AUTONOMOUS MOBILE ROBOT PATH PLANNING BASED ON ENHANCED A* ALGORITHM INTEGRATING WITH TIME ELASTIC BAND

THAI-VIET DANG

Mechatronics Department, School of Mechanical Engineering,
Hanoi University of Science and Technology, Hanoi,
Vietnam

DOI: 10.17973/MMSJ.2023_10_2023052

viet.dangthai@hust.edu.vn

Path planning is the core technology to realize mobile robot navigation. The enhanced A* algorithm with a time elastic band method (TEB) is utilized to aim at the collision problem in a complex environment. A grid-based method transfers the complex environment to a simple grid-based map. The mobile robot's position is determined based on information from the given maps. Optimizing the search point strategy and removing redundant path points reduces the path length and computational scale. The approach yields a safe trajectory in the local path using a time elastic band. The collision cost function combined with the preview deviation angle tracking method ensures the minimum collision-free distance to obstacles and the shortest path. Furthermore, the third-degree B-spline curve will improve the performance of the mobile robot path at the steering angles. Simulated results illustrate the correctness of the proposed method.

KEYWORDS

A* algorithm, mobile robot, path planning, obstacle avoidance, time elastic band (TEB), smooth trajectory.

1 INTRODUCTION

The Revolution 4.0 provides opportunities for the industrial sectors of intelligent systems based on the incredible convergence of emergent technological breakthroughs [La 2023]. Path planning is essential for achieving mobile robot navigation along the most efficient path between the beginning and goal points [Han 2017, Tran 2023]. Autonomous mobile robots (AMRs) sense their surroundings by identifying real-time barriers to moving safely [Yonggang 2022]. Depending on how well one understands the AMR environment, route planning can be divided into global and local path planning [Xunyu 2020]. Path planning has also been required to optimize multi-objective problems (MOOP) simultaneously, including shortest length, smoothness, and obstacle avoidance [Yibo 2020, Dang 2023]. These multi-objective problems can be optimized through various algorithms, such as graph-based and sampling-based algorithms [Shaher, 2022, Xunyu, 2020].

Global path planning has been typically employed in known environments using well-established algorithms such as the Dijkstra algorithm [Shaher 2022], A* algorithm [Yonggang 2022, Xunyu 2020], and rapidly exploring random tree (RRT) search algorithm [Xunyu 2020]. Conversely, local path planning approaches, such as the dynamic window approach (DWA) [Masato 2022], path planning based on game theory [Nguyen 2022], and the ant colony method [Wang 2022], have been utilized in situations where environmental knowledge is limited. Although Shaher et al. presented a straightforward and optimal global path planning algorithm based on the Dijkstra algorithm, the algorithm's performance is limited by available RAM, rendering the search slow and inefficient. To address this issue, Eshtehardian et al. introduced a technique combining RRT* and B-spline for path smoothing, effectively resolving multiple objective optimization problems (MOOPs) such as smoothness, shortest path, and collision avoidance. While this approach required more memory and processing time, it represented a more efficient search algorithm [Eshtehardian 2023]. Despite its slow convergence rate, the [Wang 2022] ant colony algorithm was a reliable path-planning method. In large and complex environments, the A* algorithm, a heuristic search technique based on the Dijkstra algorithm [Goldenberg 2017], was not a standalone solution. Yonggang et al. proposed an effective A* method combined with the three-time Bezier curve to accommodate various turning points and steering angles to address this issue. Improved A* algorithms have been widely used in global and local path planning due to their short computation speeds, path optimization, and other advantages. The dynamic window approach (DWA) [Kobayashi 2022] investigated the optimal path from samples of the current moment's surroundings, considering the AMR's kinematics and dynamics constraints. However, this approach had difficulty avoiding collisions in confined moving areas and unknown environments [Dang 2023].

The enhanced A* algorithm integrating with TEB is introduced in this study to address the difficulties mentioned above. The physical environment is first translated into a map using a grid-based method. The search point approach is optimized, removing unnecessary path points to reduce path length and computing scale. The proposed method produces a safe trajectory in the local path while ensuring the shortest path and the lowest collision-free distance to obstacles through the collision cost function and preview deviation angle tracking approach. The Timed Elastic Band (TEB) algorithm [Rösmann 2012] is a collision avoidance pathfinding for optimizing the local trajectory of a mobile robot. The mobile robot's path is smoothed at steering angles using the third-degree B-spline curve to enhance stability and track trajectory resilience.

