Journal of Robotics and Control (JRC)
Volume 6, Issue 4, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i4.27678

Hybrid Path Planning for Wheeled Mobile Robot Based on RRT-star Algorithm and Reinforcement Learning Method

Hoang-Long Pham [1], Nhu-Nghia Bui [2], Thai-Viet Dang [3*]
[1] Research Institute of Post and Telecommunication, Posts and Telecommunications Institute of Technology, Vietnam
[2, 3] Department of Mechatronics, School of Mechanical Engineering, Hanoi University of Science and Technology, Vietnam
Email: [1] Longph@ptit.edu.vn, [2] Nghia.BN215977@sis.hust.edu.vn, [3] viet.dangthai@hust.edu.vn
*Corresponding Author

Abstract*—In the field of wheeled mobile robots (WMRs), path planning is a critical concern. WMRs employ advanced algorithms to find out the feasible path from a starting point to a specific destination. The paper proposes efficient and optimal path planning for WMRs, integrating collision avoidance strategies and smoothed techniques to determine the best route during navigation. The proposed hybrid path planning consists of improved RRTstar algorithm and reinforcement learning method. Therefore, the RRT* algorithm employs random sampling in conjunction with a reinforcement learning model to purposefully guide the sampling process towards areas that demonstrate an increased likelihood of successful navigation completion. The proposed RRTstar-RL algorithm generates significantly shorter trajectories compared to the traditional RRT and RRTstar methods. Specifically, the path length with the proposed algorithm is 11.323 meters, while the lengths for RRT and RRTstar are 15.74 and 14.40 meters, respectively. Moreover, the optimization of computation time, especially when using pre-trained data, greatly speeds up the path-finding calculation process. In particular, the time needed to generate the optimal path with the RRTstar-RL algorithm is 2.02 times faster than that of RRTstar and 1.6 times faster than RRT. Finally, the proposed RRTstar-RL algorithm has been successfully verified for feasibility and effectively meets numerous objectives established during simulations and validation experiments.

Keywords*—*Wheeled Mobile Robots; Reinforcement Learning; Rrtstar; Path Planning.

I. INTRODUCTION

In recent years, mobile robots have gained significant prominence in the domains of automation and robotics, particularly in navigation tasks within environments such as warehouses and manufacturing facilities. Motion and path planning are fundamental components that enable wheeled mobile robots (WMRs) to navigate autonomously [1]-[5]. Upon acquiring a global or local map through environmental perception, the robot must formulate a feasible trajectory from its initial position to the desired destination. These robots are required to operate with both flexibility and precision to effectively execute their assigned tasks. Prior research has investigated various image processing techniques and control strategies aimed at enhancing the operational flexibility and cognitive capabilities of WMRs. Traditional control methods are often effective only when the specific characteristics of the system and the precise locations of tracked objects are known [6]-[10]. For example, visual servoing utilizes continuous visual feedback to guide a mobile robot toward a stationary target [11]-[17]. However, achieving this objective can be particularly challenging in environments characterized by complex and unpredictable behaviors, as well as uncertain disturbances [18], [19]. Such irregular fluctuations in control systems can considerably affect the efficacy and stability of control mechanisms. Consequently, it is imperative to develop a system that demonstrates exceptional tracking capabilities to improve the performance of vision-based mobile robots. A critical component in the pursuit of autonomy is motion planning, which enables wheeled mobile robots (WMRs) to determine their own trajectories [20], [21]. Once equipped with either a global or local map through environmental awareness, WMRs must formulate a feasible path from their initial position to their intended destination. This journey must comply with specific criteria, which may include reducing operational costs, identifying the most expedient route, or minimizing travel time [22]-[23].

Path planning algorithms function as proficient navigators, adeptly determining routes from initial points to destinations while skillfully circumventing obstacles [24], [25]. These algorithms can be classified into two primary categories: global and local path planning. Global path planning serves as a comprehensive strategist, identifying a sequence of critical waypoints that connect the starting point to the endpoint [26]-[30]. It utilizes three fundamental techniques: graph search, sampling search, and dynamic search. Graph search algorithms, such as Dijkstra's [31] and Astar [32], excel in low-dimensional spaces, ensuring a comprehensive exploration of potential routes. In contrast, sampling search algorithms are suited for high-dimensional spaces, providing a probabilistic assurance of pathfinding. Dynamic search, on the other hand, improves the connectivity of path nodes, albeit at the expense of some completeness. Conversely, local path planning operates as a meticulous artist, generating precise trajectories from the start node to the target node within a localized area [33]-[38]. This category encompasses traditional methods such as the Rapidly exploring Random Tree (RRT) [39], Time-Elastic Band (TEB) [40], Dynamic Window Approach (DWA) [41], [42], Artificial Potential Field (APF) [43], [44], and Neural Network Methods (NNM) [45], [46]. As we consider future developments, the trajectory of motion planning for WMRs is evident: it is transitioning from broad, coarse path planning in spatial contexts to detailed trajectory planning in temporal domains, facilitated by the continuously advancing capabilities of computational power.

In a comprehensive examination of path finding algorithms, Liu et al. [47] presented the weighted Astar algorithm to complete the WMR's understanding. Concurrently, Feng et al. [48] achieved significant advancements of the bidirectional search algorithm for optimal values. Dang et al. [49] introduced the innovative jump point search (JPS) algorithm for eliminating the redundant Astar path point, in grid maps. The hybrid JBS-A*B algorithm and improved DWA increased the safety in local areas. Building upon this progress, Esmaiel et al. [50] developed the LQR-RRTstar algorithm, which utilizes a Linear Quadratic Regulator (LQR) to determine the optimal path for extended random tree nodes within a specified timeframe, in dynamic environments. Wang et al. [51] addressed this challenge by formulating a quadratic convex optimization problem aimed at minimizing the discrepancy between the current and ideal states, thereby directly determining the optimal trajectory under dynamic constraints. Finally, Zhang et al. [52] introduced the Flat-RRTstar algorithm, specifically designed for differentially flat systems. Therefore, trajectory kinematic constraints derive optimal motion primitives between two grid states, ultimately producing suboptimal trajectories that connect two nodes.

