Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Path planning control in known environments using turning points

Areej Ghazi Abdulshaheed *, Farah Kamil

Department of Mechanical Engineering, Technical Institute of AL-Diwaniyah, AL-Furat AL-Awsat Technical University, Najaf, Iraq

ARTICLE INFO

Article history:
Received 19 July 2024
Received in revised form
21 October 2024
Accepted 21 November 2024

Keywords:
Path planning
Mobile robots
Collision-free
Local minima
Navigation reliability

ABSTRACT

The path planning problem for a wheeled mobile robot (WMR) involves determining a collision-free path from a starting point to a target destination while optimizing a specific fitness function, such as minimizing distance, cost, or both, depending on the scenario. This research introduces a novel method for generating smooth paths for mobile robots in user-defined two-dimensional environments with stationary obstacles. The proposed method addresses the issue of local minima by utilizing free segments and path-planning turning points. The approach evaluates both path length and path safety as key objectives. Simulation results demonstrate that the proposed method effectively identifies optimal paths and validates the reliability of the control strategy for mobile robot navigation.

1. Introduction

Mobile robots are used in many fields, and one of their most important uses is autonomous navigation in environmental space (Yang et al., 2022). Robots replaced humans in routine and hazardous tasks (Abdulshaheed et al., 2024). Path planning is the foundation of navigation, which identifies feasible routes from the start state to the goal state without running into any barriers (Qin et al., 2023). A possible path in a workplace with obstacles is safe, collision-free, and either optimal or suboptimal based on a few performance characteristics (path smoothness, walking path, planning time) (Zhang et al., 2022). Depending on the amount of environmental knowledge available, path planning can be broadly classified as local or global (Liu et al., 2023). Global path planning refers to the ability of robots in environments to plan their whole pathways (offline) before they start moving since they are aware of all stationary obstacles and the trajectory of all moving obstacles beforehand (Yao et al., 2023). Updating the local map to avoid impediments while working is the primary goal of local path planning (Xu et al., 2023).

Numerous studies that tackle the problem of path planning have been presented in the literature (Rafai et al., 2022; Tan et al., 2021; Jogeshwar and Lochan, 2022). To date, a wide range of methods have been used for mobile robot path planning in static situations. A novel incentive mechanism is introduced to guarantee that a mobile robot is aware of its surroundings beforehand. In addition to guaranteeing a quick convergence, innovative mathematical modeling is suggested to offer the best choice (Bulut, 2022). A mobile robot finds it difficult to navigate a path with sharp corners, so a smooth path is created once the ideal skeletal path has been found. Moreover, an actual experiment based on the multi-objective function is provided. The suggested IEGQL method is benchmarked against the traditional EGQL and A-star algorithms. A new definition of state space and actions space is included in the algorithm, along with a selection strategy that aims to

- Assist the robot in choosing the best course of action at each stage
- Speed up the learning process
- Find optimal or nearly optimal solutions
- Take advantage of a new reward function in the initialization of Q-tables

An improved adaptive ant colony algorithm (IAACO) in response to the drawbacks of the traditional ant colony algorithm (ACO) is proposed in path planning of indoor mobile robots, including a long path planning time, a non-optimal path for the slow convergence speed, and the local optimal solution characteristic of ACO (Miao et al., 2021). Li et al. (2022b) described a Forward Search

* Corresponding Author.
Email Address: areej.ghazy.idi25@atu.edu.iq (A. G. Abdulshaheed)
https://doi.org/10.21833/ijaas.2024.12.015
Corresponding author's ORCID profile:
https://orcid.org/0000-0002-2343-929X

Optimization (FSO) technique for shortening the global path provided by the searching-based Global Path Planning (GPP) method. The adjusted path is usually shorter than the original path and is like the optimal path in continuous space. Furthermore, a Subgoal-based Hybrid Path Planning (SHPP) strategy that combines the Improved Fuzzy Inference System (IFIS) and Improved Artificial Potential Field (IAPF) algorithms is proposed to smooth the global path. The key improvement to IFIS and IAPF is that subgoals rather than global goals govern planning. Another study (Li et al., 2022a) includes a bidirectional alternating search (BAS) technique in the A* algorithm, increasing search efficiency. The revised heuristic function addresses the inadequacies of the BAS-A* method. The filtering function and Bézier curves are used to lower the turning angle and path length while also smoothing the path.

The proposed algorithm's feasibility is tested using the TurtleBot3 Waffle Pi mobile robot. A new obstacle-based path-planning algorithm for a grid map is proposed (Deng et al., 2021). After convex hull optimization, the simple point set is generated. Hierarchical obstacles are used to swiftly plan a path on a small scale. The length and smoothness are optimized using the multi-objective D* Lite algorithm. Cubic Bezier curves are used to soften the path so that it fits the real robot. In Xu et al. (2023), a novel strategy was developed to find the optimum search node. The map was a rectangle that could be explored in two different directions. The adaptive cost function was used to improve safety. A Slide-Rail corner adjustment method was developed to achieve a smoother path.

A Hybrid algorithm combining the improved A* algorithm and the Dynamic Window A approach is addressed in Li et al. (2022c). In both static and dynamic contexts, the algorithm presented attempted to solve a path-planning problem for an autonomous mobile robot by finding the collision-free path that satisfies the requirements for shortest distance and path smoothness. The recommended path planning method approximates the real world by multiplying the size of the mobile robot by the size of the obstacles and formulating the problem as a moving point in free space. A method for enhancing mobile robot systems' performance for the best possible path planning is suggested by Al-Kamil and Szabolcsi (2024). The method makes use of motion capture technologies to gather movement data from the robot in real-time, create the best possible path planning schemes, and allow for remote control and activity monitoring.

Zhao et al. (2023) explored a new two-stage global route planning and path control method for unmanned surface vehicles (USVs) that uses path navigation. Using the traveling salesman problem (TSP), a global path is initially found in the first step by maximizing profit per unit of time across several task locations. The second step is to create a nonlinear multi-objective optimization model to control the route between two task locations. An enhanced NSGA-II is suggested to address the multi-objective path planning challenges to minimize path length while optimizing path safety and smoothness (Duan et al., 2024). To provide precise and efficient solutions, three problem-specific evolutionary operators were examined. Two navigation strategies were developed by Hassani et al. (2022), where the free segments algorithm is only used in complex environments, while the turning point algorithm is used as a navigational strategy in basic environments.

Zhang et al. (2024) presented an improved A-star-based path planning algorithm based on diverse turning angles to curtail unnecessary turns. The presented method integrates the artificial potential field method, introducing a unique heuristic function. This function includes a penalty term accounting for obstacle information at turning points, ensuring that these points are strategically positioned away from obstacles. A turning point method is introduced by Li et al. (2021) to facilitate improved traffic efficiency through infrastructure and vehicle cooperation. Based on the connected autonomous vehicle highway system's framework, a junction management system employing this technique is put into place. With the use of such a system, roadside infrastructure may effectively gather vehicle state data, reserve the corresponding intersection time-space occupancy, and then give vehicle input on planning and decision-making.