The paper presents an upgraded path-planning algorithm combining the A* and TEB algorithms in Section 2. The proposed algorithm overcomes the limitations of the existing algorithms in addressing complex environments with limited knowledge. In Section 3, the simulation results confirm the effectiveness of the proposed approach in the comparison with recent path planning methods. Finally, completed multi tasks of the shortest path, obstacle avoidance and smoothed trajectory is guaranteed. Moreover, future works may include the application of the proposed algorithm to real-world scenarios and further improving its performance.

2 PROPOSED METHOD

Consider a mobile robot going through a known area with obstacles specified by a grid-based method in Fig. 1. The white grid "0" permits mobility, while the black grid "1" represents a restricted zone. The objective of a path planning problem is optimal travel from the beginning position S to the destination position G. The route planning meets all objectives for the shortest length, obstacle avoidance, and smoothness simultaneously.

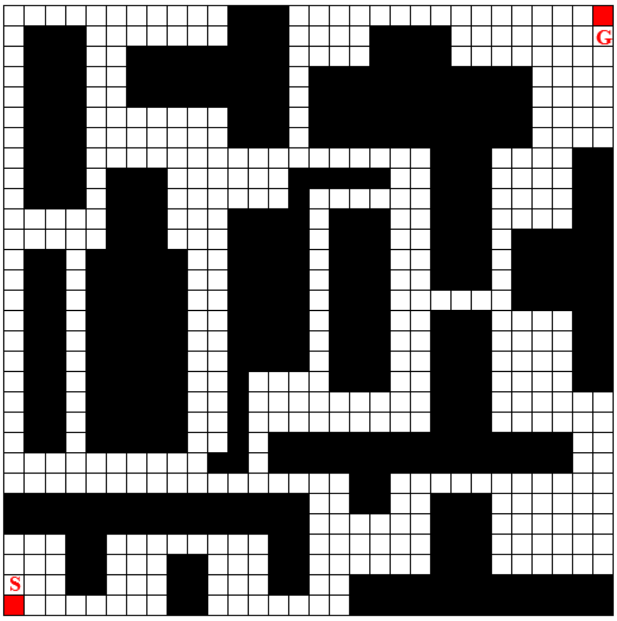


Figure 1. Grid map of the environment.

2.1. Traditional A algorithm*

The A* algorithm is a popular search algorithm that finds the shortest path between two points in a graph. It is frequently employed in navigation and routing applications, such as determining the shortest route between two points on a map. A fundamental aspect of the A* method is its utilization of a heuristic function to estimate the cost of the objective. The heuristic function is a distance-based mathematical estimation of the cost to the target from a given node. Utilizing a heuristic function, the A* algorithm can prioritize nodes likely closer to the target, enabling it to quickly identify the shortest path (see Eq. 1).

$$f(m)=h(m)+g(m) \tag{1}$$

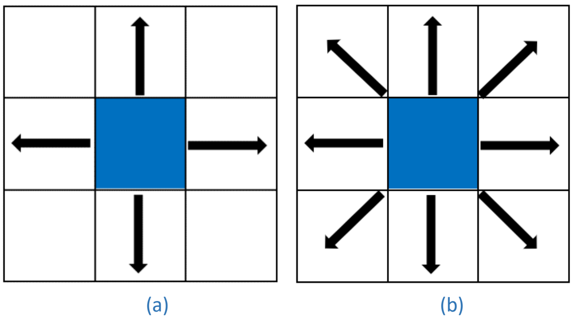
where m represents the current node, *f(m)* represents the cost evaluation function, *h(m)* represents the predicted cost from m to G, and *g(m)* represents the actual cost from m to the next node. The typical heuristic function for distance is the Manhattan distance, as indicated in Eq. 2, or the Euclidean geometric distance, as given in Eq. 3.

$$h_M(m)=|x_G-x_b|+|y_G-y_b| \tag{2}$$

and $$h_E(m)=\sqrt{(x_G-x_b)^2+(y_G-y_b)^2} \tag{3}$$

where *(x_G, y_G)* is the position of the target node, and *(x_b, y_b)* is the coordinate of any node.

Fig. 2(a) shows that the A* algorithm can only search in four directions for neighborhoods when utilizing the Manhattan distance. Fig. 2(b) presents an alternative use of Euclidean distance to study eight neighborhoods.



(a) (b)

Figure 2. Symmetric path search: **(a)** Four search directions of the neighborhood and **(b)** Eight search directions of the neighborhood.