The analysis presented above clearly indicates that various navigation strategies possess distinct advantages and disadvantages. To facilitate seamless movement and enhance the stability of WMRs while tracking their trajectories, we propose a hybrid path planning approach that integrates an improved RRTstar algorithm with RL method [53], [54]. Initially, the RRTstar algorithm is refined to leverage the benefits of reinforcement learning. To address the inefficiencies associated with the sampling process of the RRTstar algorithm, we have incorporated a reinforcement learning framework. This framework comprises an Actor, which determines the appropriate actions to be taken, and a Critic, which evaluates the effectiveness of these actions and provides feedback for necessary adjustments to the WMRs in dynamic environments. The Actor is constructed using the U-Net architecture [55], [56], which is responsible for generating probability maps, while the Critic employs the MobileNetV2 [57]-[60] architecture to assess the current policy, or the weight patterns produced by the Actor, in relation to the achievement of the reward parameter. The proposed RRTstar-RL algorithm offers significant advantages, including a reduction in inefficient sampling, the extraction of map features that incorporate trained data into the pathfinding process, and an acceleration of processing time due to a decreased need for sample review. Two basic drawbacks of RRTstar have been addressed, including the inefficient random sampling process and the lack of connectivity between samples. Finally, the feasibility of the proposed RRTstar-RL algorithm has been successfully validated, demonstrating its effectiveness in achieving various objectives established during simulations and validation experiments.

II. Proposed Method

A. RRTstar Algorithms

RRTstar algorithm represents an advancement over the conventional RRT methodology (see Fig. 1), particularly in the domains of pathfinding and optimization. Its primary aim is to determine the most efficient and feasible route for navigation from a specified starting point x_{init} to a designated destination x_{goal}. The operational framework involves the random selection of points, referred to as x_{rand}, which facilitates the expansion of the search tree. Subsequently, the algorithm identifies the vertices within the tree, denoted as x_{near}, that are closest to x_{rand}, while adhering to a predetermined distance threshold s to ensure that the trajectory from x_{near} to x_{new} remains unobstructed by obstacles. The neighboring points are then evaluated, and the parent point with the lowest associated cost is selected. Ultimately, the neighboring points are reconnected to optimize the overall path, culminating in a complete route.

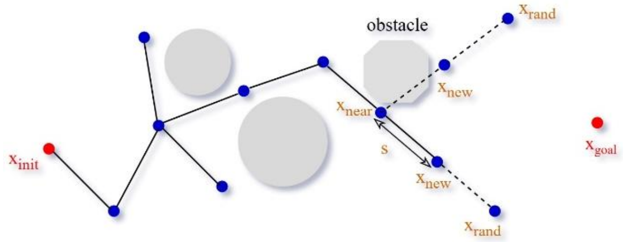

Fig. 1. Traditional RRT algorithm

The RRTstar algorithm diverges from the conventional approach of selecting the nearest point as the parent point for the new point. As illustrated in Fig. 2, this process involves drawing a circle around the nearest point $x_{nearest}$ and assessing the distance between any point within this circle and the new point x_{new}, in Fig. 2(a). If the distance from $x_{nearest}$ to x_{new} is less than the distance from x_{new} to other points (q_1 or q_2), a connection is established between $r_{nearest}$ and x_{new}, in Fig. 2(b). Additionally, it is necessary to evaluate the shortest distance between $x_{nearest}$ and q_2. Should the distance from x_{new} to q_2 be shorter, the parent of q_2 is updated to x_{new} (see Fig. 2(c)).

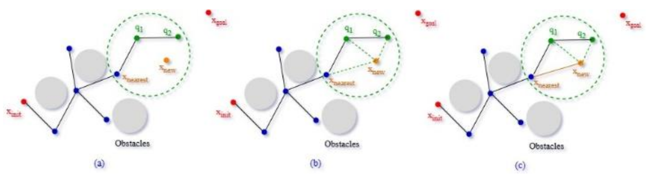

Fig. 2. RRTstar algorithm

Within the context of this research, the authors have developed the RRTstar algorithm to facilitate enhancements when utilizing reinforcement learning techniques. In particular, x_{rand} points are selected randomly from the configuration space in accordance with the established probability map, which delineates marginal probability along the X-axis, in (1) and conditional probability along the Y-axis, in (2):

$$P(x) = \sum_{y} P(x, y) \tag{1}$$

$$P(x \mid y) = \frac{P(x,y)}{P(x)} \tag{2}$$

There is always the formula:

$$x_{rand} \sim P(x,y) \in C_{free} \tag{3}$$

Where C_{free} is configuration space.

