Journal of Robotics and Control (JRC)
Volume 6, Issue 5, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i5.27038

Integration of RRT* and D* Lite with Path Smoothing for Robust Path Planning in a Dynamic Robotic Arm Environment

Hameed S. Hameed [1], Firas A. Raheem [2], Omar F. Lutfy [3]
[1, 2, 3] College of Control and Systems Engineering, University of Technology, Baghdad, Iraq
Email: [1] cse.22.03@ grad.uotechnology.iq, [2] Firas.A.Raheem@uotechnology.edu.iq, [3] Omar.F.lutfy@uotechnology.edu.iq
*Corresponding Author

***Abstract*—This paper presents a hybrid trajectory planning framework that integrates the strengths of rapidly-exploring random tree star (RRT), D* Lite (DL) algorithms, and a Gaussian filter to enable efficient and smooth navigation of a two-link robotic arm in dynamic environments. RRT* is employed to generate a globally optimal path from the initial to the goal configurations by exploring the Cartesian workspace while considering kinematic and dynamic constraints, including static obstacles. To handle environmental changes, DL is incorporated for local re-planning, allowing the trajectory to adapt in real-time when obstacles move or new ones appear, thus ensuring continuous path feasibility. The initial path produced by RRT* is incrementally optimized, and any necessary local adjustments are efficiently handled by DL without re-planning the entire path. To further enhance the quality of motion, the Shortcut smoothing + Gaussian filter is applied for path smoothing, resulting in improved trajectory continuity, computational efficiency, and robustness in the presence of dynamic obstacles. This hybrid approach offers the optimality of RRT*, the adaptability of D* Lite, and the smoothness required for practical robotic applications.**

Keywords—Hybrid Path Planning; Sampling-Based Algorithms; Dynamic Environments.

I. INTRODUCTION

Robotic arms are widely employed in industrial settings due to their high precision and versatility in performing complex tasks. Consequently, there is a growing need to develop advanced path planning strategies that optimize their motion efficiency and accuracy. In this regard, path planning plays a critical role in robotics, as it determines the motion behavior of the robot at an early stage regardless of whether the planning approach is conventional or based on intelligent algorithms [1]. Specifically, generating a minimal-length, collision-free path for a manipulator operating in a dynamic workspace with mobile obstacles can use potential-field methods (artificial, harmonic), sampling-based planners (RRT* and PRM), grid-based techniques, and AI-based strategies (evolutionary algorithms, ANNs, and DRL) [2]-[4]. In addition, the choice of the optimal path planner may be classified by energy use/energy cost [5], [6]. The task of this planner is to minimize energy consumption during motion and compute a short, safe, smooth start-to-goal path while avoiding obstacles with minimal traversal time. In addition, it reduces computation, especially in cluttered scenes by solving IK at each waypoint, verifying kinematic/dynamic feasibility, operating under real-time constraints, coordinating coupled joints, selecting the best configuration, preventing self-collisions, building a precise environment model, using robust collision checking, and maintaining a safety clearance in collaborative settings. Integrate RRT* with DL in one framework; re-plan at a proximity threshold; smooth the path with a Gaussian filter to safely bypass moving obstacles and shorten paths, speed up re-plans, lower computational demand, and require less manual shape tuning [7]-[9].

Our key contributions are utilizing RRT* to find a feasible seed path and refine its quality, integrating RRT* with DL in a single hybrid method, employing DL for path repair to avoid obstacles, and evaluating the hybrid simulation method and analyzing its performance in dynamic environments. In addition, the Shortcut smoothing + Gaussian filter is applied for application-specific path smoothing.

We organize the rest of the paper as follows. Section 2 surveys dynamic-obstacle planning with emphasis on hybrid RRT* approaches. Sec. 3 describes the two-link arm kinematics, RRT*, D*Lite, the RRT*–DL method, and the Smoothing and collision detection procedures. Section 4 presents the simulations and discussion on point-mass and two-link arm and compares the hybrid and the baseline RRT*. Section 5 gives the conclusions highlighting efficiency and safety gains of the RRT*–D* Lite approach in dynamic environments.

II. RELATED WORKS

Path planning for robotic manipulators in dynamic environments requires a solid theoretical foundation that integrates concepts from robotics kinematics, motion planning, and optimization. For two-link robotic arms, the forward and inverse kinematics define the relationship between joint angles and end-effector positions, forming the basis for trajectory generation within the workspace [10]. In complex scenarios, where both static and dynamic obstacles exist, classical approaches such as grid-based methods or potential fields often suffer from high computational cost or local minima. To this end, sampling-based algorithms, particularly RRT*, have emerged as powerful tools due to their probabilistic completeness and asymptotic optimality in finding feasible collision-free paths [11]-[13]. However, in highly dynamic scenarios, RRT* alone may be inefficient, requiring repeated re-exploration when obstacles change

position. To address this limitation, incremental search algorithms such as D* Lite provide efficient re-planning capabilities by updating paths locally without reconstructing the entire tree. This section presents the theoretical principles underlying the hybrid RRT*–D* Lite framework, which combines global optimal exploration with adaptive local re-planning to achieve smooth and collision-free trajectories for two-link robotic arms operating in constrained and dynamic environments.

For instance, Liang *et al.* [14] proposed an improved RRT* variant, AGP-RRT*, to enhance multi-axis robotic arm path planning by addressing poor directionality and slow convergence. They introduced adaptive probabilistic sampling and gravitational step-size adjustment to guide efficient search, alongside cubic B-spline smoothing for flexible path refinement. In another work, Fang *et al.* [15] proposed the XN-RRT* algorithm to improve path planning efficiency and success rate in complex pitaya (dragon fruit) harvesting environments. They enhanced sampling via normal distribution, improved tree expansion using a potential field method, and refined the path with a greedy algorithm and B-spline smoothing. In addition, Hameed *et al.* [16] introduced the APF-IRRT*-HS algorithm, enhancing RRT* by integrating artificial potential fields and Halton sequence sampling with a modified goal-biased strategy. The method, tested on mass point and two-link robots, showed superior performance in path length, computation time, and iteration count. Moreover, Hameed *et al.* [17] proposed the APF-IRRT*-SB, combining APF with an enhanced RRT* using the Sobol-Burkardt sequence and a probabilistic goal strategy for efficient path planning. Evaluated on mass point and two-link robot scenarios, it improved path length and computational efficiency across varying environments. Additionally, Yang *et al.* [18] introduced the MMD-RRT, an improved RRT-based method with multi-mode dynamic sampling and adaptive step sizing to reduce randomness and improve search efficiency. A greedy strategy was applied for path optimization by removing redundant nodes. Li *et al.* [19] proposed the RCM strategy to enhance RRT-based robotic arm path planning by integrating probabilistic sampling, a reward mechanism, and the B-spline interpolation. This approach reduces redundant nodes, improves smoothness, and enhances search efficiency. Gu *et al.* [20] proposed the LRRT* algorithm using an improved Lévy flight strategy and effective region sampling to overcome blind sampling and slow planning in RRT*. The method consists of two stages: initial path finding using goal-oriented exploration and optimization using localized sampling and node rejection. Wang *et al.* [10] proposed the T-ABA* algorithm, integrating the Adaptive Bidirectional A* with Transformer models to improve robotic arm path planning and dynamic obstacle avoidance. The Transformer enhances heuristic adjustment and collision prediction, ensuring smoother, more efficient paths and reducing computational overhead. The simulations showed that the T-ABA* effectively optimizes both single- and dual-arm tasks by reducing joint revolutions, smoothing turns, and preventing inter-arm collisions. Zhang *et al.* [21] introduced the Fast-IBI - RRT* algorithm to improve path planning for robotic arms by enhancing sampling with probabilistic target region sampling and redundant node deletion. It further optimizes path quality through adaptive step sizing and a reconnecting grandfather node strategy. Experimental results in multiple environments confirmed reduced planning time, improved smoothness, and lower path cost compared to IBi - RRT*. Balint *et al.* [22] tackled the issue of sample inefficiency in Deep RL by integrating it with RRT, using RRT to generate diverse experiences that enrich the training buffer and improve efficiency; applied to autonomous vehicle trajectory tracking, this hybrid method outperformed conventional Deep RL across multiple metrics. Tahmasbi *et al.* [23] proposed Zonal RL-RRT, which combines kd-tree zone partitioning with Q-learning for high-level planning, achieving up to 3× faster performance than RRT/RRT* and 1.5× better than heuristic or learning-based methods across 2D–6D spaces, and validated its. Finally, Chao *et al.* [24] proposed the DL-RRT*, which is a hybrid path planning method combining RRT* and D* Lite. In dynamic, radiation-rich scenarios, this approach cuts down on path re-planning latency by using D* Lite's grid search to establish high-quality starter paths and boost the RRT*'s convergence speed.