The search process begins at S and then expands to surrounding nodes. Calculate the generation values following. The generating values are then evaluated using the heuristic function. Finally, select the node with the smallest generation value as the next parent node. The search procedure will be repeated until the target node G is located. In a large-scale map, the traditional A* search will generate many path nodes and consume significant memory.

2.2. Obstacle avoidance

A safety area surrounds an impediment that an autonomous system should not penetrate to prevent a collision. The size and shape of the safety zone will be determined by the features of the impediment and the system's capabilities. For instance, an obstacle with a complex shape or an autonomous system with limited agility may necessitate a greater safety zone. Improved A* path planning algorithms can calculate safe and efficient obstacle avoidance routes. A new environment model will include additional hazard zones around obstructions in Fig. 3.

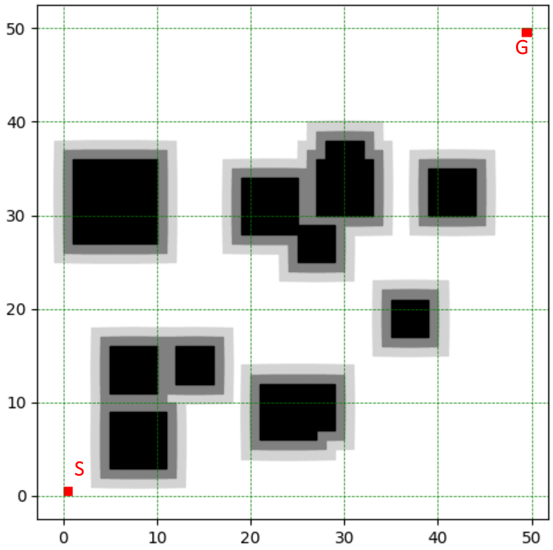


Figure 3. New grid map with risk grey zones at two serious levels of collision distance to obstacles (light and dark grey).

The risk cost function k(m) is added to the cost assessment function in equation 1 to provide Eq. 4.

$$f(m)=h(m)+g(m)+k(m) \tag{4}$$

where k(m) is the risk cost function depending on the added distance to the obstacle. Safety regions are an important issue in the design of obstacle avoidance algorithms for autonomous systems. They aid in preventing collisions and ensuring the safe operation of the autonomous system in its environment.

- In Fig. 3, the safety zones surrounding an obstacle can be separated into two levels of risk:
- Inner risk zone (dark grey): This is the closest region to the obstacle, non allowing mobile motion. The Inner danger zone will maintain a safe distance from the obstacle, regardless of the robot's trajectory or navigation system faults. Typically, the inner risk zone is greater than the robot itself to offer a suitable buffer zone for the robot's movement.
- Outer risk zone (light grey): The territory surrounding the inner risk zone that the mobile robot can enter under specific circumstances while keeping a safe distance. Nonetheless, the robot must exercise caution while in the outside risk zone because it is still a high-risk location.

By dividing the static safety region around an obstacle into Inner and outer risk zones, the mobile robot can navigate obstacles with greater safety and reliability. The inner risk zone provides a fail-safe buffer zone that ensures that the robot does not collide with the obstacle, while the outer risk zone provides a more flexible safety zone that allows the robot to navigate the obstacle more efficiently.

In a complicated environment, the search process will produce many nodes, increasing the computing scale, memory usage, and inefficiency. As a result of redundant path sites, the movement distance and computational scale will grow. Therefore, removing duplicate spots and smoothing the path are crucial components of our mobile robot's path planning.

2.3. Jump Point Search (JPS)

JPS is an algorithm that identifies the shortest path between two points in a grid-based environment. It is a variant of the A* algorithm aimed to be more effective by reducing the search space and removing unneeded nodes.

The JPS algorithm operates outward from the initial node in a grid-like pattern. At each step, the algorithm chooses a "jump point" from the current node and advances immediately instead of traversing all the nodes in between. A jump point is a node selected as a potential next step based on certain criteria and located on the shortest path between the start and the objective. For instance, a jump point may be selected if it is the first node in a particular direction (north, south, east, or west) blocked by an obstruction.

