує додаткових коригувань чи змін для роботи із рухомими перешкодами.

Висновки.

Навігація та планування шляху являють собою одні із центральних проблем у розробці мобільних та автономних роботів. Дослідження у цій галузі проводять вже десятки років і було запропоновано кілька методологій для вирішення цих задач, проте підходи та алгоритми досі потребують покращень за різними показниками. У галузі ці підходи поділяють на класичні чи детерміновані та недетерміністичні або евристичні. У статті наведено короткий огляд типових представників обох класів, а також розширений огляд методів класичного підходу на основі штучних потенціальних полів.

Важливими характеристиками алгоритмів виявлення та уникнення перешкод є збіжність, час обчислень та вимоги до об'єму пам'яті у системі. Потреба у збіжності впливає із вимоги до отримання стабільного чи бажаного стану системи. Цей час варіюється у залежності від обраного алгоритму, природи задачі та початкових умов. Основна мета це зменшувати час збіжності – тобто прийти до бажаного стану як можна швидше. Вимоги до часу обчислень та потрібного об'єму пам'яті впливають із того, що робот повинен реагувати на робоче середовище та зміни в ньому у режимі реального часу, а зазвичай автономні роботи мають доволі обмежені апаратні ресурси. Тому це також важливі характеристики у підборі методу під конкретну задачу та робота. Розглянута модифікація класичного методу штучних потенціальних полів із використанням функції гауса для опису репульсивних сил є прикладом оптимізації методу для роботи у системах із обмеженими ресурсами. Так як на момент написання статті безпілотні літальні апарати із обмеженими ресурсами починають широко застосовуватись – такого типу оптимізації набувають практичної цінності.

Серед розглянутих методів евристичні є відносно новими та починають все більше знаходити практичне застосування. Дослідження на момент написання статті фокусуються на оптимізації існуючих алгоритмів та гібридизації для покращення ефективності. Прикладом такої гібридизації є метод штучних потенціальних полів із використанням нечіткої логіки. Тобто поєднується класичний алгоритм методу штучних потенціальних полів та евристичний підхід – нечітка логіка. Це призводить до певного ускладнення методу, проте вирішує типові проблеми класичного алгоритму у вигляді локальних мінімумів та збільшує оптимальність і плавність шляху.

Також варто звернути увагу на суттєвий прогрес у розробці різноманітних сенсорів. Доволі багато алгоритмів виявлення та уникнення перешкод фокусуються на роботі лише із одним типом сенсорів: наприклад ультразвукових сенсорів відстані чи LIDAR або камери. Кожен із сенсорів та відповідних алгоритмів мають свої переваги та недоліки. Перспективним виглядає застосування кількох типів сенсорів та кількох алгоритмів із подальшим об'єднанням результатів різних алгоритмів для отримання більш оптимального кінцевого результату. Але варто зазначити, що такий підхід буде потребувати більш складного апаратного забезпечення.

Оскільки роботи все більше стають частиною повсякденного життя – цілком можливо, що вони почнуть працювати групами та взаємодіяти для вирішення поставлених задач. Розробки колективних методів для уникнення перешкод та взаємодії між роботами у єдиному робочому середовищі також є перспективним напрямком досліджень.

Підсумовуючи вище сказане – доходимо до висновку, що поступова роботизація багатьох процесів у повсякденному житті чи виробництві породжує високу потребу у дослідженнях у галузі мобільної робототехніки в цілому та методів для виявлення та уникнення перешкод та планування шляху зокрема.