The process of finding the nearest point consists of the k neighbors closest to the x_{rand} point in the tree:

$$N_k(x_{rand}) = argmin_{x_i \in T} \parallel x_{rand} - x_i \parallel_2, i = 1 \ldots k \tag{4}$$

where T is the set of existing points in the search tree. $\parallel x_{rand} - x_i \parallel_2$ is the Euclidean distance between x_{rand} and x_i in space. Points x_{new} are created in the direction from x_{near} to x_{rand} satisfying the allowed distance:

$$x_{new} = x_{near} + min(range, \parallel x_{rand} - x_{near} \parallel) \cdot \frac{x_{rand} - x_{near}}{\parallel x_{rand} - x_{near} \parallel} \tag{5}$$

The valid points are checked for collisions with obstacles to select no collision points. Then the optimal parent is selected based on selecting points from k neighbors so that the total cost from the current point is minimum. The selection function is illustrated as follows:

$$x_{parent} = argmin_{c(x_i)} + \parallel x_i - x_{new} \parallel \tag{6}$$

where c is the distance from the current point to x_i. The parents of the points are updated if traversing x_{new} costs less.

$$\forall x_i \in N_k(x_{new}): if\ c(x_{new}) + \parallel x_{new} - x_i \parallel < c(x_i) \Rightarrow update\ parent\ of\ x_i \tag{7}$$

B. *Reinforcement Learning (RL) Method*

The primary challenge lies in estimating the RRTstar function with a limited number of iterations, utilizing a floating-point value for each iteration. While RRTstar can identify optimal motion paths, its sampling process is characterized by inefficiency, as it does not leverage any information regarding the environment and fails to derive insights from previously solved problems. To address these limitations, the authors have incorporated a reinforcement learning framework into the RRTstar algorithm. This integration involves an Actor that determines the appropriate actions to take, while a Critic evaluates the effectiveness of these actions and provides feedback for necessary adjustments. The overall architecture of this enhanced framework is depicted in Fig. 3.

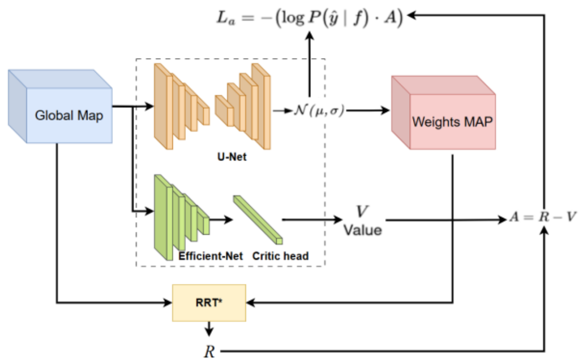

Fig. 3. The architecture of RRTstar-RL algorithm

This study diverges from the conventional approach of employing random sampling of points in RRTstar by utilizing a RL model to strategically direct sampling towards regions that exhibit a higher navigation completion rate. During the training phase, the model generates weight maps that correspond to the processed map data, thereby optimizing both path cost and computational efficiency. The RL model is predicated on three fundamental components: state, action, and reward. The state is defined as the input environment, which encompasses information regarding the starting point, destination, and obstacles. These elements are integrated into a corresponding dataset for both training and inference purposes, with each instance encoded as a tensor of dimensions (3, *H*, *W*). The action is conceptualized as a weight map, wherein each pixel represents the probability of sampling a point at its respective coordinates. The computational framework is constructed based on a normal distribution, denoted as $N(\mu, \sigma^2)$, where:

$$\mu, \sigma^2 = Actor(map) \in R^{H \times W} \tag{8}$$

In (8), the parameters of the normal distribution are processed through the Sigmoid function, resulting in a range of values between 0 and 1, thereby generating a probability map. The reward function is subsequently determined using the following formula:

$$Reward = 1000 - Cost \tag{9}$$

A reward value of zero signifies the absence of a satisfactory path within the environment. The model consists of two primary components: the Actor and the Critic. The Actor is designed using the U-Net architecture, which is responsible for generating probability maps, or, in other terms, producing actions based on the given input. In contrast, the Critic utilizes the MobileNetV2 architecture to assess the current policy, or the weight patterns generated by the Actor (π) in relation to the attainment of the reward parameter. This dual structure enhances the model's capability and efficiency in pathfinding within the environment. Unet is inherently a flexible model with input heads, accompanied by stable performance thanks to its symmetric architecture. The setup and customization with input data from the environment are highly valued. MobileNetv2 is specially designed for high speed and small model size, accompanied by minimal computational parameters. This architecture is suitable for accelerating the computation process and evaluating the effectiveness of the generated actions. From there, the optimal navigation process is selected. The evaluation process is conducted through the value function, as outlined below:

$$V(s) = E_\pi[\sum_{t=0}^{\infty} \gamma^t r_t] \tag{10}$$

where $V(s)$ is the value function at the corresponding state s. E_π is the average expectation under policy π mapped from the state. $\gamma \in [0,1)$ is the discount factor. The reward at state t is denoted as rt. The function $V(t)$ is calculated to compare the actual rewards, select the advantages (A) and participate in calculating the Critic's loss:

$$A = Reward - V(t) \tag{11}$$

Therefore, the simulation of probability distributions and maps in the figure is illustrated in Fig. 4 and Fig. 5, respectively.

					$P_{y\|x=x}(y)$
0.006	0.006	0.006	0.02	0.1	0.431
0.006	0.006	0.006	0.1	0.1	0.431
0.02	0.1	0.1	0.02	0.02	0.086
0.02	0.1	0.02	0.006	0.006	0.026
0.1	0.02	0.02	0.006	0.006	0.026
0.232	0.232	0.152	0.152	0.232	

$P_x(x)$

Fig. 4. Probability map

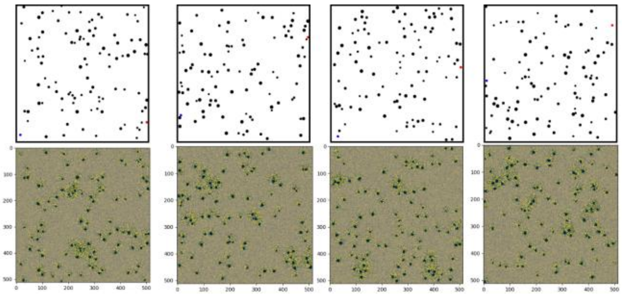

Fig. 5. The training scenarios in the dataset (above) and the trained weight maps (below). With a color scale from light to dark indicating accessible regions (light) and slow collision probability regions (dark)

The correlation between values and optimal completion rates during the pathfinding process facilitates the rapid convergence of the RRT-star algorithm. The integration of learned data minimizes the formation of redundant and inefficient search trees. Ultimately, this leads to the identification of the optimal path from the initial point to the target destination. In conclusion, the authors employ a RL framework in conjunction with RRT-star, which offers significant advantages, including the reduction of inefficient sampling, the extraction of map features that integrate trained data into the pathfinding process within the environment, and an acceleration of processing time due to a decreased necessity for sample review.