To update the obstacle avoidance algorithm based on A* with JPS, we need to modify the A* algorithm to include the jump point search technique. Here's a general outline of the steps involved such as follows :

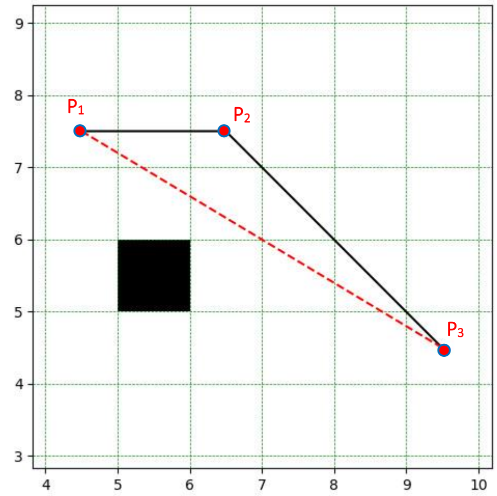


Figure 4. Diagram of removing redundant path points.

The search results of the traditional A* algorithm are a group of path points P_k. For each set of three path points in Fig. 4, we can link the line P_1P_3 as an illustration (red dotted line). Then, evaluate the condition for obstacle avoidance. If no obstacle grid is on the line, the path P_1P_3 will be changed to $P_1P_2P_3$ (black continuous line). Deleting superfluous points will be repeated until the target grid is located. Thus, enhanced path planning would be entirely attained. The JPS algorithm can locate the shortest path faster than the A* algorithm because it avoids studying many grid nodes that are not on the shortest path. Finally, by incorporating the JPS technique into the A* algorithm, we can reduce the computational complexity of the search process and improve the efficiency of the path planning algorithm. It is also simple to develop and can be particularly useful in obstacle avoidance scenarios, where the search space can be large and complex.

2.4. Smoothness

Smoothness in mobile robot route planning refers to the absence of abrupt changes in direction or velocity along a robot's path. Mobile robots are like smooth road since it is more efficient and easier to navigate. It may also be safer since abrupt changes in direction, or velocity may lead the robot to lose stability or collide with obstructions. These techniques can smooth the path by reducing the frequency of abrupt changes in direction or speed. This can be accomplished by fitting the path to a smooth curve or by minimizing the path's "roughness" through optimization techniques in Fig. 5.

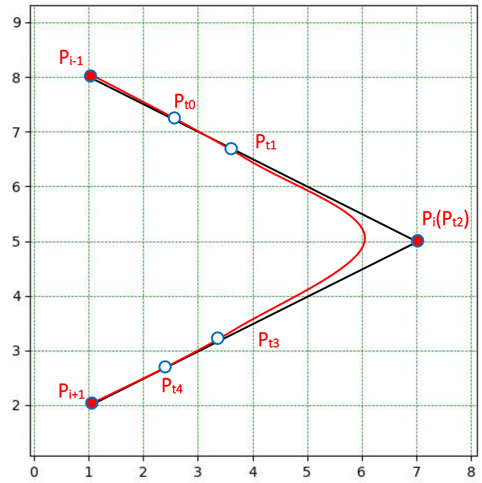


Figure 5. The B-spline transition curve (red curve) replacing with one segment of path $P_1P_2P_3$ (black line)

A B-spline curve and a straight line will generate a trajectory-creation algorithm to aid the mobile robot's navigation. Eq. 5 depicts the expression of the third-degree B-spline curve.

$$B(t)=\sum_{i=0}^{4} N_{i3}(u)Pt_i. \tag{5}$$

where, P_{ti} are the control path points, and $N_{i3}(u)$ are the third-degree B-spline basis functions. Therefore, the trajectory must be shorter than the planning path.

Consequently, smoothness in mobile robot path planning is a significant component in the design of autonomous systems, and it can improve the efficiency, dependability, and safety of the robot's motion.

2.5. Time Elastic Band (TEB)

The TEB algorithm includes geometric constraints based on the mobile robot kinematics model in the multi-objective constraints in Eq. 6, such as below :

$$u(t)=\begin{bmatrix} V(t) \\ \omega(t) \end{bmatrix}=\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{D} & \frac{1}{D} \end{bmatrix}\begin{bmatrix} V_L \\ V_R \end{bmatrix} \tag{6}$$

where, V_L and V_R present the mobile robot speeds of the left and right wheels, and D presents the distance between the two wheels.

The TEB algorithm is developed from the Elastic Band algorithm (EB) in Eq. 7.

$$\begin{cases} Q=\{S_i\}, i=1,2,...n \\ \tau=\{\Delta T_i\}, i=1,2,...n-1 \end{cases} \text{with } n\in N \tag{7}$$

where, S_i presents the pose at the time *i*, *Q* presents the pose sequence, ΔT_i presents the time interval between adjacent poses, and τ is the time interval sequence, respectively.