The issues of decision-making at highly unpredictable crossings of varied forms are addressed by Shu et al. (2021) with their proposed hierarchical planning and decision-making system based on the critical turning point (CTP). Behavior-oriented pathways can be generated using the proposed CTP approach calibrated with naturalistic driving datasets. The described two-dimensional partially observable Markov decision-making problem can be handled in real-time by using CTPs in place of lateral accelerations along with longitudinal accelerations as choice variables. Another study proposes a global path-planning algorithm based on the directionality of line segment features and the feature map (Ren et al., 2022).

The suggested method uses the distances between the point (robot) and the line segment (obstacle) to evaluate the robot-obstacle relationship. Additionally, a turning point for path planning based on the map-matching method is presented in Zhang et al. (2021). He offers the idea of vehicle turning points to apply map-matching pieces appropriately. It also suggests a revolutionary point-based offline map-matching technique. Numerous offline map-matching techniques have been put forth to address the situation of trajectory data with a low sampling rate (Ou et al., 2022; Wang et al., 2023; Safarzadeh and Wang, 2024; Yu et al., 2022). These matching algorithms usually first set numerous candidate-matched positions for each trajectory point and then compute the shortest path between each pair of candidate positions at every two subsequent points to get the precise travel path

between those places. These methods have a considerable computation time due to several shortest-path calculations, even if they can achieve excellent matching accuracy.

A local minimum is considered a common problem in path planning. A novel method for local minimum avoidance is introduced (Szczepanski et al., 2022). It is based on the placement of virtual obstacles called top quarks in critical areas. These obstacles provide additional repulsive force for the Artificial Potential Field (APF) based path planner. Considering the predicted stagnation-free path of the autonomous ground vehicle AGV, the new temporary goal for APF is selected. Another research (Wu et al., 2023) proposed a method based on a deterministic annealing strategy to improve the potential field function by introducing a temperature parameter to increase the robot's obstacle avoidance efficiency. The annealing and tempering strategies prevent the robot from being trapped at the local minima and allow it to continue toward its destination. The initial path is optimized using an annealing algorithm to enhance the overall performance.

Our contribution is the creation of a novel algorithm for the robot path planning problem in a known environment with static obstacle avoidance. The benefit of this planning algorithm is that it guarantees both path safety and path brevity. Additionally, the suggested algorithm exhibits reactive behavior in locating a smooth path that is free of collisions. Conversely, the mobile robot needs to follow the path without running into any obstructions. Thus, to provide stability, responsiveness, and robustness, a turning mode control is suggested.

The following is how this paper is organized: The background, relevance, and literature evaluation of the topic are covered in the first section, which is the introduction. The mobile robot's kinematics are shown in the second section. The algorithm's principle is explained in the third section, along with the analysis and description of issues like obstacle presentation, choosing a turning point search direction, avoiding obstacle endpoints, and the idea and procedure of path optimization. Using a comparison simulation experiment, the fourth portion demonstrates the algorithm's clear superiority in terms of increasing computational efficiency. Lastly, the paper's conclusion will be presented in Section 7.

2. Kinematics of a mobile robot

Nonholonomic mechanical system requires the definition of two distinct coordinate systems to characterize its position. The first is the fixed inertial coordinate system {*X, Y*} in the plane or environment that the WMR moves in. This frame serves as the standard frame.

The second one is the mass center coordinate system's (X_c, Y_c) orientation angle (θ), and it is a local frame connected to the WMR called the robot coordinate system (X_r, Y_r). The parameter of the Differential Wheeled Robot (DWR) is shown in Fig. 1 and listed in Table 1.

The DWR shown in Fig. 1 presents three constraints. The first constraint is the velocity in the inertial frame which can be written as:

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0. \tag{1}$$

The other two constraints have to do with how the wheels rotate, and are given by:

$$\dot{y}\cos\theta + \dot{x}\sin\theta + a\dot{\theta} - r\dot{\phi}_r = 0 \tag{2}$$
$$\dot{x}\cos\theta + \dot{y}\sin\theta + a\dot{\theta} - r\dot{\phi}_l = 0. \tag{3}$$

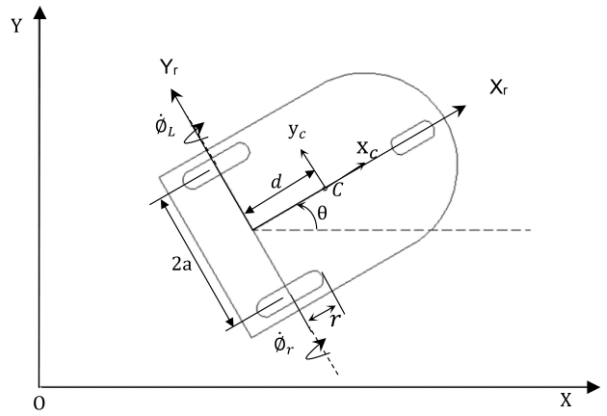


Fig. 1: Coordinate system and WMR

Table 1: DWR parameters

Parameter	Description
C	Mass center of guidance point
d	The separation between C and the point where the wheels' axis and the symmetry axis cross
r	Right and left wheel radius
2a	The separation between the symmetry axis and the actuated wheels
$\dot{\phi}_r, \dot{\phi}_L$	Angular velocity of the right and left wheels
m_c	Mass of the DWR without wheel and motor
m_w	The mass of each wheel and motor assembly
m_t	The total mass of the DWR
I_w	The moment of inertia of each wheel and motor about the vertical axis
I_m	Each wheel's and motor's moment of inertia to the vertical axis parallel to the wheel plane
I	Total inertia moment of the DWR

Let $\bar{q} = (x, y, \theta)^T$ denote the posture vector. where, (*x,y*) denotes the position of the mobile robot, and θ is the angle between the X-coordinate and the heading position. Where: $\gamma = (v, \omega)^T$ is the control vector; v is the linear velocity; ω is the angular velocity. All symbols that have been used in the equations should be defined in the following text.

$$\dot{\bar{q}} = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{4}$$

For the Differential Drive Mobile Robot (DDMR), the generalized coordinates are selected as

$$q = [\bar{q}^T \ \emptyset^T]^T = [x\ y\ \theta\ \emptyset_r\ \emptyset_L]^T \tag{5}$$

Eqs. 1-3 can be described in matrix form as:

$$A(q)\dot{q} = 0. \tag{6}$$

The three constraints can be reformulated as Eq. 4,

$$A(q)\dot{q} = \begin{bmatrix} -\sin(\theta) & \cos(\theta) & -d & 0 & 0 \\ -\cos(\theta) & -\sin(\theta) & -a & r & 0 \\ -\cos(\theta) & -\sin(\theta) & a & 0 & r \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\emptyset}_r \\ \dot{\emptyset}_l \end{bmatrix}. \quad (7)$$

We assumed the mobile robot had driving wheels. Here, the kinematic model of a nonholonomic mobile robot is given:

$$v = \frac{1}{2}(V_R + V_L) \quad (8)$$

where, V_R, V_L are the velocity vectors to the right and left of the robot.