- [1] *V. Lumelsky, A. Stepanov*: Path planning strategies for a point mobile automation moving amidst unknown obstacles of arbitrary shape, *Algorithmica* 2, 403–430, 1987.
- [2] *Borenstein, J.; Koren, Y.* The Vector Field Histogram–Fast Obstacle Avoidance For Mobile Robots. *Robot. Autom. IEEE Trans.* 1991, 7, 278 – 288. <https://doi.org/10.1109/70.88137>.
- [3] *D. Fox, W. Burgard, S. Thrun*, The dynamic window approach to collision avoidance, *IEEE Robotics Au-tom. Mag.* 4(1), 23–33, 1997.
- [4] *J. Mingue*, The obstacle restriction method: Obstacle avoidance in difficult scenarios, *IEEE/RSJ Int. Conf. Intell. Robot Syst. (IROS)*, 2005.
- [5] *O. Khatib*, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '85)*, vol. 2, pp. 500–505, 1985.
- [6] *J. W. Park, H. J. Kwak, Y. C. Kang, and D. W. Kim*. “Advanced Fuzzy Potential Field Method for Mobile Robot Obstacle Avoidance,” *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 6047906, 13 pages, 2016. <https://doi.org/10.1155/2016/6047906>
- [7] *Cho, Jang-Ho, Pae, Dong-Sung, Lim, Myo-Taeg, Kang, Tae-Koo*, A Real-Time Obstacle Avoidance Method for Autonomous Vehicles Using an Obstacle-Dependent Gaussian Potential Field, *Journal of Advanced Transportation*, 2018, 5041401, 15 pages, 2018. <https://doi.org/10.1155/2018/5041401>
- [8] *I. Berizka, R. Romanyshyn, O. Savitskyi*, “Customizable IoT Solution Based on ESP32 MCU”, *International Scientific and Practical Conference “Electronics and Information Technologies” (ELIT-2022)*, Issue 19. P. 75–85, 2022.
- [9] *Bruno Siciliano, Oussama Khatib*, “Springer Handbook of Robotics”, 2nd Edition, 2016.
- [10] *Dijkstra, E.W.* A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: His Life, Work, and Legacy*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 287–290.
- [11] *Sabo, C.; Cohen, K.* Fuzzy logic unmanned air vehicle motion planning. *Adv. Fuzzy Syst.* 2012, 2012, 13–13.
- [12] *Gonzalez, R.; Kloetzer, M.; Mahulea, C.* Comparative study of trajectories resulted from cell decomposition path planning approaches. In *Proceedings of the 2017 21st International Conference on System Theory, Control and Computing (ICSTCC)*, Sinaia, Romania, 19–21 October 2017; pp. 49–54.
- [13] *Kirono, S.; Arifianto, M.I.; Putra, R.E.; Musoleh, A.; Setiadi, R.* Graph-based modeling and dijkstra algorithm for searching vehicle routes on highways. *Int. J. Mech. Eng. Technol. (IJMET)* 2018, 9, 1273–1280.
- [14] *Wang, C.; Cheng, C.; Yang, D.; Pan, G.; Zhang, F.* Path planning in localization uncertain environment based on Dijkstra method. *Front. Neurorobot.* 2022, 16, 821991.
- [15] *Y.Singh S.Sharma, R.Sutton, D.* Optimal path planning of an unmanned surface vehicle in a real-time marine environment using a dijkstra algorithm. *Mar. Navig.* 2017, 399–402.
- [16] *Chen, R.; Hu, J.; Xu, W.* An RRT-Dijkstra-based path planning strategy for autonomous vehicles. *Appl. Sci.* 2022, 12, 11982.
- [17] *Broumi, S.; Bakal, A.; Talea, M.; Smarandache, F.; Vladareanu, L.* Applying Dijkstra algorithm for solving neutrosophic shortest path problem. In *Proceedings of the 2016*