Therefore, we describe all steps of integrating TEB with the traditional A* algorithm, such as follows: First, generate a global path using traditional A* path planning. Then, divide the global path into smaller local paths the mobile robot can follow. Next, apply the TEB algorithm to each local path to optimize the robot's trajectory while maintaining a safe distance from obstacles (see Fig. 6). Select the optimal local path based on the TEB algorithm's cost function. Finally, execute the selected local path using the robot's motion control system.

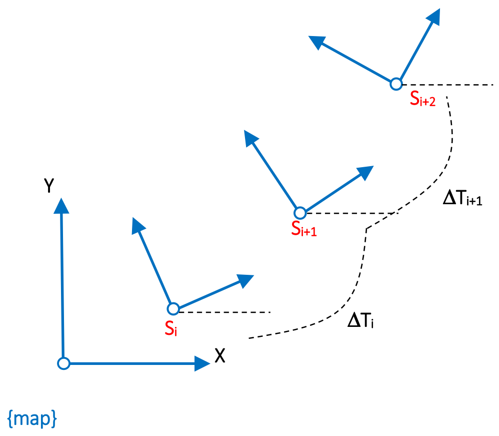


Figure 6. The configurations of trajectory sequence B and the time interval sequence.

In Fig. 6, the configurations of trajectory sequence *B* is represented as:

$$B=(Q,t)=[S_1,\Delta T_1,S_2,\Delta T_2,...,\Delta T_n,S_n] \tag{8}$$

Then, the TEB's multi-objective optimization function is defined as follows :

$$f(B)=\sum_k \gamma_K f_K(B) \tag{9}$$

where, $f_k(B)$ presents each constraint function, and γ_K represents the weight corresponding to the constraint function.

The objective functions are based on the minimum distance $d_{min,j}$ between the TEB and the A*'s path points or obstacles (z_j) in Fig. 7.

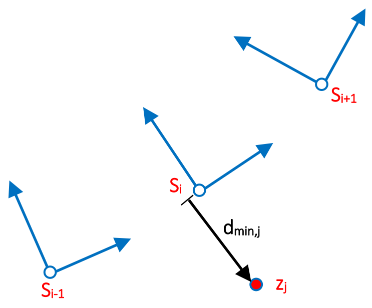


Figure 7. The minimum distance $d_{min,j}$ between the TEB as the A* path points or obstacles (zj).

The maximum and minimum allowable distance will limit the distance between the TEB and obstacles. These two parameters will be included in the penalty functions $_{path}$ and f_{ob} expressed in equation 10 :

$$\begin{aligned} f_{path} &= e_{\Gamma}\left(d_{\min,j}, r_{p,max}, \varepsilon, S, n\right) \\ f_{ob} &= e_{\Gamma}\left(-d_{\min,j}, r_{o,\min}, \varepsilon, S, n\right) \end{aligned} \tag{10}$$

The TEB algorithm works by modeling the mobile robot's movement as a time-varying trajectory and using a cost function to evaluate the trajectory's safety and efficiency. The cost function considers factors such as speed, acceleration, and proximity to obstacles and seeks to minimize the overall cost of the trajectory.

By optimizing each local path using the TEB algorithm, we can ensure that the robot follows a safe and efficient trajectory while navigating around obstacles. This robot can be particularly useful in dynamic environments where obstacles may move or change position over time, as in Fig. 8.

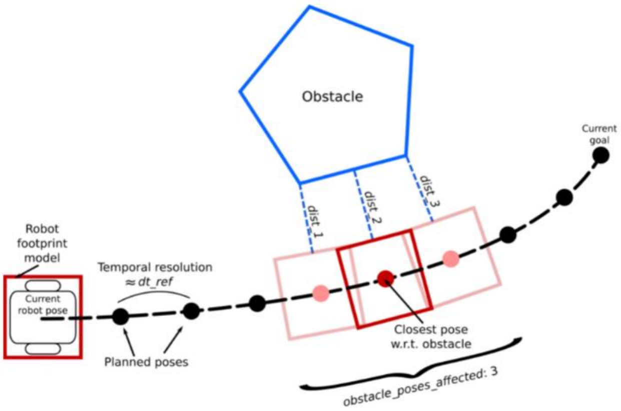


Figure 8. Mobile robot path planning based on improved A* using TEB in the obstacle environment.