$$\dot{x} = \frac{V_R+V_L}{2} cos\,\theta \quad (9)$$

$$\dot{x} = \frac{V_R+V_L}{2} cos\,\theta \quad (10)$$

$$\dot{y} = \frac{V_R+V_L}{2} sin\,\theta \quad (11)$$

$$\dot{\theta} = \frac{V_R-V_L}{2} \quad (12)$$

3. Dynamic model

The DDMR dynamic model is crucial for both designing different motion control algorithms and doing a simulation study of the DDMR's motion. One effective way to formulate the equations of motion of mechanical systems is to use the Lagrange formulation for the case of DDMR with standard fixed wheels (Rodríguez-Molina et al., 2022).

$$\frac{d}{dt}\left(\frac{\partial T}{\partial q}\right) - \frac{\partial T}{\partial q} = -A^T(q)\rho + E(q)\tau \quad (13)$$

where, T represents the Kinetic energy. *A(q)* is a *p×n* matrix associated with nonholonomic constraints, ρ is a *p×1* vector of constraint forces, τ is a *p×1* vector of control or input torque, ρ as in Dhaouadi and Hatab (2013) which represents the Lagrange multipliers vector. The total Kinetic energy of DDMR is expressed by:

$$T = \frac{1}{2} q^T\, H(q)\dot{q} \quad (14)$$

where, H(q) is the *n×n* positive definite symmetric inertia matrix-Eq. 13 can be written as:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} = -A^T(q)\rho + E(q)\tau \quad (15)$$

where, $C(q,\dot{q})$ is the matrix of Coriolis and centripetal Torques, considering a dynamic system with uncertainties and disturbances. In this model, the matrix *A(q)* encapsulates the restrictions on movement. of the DDMR, supposing that the DDMR only travels in the direction of the symmetry axis and that there is no sliding.

$$A(q) = \begin{bmatrix} -\sin(\theta) & \cos(\theta) & -d & 0 & 0 \\ -\cos(\theta) & -\sin(\theta) & -a & r & 0 \\ -\cos(\theta) & -\sin(\theta) & a & 0 & r \end{bmatrix}, \quad (16)$$

the matrices $H(q)$, $C(q,\dot{q})$ and $E(q)$ are given by:

$$H(q) = \begin{bmatrix} m_t & 0 & m_t d\sin\theta & 0 & 0 \\ 0 & m_t & -m_t d\cos\theta & 0 & 0 \\ m_t d\sin(\theta) & -m_t d\cos(\theta) & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix} \quad (17)$$

where,

$$m_t = m_c + 2m_w,\ I = m_c d^2 + I_c + 2m_w(d^2 + a^2) + 2I_m$$

$$C(q,\dot{q})q = \begin{bmatrix} m_t d\dot{\theta}^2\cos\theta \\ m_t d\dot{\theta}^2\sin\theta \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (18)$$

$$E(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (19)$$

Finally, the vector of Torque is given by

$$\tau = \begin{bmatrix} \tau_r \\ \tau_L \end{bmatrix}, \quad (20)$$

where, τ_r and τ_L represent respectively the torques on the torques on the right and left wheels of the DDMR. To eliminate the constraint forms $A^T(q)\rho$ in Eq. 15 since the Lagrange multipliers ρ are unknown. This is done by defining the velocity of the right and left wheels. Where:

$$\gamma = \begin{bmatrix} \dot{\emptyset}_r \\ \dot{\emptyset}_L \end{bmatrix}, \quad (21)$$

by generalizing the coordinate velocities, we get

$$\dot{q} = S(q)\gamma, \quad (22)$$

$$S(q) = \frac{1}{2}\begin{bmatrix} r\cos\theta & r\cos\theta \\ r\sin\theta & r\sin\theta \\ \frac{r}{a} & \frac{r}{a} \\ 2 & 0 \\ 0 & 2 \end{bmatrix}. \quad (23)$$

where, $S(q)$ refers to the null spaces of the constraint matrix $A(q)$.

$$S^T(q)A^T(q) = 0. \quad (24)$$

By differentiating Eq. 22 and substituting the result into Eq. 15, and then pre-multiplying by $S^T(q)$ we can get.

$$S^T(q)\,H(q)S(q)\dot{\gamma} + S^T(q)\left[H(q)\dot{S}(q) + C(q,\dot{q})\,S(q)\right]\gamma = S^T(q)\,E(q)\tau, \quad (25)$$

Eq. 25 can be rewritten as:

$$\bar{H}(q)\dot{\gamma} + \bar{C}(q,\dot{q})\gamma = \bar{E}(q)\tau, \quad (26)$$

where,

$$\bar{H}(q) = S^T(q)\,H(q)S(q), \quad (27)$$

$$\bar{C}(q,\dot{q}) = S^T(q)\left[H(q)\dot{S}(q) + C(q,\dot{q})\,S(q)\right], \quad (28)$$

$$\bar{E}(q) = S^T(q)\,E(q). \quad (29)$$

4. Path planning algorithm

The suggested algorithm's basic idea is to discover the shortest route between the robot's starting position and the intended destination. The mobile robot will proceed in a straight route if there are no obstacles in the way between the robot P_s current position and its destination P_T. On the other hand, a path issue occurs when the robot encounters barriers that hinder its mobility. The robot needs to figure out a safe route between the barriers to avoid colliding.

To resolve this problem, a path strategy's guiding principles are based on identifying the ends of each obstacle segment and fixing it as the site of collision. The goal of this method is to ensure that the robot turns safely away from this spot. This technique divides the path-planning problem into several steps. The control trajectory for a straight path between two points is given by:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \quad (30)$$

For a specific point trajectory

$$p(x, y) = (x + d\cos\theta\,, y + d\sin\theta). \quad (31)$$

4.1. Representation of obstacles

An object with numerous edges can be used to depict a regular form. Given that, a robot can only observe a section of the shape of an obstacle, as Fig. 2 illustrates.

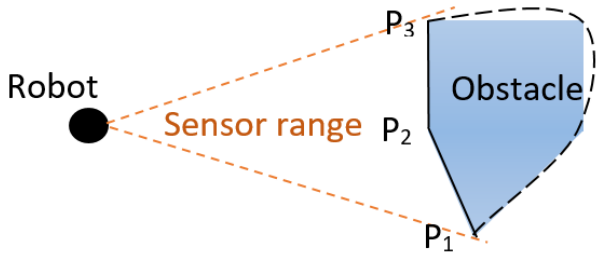


Fig. 2: Configuration of the obstacle boundaries detected by the robot

A series of segments can be used to approximate the visible portion of the obstacle contour S_i^j where j=1, 2, 3, ..., n, where n represents the number of segments of the obstacle. Each segment can be represented by two points p_i^j and p_i^{j+1} and each point has its coordinates $(x_{p_i^j}, y_{p_i^j})$ and $(x_{p_i^{j+1}}, y_{p_i^{j+1}})$ respectively.