In this regard, the majority of prior research has overlooked the process of path reconfiguration and the influence and efficiency of localized path adjustments. To address this issue, a combined RRT* and D* Lite methodology is introduced, integrating the advantages of the two planning methods. While RRT* restarts the planning process entirely upon encountering obstacles, the hybrid method enables rapid localized path corrections, minimizing the duration of path recalculations. Although it requires upfront computational effort to construct the search structure, successive path updates are both swift and reliable, promoting improved completion rates and enhanced real-time adaptability. The approach exploits RRT* for fast exploration and D* Lite for adaptive, real-time route modifications, while applying a Gaussian filter to smooth out trajectory irregularities, making it well-suited for responsive and flexible navigation in changing scenarios [25].

III. THE PROPOSED METHOD

The proposed method integrates the strengths of RRT* and D* Lite to achieve smooth and collision-free path planning in dynamic environments [26]. More specifically, the RRT* is first employed to construct a global near-optimal path by exploring the free space within the workspace. While this step ensures asymptotic optimality, its efficiency decreases when obstacles move, requiring repeated re-exploration. To address this limitation, the D* Lite is incorporated as a local re-planning module that updates only the affected portions of the path when dynamic changes occur. This hybrid framework allows the algorithm to balance global exploration and local adaptability, ensuring safe trajectory generation even in highly constrained spaces. By combining these two algorithms, the method reduces the path length, the computation time, and the number of iterations while maintaining robustness against dynamic obstacles [27].

A. Robotic Arm Kinematics

The workspace coordinates of the two-joint manipulator, as shown in Fig. 1, are determined by the joint angles θ_1 and θ_2 [28], [29]. Every joint setup map to a distinct pose and location in the task space. In the case of a two-segment

manipulator with revolute joints, the robot's pose is entirely defined by θ_1 (from the base) and θ_2 (from the preceding segment). This 2D configuration maps joint angles (θ_1, θ_2) to a unique (x, y) tool-tip location. Computing the forward kinematics is straightforward, yields a single result, and determines end-effector position from angles. On the other hand, the inverse kinematics is used to find joint angles for a desired position in the workspace. However, it may yield no solution, a single solution, or multiple possible configurations [30], [31]. Joint angle constraints and kinematic singularities must also be considered to generate and perform movements in the task space while ensuring that unreachable or invalid targets are avoided [32], [33].

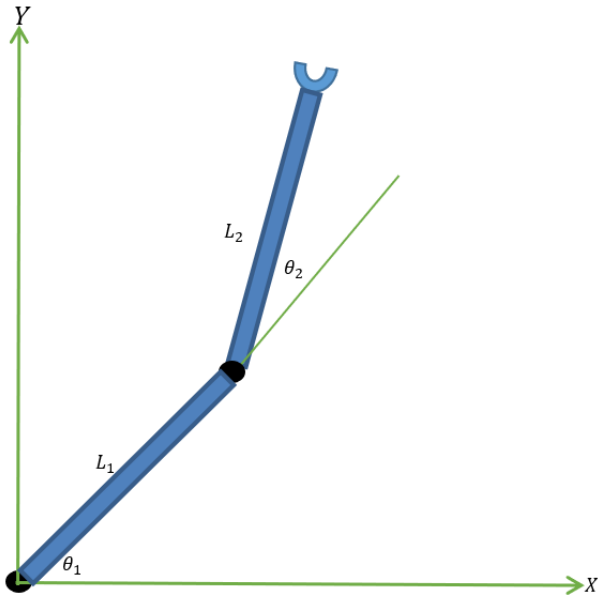

Fig. 1. The two-link robotic arm

Specifically, we adopt this method to verify whether the goal position can be accessed, determine the joint configurations required to move the end-effector to coordinates (x, y), and modify the target point if it exceeds the manipulator's reach or falls beyond the robot's operational area [34], [35].

Then, we compute θ_2 as follows:

$$cos(\theta_2) = \frac{r^2 - L1^2 - L2^2}{2.L1.L2} \quad (1)$$

Where, $r = \sqrt{x^2 + y^2}$, L_1 is the length of link 1, and L_2 is the length of link 2.

$$\theta_2 = cos^{-1}\left(\frac{r^2 - L1^2 - L2^2}{2.L1.L2}\right) \quad (2)$$

$$\theta_1 = tan^{-1}\left(\frac{y}{x}\right) - tan^{-1}\left(\frac{L2.sin(\theta_2)}{L1 + L2.cos(\theta_2)}\right) \quad (3)$$

B. Random Tree Star (RRT*)

The rapidly exploring random tree star (RRT*) is an optimal extension of the RRT algorithm that incrementally builds a search tree while improving the quality of the generated path [36]-[40].

The algorithm begins by randomly sampling a state X_{rand} in the search space and identifying the nearest node X_{near} from the existing tree using a distance metric [41], [42]. Then, a steering function generates a new node X_{new} toward the sample, constrained by a maximum step size and obstacle avoidance rules. The RRT* considers a neighborhood set X_{near} around X_{new}, and the parent node is selected to minimize the cumulative cost-to-come [43]-[45]:

$$X_{parent} = \arg min_{x \in X_{near}[c(x) + \|x - x_{new}\|} \quad (4)$$

where $c(x)$ represents the accumulated path cost from the start node. After connecting X_{new}, a rewiring step checks whether re-routing existing nodes in X_{near} X_{new} reduces their cost, and updates the tree accordingly [39], [46]-[48]. Through this continuous optimization, the algorithm ensures that the path cost is gradually reduced, leading to convergence toward the optimal solution. The RRT* therefore guarantees both probabilistic completeness (a solution will be found if it exists) and asymptotic optimality (the solution approaches the global optimum as the number of iterations tends to infinity) [49], [50]. These properties make the RRT* a widely used and powerful tool in robotic path planning, particularly for navigating complex, high-dimensional, and obstacle-rich environments [51]-[54].

C. D*Lite

The D* Lite algorithm is an incremental heuristic search method that efficiently re-plans paths in dynamic environments [55]. It maintains two values for each node: the g-value (cost-to-come) and the rhs-value (one-step look-ahead cost). The shortest-path condition is satisfied when $g(s) = rhs(s)$ for all nodes, and the rhs-value is updated according to [56], [57]:

$$rhs(s) = min_{s' \in Succ(s)}[c(s, s') + g(s')] \quad (5)$$

Where $c(s, s')$ denotes the transition cost between nodes. A priority queue orders the nodes using keys based on path cost and heuristic estimates, ensuring efficient updates [58]. When an edge cost changes due to moving obstacles, only affected nodes are recalculated instead of re-computing the entire path. The algorithm expands inconsistent nodes $g(s) \neq rhs(s)$ and rewires their neighbors to restore consistency [51], [59]. This incremental update mechanism allows fast re-planning and ensures that the computed path remains optimal with respect to the new environment [60]-[63]. Fig. 2 illustrates this process, where the blue dashed path shows the initial shortest route from the start to the goal, and when the red nodes representing dynamic obstacles appear, the original path becomes blocked.

The D* Lite then performs a local update and generates a green re-planned path, efficiently navigating around the new obstacles without reconstructing the entire search tree [64]-[66]. This property makes the D* Lite highly suitable for real-time robotic navigation in environments with dynamic changes [67].