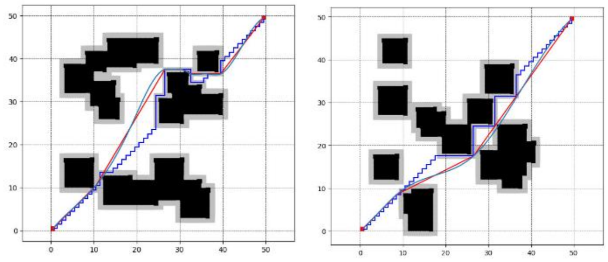
3 SIMULATION RESULTS

Initially, a mobile robot equipped with Lidar and Jetson nano-embedded computer is used to construct the grid map (see Fig. 9).

Figure 9. A mobile robot equipped with Lidar and Jetson Nano.

To validate the efficacy of the revised method, the following tests are conducted: 50 x 50 grid environments (start point S (0,0) and goal point (50,50)) are used as simulated test maps, and the traditional A* and improved A* algorithms are compared. For experimental simulation, Intel(R) Core (TM) I7-8750h CPU @2.20ghz 2.21ghz, Ram 8.00GB, 64 operating systems, Windows 10 home English version, and Visual Studio 1.74 are utilized.

Then, using an enhanced A* algorithm, optimal path planning for a mobile robot is designed from start point S to goal point G. (see blue lines in Figure 10). As a result of the findings, an enhanced path-based JPS will be developed (see red lines in Figure 10). Next, the B-spline curve of the third degree is utilized to construct the smoothed trajectory (see green lines in Fig. 10). Fig. 11 depicts the observation of the ROS mobile robot in an indoor setting.



(a) (b)

Figure 10. Mobile robot path planning with one level of risk region: (a) three large size obstacles, and (b) two large size obstacles (red line is improved A* algorithm combining with safe regions and JPS and a green line is smoothed improved path).

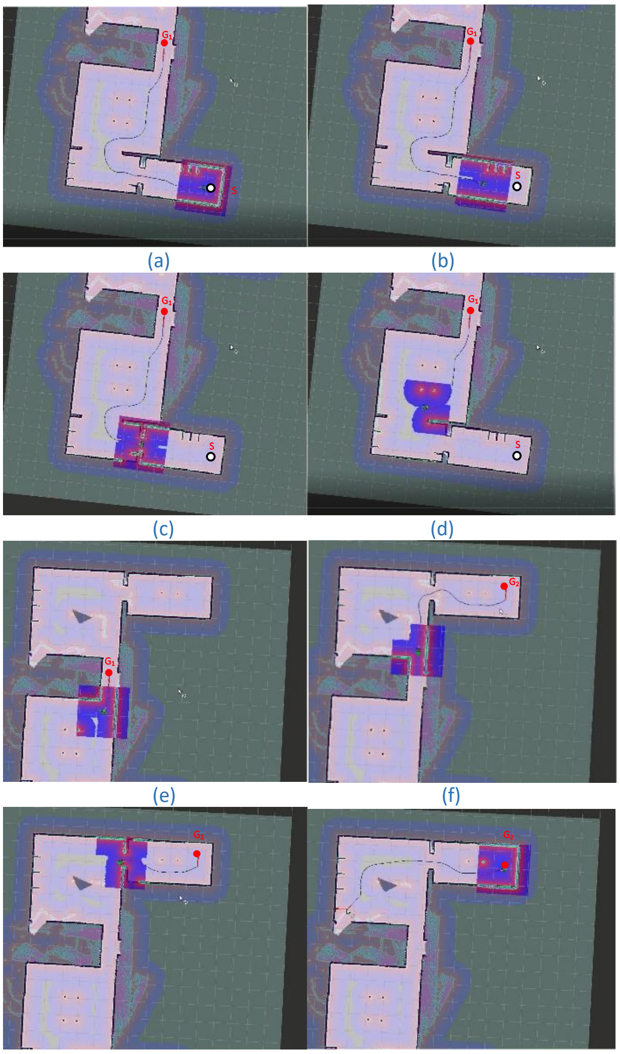
Two scenarios with one level of risk region are depicted in Fig. 10. The traditional A* path (blue line) can avoid the obstacles; however mobile robot's path has still moved to the risk region around the obstacle (grey areas). To solve the problems, the improved A* algorithm with a safety cost function (grey areas) ensures a safe collision distance from the obstacle. In addition, after removing redundant path points, the new path will be generated via jump point search (see red lines). Furthermore, the performance of mobile robot steering angles will be enhanced by a smoothed algorithm based on a B-spline curve of the third degree (see green lines). The smoothing method permits the mobile robot to enter the risk region (areas) at the steering angles. Finally, the development of trajectories has continued to satisfy all MOOPs, including minimum length, smoothness, and obstacle avoidance.