Remark 4.1: It is presumed that the obstacle is made up of a series of segments. If two obstacles are sharing a point, then there must be an intersecting point between them. The trajectory path will consider the following point and ignore this point from the path points list. The discussion can be made in several sub-sections.

4.2. Selection of free segments

The robot will identify segments that are free if the distance between the endpoints of two nearby obstacles is greater than the robot's diameter. If two obstacles intersect, a block segment is present and will be shown as a star. The two endpoints of the free segments are indicated when the robot has finished scanning the search region. A circle with a radius of R encircles each endpoint. This circle denotes the area of risk where a robot and the obstacle's edge could conspire. To prohibit collaboration between the robot and the obstacle, R needs to be greater than the robot's radius ($R > r$), where r is the radius of the robot, as Fig. 2 illustrates.

4.3. Elimination of the dangerous path

In a random environment, the distance between two obstacles could be bigger than the diameter of the robot, which can be noted as the free segment, or smaller, which is called the dangerous zone where colliding could occur. In other cases, the two obstacles could be connected at one point or more (Fig. 3). In this case, the intersection points could be identified by comparing the trajectory of each point of the obstacle P_i with the rest of the obstacles. If two points of different obstacles have the same trajectory (x_i, y_i), they will be removed from the OB_list, which refers to the obstacle point's list (see algorithm 2).

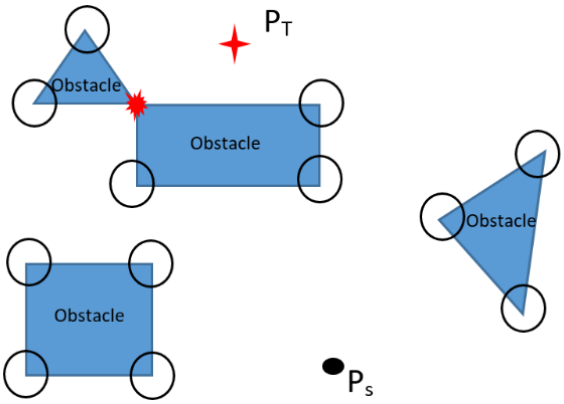


Fig. 3: Complex environment with multi-types of obstacle shapes

4.4. Determining the safe path

The safe path is one of the routes the robot can take to get to the target. The shortest path between these paths is chosen because they are identifiable.

4.4.1. Finding nearest objects to the start points

Since the obstacle is represented by serial segments (p_1, p_2, ..., p_n), the selection of the first segment depends on the relative distance between the robot and the object. Each free segment is represented by a point (x_i^j, y_i^j), see Fig. 4. A comparison between the y-axis of each point of the

objects and the y_s is established to find the nearest object to the robot.

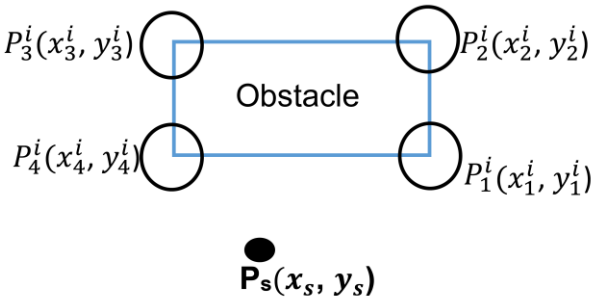


Fig. 4: Obstacle segments

4.4.2. Selection of the turning points

The robot measures the distance between itself and the dangerous circle's tangent line by scanning the surrounding area. The robot enters obstacle boundary mode when it reaches the risky circle. The robot must veer along the path of the obstruction. For example, the robot needs to follow the contours $[\bar{P}_i^1, \bar{P}_i^2, \dots., P_T]$ to reach the goal. It can be illustrated in Fig. 5. Firstly, find out the point $\bar{P}_i^1$ at a distance of R to the point $\bar{P}_i^2$ on the extension of a segment from the endpoint $\bar{P}_i^2$ to $\bar{P}_i^3$. See Algorithm 2.

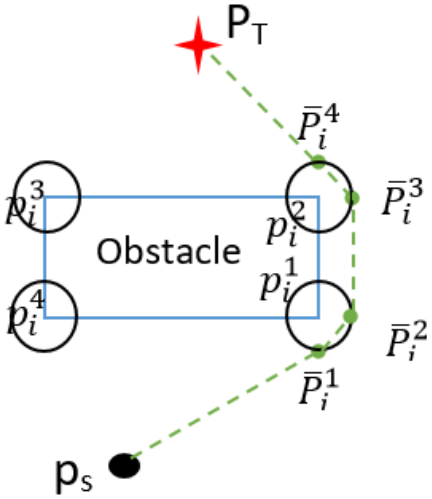


Fig. 5: Path around the dangerous circle

Remark 4.4: A new path trajectory is added if the line between any point of the obstacle and the robot crosses with another obstacle. The path trajectory should be chosen from the obstacles that cross with the line P_sP_T.

4.4.3. Path trajectory selection

Initially, the collection of obstacles identified by the robot's sensors can be described as P= {P_i} for $1 \leq i \leq N$. Where N is the number of obstacles. The joint point set is identified as P_i= {p_i^j } for $1 \leq j \leq n$. Where n is the number of points. Each segment S_i^j is defined by its endpoints $S_i^j = (p_i^j, p_i^{j+1})$. Let path_list= {$\bar{P}_k, P_T$} for $1 \leq i \leq M$ is the path set points. Denote dis (P_k, P_{k+1}) for 1≤ i≤ m-1. The principle that determines the distance between two points P_k and P_{k+1}, where,

$$\text{Dis } (\{Ps\} \cup P\_\text{list}) = \text{dis}(P_s, \bar{P}_l) + \sum_{k=1}^{m-1} \text{dis}(\bar{P}_k, \bar{P}_{k+1}) + \text{dis}(\bar{P}_m, P_T)$$

Dis ({Ps} ∪ P_list) is the function that uses P_list to calculate the total path cost from the robot's current locations (P_s) to the end destination (PT). See Algorithm 1. Path_L and path_R are defined as the two possible paths to the left and right of the obstacle, which lie between the start point and the target being initialized as Path_L=path_R= {P_T}.

4.5. Optimum path selection

If the robot can see the final target, then P_sP_T has no intersection with all obstacles, and the optimal path is the straight line P_sP_T, (Fig. 6). Otherwise, if P_sP_T has intersections with obstacles, the possible optimal trajectory might be from the right region of P_sP_T or the left region. Once the precise obstacle has been identified (in Fig. 7, it is P_i, P_{i+1}) since the segment p_i^j, p_i^{j+1} in P_i={$p_i^1, p_i^2, \dots \dots, p_i^4$} connects with P_sP_T, path trajectory of both sides of P_sP_T is detected. For example, if the right region of P_sP_T is taken, the close point p_i^{j+2} is selected. If the robot is not able to reach the final target after passing p_i^{j+2}, p_i^{j+3}, the robot should go crossing p_i^{j+4}, p_i^{j+5} to observe P_T. Consequently, the previous point would be stored on list_R. Where list_R= {$p_i^{j+2}, p_i^{j+3}, p_i^{j+4}, p_i^{j+5}, P_T$}, and by repeating the same process to the left region, one obtains list_L= { $p_i^{j+1}, p_i^j, p_{i+1}^j, p_{i+1}^{j+1}$, p_{i+1}^{j+2}, p_{i+1}^{j+3}, P_T}. Finally, the optimum path could be obtained by calculating the path cost of these two lists, i.e., If dis(P_s, list_R) > dis(P_s, list_L), then opt_path=list_L otherwise opt_path=list_R.