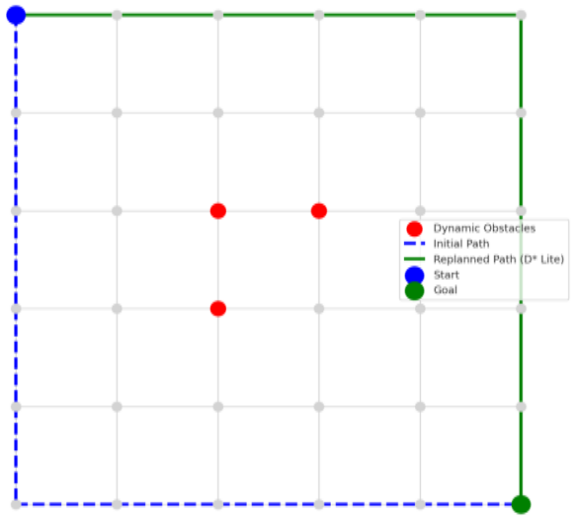

Fig. 2. D*Lite re-planning

D. The Hybrid Approach (RRT-D*Lite)*

The RRT* method is particularly suitable for high-dimensional global path planning, efficiently identifying initial paths even within complex and dynamic environments [68], [69]. It iteratively enhances its solution over time, moving towards an optimal path. With sufficient computation, it guarantees convergence to the shortest feasible trajectory [70]-[72]. In contrast, the D* Lite excels at local adaptive planning, adjusting trajectories in response to changing obstacle configurations [58]. This feature makes the DL highly effective in dynamic scenarios, maintaining feasible paths when the environment evolves. The hybrid RRT*-DL approach combines these strengths: while RRT* handles global exploration, DL efficiently manages local modifications. Although the hybrid system incurs an initial cost to construct a planning grid, it compensates with rapid, incremental updates that reduce total runtime. When obstacles disrupt the RRT* path, the pure RRT* may struggle to recover promptly, increasing collision risk. In comparison, the hybrid model improves reliability under frequent environmental shifts. Designed for real-time execution, this method typically achieves better responsiveness and higher success rates, even if the resulting paths are marginally longer.

The flowchart in Fig. 3 outlines the hybrid RRT*+DL path planning process for robotic navigation in dynamic environments. The algorithm begins by initializing the environment parameters, including static and dynamic obstacles, and then builds an occupancy grid. The DL is initialized with start and goal cells for local path correction. The system then creates the figures, plots the obstacles, and enters a main loop iterating over a set number of nodes. In each loop, dynamic obstacles oscillate and update the occupancy grid. If any grid cell changes, the DL locally updates its vertices. The RRT* then tries to expand toward the goal. If a collision-free path is found, the timing updates and the next iteration begins. Concurrently, the robot executes its motion by reconstructing the RRT* tree path, applying shortcutting and Gaussian smoothing, and updating the DL start state. The DL algorithm checks for path feasibility. If a valid path is found, it is smoothed and converted to world coordinates. The robot then snaps to the nearest path point and moves based on its speed. If the robot reaches the goal (within a tolerance), the loop ends, and performance metrics are printed. If not, the iteration and the time counter are updated for the next cycle. This hybrid method leverages the RRT* for global planning and the DL for efficient local re-planning, making it ideal for dynamic environments as shown in the following Pseudo-code:

The Hybrid RRT + D*L* Path Planning Pseudo-code,

```
START Robot_Path_Planning
1. Initialize all parameters:
   - Static and dynamic obstacles
   - Start/goal positions
   - Robot settings
2. Build the occupancy grid with resolution
   - Initialize D* Lite with startCell and goalCell
3. Create the initial figure
   - Draw the static/dynamic obstacles and
start/goal
4. FOR iter = 1 to numNodes DO:
    a. Update positions of dynamic obstacles
       (oscillating)
    b. Redraw dynamic obstacles in both subplots
    c. Rebuild the occupancy grid
    d. IF any grid cells are changed THEN:
       FOR each changed cell:
       Call dstar_lite_update_vertex(cell)
       END FOR
    E ND IF
    e. Proceed with RRT* expansion up to
       maxAttempts:
        - Add a node to the tree
        - Draw the edge
    f. IF a collision-free path to the goal is
       found THEN:
       Update time and drawing
       CONTINUE to the next iteration
       ELSE:
       CONTINUE to the next iteration
       END IF
       END FOR
5. Reconstruct the path to the closest node to
the goal
6. Apply shortcutting + Gaussian smoothing
   (safety-checked)
7. Set D* Lite start to the current robot position
8. Call dstar_lite_compute_path()
9. IF D* Lite path is found THEN:
    Convert cells to world coordinates
    Apply smoothing and plot
ELSE:
    Keep the previous final path
END IF
10. IF the final path is non-empty THEN:
    Snap the robot to the nearest point on the
path
    Advance by robot_speed
    Update the robot marker
ELSE:
    There is no motion in this iteration
END IF
11. IF the goal is reached within the tolerance
THEN:
    Stop the loop
    Compute and print the metrics
    END
ELSE:
    Update time, iter++
    Draw a robot marker
    RETURN to Step 4
END.
```

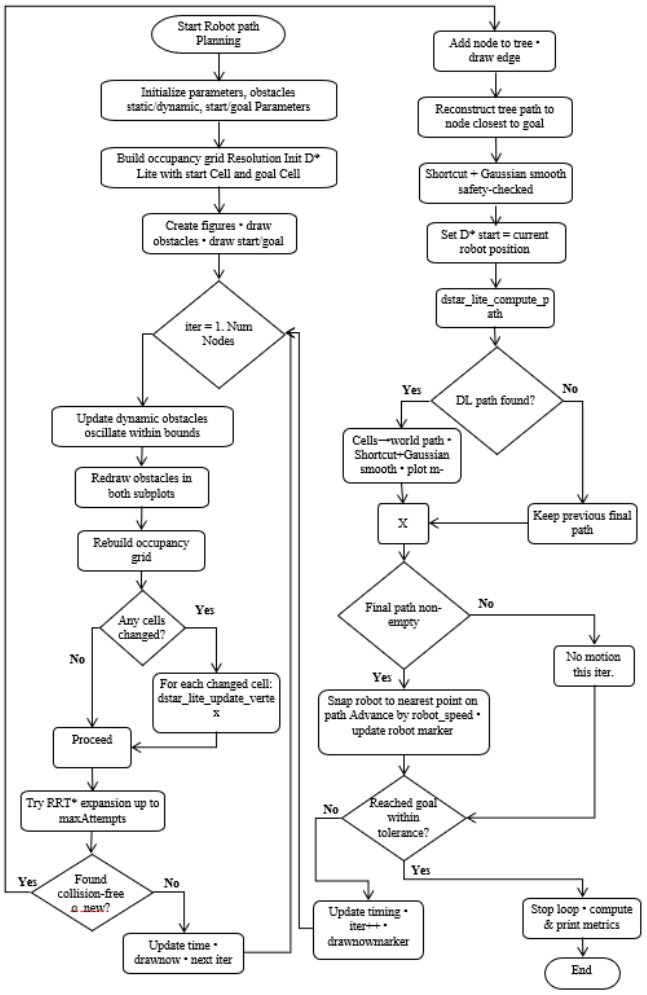

Fig. 3. Flowchart for hybrid approach (RRT*-DL)

E. Gaussian Filter for Smoothness

The Gaussian function refines the initial RRT* trajectory, which might have abrupt changes or irregularities due to stochastic sampling [73], [74]. This refinement guarantees that the path is more consistent and suitable for robotic arm execution. Moreover, minor irregularities or path noise are minimized to enhance the robot's movement steadiness and suppress disturbances [75]-[77]. The sigma (σ) parameter adjusts the intensity of the filtering process; where increasing σ leads to stronger path refinement, but may overly generalize the route, possibly causing obstacle hits or inefficient paths. The Gaussian function is generally expressed mathematically, with σ defining the spread of the Gaussian profile [78]. It refines the trajectory by convolving the function over a series of waypoints, and the length of the smoothing interval is determined accordingly.

$$G(x) = \left[\frac{1}{\sqrt{2\pi\rho^2}}\right].Exp\left(\frac{-x^2}{\sigma^2}\right) \quad (6)$$

Where $G(x)$, $G(y)$ is the value of the Gaussian function at point x, y, respectively, ρ^2 *or* σ^2 is the variance of the distribution (σ is the standard deviation), and $\sqrt{2\pi\rho^2}$ is a normalization factor to ensure that the total area under the curve is 1. $Exp(-x^2/\sigma^2)$ has higher values for points near the center (x = 0) and lower values farther from it. Each point on the path is recalculated as a weighted average of its neighbors. The weights are determined by the Gaussian function, where points closer to the center contribute more, and points farther away contribute less. The width of the bell curve is controlled by σ (the standard deviation), which affects how much smoothing is applied. In particular, σ determines the spread of the Gaussian function, and it refines the trajectory by convolving this function across a set of points along the route. The length of the smoothing window is calculated as: window length = 2 × [3σ] + 1. The overall smoothness of the trajectory is evaluated using the cumulative curvature, which reflects how sharply or frequently the path direction changes.