- International Conference on Advanced Mechatronic Systems (ICAMechS), Melbourne, Australia, 30 November–3 December 2016; pp. 412–416.
- [18] *Dhulkeft, E.; Durdu, A.; Terzio ğlu, H.* Dijkstra Algorithm Using Uav Path Planning. *Konya J. Eng. Sci.* 2020, 8, 92–105.
- [19] *Yufka, A.; Parlaktuna, O.* Performance comparison of bug algorithms for mobile robots. In *Proceedings of the 5th International Advanced Technologies Symposium*, Karabuk, Turkey, 13–15 May 2009; pp. 13–15.
- [20] *Neloy, M.; Das, M.; Barua, P.; Pathak, A.; Rahat, S.U.* An intelligent obstacle and edge recognition system using bug algorithm. *Am. Sci. Res. J. Eng. Technol. Sci.* 2020, 64, 133–143.
- [21] *Wang, X.; Yin, Y.; Jing, Q.* Maritime Search Path Planning Method of an Unmanned Surface Vehicle Based on an Improved Bug Algorithm. *J. Mar. Sci. Eng.* 2023, 11, 2320.
- [22] *Dong, T.; Zhang, Y.; Xiao, Q.; Huang, Y.* The Control Method of Autonomous Flight Avoidance Barriers of UAVs in Confined Environments. *Sensors* 2023, 23, 5896.
- [23] *LaValle, S.* Rapidly-Exploring Random Trees: A New Tool for Path Planning; Research Report 9811; 1998. Available online: <https://msl.cs.illinois.edu/~lavalle/papers/Lav98c.pdf>
- [24] *Jang, D.u.; Kim, J.s.* Development of Ship Route-Planning Algorithm Based on Rapidly-Exploring Random Tree (RRT*) Using Designated Space. *J. Mar. Sci. Eng.* 2022, 10, 1800.
- [25] *Pérez-Hurtado, I.; Martínez-del Amor, M.Á.; Zhang, G.; Neri, F.; Pérez-Jiménez, M.J.* A membrane parallel rapidly-exploring random tree algorithm for robotic motion planning. *Integr. Comput.-Aided Eng.* 2020, 27, 121–138.
- [26] *Shi, Y.; Li, Q.; Bu, S.; Yang, J.; Zhu, L.* Research on intelligent vehicle path planning based on rapidly-exploring random tree. *Math. Probl. Eng.* 2020, 2020, 1–14.
- [27] *Kang, J.-G.; Lim, D.-W.; Choi, Y.-S.; Jang, W.-J.; Jung, J.-W.* Improved RRT-Connect Algorithm Based on Triangular Inequality for Robot Path Planning. *Sensors* 2021, 21, 333. <https://doi.org/10.3390/s21020333> .
- [28] *Wei, K.; Ren, B.* A Method on Dynamic Path Planning for Robotic Manipulator Autonomous Obstacle Avoidance Based on an Improved RRT Algorithm. *Sensors* 2018, 18, 571. <https://doi.org/10.3390/s18020571>
- [29] *Luo, S.; Zhang, M.; Zhuang, Y.; Ma, C.; Li, Q.* A survey of path planning of industrial robots based on rapidly exploring random trees. *Front. Neurorobot.* 2023, 17, 1268447
- [30] *Jang, D.u.; Kim, J.s.* Development of Ship Route-Planning Algorithm Based on Rapidly-Exploring Random Tree (RRT*) Using Designated Space. *J. Mar. Sci. Eng.* 2022, 10, 1800
- [31] *Löfgren, K.* Rapidly-Exploring Random Trees for real-time combined Exploration and Path Planning. 2023
- [32] *CHANG Xin-xin, HU Wei, JI Shu-de, YUE Yu-mei.* Obstacle Avoidance of Mobile Robot Based on Improved Dynamic Window Method[J]. *Modular Machine Tool & Automatic Manufacturing Technique*, 2021, 0(7): 33-36,39 <https://doi.org/10.13462/j.cnki.mmtamt.2021.07.008>
- [33] *Kim, J., Yang, GH.* Improvement of Dynamic Window Approach Using Reinforcement Learning in Dynamic Environments. *Int. J. Control Autom. Syst.* 20, 2983–2992 (2022). <https://doi.org/10.1007/s12555-021-0462-9>

- [34] *Hart, P.E.; Nilsson, N.J.; Raphael, B.* A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* 1968, 4, 100–107
- [35] *Yao, J.; Lin, C.; Xie, X.; Wang, A.J.; Hung, C.C.* Path Planning for Virtual Human Motion Using Improved A* Star Algorithm. In *Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations, Las Vegas, NV, USA, 12–14 April 2010*; pp. 1154–1158
- [36] *Guan, W.; Wang, K.* Autonomous collision avoidance of unmanned surface vehicles based on improved A-star and dynamic window approach algorithms. *IEEE Intell. Transp. Syst. Mag.* 2023, 113, 102755
- [37] *Tang, G.; Tang, C.; Claramunt, C.; Hu, X.; Zhou, P.* Geometric A-star algorithm: An improved A-star algorithm for AGV path planning in a port environment. *IEEE Access* 2021, 9, 59196–59210
- [38] *Gao, X.; Jia, Q.; Sun, H.; Chen, G.* Research on path planning for 7-DOF space manipulator to avoid obstacle based on A* algorithm. *Sens. Lett.* 2011, 9, 1515–1519
- [39] *Bremermann, H.J.* *The Evolution of Intelligence: The Nervous System as a Model of Its Environment*; University of Washington, Department of Mathematics: Washington, DC, USA, 1958
- [40] *Adzhar, N.; Salleh, S.; Yusof, Y.; Ahmad, M.A.* Routing problem in rectangular mesh network using shortest path based Greedy method. In *Proceedings of the Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2019; Volume 1358, p. 012079.
- [41] *Kumar, A.; Kumar, P.B.; Parhi, D.R.* Intelligent navigation of humanoids in cluttered environments using regression analysis and genetic algorithm. *Arab. J. Sci. Eng.* 2018, 43, 7655–7678
- [42] *Roberge, V.; Tarbouchi, M.; Labonté, G.* Fast genetic algorithm path planner for fixed-wing military UAV using GPU. *IEEE Trans. Aerosp. Electron. Syst.* 2018, 54, 2105–2117
- [43] *Liu, X.; Jiang, D.; Tao, B.; Jiang, G.; Sun, Y.; Kong, J.; Tong, X.; Zhao, G.; Chen, B.* Genetic algorithm-based trajectory optimization for digital twin robots. *Front. Bioeng. Biotechnol.* 2022, 9, 793782
- [44] *Li, D.; Wang, L.; Cai, J.; Wang, A.; Tan, T.; Gui, J.* Research on mobile robot path planning based on improved genetic algorithm. *Int. J. Model. Simul. Sci. Comput.* 2023, 14, 2341030
- [45] *A. Sepehri and A. M. Moghaddam,* "A Motion Planning Algorithm for Redundant Manipulators Using Rapidly Exploring Randomized Trees and Artificial Potential Fields," in *IEEE Access*, vol. 9, pp. 26059-26070, 2021, doi: [10.1109/ACCESS.2021.3056397](https://doi.org/10.1109/ACCESS.2021.3056397)
- [46] *Benzaouia, A., El Hajjaji, A.* (2014). Introduction to Takagi–Sugeno Fuzzy Systems. In: *Advanced Takagi–Sugeno Fuzzy Systems. Studies in Systems, Decision and Control*, vol 8. Springer, Cham. https://doi.org/10.1007/978-3-319-05639-5_1
- [47] *Berizka I.A., Karbovnyk I.D,* Mathematical Model of Modified Real-Time Obstacle Avoidance Method Based on Laplace Artificial Potential Field. *Прикладні проблеми комп'ютерних наук, безпеки та математики*, вип 3, 12–22, 2024. <https://apcssm.vnu.edu.ua/index.php/Journalone/article/view/123>
- [48] *Yu J, Su Y and Liao Y,* The Path Planning of Mobile Robot by Neural Networks and Hierarchical Reinforcement Learning. *Front. Neurobot.* 14:63. 2020. <https://doi.org/10.3389/fnbot.2020.00063>