The routine to generate the list of opt_path selections is given in Algorithm 1 (Algorithm of Path Planning).

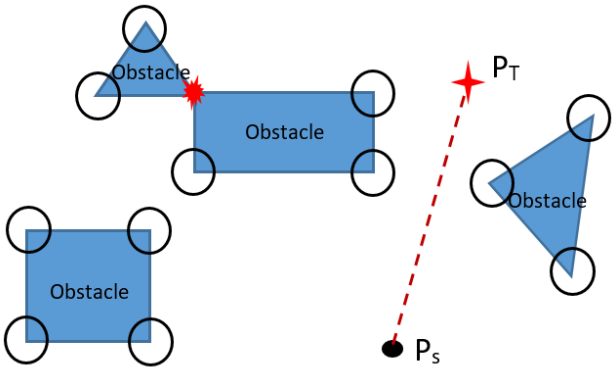


Fig. 6: Optimum path of free segment

4.6. A local minima problem

This problem could occur if each segment is in danger or if obstacles are preventing the robot from moving. A nearby minimum is characterized,

pertinently, by a scenario in which an object becomes stuck and cannot be moved. The item becomes stuck in a configuration where it has not yet reached the target but has nearly reached zero velocity. The robot moves far away from those obstructions until it reaches its destination to escape such a scenario.

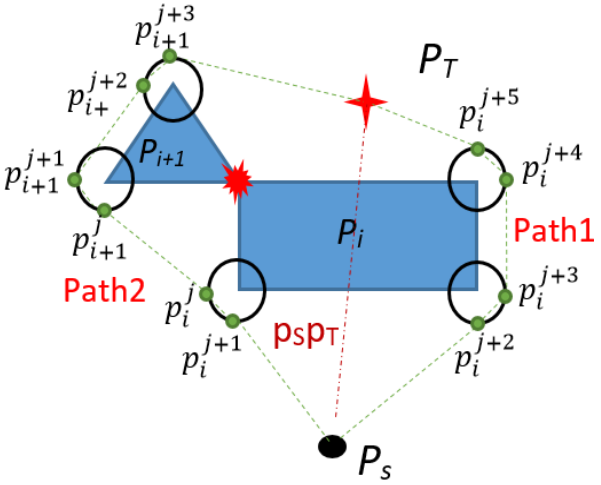


Fig. 7: Two suggested paths around obstacles

Algorithm 1: Path planning

1: Function opt_path (P_T (x_f, y_f), Int. conditions)
2: For each P do
3: P_list= O-generation (P, P_s (x_s, y_s), P_T (x_f, y_f))
4: if $\exists \bar{P}_i^j, \bar{P}_i^{j+1} \in P\_list.\ \bar{P}_i^j \bar{P}_i^{j+1} \cap P_T P_G \neq \emptyset$ *then*
5: select P_i, break
6: end if
7: end for
8: path_L=path_R={P_T}
9: for k=j:1:m do # m is the number of points on P_i
10: if $P_i^j > P_s P_T$ # check the position of the obstacle points
11: path_L={p_i^j} U path_L
12: else
13: path_R={p_i^j} U path_R
14: end if
15: end for

Algorithm 2: Turning point

1: function O-generation (P, P_s(x_s,y_s), P_T(x_t,y_t))
2: for each P do
3: for k=j:1:m
4: ints_list=int_points (P) # intersection points list
5: OB_list= P_i – int_list # removing the intersection points from the obstacles points list
6: get segment $p_i^j p_i^{j+1}$ from OB_list
7: compare p_i^j with p_i^{j+1} with p_i^{j+2}
8: compute points : ($\bar{P}_i$s.t)
9: end for

Algorithm 3: Intersection point

1: function int_point (P)
2: for L=i:1:n # n is the number of obstacles
3: for k=j:1:m # m is the number of points on Pi
4: if $P_i \cap P_{i+1} \neq \emptyset$ then
5: Ints={ $p_i(x, y)$}
6: else
7: ints= 0
8: end for
9: end for

5. Simulation results

The environment's construction is regarded as crucial to the motion planning process in mobile robot navigation. This section contains some simulation results that show the fundamental capabilities of the suggested algorithm. Fig. 8 displays the outcomes of all simulations, which include five randomly placed obstacles in the area. The starting center coordinate of static obstacles is shown in Table 2. The simulation runs in situations where the robot starts from the same point while the target (x_t, y_t) positions are changed.

Table 2: Center coordinate of obstacles

obstacles	Xobs	Yobs
Obstacle1	150	150
Obstacle2	66.66	250
Obstacle3	350	150
Obstacle4	250	250
Obstacle5	325	425

Xobs and Yobs coordinate of the center points of static obstacles

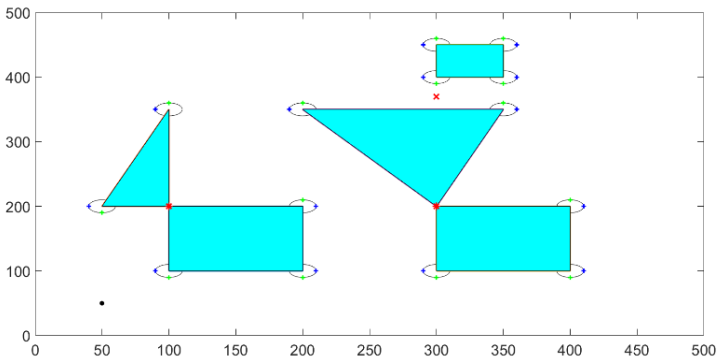
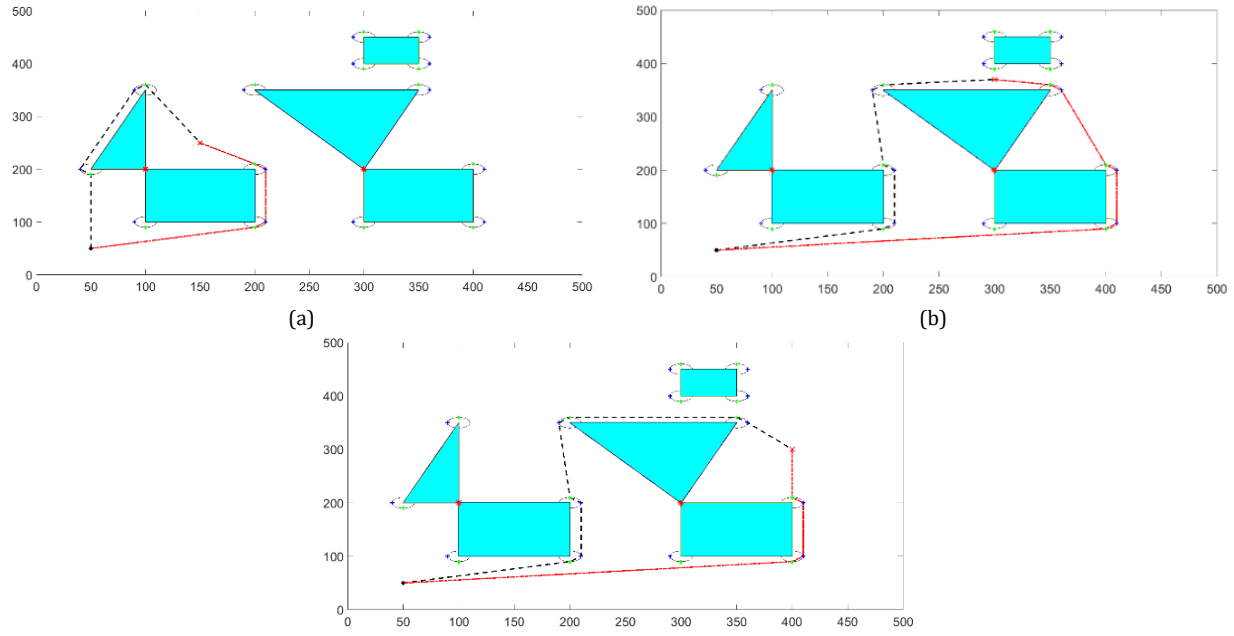