III. Results and Discussion

The experimental robot model is a three-wheeled mobile robot equipped with Lidar and computer, in Fig. 6.

(a): The WMR in the practical evironment (b): WMR's equipments

Fig. 6. Practical three wheeled mobile robot

Utilizing the established and trained environment, the authors assess and contrast the trajectories produced by the proposed model with those generated by the RRTstar algorithm. The findings are presented in Fig. 7.

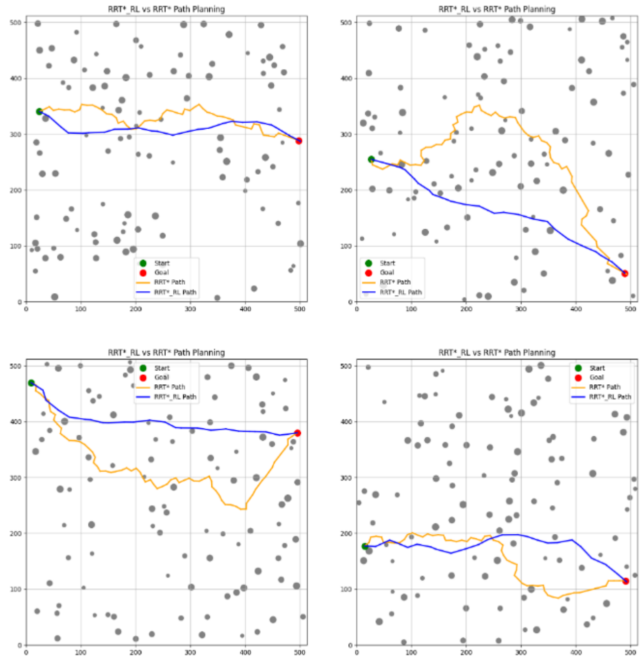

Fig. 7. Comparison of the path results calculated from the proposed method (blue) with the RRT-Star algorithms

Based on the probability of regions in the map, the proposed RRTstar-RL algorithm produces significantly shorter trajectories than the RRTstar algorithm. In Fig. 8, the paths generated based on the reinforcement learning model have an average length of 11.323 meters. While the experiments with RRTstar [61] have an average lenght of 15.74 meters and the traditional RRT algorithm [62] has a length of 14.4 meters. It is clear that the proposed RRTstar-RL algorithm produces paths that are closer to the ideal path than the unimproved approaches. In the cases of densely obstructed maps, RRTstar sometimes fails to find any path from the starting point to the destination while the proposed model almost does not record the cases of not finding a path. The authors conclude that the use of reinforcement learning combined with the RRTstar algorithm significantly improves the path finding efficiency in diverse environmental scenarios. Some potential and advantages of integrating the proposed method into path planning problems for autonomous systems, especially intelligent mechatronic systems [63], [64].

Visual analyses employing metrics such as computation time, path length, and optimality have been systematically conducted to demonstrate the superior performance of the proposed method. Notably, the optimization of computation time, particularly when utilizing pre-trained data, significantly accelerates the path-finding process. Specifically, the RRTstar-RL algorithm generates the optimal path 2.02 times faster than RRTstar and 1.6 times faster than RRT. Furthermore, improvements in path length have been empirically validated through comparative tables. Based on the referenced studies, the authors conclude that integrating reinforcement learning within the RRTstar framework substantially enhances path finding efficiency across diverse environments, yielding shorter travel distances and significantly reduced computation times, while effectively optimizing parameter utilization. The reinforcement learning model processes 5.8 million parameters and directly outputs the optimal movement trajectory, a procedure considerably faster than traditional path planning algorithms that rely on exhaustive sampling and map-based path searches. Additionally, the relatively small computational footprint of RRTstar-RL facilitates its deployment on resource-constrained systems with minimal computational overhead.