**PATH PLANNING AND OBSTACLE AVOIDANCE METHODS
FOR AUTONOMOUS MOBILE ROBOTS****I. Berizka**

*Ivan Franko National University of Lviv,
107 Tarnavskoho Str., Lviv, 79005, Ukraine
ihor.berizka@lnu.edu.ua*

Navigation and path planning are among the central problems in the development of mobile and autonomous robots. Research in this field has been conducted for decades, and several methodologies have been proposed to solve these problems. In the field, these approaches are divided into classical or deterministic and non-deterministic or heuristic methods. The article provides a brief overview of typical representatives of both classes, as well as an extended review of methods based on artificial potential fields.

Important characteristics of obstacle detection and avoidance algorithms include convergence, computation time, and memory requirements in the system. The need for convergence arises from the requirement to achieve a stable or desired state of the system. This time varies depending on the chosen algorithm, the nature of the task, and the initial conditions. The main goal is to reduce convergence time, i.e., to reach the desired state as quickly as possible. Computation time and memory requirements are important because the robot must respond to the working environment and changes in it in real-time, and autonomous robots usually have quite limited hardware resources. Therefore, these are also important characteristics when selecting a method for a specific task and robot. The modification of the classical artificial potential field method using the Gaussian function to describe repulsive forces is an example of optimizing the method for systems with constrained resources. As of the writing of the article, unmanned aerial vehicles with limited resources are beginning to be widely used, making such optimizations practically valuable.

Among the considered methods, heuristic ones are relatively new and are increasingly finding practical application. Research at the time of writing focuses on optimizing existing algorithms and hybridization to improve efficiency. An example of such hybridization is the artificial potential field method using fuzzy logic. This combines the classical artificial potential field method with a heuristic approach—fuzzy logic. This leads to some complexity in the method but solves typical problems of the classical algorithm, such as local minima, and increases the optimality and smoothness of the path.

Most of obstacle detection and avoidance algorithms are working with only one type of sensor, such as ultrasonic distance sensors, LIDAR, or cameras. Each sensor technology and corresponding algorithms have their advantages and disadvantages. A promising approach is to use several types of sensors and algorithms, combining the results of different algorithms to achieve a more optimal final result, so called sensor fusion. However, it should be noted that this approach will require more sophisticated hardware.

As robots increasingly become part of everyday life, it is quite possible that they will start working in collaboratively and interacting to solve assigned tasks. The development of collaborative methods for obstacle avoidance and interaction between robots in a single working environment is also a promising research direction.

In summary, the gradual robotization of many processes in everyday life or production generates a high demand for research in the field of mobile robotics in general and methods for obstacle detection, avoidance and path planning in particular.

Key words: robotics, obstacle avoidance, path planning, artificial potential field, autonomous robots, mobile robots.

*Стаття надійшла до редакції 12.09.2023
Прийнята до друку 15.10.2023*