Fig. 8: Environment mapping

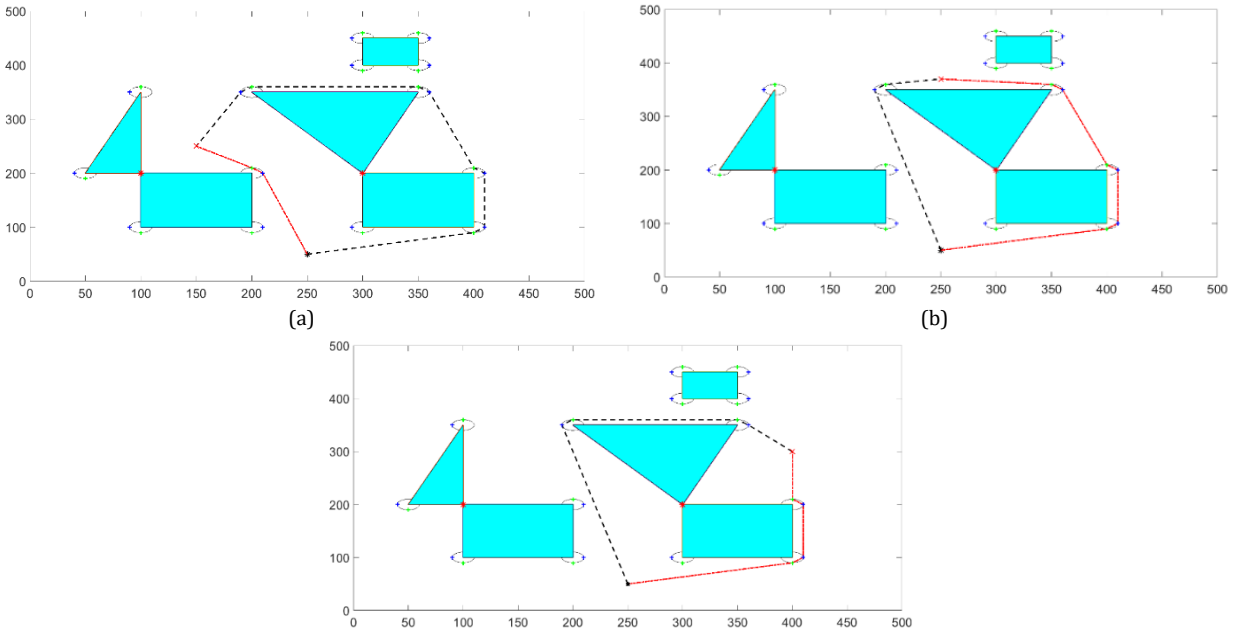
This section covers the scenario in which the robot starts at the starting point (x_s, y_s)=(50, 50), while the target location is selected to be (x_t, y_t)=(150, 250), (300, 370), and (400, 250), as shown in Figs. 9a, 9b, and 9c, in which certain segments are free while others are not. It is observed that the robot can trace two possible paths to reach its destination. The shortest path is denoted in red. The robot avoids collisions with obstacles by turning around the unsafe zone at the end of the obstacle segment. The path navigation will adjust in tandem with the robot's position adjustments.



(a) (b) (c)

Fig. 9: Path planning with (x_s, y_s)= (50, 50), (a) Safe path navigation with (x_t, y_t)= (150, 250), (b) Safe path navigation with (x_t, y_t)= (300, 370), and (c) Safe path navigation with (x_t, y_t)= (400, 300)

Fig. 10 shows how a mobile robot can navigate in different positions (x_s, y_s)= (250, 50) (see Figs. 10a, 10b, and 10c). Meanwhile, the intended destination remains the same as before. Once more, the robot managed to avoid the hazardous area and go along the short, safe path.



(a) (b) (c)

Fig. 10: Path planning with (x_s, y_s)= (250, 50), (a) Safe path navigation with (x_t, y_t)= (150, 250), (b) Safe path navigation with (x_t, y_t)= (250, 370), and (c) Safe path navigation with (x_t, y_t)= (400, 300)

Another simulation finding is shown in the case where the robot begins at location (x_s, y_s)=(350, 50) (see Figs. 11a, 11b, and 11c. Each time the goal position is altered, the robot's trajectory is adjusted. The robot keeps turning only around the safe segments and avoiding the hazardous ones until it reaches its destination. Figs. 12a, 12b, and 12c demonstrate how the mobile robot makes sure to reach the distinction while dodging various obstacles. In this instance, a local minimum issue is identified. As a result, the robot avoids obstructions and moves straight towards the goal.

All simulation results show that the developed method is very reactive because the robot was able to avoid obstacles both in safe and risky areas and when the robot and target location changed.