Where Larger σ means more smoothing, but this may over-simplify the curves. On the other hand, smaller σ means less smoothing, preserving sharp turns.

$$Path\ Smoothness\ = \sum_{i=1}^{N-2}\left|arctan\left(\frac{P_{i+2}-P_{i+1}}{P_{i+1}-P_i}\right)\right| \quad (6)$$

Where P_i is a point (position) along the path, $P_{i+1} - P_i$ is the vector representing a path segment between two consecutive points, $(P_{i+2} - P_{i+1})/(P_{i+1} - P_i)$ is the ratio of two consecutive path segments (and this is not the algebraic division but the change in direction), the $\arctan(.)$ returns the angle between the two segments (the change in direction), and $\sum_{i=1}^{N-2}()$ is the sum over all path bends from start to end. The total sum smooths out abrupt direction and speed changes, leading to more fluid robotic movement, better energy usage, and less mechanical stress on actuators. The D* Lite generates a series of distinct positions, which can be further refined to ensure seamless and collision-free motion across all path segments. In this regard, tracking and tuning these metrics offer a solid foundation for evaluating and improving the consistency and effectiveness of the path-planning algorithm [7], [79]. When the robot resumes motion after a pause, it might follow a shorter route once the obstacle has moved. To enhance collision detection, each robotic arm link is divided into smaller parts, and the system checks whether any of these parts fall within a defined safety margin that triggers pausing to avoid collisions [79], [80].

IV. RESULTS AND DISCUSSION

The simulation will concentrate on path generation for a point mass and a two-joint robotic manipulator within a time-varying environment. It evaluates the performance of both methods, specifically, the standalone RRT* and the combined RRT*-DL algorithm. Two simulation scenarios are considered: one involving the point mass and another involving the two-link robotic arm.

A. Case Study 1. Mass Point

This case study evaluates the RRT* algorithm for path planning of a mass point in a 2D environment with a fixed maze and dynamic obstacles. The workspace is the entire rectangular area, which includes both the obstacles and the available regions for motion. Within this workspace, the free space is the region where the mass point can move safely without collisions. The blue rectangular blocks represent the

static maze obstacles**,** which are fixed and permanently reduce the free space. The red vertical bars indicate the dynamic obstacles, which change position over time and create temporary restrictions that the mass point must avoid during motion, as shown in Fig. 4. The RRT* incrementally expands a tree from the start configuration to generate a feasible and collision-free trajectory. Particularly, the method provides an initial solution but often requires higher computation and a larger number of iterations. The simulation results highlight the path length, the computational time, and the limitations of the RRT* in handling dynamic environments, as shown in Fig. 5. When we use the hybrid RRT*–D* Lite algorithm, the path planning of the mass point is done under the same maze and obstacle settings, as shown in Fig. 6.

The hybrid approach combines the global exploration of the RRT* with the fast local re-planning ability of the D* Lite. This integration enables adaptive trajectory correction when obstacles interfere with the initial path. The hybrid method proves to be an amazing approach, achieving significant reductions in computation time, path length, and number of iterations compared to the standalone RRT*, as illustrated in Fig. 5.

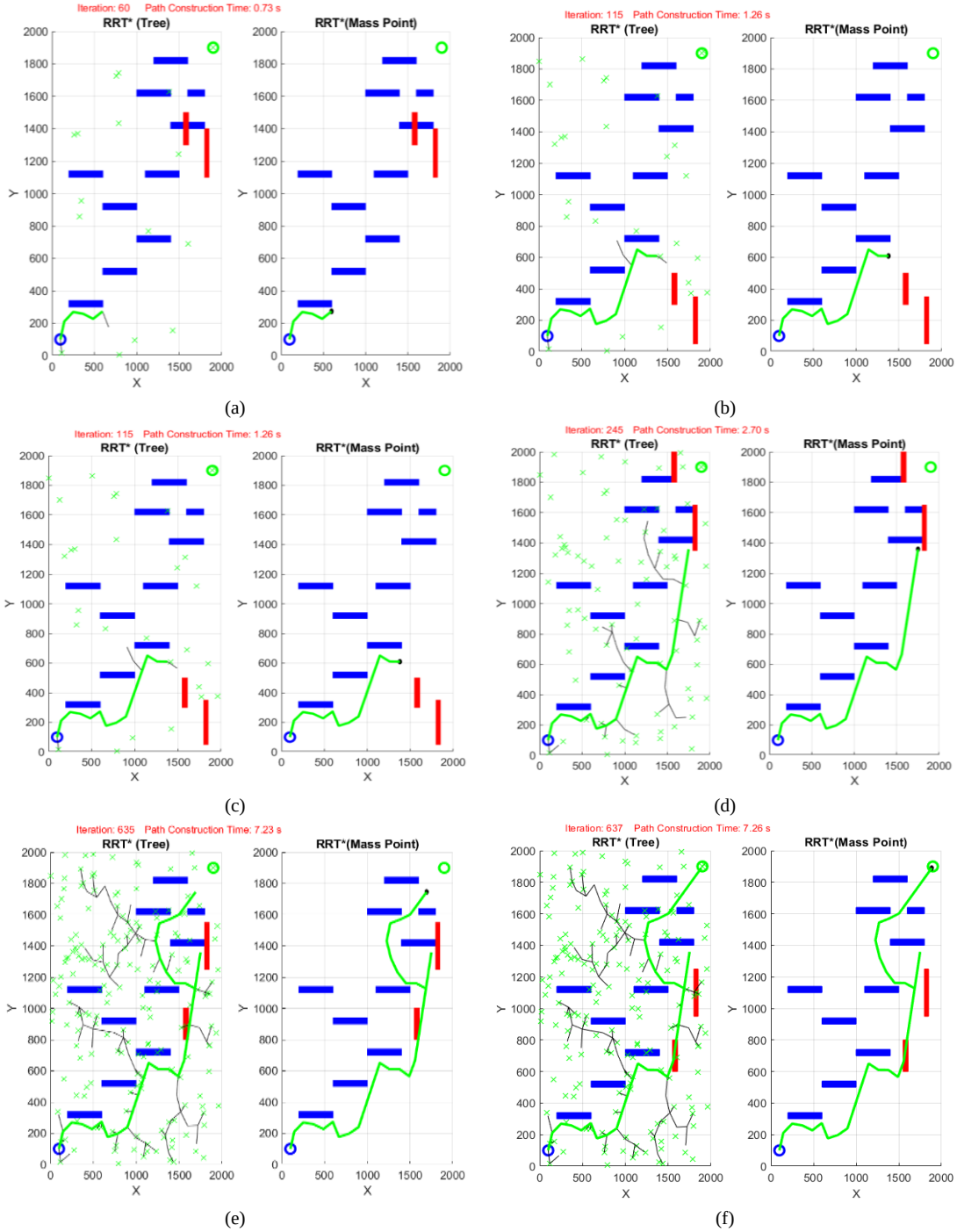

Fig. 4. Mass point path planning using the RRT: (a) initial tree expansion in the static maze with dynamic obstacles (b), (c), (d), and (e) intermediate path growth toward the goal with the static maze and dynamic obstacle avoidance, (f) complete collision-free path from start to goal

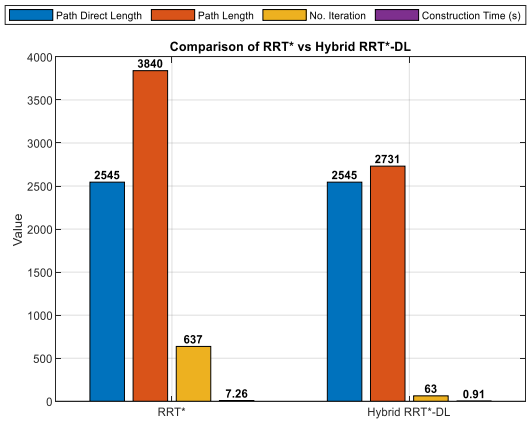

Fig. 5. The RRT* versus the hybrid approach for the mass point (the path length, no. of iterations, and path construction time)