Table 1 shows comparison results between the traditional A* algorithm and the proposed A* algorithm through the number of path nodes, average turning angles, and path length in both above two scenarios, such as follows:

Method	Nodes	Average turning angles (°)	Path length
Traditional A* [Shaher 2022]	70	42.56	72.5
Improved Dijkstra [Shaher 2022]	36	32.78	63.6
Proposed method	6	13.68	56.2

The proposed method has significantly reduced the calculation path nodes, average turning angles, and path length in mobile robot's movement compared to traditional A* [Yonggang 2022] and improved the Dijkstra algorithm [Shaher 2022]. The method deletes redundant path points in the A* algorithm and smoothens the trajectory. The mobile robot's trajectory has been smoother and more robust when the average turning angle is reduced. In addition, the proposed path planning shortens the average distance between nodes to support the optimal mobile robot navigation strategy design.

Fig. 11 (snapshots (a)-(h)) depicts a mobile robot following a global path determined by an enhanced smooth A* algorithm. The experimental test was conducted on a mobile robot using LiDAR in an actual ROS environment. In addition, the search area's safe cost ensured the mobile robot's robust movement while avoiding obstacles.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 11. The observation of the ROS mobile robot from the star point to the goal point.

All snapshots of Fig. 11 present the mobile robot's moving process, including as follows:

- In Fig. 11a, at the start point S, a mobile robot will move to the first goal point G_1. The global path is completely built.
- Where Fig. 11b, in each local area, the mobile robot discovers the obstacles along the right wall to avoid based on the search region (see blue area).
- Then, the mobile robot determines the door's position and adjusts the path to move through door 1 successfully, as in Fig. 11c. Using a smooth algorithm to turn right at the corner ensures a smooth path.
- Next, the mobile robot moves with the shortest distance and maintains a safe collision distance from the new four obstacles at the room's center in Fig. 11d.
- Furthermore, the mobile has successfully moved through the narrow corridor and reached the first goal G_1, in Fig. 11e.
- From Fig. 11f to Fig. 11h, the improved A* has been tested with more than one goal point G_2 in the small room. Based on the global path, at each local pathfinding, the mobile robot tracks the path successfully, turns right, and moves through door 2. Finally, arrives at goal point G_2.

The mobile robot tracks the path after removing redundant points, successfully avoids the new obstacles, and then smoothly reaches goal points G_1 and G_2 in both required scenarios. Therefore, the results illustrate the improved A* algorithm successfully achieves simultaneous path tracking and obstacle avoidance.

4 CONCLUSIONS

The paper presents the improved A* algorithm combining with TEB algorithm for mobile robot path planning. Firstly, the mobile robot's global path is planned using the traditional A* algorithm. Next, path length and computing complexity are decreased by optimizing the search point technique and deleting redundant path points. In addition, in each local area, the movement of mobile robots has been made safer by establishing risk zones around obstacles. The TEB algorithm will optimize each mobile robot's local path to ensure a safe and efficient trajectory while navigating obstacles.

Moreover, a smoothed algorithm will enhance the performance of mobile robot steering angles based on a B-spline curve of the third degree. Simulations conducted in grid-based systems reveals the algorithm's advantages in terms of reduced memory consumption and optimization of computational calculation speed. Ultimately, path planning has been accomplished in an actual ROS environment.

ACKNOWLEDGMENTS

The authors express grateful thankfulness to Vietnam-Japan International Institute for Science of Technology (VJIIST), School of Mechanical Engineering, HUST, Vietnam and Shibaura Institute of Technology, Japan.

REFERENCES

[La 2023] La Thi Ngoc Anh et al. 2023 “Research MS Access Software for Application in Smart Garment Factory Machine Management.” *MM Science Journal:* 6561-6567, https://doi.org/10.17973/MMSJ.2023_06_202304.

[Han 2017] Han, J., and Seo, Y. 2017. "Mobile robot path planning with surrounding point set and path improvement." *Application of Software Computer* 57: 35-47. https://doi.org/10.1016/j.asoc.2017.03.035.