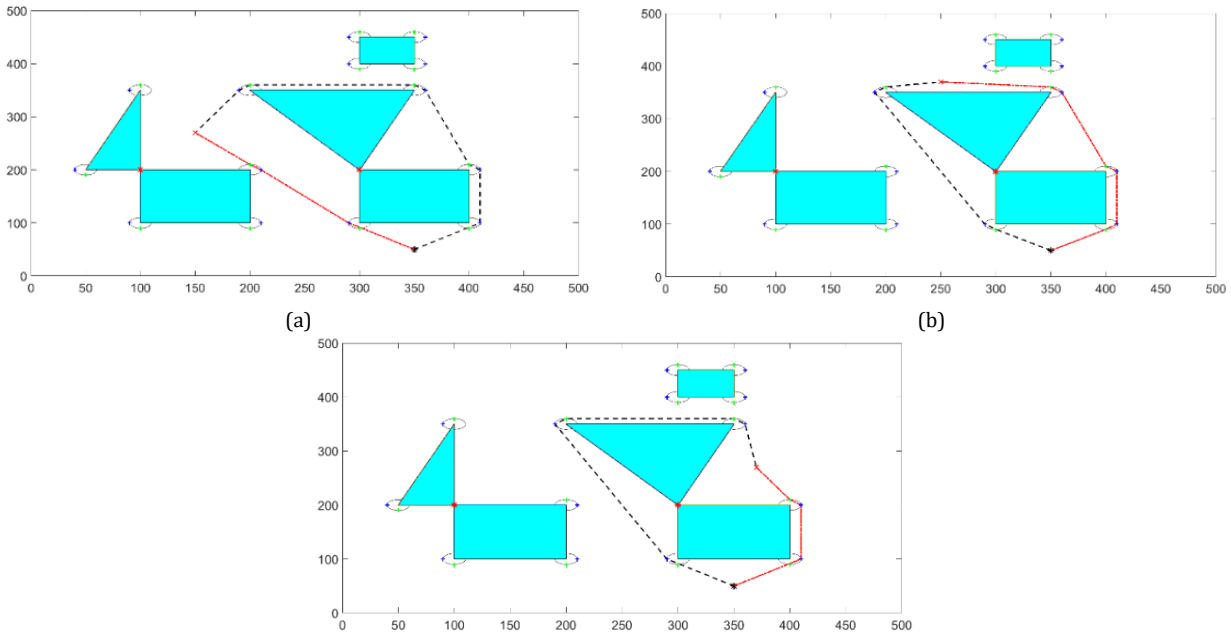


Fig. 11: Path planning with (xs, y_s)= (350, 50), (a) Safe path navigation with (x_t, y_t)=(150, 270), (b) Safe path navigation with (x_t, y_t)=(250, 370), and (c) Safe path navigation with (x_t, y_t)=(370, 270)

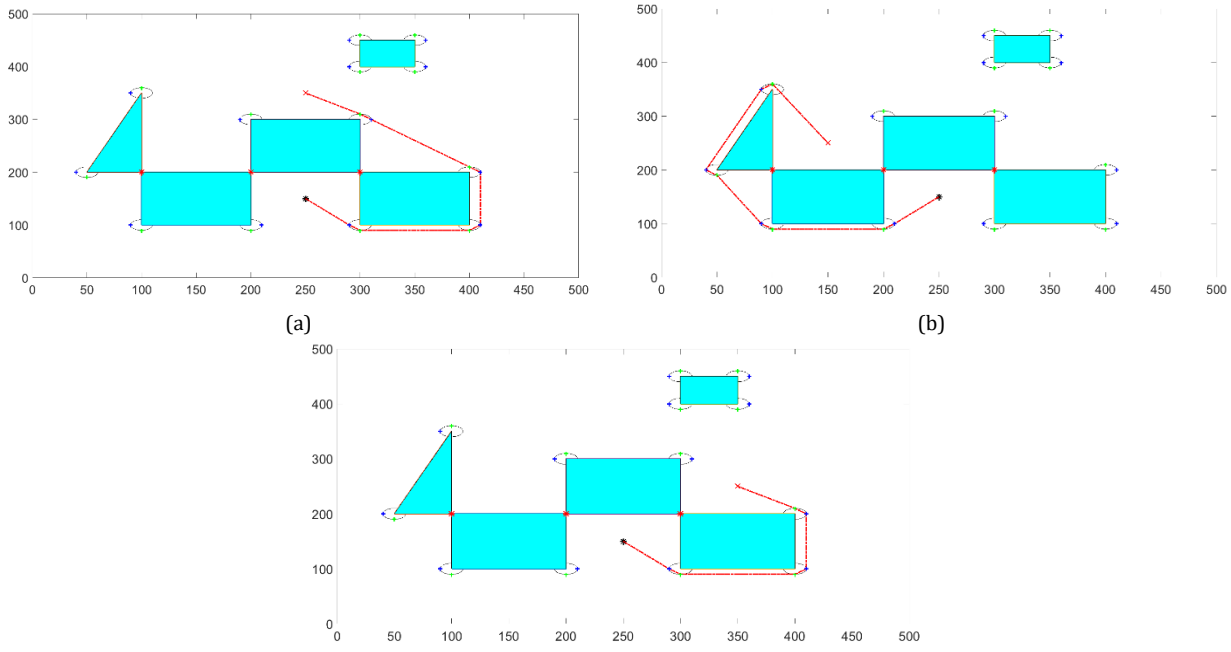


Fig. 12: Local minimum, (a) Safe path navigation with (x_t, y_t)= (250, 350), (b) Safe path navigation with (x_t, y_t)= (150, 300), and (c) Safe path navigation with (x_t, y_t)= (350, 250)

6. Discussion

Compared to the work developed by Hassani et al. (2022), which considers only the free segments where there is no conjunction between two obstacles or dangerous segments, the presented algorithm in this research succeeds in element all dangerous points to reduce the unwanted path selections as shown in the simulation results. The other difference is that the old work combined two approaches to overcome the navigation of mobile robots in complex environments, which are the turning point with simple environments and free segments with complex environments without considering the local minimum problem. This is in contrast to the algorithm presented in this research, which successfully applies to all types of environments, including environments with minimum local problems.

7. Conclusion

This paper proposes a novel method for planning mobile robots' smooth paths in two-dimensional user-defined environments with static obstacles. This work also provides a solution to the local minima problem. This paper presents an algorithm that uses free segments to find turning points. It

addresses the length of the path and its safety. The benefit of the created algorithm is that, in the given environment, the robot can always proceed from its initial location to the final position in a safe and quickest path, irrespective of the configuration of the obstacles or changes in the goal position. On the other hand, the suggested turning point mode control is a crucial approach to system management. This method performs well in tracking, exhibiting robustness, stability, and a short path. Matlab simulation results are used to illustrate that. The suggested approach is a useful substitute for resolving the trajectory tracking and path planning issues. Future research on determining multiple robot route controllers would be worthwhile.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Abdulshaheed AG, Hussein MB, Kadhom MA, Al Dulaimi ZM, and Gburi FH (2024). Step climbing of snake robot using script gait and traveling wave locomotion. AIP Conference Proceedings, AIP Publishing, 3051(1): 040017. https://doi.org/10.1063/5.0191637

Al-Kamil SJ and Szabolcsi R (2024). Optimizing path planning in mobile robot systems using motion capture technology. Results in Engineering, 22: 102043. https://doi.org/10.1016/j.rineng.2024.102043

Bulut V (2022). Optimal path planning method based on epsilon-greedy Q-learning algorithm. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 44: 106. https://doi.org/10.1007/s40430-022-03399-w

Deng X, Li R, Zhao L, Wang K, and Gui X (2021). Multi-obstacle path planning and optimization for mobile robot. Expert Systems with Applications, 183: 115445. https://doi.org/10.1016/j.eswa.2021.115445

Dhaouadi R and Hatab AA (2013). Dynamic modelling of differential-drive mobile robots using Lagrange and Newton-Euler methodologies: A unified framework. Advances in Robotics and Automation, 2(2): 1-7.