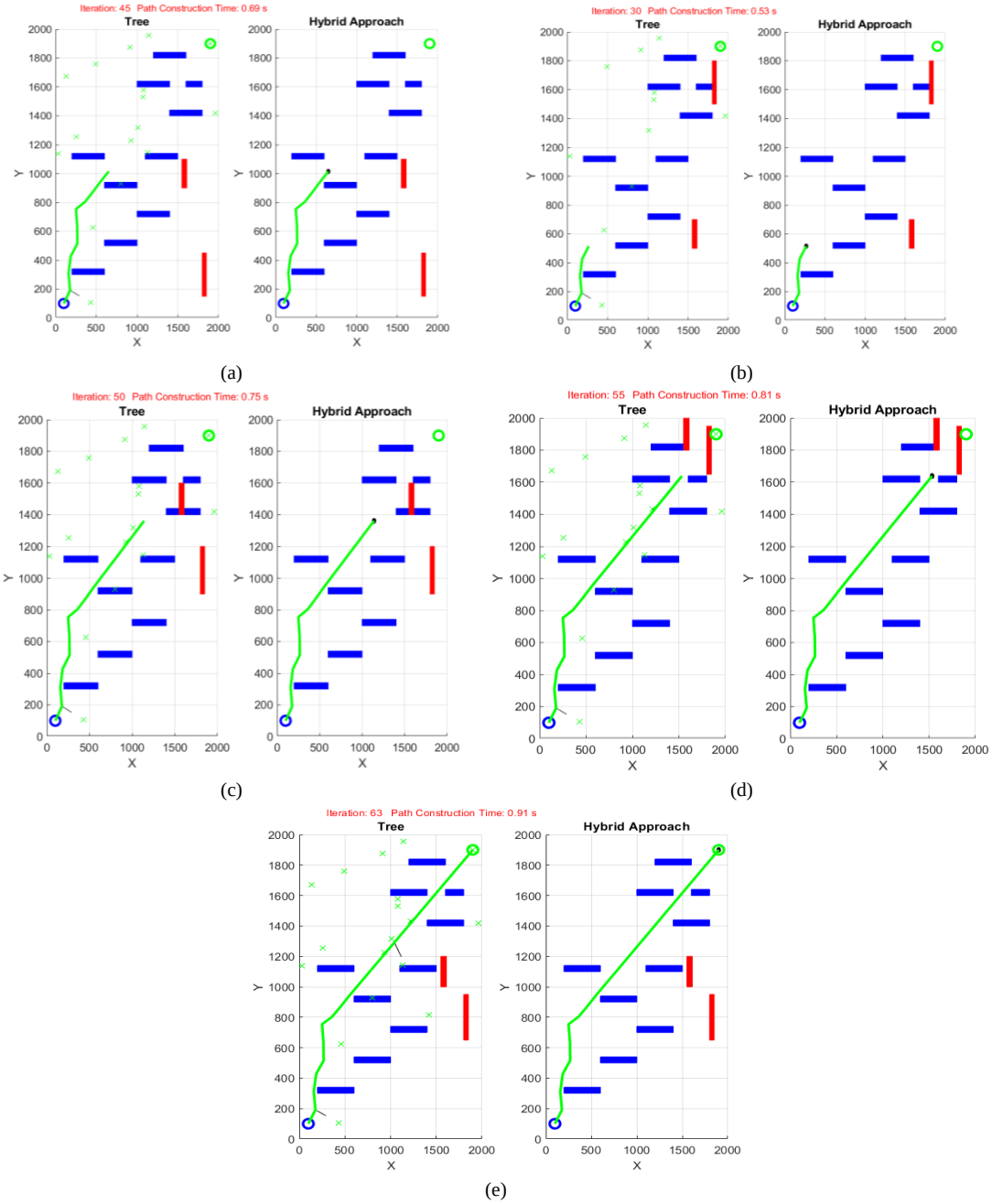

Fig. 6. Mass point path planning using the hybrid approach: (a) initial tree expansion in the static maze with dynamic obstacles (b), (c), and (d) intermediate path growth toward the goal with the static maze and dynamic obstacle avoidance and (e) complete collision-free path from start to goal

B. Case Study: The Two-Link Robotic Arm

This case study investigates the performance of the RRT* algorithm for path planning of a two-link robotic arm in a constrained environment. The workspace contains a fixed maze together with three ball-shaped dynamic obstacles, as shown in Fig. 7. The left panel of Fig. 7 illustrates the *RRT** within the robot's workspace.

The green region represents the *free space*, which indicates all the feasible positions where the robotic arm can operate without collision. In contrast, the blue rectangular shapes correspond to the static maze, which imposes permanent obstacles in the environment and defines the restricted zones. Additionally, the red circular regions depict the dynamic obstacles, which move periodically and create time-varying constraints in the workspace. The RRT* tree (blue crosses) explores the free space and gradually converges to construct a collision-free path (the red curve) from the starting point to the goal. The right panel shows the **t**wo-link robotic arm motion following the planned path. In addition, the black lines represent the two links of the manipulator, while the red trajectory corresponds to the end-effector's movement toward the goal (the green circle). During the motion, the arm avoids both the static maze and the dynamic obstacles, ensuring safe operation within the free space. This visualization highlights the complexity of planning in dynamic environments and the ability of the RRT* to find a feasible solution within constrained workspaces. The RRT* algorithm incrementally expands a tree in the configuration space to generate feasible joint trajectories while avoiding collisions.