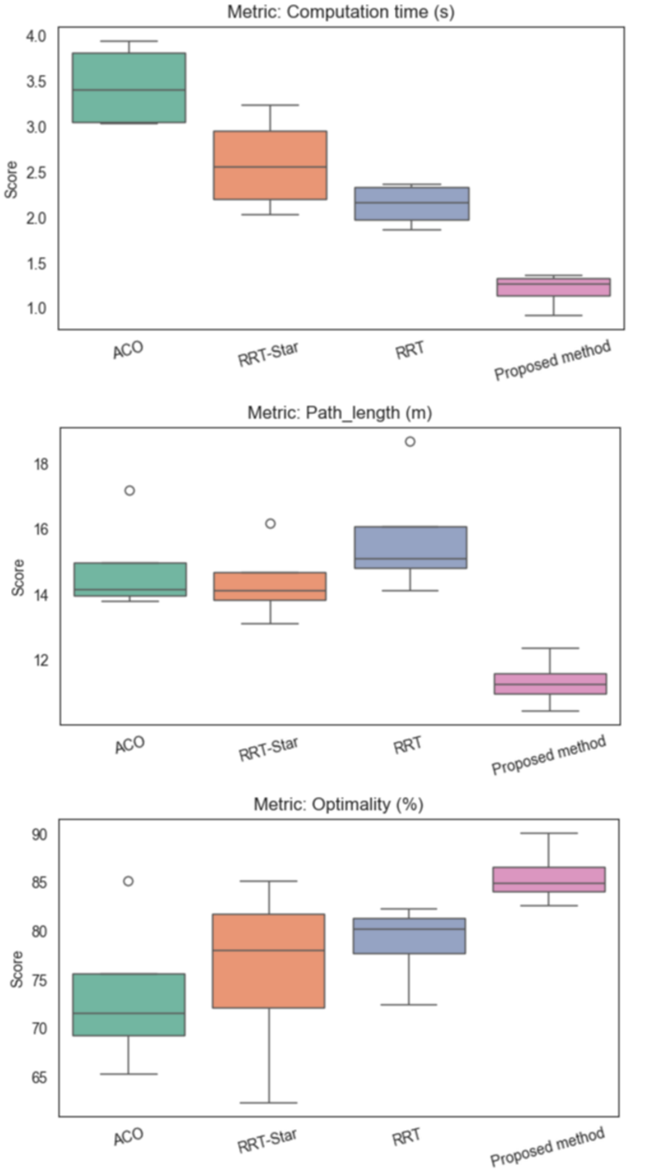

Fig. 8. Comparison of the proposed RRT-star-RL algorithm with other methods based on the metrics: computing time, path length and optimality

In order to enhance the accuracy of the proposed methodology, the RRTstar-RL algorithm has been developed to improve the efficiency of data processing within dynamic robotic environments. This algorithm is designed to effectively plan paths that ensure successful navigation to designated destinations while eliminating superfluous waypoints, thereby achieving the shortest possible path distance. Furthermore, the RRTstar-RL algorithm is integrated with a trajectory smoothing technique utilizing B-spline, which contributes to the stability of the trajectory tracking process and minimizes the error associated with the three-wheel mobile robot's turning angles, in Fig. 9.

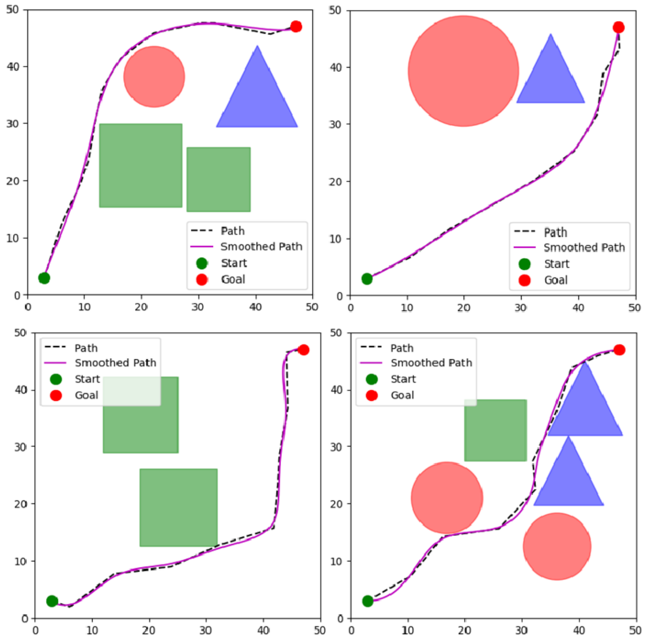

Fig. 9. The processing of the calculated trajectories is based on the proposed RRTstar-RL algorithm

Finally, to enhance the efficacy of implementing the petal-shaped complex trajectory tracking control process in practical settings. Fig. 10 illustrates that, utilizing the navigation plan derived from the proposed RRTstar-RL algorithm, the three-wheeled mobile robot demonstrates stable movement and precise navigation to each peak of the petal trajectory. This is achieved while maintaining stable trajectory tracking and minimizing error, as indicated by the black line during its motion.

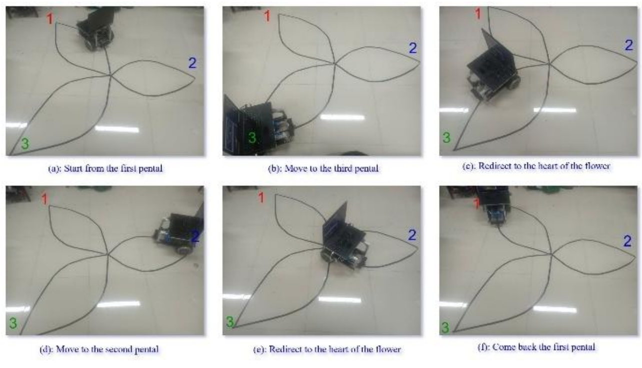

Fig. 10. Three-wheel mobile robot's three-petal trajectory tracking process

IV. CONCLUSIONS

The paper proposes an efficient and optimal path planning approach for WMRs, which incorporates collision avoidance strategies and smoothing techniques to identify the most effective navigation route. The proposed hybrid path planning framework integrates an enhanced RRTstar algorithm with a reinforcement learning methodology. The path length achieved with the proposed algorithm decreased by 28% and 21% compared to the RRT and RRTstar algorithms respectively. Furthermore, the optimization of computational time, particularly when utilizing pre-trained data, substantially accelerates the path-finding calculation process. Notably, the time required to generate the optimal path using the RRTstar-RL algorithm is 2.02 times faster than that of RRTstar and 1.6 times faster than RRT. Ultimately, the proposed RRTstar-RL path planning algorithm has been successfully validated for feasibility and effectively fulfills multiple objectives established during simulation and validation experiments. The challenges of optimizing the model to achieve fast inference speed, minimizing computational parameters are the premise for future jobs.

REFERENCES

[1] V. T. Nguyen, D. N. Duong, and T. V. Dang, "Optimal Two-Wheeled Self-Balancing Mobile Robot Strategy of Navigation using Adaptive Fuzzy controller-based KD-SegNet," *Intelligent Service Robotics*, pp. 1-25, 2025, doi: 10.1007/s11370-025-00606-0.