[Tran 2023] Hoai-Linh Tran, Thai-Viet Dang. " An Ultra Fast Semantic Segmentation Model for AMR's Path Planning." *Journal of Robotics and Control* 4(3):424-430. https://doi.org/ 10.18196/jrc.v4i3.18758.

[Yonggang 2022] Yonggang, L. et al. 2022. "A Mobile Robot Path Planning Algorithm Based on Improved A* Algorithm and Dynamic Window Approach." *IEEE Access* 10: 57736-57747. https://doi.org/10.1155/2022/2183229.

[Xunyu 2020] Xunyu, Z., Jun, T., Huosheng, H., and Xiafu, P. 2020. "Hybrid Path Planning Based on Safe A* Algorithm and Adaptive Window Approach for Mobile Robot in Large-Scale Dynamic Environment." *Journal of Intelligent & Robotic Systems* 99(2): 65-77. https://doi.org/10.1007/s10846-019-01112-z.

[Yibo 2020] Yibo, L., Dingguang, D., and Xiaonan, G. 2020. “Mobile Robot Path Planning based on Improved Genetic Algorithm With A-star Heuristic Method.” In: IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC 2020), pp. 1306-1311. https://doi.org/10.1109/ITAIC49862.2020.9338968.

[Dang 2023] Dang, T. V., and Bui, N. T. 2023. "Multi-Scale Fully Convolutional Network-Based Semantic Segmentation for Mobile Robot Navigation." *Electronics* 12(3): 533. https://doi.org/10.3390/electronics12030533

[Shaher 2022] Shaher, A., Sahbi, B., and Lioua, K. 2022. "Improved Dijkstra Algorithm for Mobile Robot Path Planning and Obstacle Avoidance." *Computers, Materials & Continua* 72(3): 5939-5954. https://doi.org/10.32604/cmc.2022.028165

[Eshtehardian 2023] Eshtehardian, S. A., and Khodaygan, S. 2023. "A continuous RRT* based path planning method for non-holonomic mobile robots using B spline curves." *Journal of Ambient Intelligence and Humanized Computing*. 14(5): 8693-8702. https://doi.org/10.1007/s12652-021-03625-8.

[Masato 2022] Masato, K., and Naoki, M. 2022. "Local Path Planning: Dynamic Window Approach with Virtual Manipulators Considering Dynamic Obstacles." *IEEE Access* 10: 17018-17029. https://doi.org/10.1109/ACCESS.2022.3150036.

[Nguyen 2022] Nguyen, L. V., Phung, M. D., and Ha, Q. P. 2022. "Game Theory-Based Optimal Cooperative Path Planning for Multiple UAVs." *IEEE Access* 10: 108034-108045. https://doi.org/10.1109/ACCESS.2022.3213035.

[Wang 2021] Wang, W., Zhao, J., Li, Z., and Huang, J. 2021. "Smooth Path Planning of Mobile Robot Based on Improved Ant Colony Algorithm." *Journal of Robotics* 2021(3): 1-10. https://doi.org/10.1155/2021/4109821.

[Goldenberg 2017] Goldenberg, M. 2017. "The Heuristic Search Research Framework." *Knowledge-Based Systems* 129: 1-3. https://doi.org/10.1609/socs.v9i1.18444.

[Kobayashi 2022] Kobayashi, M., Motoi, N. 2022. "Local Path Planning: Dynamic Window Approach with Virtual Manipulators Considering Dynamic Obstacles." *IEEE Access* 10: 17018-17029. https://doi.org/10.1109/ACCESS.2022.3150036.

[Dang 2023] Dang, T. V., and Bui, N. T. 2023. " Obstacle Avoidance Strategy for Mobile Robot based on Monocular Camera." Electronics 12(8): 1932. https://doi.org/10.3390/electronics12081932.

[Rösmann 2012] Rösmann, C., Feiten, W., Wösch, T.,Hoffmann, F., and Bertram, T. 2012 "Trajectory modification considering dynamic constraints of autonomous robots." In Proceedings of the ROBOTIK 2012, 7th German Conference on Robotics, Munich, Germany, 21–22 May 2012; pp. 1-6.

CONTACTS:

Dr. Thai-Viet Dang
School of Mechanical Engineering/Hanoi University of Science and Technology, Department of Mechatronics
No 1 Dai Co Viet Rd. Hai Ba Trung, Hanoi, 10000, Vietnam
(+084) 0989458581, viet.dangthai@hust.edu.vn, https://orcid.org/my-orcid?orcid=0000-0002-1496-2492.