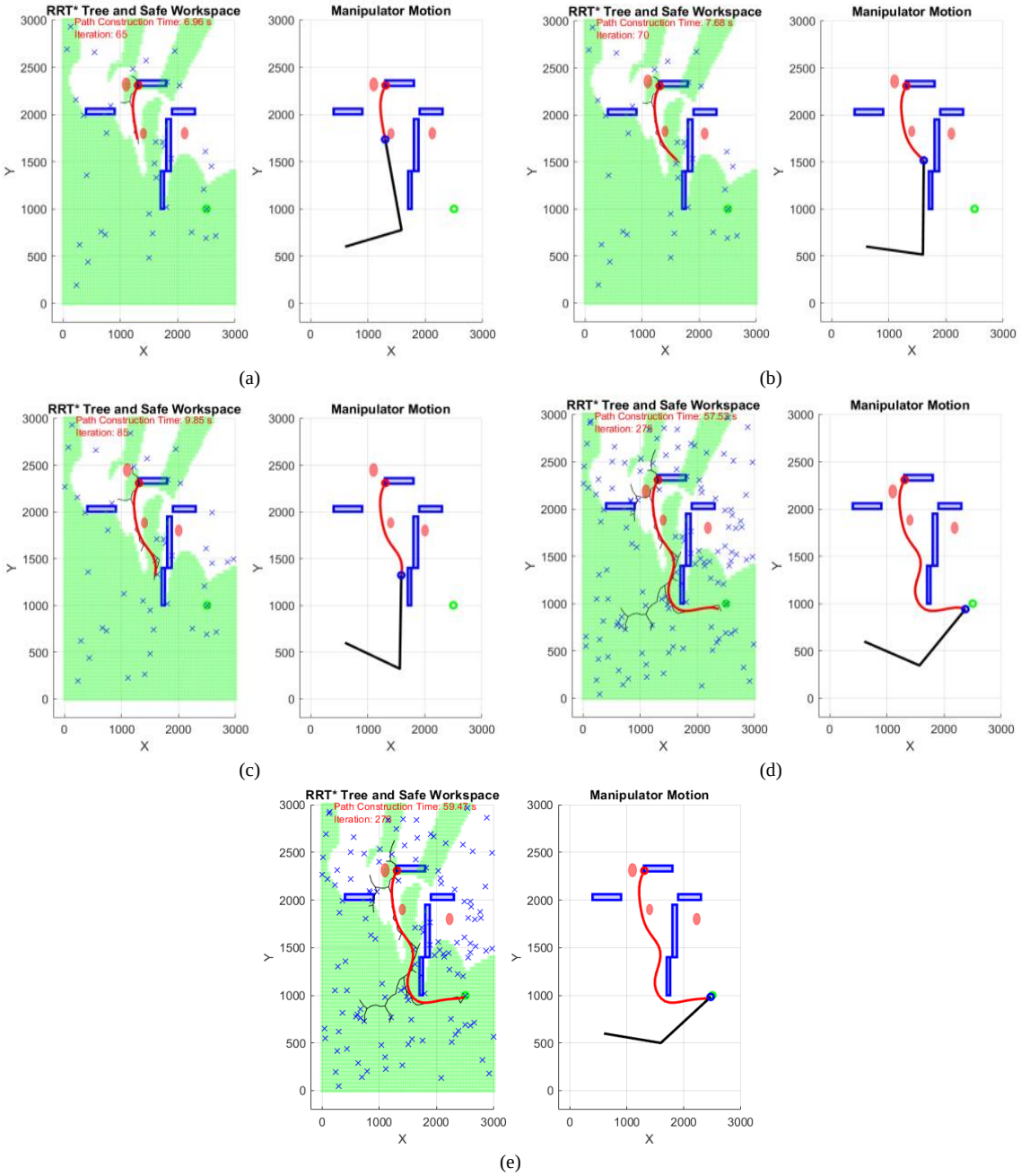

Fig. 7. Two-link robotic arm path planning using the RRT* (a) tree expansion and safe workspace in the static maze with dynamic obstacles (b), (c), and (d) manipulator motion following the planned path while avoiding obstacles and (e) complete collision-free path from start to goal

Although it provides a collision-free path, the method requires high computational effort and many iterations for convergence. The simulation results analyze the constructed path and the total computation time and highlight the limitations of the RRT* when applied to articulated robotic systems (see Fig. 8) when we use the hybrid RRT*–D* Lite algorithm for path planning of the two-link robotic arm in a dynamic workspace. The environment contains the same fixed maze and dynamic obstacles, as shown in Fig. 9. The RRT* component first constructs a global near-optimal path in the configuration space of the arm. When dynamic obstacles interfere, the D* Lite module performs local re-planning to adapt the arm's motion in real time. In this regard, the hybrid method proves to be highly effective, significantly reducing computation time, path length, and iteration count compared to the standalone RRT*, as illustrated in Fig. 8.

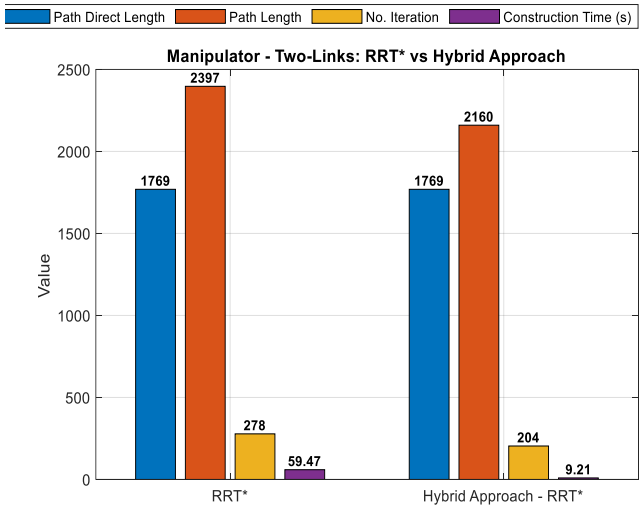

Fig. 8. The RRT* versus the hybrid approach for the mass point (the path length, no. of iterations, and path construction time)

For the mass point scenario with a fixed maze and two rectangular dynamic obstacles, the hybrid RRT*–D* Lite approach demonstrates remarkable improvements over the standalone RRT*. More precisely, the results show a 29% reduction in path length**,** indicating that the hybrid method generates a more direct and efficient trajectory. In terms of computational demand, the number of iterations is reduced by 90%, reflecting the effectiveness of local re-planning in avoiding unnecessary tree expansion. Moreover, the path construction time is reduced by 87%, highlighting the ability of the hybrid method to achieve faster convergence (see Fig. 10). These improvements clearly demonstrate that the hybrid approach is an outstanding solution for real-time path planning of point-mass systems.

For the two-link robotic arm case with a fixed maze and three ball-shaped dynamic obstacles, similar performance benefits are observed. In particular, the hybrid RRT*–D* Lite achieves a 10% reduction in path length, demonstrating a moderate yet meaningful improvement in trajectory efficiency for articulated systems. The number of iterations decreases by 27%**,** which reduces computational overhead during planning in the robot's configuration space. Most notably, the construction time is reduced by 85%**,** confirming the hybrid approach's ability to significantly accelerate planning even in higher-dimensional state spaces (see Fig. 10). This outcome validates the hybrid method as a powerful framework for robotic manipulators, where rapid re-planning is crucial to avoid collisions and ensure smooth motion.

V. CONCLUSION

This study introduces a novel integration of RRT* and D* Lite tailored for robotic arms in static and dynamic environments, achieving significant improvements in runtime, path length, and search efficiency. In addition, the use of a re-plan threshold enhances adaptability to obstacle changes while maintaining efficiency. The results demonstrated that the hybrid approach consistently outperforms the standalone RRT*, achieving significant reductions in the path construction time, the path length, and the iteration count. For the mass point, the hybrid method achieved up to 29% reduction in path length, 90% reduction in iterations, and 87% reduction in time, while for the two-link robotic arm, it achieved 10% reduction in path length, 27% reduction in iterations, and 85% reduction in time. These improvements highlight the hybrid approach as a highly effective and adaptive strategy for real-time navigation in dynamic and constrained workspaces. Future research will focus on feasible extensions supported by the current results, thereby enhancing both the rigor and practical relevance of this approach

Future research can extend this work in several directions. First, the integration of uncertainty modeling and sensor-based perception could enhance robustness in real-world environments where obstacle motion is unpredictable. Second, applying advanced path smoothing and optimization techniques may further improve trajectory quality, especially for articulated manipulators. Finally, extending the hybrid framework to multi-robot systems would allow coordination and collision avoidance among multiple agents.