[2] T. T. Le, K. K. Phung Cong, and T. V. Dang, "An Improved Coverage Path planning for Service robots based on Backtracking method," *MM Science Journal*, vol. 10, pp. 7464-7468, 2024, doi: 10.17973/MMSJ.2024_10_2024063.

[3] A. Abadi *et al*., "Robust Tracking Control of Wheeled Mobile Robot Based on Differential Flatness and Sliding Active Disturbance Rejection Control: Simulations and Experiments," *Sensor*, vol. 24, no. 2849, 2024, doi: 10.3390/s24092849.

[4] M. Y. Silaa, A. Bencherif, and O. Barambones, "Indirect Adaptive Control Using Neural Network and Discrete Extended Kalman Filter for Wheeled Mobile Robot," *Actuators*, vol. 13, no. 51, 2024, doi: 10.3390/act13020051.

[5] J. Lin *et al*., "Combined Localization Method for Multimodal Wheel-Track Robots in Sheltered Space," *IEEE Access*, vol. 12, pp. 47271-47282, 2024, doi: 10.1109/ACCESS.2024.3364068.

[6] B. Kazed and A. Guessoum, "A Lyapunov based posture controller for a differential drive mobile robot," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 13, no. 1, pp. 1-10, 2024, doi: 10.11591/ijra.v13i1.

[7] S. Sachan and P. M. Pathak, "Addressing unpredictable movements of dynamic obstacles with deep reinforcement learning to ensure safe navigation for omni-wheeled mobile robot," *Proc. I. Mech. E. Part C: J. Mechanical Engineering Science*, vol. 239, no. 4, pp. 1267-1293, 2024, doi: 10.1177/09544062241281115.

[8] S. Yang *et. al*., "A RISE-based asymptotic prescribed performance trajectory tracking control of two-wheeled self-balancing mobile robot," *Nonlinear Dyn*., vol. 112, pp. 15327-15348, 2024, doi: 10.1007/s11071-024-09569-w.

[9] H. Xue, S. Lu, and C. Zhang, "An Adaptive Control Based on Improved Gray Wolf Algorithm for Mobile Robots," *Applied Sciences*, vol. 14, no. 7092, 2024, doi: 10.3390/app14167092.

[10] C. Li and Z. Li, "Dynamic Modeling and Disturbance-Observer-Enhanced Control for Mecanum-Wheeled Vehicles Under Load and Noise Disturbance," *Mathematics*, vol. 13, no. 789, 2025, doi: 10.3390/math13050789.

[11] T. V. Dang and N. T. Bui, "Multi-Scale Fully Convolutional Network-Based Semantic Segmentation for Mobile Robot Navigation," *Electronics*, vol. 12, no. 3, p. 533, 2023, doi: 10.3390/electronics12030533.

[12] T. V. Dang and N. T. Bui, "Obstacle Avoidance Strategy for Mobile Robot based on Monocular Camera," *Electronics*, vol. 12, no. 8, p. 1932, 2023, doi: 10.3390/electronics12081932.

[13] H. T. Linh and T. V. Dang, "An ultra-fast Semantic Segmentation Model for AMR's Path Planning," *Journal of Robotics and Control*, vol. 4, no. 3, pp. 424-430, 2023, doi: 10.18196/jrc.v4i3.18758.

[14] T. V. Dang and N. T. Bui, "Design the Abnormal Object Detection System Using Template Matching and Subtract Background Algorithm," *Proceedings of the 3rd Annual International Conference on Material, Machines and Methods for Sustainable Development*, pp. 87-95, 2024, doi: 10.1007/978-3-031-57460-3_10.

[15] T. V. Dang, N. N. Bui, and N. T. Bui, "Binary-SegNet: Efficient Convolutional Architecture for Semantic Segmentation Based on Monocular Camera," *From Smart City to Smart Factory for Sustainable Future: Conceptual Framework, Scenarios, and Multidiscipline Perspectives*, pp 275-285, 2024, doi: 10.1007/978-3-031-65656-9_28.

[16] V. T. Nguyen, N. N. Bui, D. M. C. Tran, P. X. Tan, and T. V. Dang, "FDE-Net: Lightweight Depth Estimation for Monocular Cameras," *The 13th International Symposium on Information and Communication Technology (SOICT 2024)*, pp. 3-13, 2025, doi: 10.1007/978-981-96-4282-3_1.

[17] T. V. Dang, V. D. Ngo, M. Q. Ngo, and N. T. Bui, "OD-CT3D: Object Detection Model based on CentreTrack 3D for Mobile Robot Global Path Planning," *5th International Conference on Intelligent Systems & Networks*, 2025.

[18] Y. Zheng *et al*., "Adaptive fuzzy sliding mode control of uncertain nonholonomic wheeled mobile robot with external disturbance and actuator saturation," *Information Sciences*, vol. 663, no. 120303, 2024.

[19] Y. Wu and W. Yu, "Asymptotic tracking control of uncertain nonholonomic wheeled mobile robot with actuator saturation and external disturbances," *Neural Computing and Applications*, vol. 32, no. 2, 2020, doi: 10.1007/s00521-019-04373-9.

[20] A. D. Nguyen, T. D. Vu, Q. A. Vu, and T. V. Dang, "Research on Modeling and Object Tracking for Robot Arm based on Deep Reinforcement Learning," *MM Science Journal*, vol. 6, pp. 8459-8463, 2025, doi: 10.17973/MMSJ.2025_06_2025059.

[21] B. Shi *et al*., "An intelligence enhancement method for USV navigation visual measurement based on variable gradient soft-threshold correction," *Measurement*, vol. 242, no. 116201, 2025, doi: 10.1016/j.measurement.2024.116201.

[22] T. V. Dang, "Optimization Hybrid Path Planning based on A-star Algorithm combining with DWA," *MM Science Journal*, vol. 10, pp. 7551-7555, 2024, doi: 10.17973/MMSJ.2024_10_2024077.