Duan P, Yu Z, Gao K, Meng L, Han Y, and Ye F (2024). Solving the multi-objective path planning problem for mobile robot using an improved NSGA-II algorithm. Swarm and Evolutionary Computation, 87: 101576. https://doi.org/10.1016/j.swevo.2024.101576

Hassani I, Ergui I, and Rekik C (2022). Turning point and free segments strategies for navigation of wheeled mobile robot. International Journal of Robotics and Control Systems, 2(1): 172-186. https://doi.org/10.31763/ijrcs.v2i1.586

Jogeshwar BK and Lochan K (2022). Algorithms for path planning on mobile robots. IFAC-Papers Online, 55(1): 94-100. https://doi.org/10.1016/j.ifacol.2022.04.016

Li C, Huang X, Ding J, Song K, and Lu S (2022a). Global path planning based on a bidirectional alternating search A* algorithm for mobile robots. Computers and Industrial Engineering, 168: 108123. https://doi.org/10.1016/j.cie.2022.108123

Li H, Zhao T, and Dian S (2022b). Forward search optimization and subgoal-based hybrid path planning to shorten and smooth global path for mobile robots. Knowledge-Based Systems, 258: 110034. https://doi.org/10.1016/j.knosys.2022.110034

Li S, Shu K, Zhou Y, Cao D, and Ran B (2021). Cooperative critical turning point-based decision-making and planning for CAVH intersection management system. IEEE Transactions on Intelligent Transportation Systems, 23(8): 11062-11072. https://doi.org/10.1109/TITS.2021.3099484

Li Y, Jin R, Xu X, Qian Y, Wang H, Xu S, and Wang Z (2022c). A mobile robot path planning algorithm based on improved A* algorithm and dynamic window approach. IEEE Access, 10: 57736-57747. https://doi.org/10.1109/ACCESS.2022.3179397

Liu L, Wang X, Yang X, Liu H, Li J, and Wang P (2023). Path planning techniques for mobile robots: Review and prospect. Expert Systems with Applications, 227: 120254. https://doi.org/10.1016/j.eswa.2023.120254

Miao C, Chen G, Yan C, and Wu Y (2021). Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm. Computers and Industrial Engineering, 156: 107230. https://doi.org/10.1016/j.cie.2021.107230

Ou Y, Fan Y, Zhang X, Lin Y, and Yang W (2022). Improved A* path planning method based on the grid map. Sensors, 22(16): 6198. https://doi.org/10.3390/s22166198 **PMid:36015963 PMCid:PMC9416044**

Qin H, Shao S, Wang T, Yu X, Jiang Y, and Cao Z (2023). Review of autonomous path planning algorithms for mobile robots. Drones, 7(3): 211. https://doi.org/10.3390/drones7030211

Rafai ANA, Adzhar N, and Jaini NI (2022). A review on path planning and obstacle avoidance algorithms for autonomous mobile robots. Journal of Robotics, 2022: 2538220. https://doi.org/10.1155/2022/2538220

Ren G, Liu P, and He Z (2022). A global path planning algorithm based on the feature map. IET Cyber-Systems and Robotics, 4(1): 15-24. https://doi.org/10.1049/csy2.12040

Rodríguez-Molina A, Herroz-Herrera A, Aldape-Pérez M, Flores-Caballero G, and Antón-Vargas JA (2022). Dynamic path planning for the differential drive mobile robot based on online metaheuristic optimization. Mathematics, 10(21): 3990. https://doi.org/10.3390/math10213990

Safarzadeh R and Wang X (2024). Map matching on low sampling rate trajectories through deep inverse reinforcement learning and multi-intention modeling. International Journal of Geographical Information Science, 38(12): 2648-2683. https://doi.org/10.1080/13658816.2024.2391411

Shu K, Yu H, Chen X, Li S, Chen L, Wang Q, Li L, and Cao D (2021). Autonomous driving at intersections: A behavior-oriented critical-turning-point approach for decision making. IEEE/ASME Transactions on Mechatronics, 27(1): 234-244. https://doi.org/10.1109/TMECH.2021.3061772

Szczepanski R, Tarczewski T, and Erwinski K (2022). Energy efficient local path planning algorithm based on predictive artificial potential field. IEEE Access, 10: 39729-39742. https://doi.org/10.1109/ACCESS.2022.3166632

Tan CS, Mohd-Mokhtar R, and Arshad MR (2021). A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. IEEE Access, 9: 119310-119342. https://doi.org/10.1109/ACCESS.2021.3108177

Wang Y, Xiong R, Tang P, and Liu Y (2023). Fast and reliable map matching from large-scale noisy positioning records. Journal of Computing in Civil Engineering, 37(1): 04022040. https://doi.org/10.1061/(ASCE)CP.1943-5487.0001054

Wu Z, Dai J, Jiang B, and Karimi HR (2023). Robot path planning based on artificial potential field with deterministic annealing. ISA Transactions, 138: 74-87. https://doi.org/10.1016/j.isatra.2023.02.018 **PMid:36822875**

Xu X, Zeng J, Zhao Y, and Lü X (2023). Research on global path planning algorithm for mobile robots based on improved A*. Expert Systems with Applications, 243: 122922. https://doi.org/10.1016/j.eswa.2023.122922

Yang L, Fu L, Li P, Mao J, and Guo N (2022). An effective dynamic path planning approach for mobile robots based on ant colony fusion dynamic windows. Machines, 10(1): 50. https://doi.org/10.3390/machines10010050

Yao M, Deng H, Feng X, Li P, Li Y, and Liu H (2023). Global path planning for differential drive mobile robots based on improved BSGA* algorithm. Applied Sciences, 13(20): 11290. https://doi.org/10.3390/app132011290

Yu L, Zhang Z, and Ding R (2022). Map-matching on low sampling rate trajectories through frequent pattern mining. Scientific Programming, 2022: 3107779. https://doi.org/10.1155/2022/3107779

Zhang D, Chen C, and Zhang G (2024). AGV path planning based on improved A-star algorithm. In the IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference, IEEE, Chongqing, China, 7: 1590-1595. https://doi.org/10.1109/IAEAC59436.2024.10503919

Zhang D, Dong Y, and Guo Z (2021). A turning point-based offline map matching algorithm for urban road networks. Information Sciences, 565: 32-45. https://doi.org/10.1016/j.ins.2021.02.052

Zhang Z, He R, and Yang K (2022). A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm. Advances in Manufacturing, 10: 114-130. https://doi.org/10.1007/s40436-021-00366-x

Zhao L, Bai Y, and Paik JK (2023). Global-local hierarchical path planning scheme for unmanned surface vehicles under dynamically unforeseen environments. Ocean Engineering, 280: 114750. https://doi.org/10.1016/j.oceaneng.2023.114750