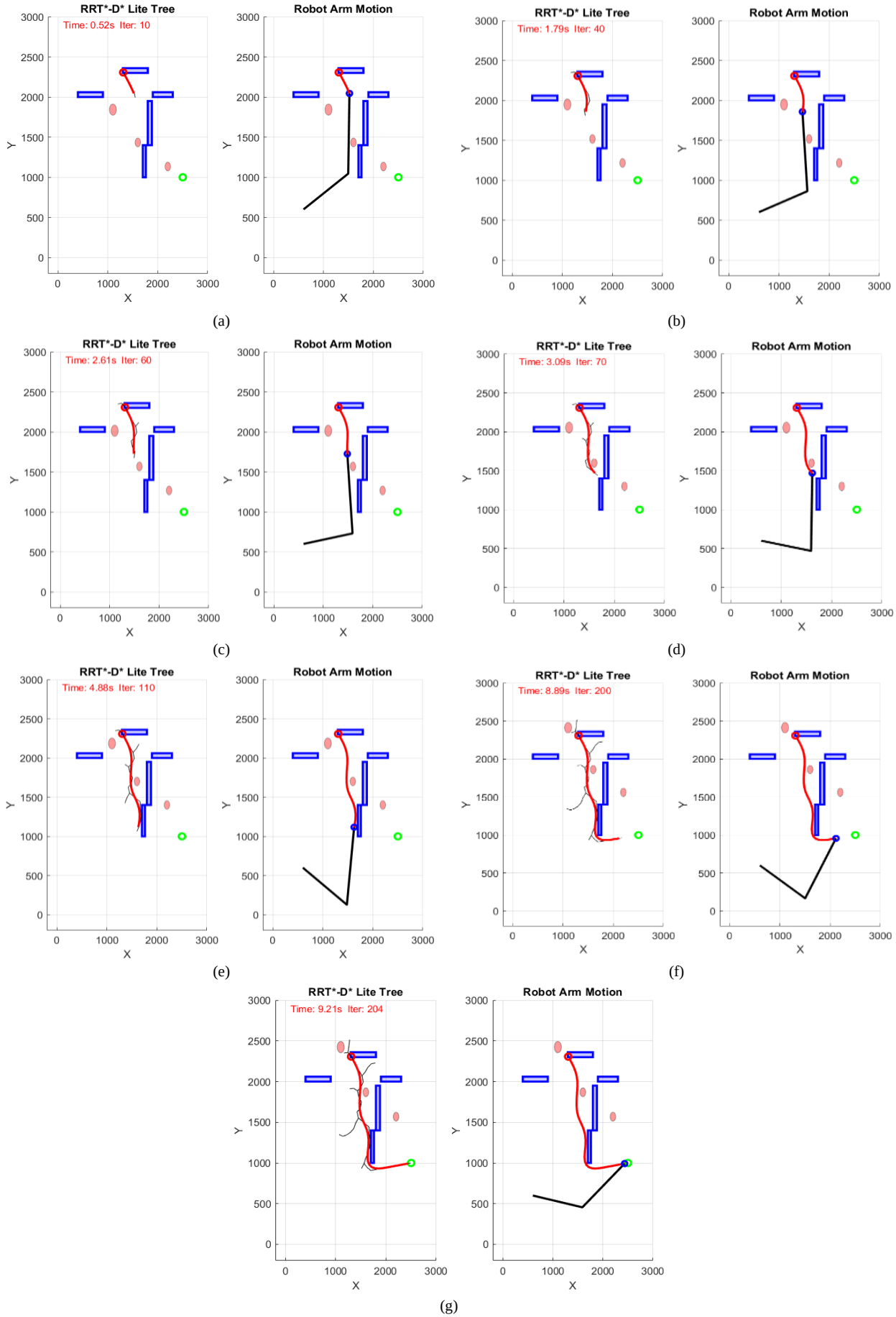

Fig. 9. Two-link robotic arm path planning using the hybrid approach (a) tree expansion and safe workspace in the static maze with dynamic obstacles (b), (c) (d), (e), and (f) manipulator motion following the planned path while avoiding obstacles (g) complete collision-free path from start to goal

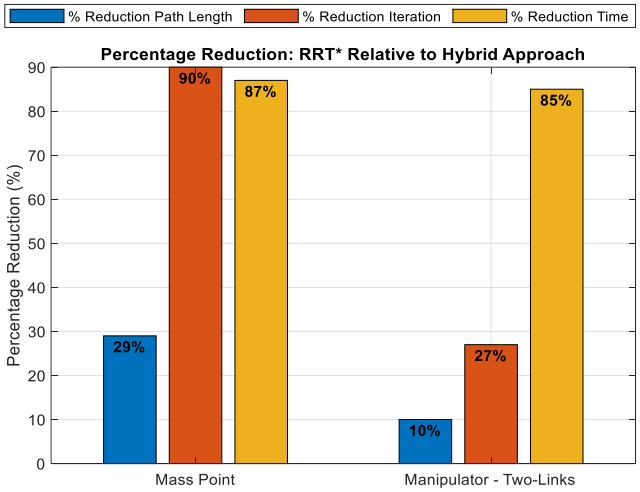

Fig. 10. Percentage reduction for the RRT* relative to the hybrid approach

REFERENCES

[1] M. Zhu, M. Kong, Y. Wen, S. Gu, B. Xue, and T. Huang, "A multi-objective path planning method for ships based on constrained policy optimization," *Ocean Engineering*, vol. 319, p. 120165, 2025, doi: 10.1016/j.oceaneng.2024.120165.

[2] F. A. Raheem and M. I. Abdulkareem, "Development of Path Planning Algorithm Using Probabilistic Roadmap Based on Modified Ant Colony Optimization," *World Journal of Engineering and Technology*, vol. 7, no. 4, pp. 583–597, 2019, doi: 10.4236/wjet.2019.74042.

[3] B. Li and B. Chen, "An Adaptive Rapidly-Exploring Random Tree," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 2, pp. 283–294, 2022, doi: 10.1109/JAS.2021.1004252.

[4] A. Hentout, A. Maoudj, and M. Aouache, "A review of the literature on fuzzy-logic approaches for collision-free path planning of manipulator robots," *Artificial Intelligence Review*, vol. 56, no. 4, pp. 3369–3444, 2023, doi: 10.1007/s10462-022-10257-7.

[5] C. Miao, G. Chen, C. Yan, and Y. Wu, "Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm," *Computers & Industrial Engineering*, vol. 156, p. 107230, 2021.

[6] W. Burzyński and W. Stecz, "Trajectory planning with multiplatform spacetime RRT*," *Applied Intelligence*, vol. 54, no. 19, pp. 9524–9541, 2024, doi: 10.1007/s10489-024-05650-4.

[7] S. Davarzani and M. T. Ejaz, "A 2D path-planning performance comparison of RRT and RRT* for unmanned ground vehicle," *IAES International Journal of Robotics and Automation*, vol. 13, no. 1, pp. 105–112, 2024, doi: 10.11591/ijra.v13i1.pp105-112.

[8] L. Wang, X. Yang, Z. Chen, and B. Wang, "Application of the Improved Rapidly Exploring Random Tree Algorithm to an Insect-like Mobile Robot in a Narrow Environment," *Biomimetics*, vol. 8, no. 4, p. 374, 2023, doi: 10.3390/biomimetics8040374.

[9] B. Xing and B. Li, *Motion Control and Path Planning of Marine Vehicles*. MDPI-Multidisciplinary Digital Publishing Institute, 2024, doi: 10.3390/books978-3-0365-9781-2.

[10] P. Wang, M. Ghergherehchi, J. Kim, M. Zhang, and J. Song, "Transformer-based path planning for single-arm and dual-arm robots in dynamic environments," *International Journal of Advanced Manufacturing Technology*, pp. 1–19, 2025, doi: 10.1007/s00170-025-16144-z.

[11] L. Zhu, P. Duan, L. Meng, and X. Yang, "GAO-RRT*: A path planning algorithm for mobile robot with low path cost and fast convergence," *AIMS Mathematics*, vol. 9, no. 5, pp. 12011–12042, 2024, doi: 10.3934/math.2024587.

[12] Z. Zhou, L. Wang, Y. Xue, X. Ao, L. Liu, and Y. Yang, "RETRACTED CHAPTER: Path Planning of Mobile Robot Based on Improved A* Algorithm," *Lecture Notes in Electrical Engineering*, vol. 1090 LNEE, no. 7, pp. 617–626, 2023, doi: 10.1007/978-981-99-6882-4_50.

[13] X. Zhu, B. Yan, and Y. Yue, "Path planning and collision avoidance in unknown environments for usvs based on an improved D* Lite," *Applied Sciences (Switzerland)*, vol. 11, no. 17, p. 7863, 2021, doi: 10.3390/app11177863.

[14] J. Liang, W. Luo, and Y. Qin, "Path Planning ofMulti-Axis Robotic Arm Based on Improved RRT?," *Computers, Materials and Continua*, vol. 81, no. 1, pp. 1009–1027, 2024, doi: 10.32604/cmc.2024.055883.

[15] X. Liu, L. Kang, Z. Shan, C. Su, and Y. Liu, "Path planning of robotic arm based on RTSR-RRT* algorithm," *Yi Qi Yi Biao Xue Bao/Chinese Journal of Scientific Instrument*, vol. 46, no. 3, pp. 65–73, 2025, doi: 10.19650/j.cnki.cjsi.J2513744.

[16] M. T. Hameed, F. A. Raheem, and A. R. Nasser, "Enhanced RRT* with APF and Halton Sequence for Robot Path Planning," *Journal of Robotics and Control (JRC)*, vol. 6, no. 2, pp. 493–513, 2025, doi: 10.18196/jrc.v6i2.24921.

[17] M. T. Hameed, F. A. Raheem, and A. R. Nasser, "Hybrid Path Planning for Robotics: APF-RRT Enhancement with Sobol Sequence Integration," *International Journal of Intelligent Engineering and Systems*, vol. 18, no. 3, pp. 556–568, 2025, doi: 10.22266/ijies2025.0430.38.

[18] P. Yang, H. Li, S. Zhu, S. Tan, Y. Lv, and L. Cao, "MMD-RRT: a path planning strategy for robotic arm with improved RRT algorithm in unstructured environments," *Measurement Science and Technology*, vol. 36, no. 7, 2025, doi: 10.1088/1361-6501/ade557.

[19] X. Li, Y. Wang, and L. Yu, "Seven-Degrees-of-Freedom Robotic Arm Path Planning Based on Improved RRT," *Lecture Notes in Electrical Engineering*, vol. 1284 LNEE, pp. 229–238, 2024, doi: 10.1007/978-981-97-8654-1_24.