[23] T. V. Dang and D. S. Nguyen, "Optimal Navigation Based on Improved A* Algorithm for Mobile Robot," *Intelligent Systems and Networks*, pp. 574-580, 2023, doi: 10.1007/978-981-99-4725-6_68.

[24] T. P. Nguyen, H. Nguyen, and N. Q. T. Ha, "Towards sustainable scheduling of a multi-automated guided vehicle system for collision avoidance," *Computers and Electrical Engineering*, vol. 120, no. 109824, 2024, doi: 10.1016/j.compeleceng.2024.109824.

[25] C. C. Huang, C. H. Huang, and J. S. Shaw, "Development of an AMR Applying Cartographer Combined with Visual Odometry for Navigation," *Journal of Applied Science and Engineering*, vol. 28, no 1, pp. 35-40, 2024, doi: 10.6180/jase.202501_28(1).0004.

[26] Z. Lin, L. Lu, Y. Yuan, and H. Zhao, "A novel robotic path planning method in grid map context based on D* lite algorithm and deep learning," *J. Circuits Syst. Comput.*, vol. 33, no. 4, p. 2450057, 2023, doi: 10.1142/S0218126624500579.

[27] L. Liu *et al*., "Global dynamic path planning fusion algorithm combining jump-A* Algorithm and dynamic window approach," *IEEE Access*, vol. 9, pp. 19632-19638, 2021, doi: 10.1109/ACCESS.2021.3052865.

[28] Z. Xunyu, T. Jun, H. Huosheng, and P. Xiafu, "Hybrid path planning based on safe A* Algorithm and adaptive window approach for mobile robot in large-scale dynamic environment," *J. Intell. Rob. Syst*., vol. 99, no. 2, pp. 65-77, 2020, doi: 10.1007/s10846-019-01112-z.

[29] C. C. Hsu, Y. J. Chen, M. C. Lu, and L. S. An, "Hybrid path planning incorporating global and local search for Mobile robot," *Conference Towards Autonomous Robotic Systems*, vol. 7429, 2012, doi: 10.1007/978-3-642-32527-4_50.

[30] M. Imran and F. Kunwar, "A Hybrid path planning technique developed by integrating global and local path planner," *2016 International Conference on Intelligent Systems Engineering (ICISE)*, pp. 118-122, 2016, doi: 10.1109/INTELSE.2016.7475172.

[31] L. S. Liu *et al*., "Path planning for smart car based on Dijkstra algorithm and dynamic window approach," *Wirel. Commun. Mob. Comput*., vol. 4, pp. 1-12, 2021, doi: 10.1088/10.1155/2021/8881684.

[32] R. Song, Y. Liu, and R. Bucknall, "Smoothed A∗ algorithm for practical unmanned surface vehicle path planning," *Appl. Ocean Res*., vol. 83, no. 6, pp. 9-20, 2019, doi: 10.1016/j.apor.2018.12.001.

[33] L. Zhao, G. Li, and H. Zhang, "Global and Local Awareness: Combine Reinforcement Learning and Model-Based Control for Collision Avoidance," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 5, pp. 422-432, 2024, doi: 10.1109/OJITS.2024.3424587.

[34] Y. Zhu, J. Zhu, and P. Zhang, "Local obstacle avoidance control for multi-axle and multi-steering-mode wheeled robot based on window-zone division strategy," *Robotics and Autonomous Systems*, vol. 183, no. 104843, 2025, doi: 10.1016/j.robot.2024.104843.

[35] L. Xiang, X. Li, H. Liu, and P. Li, "Parameter Fuzzy Self-Adaptive Dynamic Window Approach for Local Path Planning of Wheeled Robot," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 1-6, 2021, doi: 10.1109/OJITS.2021.3137931.

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

[37] M. Kobayashi, H. Zuski, T. Nakamura, and N. Motoi, "Local Path Planning: Dynamic Window Approach With Q-Learning Considering Congestion Environments for Mobile Robot," *IEEE Access*, vol. 11, pp. 96733-96742, 2023, doi: 10.1109/ACCESS.2023.3311023.

[38] M. Kramer and T. Bertram, "Improving Local Trajectory Optimization by Enhanced Initialization and Global Guidance," *IEEE Access*, vol. 10, pp. 29633-29645, 2022, doi: 10.1109/ACCESS.2022.3159233.

[39] T. V. Dang, "Research and design of a path planning using an improved RRT* algorithm for an autonomous mobile robot," *MM Science Journal*, vol. 10, pp. 6712-6716, 2023, doi: 10.17973/MMSJ.2023_10_2023051.

[40] T. V. Dang, "Autonomous mobile robot path planning based on enhanced A* algorithm integrating with time elastic band," *MM Science Journal*, vol. 10, pp. 6717-6722, 2023, doi: 10.17973/MMSJ.2023_10_2023052.

[41] T. V. Dang, D. S. Nguyen, and N. T. Bui, "Hybrid Path Planning for Mobile Robot based on Improved A* Fusion Dynamic Window Approach," *Proceedings of the International Conference on Intelligent Systems and Networks*, pp. 82-88, 2024, doi: 10.1007/978-981-97-5504-2_10.

[42] Y. Sun *et al*., "Local Path Planning for Mobile Robots Based on Fuzzy Dynamic Window Algorithm," *Senso*r, vol. 23, no. 8260, 2023, doi.: 10.3390/ s23198260

[43] Y. Chen, G. Bai, Y. Zhan, X. Hu, and J. Liu, "The USV Based on Improved ACO-APF Hybrid Algorithm with Adaptive Early-Warning," *IEEE Access*, vol. 9, pp. 40728-40742, 2021, doi: 10.1109/ACCESS.2021.3062375.

[44] Y. Ji, L. Ni, C. Zhao, C. Lei, and Y. Du, "TriPField: A 3D Potential Field Model and Its Applications to Local Path Planning of Autonomous Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, 2023, doi: 10.1109/TITS.2022.3231259.

[45] D. Li, W. Yin, W. E. Wong, M. Jian, and M. Chau, "Quality-Oriented Hybrid Path Planning Based on A∗ and Q-Learning for Unmanned Aerial Vehicle," *IEEE Access*, vol. 10, pp. 7664-7674, 2021, doi: 10.1109/ACCESS.2021.3139534.

[46] C. Wang, X. Yang, and H. Li, "Improved Q-Learning Applied to Dynamic Obstacle Avoidance and Path Planning," *IEEE Access*, vol. 10, pp. 92879-92888, 2022, doi: 10.1109/ACCESS.2022.3203072.

[47] Y. Liu, C. Wang, and H. Wu, "Mobile Robot Path Planning Based on Kinematically Constrained A-Star Algorithm and DWA Fusion Algorithm," *Mathematics*, vol. 11, no. 4552, 2023, doi: 10.3390/math11214552.

[48] Y. Feng, W. Zhang, and J. Zhu, "Application of an Improved A* Algorithm for the Path Analysis of Urban Multi-Type Transportation Systems," *Appl. Sci*., vol. 13, no. 13090, 2023, doi: 10.3390/app132413090.

[49] T. V. Dang and P. X. Tan, "Hybrid Mobile Robot Path Planning Using Safe JBS-A*B Algorithm and Improved DWA Based on Monocular Camera," *Journal of Intelligent & Robotic Systems*, vol. 110, no. 151, pp. 1-21, 2024, doi: 10.1007/s10846-024-02179-z.

[50] H. Esmaiel, G. Zhao, Z. A. Qasem, J. Qi, and H. Sun, "Double-Layer RRT* Objective Bias Anytime Motion Planning Algorithm," *Robotics*, vol. 13, no. 41, 2024, doi: 10.3390/robotics13030041.

[51] H. Wang, X. Zhou, and J. Li, "Improved RRT* Algorithm for Disinfecting Robot Path Planning," *Sensors*, vol. 24, no. 1520, 2024, doi: 10.3390/s24051520.

[52] Q. Zhang, Y. Liu, and J. Qin, "An Informed-Bi-Quick RRT* Algorithm Based on Offlne Sampling: Motion Planning Considering Multiple Constraints for a Dual-Arm Cooperative System," *Actuators*, vol. 13, no. 75, 2024, doi: 10.3390/act13020075.

[53] H. Han, J. Wang, L. Kuang, X. Han, and H. Xue, "Improved Robot Path Planning Method Based on Deep Reinforcement Learning," *Sensor*, vol. 23, no. 5622, 2023, doi: 10.3390/s23125622.

[54] C. H. Nguyen, Q. A. Vu, K. K. Phung Cong, T. V. Dang, "Optimal obstacle avoidance strategy using deep reinforcement learning based on stereo camera," *MM Science Journal*, vol. 10, pp. 7556-7561, 2024, doi: 10.17973/MMSJ.2024_10_2024078.

[55] C. Copurkaya, E. Meriç, F. P. Akbulut, and C. Catal, "A multi-pretraining U-Net architecture for semantic segmentation," *Signal Image and Video Processing*, vol. 19, no. 8, 2025, doi: 10.1007/s11760-025-04125-4.

[56] K. Rezaee *et al*., "Hand gestures classification of sEMG signals based on BiLSTM-metaheuristic optimization and hybrid U-Net-MobileNetV2 encoder architecture," *Scientific Reports*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-024-82676-1.

[57] Q. Zhu *et al*., "A study on expression recognition based on improved mobilenetV2 network," *Scientific Reports*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-024-58736-x.

[58] Z. Li, "Enhancing Tea Leaf Disease Identification with Lightweight MobileNetV2," *Computers, Materials & Continua*, vol. 80, no. 1, pp. 679-694, 2024, doi: 10.32604/cmc.2024.051526.

[59] T. V. Dang, D. M. C. Tran, and P. X. Tan, "IRDC-Net: Lightweight Semantic Segmentation Network Based on Monocular Camera for Mobile Robot Navigation," *Sensors*, vol. 23, no. 15, p. 6907, 2023. doi: 10.3390/s23156907

[60] T. V. Dang, X. T. Phan, and N. N. Bui, "KD-SegNet: Efficient Semantic Segmentation Network with Knowledge Distillation Based on Monocular Camera," *Computers, Materials & Continua*, vol. 82, no. 2, pp. 2001-2026, doi: 10.32604/cmc.2025.060605.

[61] M. Zhang, S. Liu, Q. Zhou, and X. Han, "A novel path planning scheme based on Fast-IBi-RRT* algorithm for industrial robots," *Applied Intelligence*, vol. 55, no. 11, 2025, doi: 10.1007/s10489-025-06694-w.

[62] S. Lei *et al*., "Research on improved RRT path planning algorithm based on multi-strategy fusion," *Scientific Reports*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-92675-5.

[63] V. T. Nguyen, C. D. Do, T. V. Dang, T. L. Bui, and P. X. Tan, "A Comprehensive RGB-D Dataset for 6D Pose Estimation for Industrial Robots Pick and Place: Creation and Real-World Validation," *Results in Engineering*, vol. 24, no. 103459, 2024, doi: 10.1016/j.rineng.2024.103459.

[64] T. V. Dang, D. M. C. Tran, N. N. Bui, and P. X. Tan, "ELDE-Net: Efficient Light-weight Depth Estimation Network for Deep Reinforcement Learning-based Mobile Robot Path Planning," *Computers, Materials & Continua*, 2025.