[20] Y. Gu *et al.*, "LRRT*: A robotic arm path planning algorithm based on an improved Levy flight strategy with effective region sampling RRT," *Plos One*, vol. 20, p. e0324253, 2025, doi: 10.1371/journal.pone.0324253.

[21] S. Liu and M. Zhang, "A Novel Path Planning Scheme Based on Improved Bi-RRT* Algorithm," *Proceedings of International Conference on Artificial Life and Robotics*, vol. 55, no. 11, pp. 79–82, 2025, doi: 10.5954/icarob.2025.os3-1.

[22] K. Balint, A. B. Gergo, and B. Tamas, "Deep Reinforcement Learning combined with RRT for trajectory tracking of autonomous vehicles," *Transportation Research Procedia*, vol. 78, pp. 246–253, 2024, doi: 10.1016/j.trpro.2024.02.032.

[23] A. Tahmasbi, M. Faghfoorian, S. Khodaygan, and A. Bera, "Zonal RL-RRT: Integrated RL-RRT Path Planning with Collision Probability and Zone Connectivity," *arXiv preprint arXiv:2410.24205*, 2024.

[24] N. Chao, Y. kuo Liu, H. Xia, M. jun Peng, and A. Ayodeji, "DL-RRT* algorithm for least dose path Re-planning in dynamic radioactive environments," *Nuclear Engineering and Technology*, vol. 51, no. 3, pp. 825–836, 2019, doi: 10.1016/j.net.2018.11.018.

[25] T. Cadete, V. H. Pinto, J. Lima, G. Gonçalves, and P. Costa, "Dynamic AMR Navigation: Simulation with Trajectory Prediction of Moving Obstacles," *2024 7th Iberian Robotics Conference (ROBOT)*, pp. 1-7, 2024, doi: 10.1109/ROBOT61475.2024.10797420.

[26] F. Rastgar, H. Masnavi, B. Sharma, A. Aabloo, J. Swevers, and A. K. Singh, "PRIEST: Projection Guided Sampling-Based Optimization for Autonomous Navigation," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2630–2637, 2024, doi: 10.1109/LRA.2024.3357311.

[27] H. Fei *et al.*, "Three-Dimensional Path Planning for Unmanned Aerial Vehicles Based on Hybrid Multi-Strategy Dung Beetle Optimization Algorithm," *Agriculture (Switzerland)*, vol. 15, no. 11, p. 1156, 2025, doi: 10.3390/agriculture15111156.

[28] F. A. Raheem and U. I. Hameed, "Interactive Heuristic D* Path Planning Solution Based on PSO for Two-Link Robotic Arm in Dynamic Environment," *World Journal of Engineering and Technology*, vol. 7, no. 1, pp. 80–99, 2019, doi: 10.4236/wjet.2019.71005.

[29] S. Azubairi, A. Petunin, H. L. Alwan, M. M. Msallam, and A. Humaidi, "Dynamic Processing 2D Maps Method for Robot's Trajectory Planning," *Proceedings of Engineering and Technology Innovation*, vol. 30, pp. 79–89, 2025, doi: 10.46604/peti.2024.14508.

[30] L. X. Zhang, X. J. Meng, Z. J. Ding, and T. S. Wang, "Two Stage Path Planning Method for Co-Worked Double Industrial Robots," *IEEE Access*, vol. 11, pp. 126995–127010, 2023, doi: 10.1109/ACCESS.2023.3332310.

[31] M. L. Muhammed, E. H. Flaieh, and A. J. Humaidi, "Embedded System Design of Path Planning for Planar Manipulator Based on

Chaos a* Algorithm With Known-Obstacle Environment," *Journal of Engineering Science and Technology*, vol. 17, no. 6, pp. 4047–4064, 2022.

[32] F. A. Raheem and H. Z. Khaleel, "Static Stability Analysis of Hexagonal Hexapod Robot for the Periodic Gaits," *IJCCCE*, vol. 14, no. 3, p. 10, 2016.

[33] M. L. Muhammed, A. J. Humaidi, and E. H. Flaieh, "a Comparison Study and Real-Time Implementation of Path Planning of Two Arm Planar Manipulator Based on Graph Search Algorithms in Obstacle Environment," *ICIC Express Letters*, vol. 17, no. 1, pp. 61–72, 2023, doi: 10.24507/icicel.17.01.61.

[34] C. K. Mun. *Controller design of a robotic orthosis using sinusoidal-input describing function model*. University of Nottingham (United Kingdom), 2020.

[35] M. Shahriari-Kahkeshi and M. Mahdavi, "Fuzzy Control of an Uncertain Robot Manipulator with Input Constraint," in *Proceedings - 2019 6th International Conference on Control, Instrumentation and Automation, ICCIA 2019*, pp. 1–6, 2019, doi: 10.1109/ICCIA49288.2019.9030882.

[36] Y. Li, Y. Yang, K. Liu, and C. Y. Wen, "Intelligent Joint Space Path Planning: Enhancing Motion Feasibility with Goal-Driven and Potential Field Strategies," *Sensors*, vol. 25, no. 14, p. 4370, 2025, doi: 10.3390/s25144370.

[37] Y. Li, W. Wei, Y. Gao, D. Wang, and Z. Fan, "PQ-RRT*: An improved path planning algorithm for mobile robots," *Expert Systems with Applications*, vol. 152, p. 113425, 2020, doi: 10.1016/j.eswa.2020.113425.

[38] J. Chen, Y. Zhao, and X. Xu, "Improved RRT-Connect Based Path Planning Algorithm for Mobile Robots," *IEEE Access*, vol. 9, pp. 145988–145999, 2021, doi: 10.1109/ACCESS.2021.3123622.

[39] A. Huang *et al.*, "A motion planning method for winter jujube harvesting robotic arm based on optimized Informed-RRT* algorithm," *Smart Agricultural Technology*, vol. 10, p. 100732, 2025, doi: 10.1016/j.atech.2024.100732.

[40] C. Lei, J. Li, Y. Deng, and X. Tan, "RRT*ASV: Improved RRT* path planning method for Ackermann steering vehicles," *Expert Systems with Applications*, vol. 279, p. 127349, 2025, doi: 10.1016/j.eswa.2025.127349.

[41] Q. Gao, Q. Yuan, Y. Sun, and L. Xu, "Path planning algorithm of robot arm based on improved RRT* and BP neural network algorithm," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, p. 101650, 2023, doi: 10.1016/j.jksuci.2023.101650.

[42] J. Qi, H. Yang, and H. Sun, "MOD-RRT*: A Sampling-Based Algorithm for Robot Path Planning in Dynamic Environment," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 8, pp. 7244–7251, 2021, doi: 10.1109/TIE.2020.2998740.

[43] X. Du, P. Luo, and X. Lv, "Path Planning Algorithm Based on Improved RRT and Artificial Potential Field," *2024 8th International Conference on Electrical, Mechanical and Computer Engineering, ICEMCE 2024*, vol. 24, no. 12, pp. 1767–1774, 2024, doi: 10.1109/ICEMCE64157.2024.10861913.

[44] W. Zhu and Z. Chen, "Research on Path Planning for Mobile Charging Robots Based on Improved A* and DWA Algorithms," *Electronics (Switzerland)*, vol. 14, no. 12, 2025, doi: 10.3390/electronics14122318.

[45] Y. Jung and B. S. Oh, "DC-RRT*: Dubins-guided Curvature RRT* for 3D Path Planning of Unmanned Aerial Vehicles," *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3600283.

[46] B. H. Meng, I. S. Godage, and I. Kanj, "RRT*-Based Path Planning for Continuum Arms," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6830–6837, 2022, doi: 10.1109/LRA.2022.3174257.

[47] S. J. Fusic and R. Sitharthan, "Improved RRT∗Algorithm-Based Path Planning for Unmanned Aerial Vehicle in a 3D Metropolitan Environment," *Unmanned Systems*, vol. 12, no. 5, pp. 859–875, 2024, doi: 10.1142/S2301385024500225.

[48] H. Fan, J. Huang, X. Huang, H. Zhu, and H. Su, "BI-RRT*: An improved path planning algorithm for secure and trustworthy mobile robots systems," *Heliyon*, vol. 10, no. 5, 2024, doi: 10.1016/j.heliyon.2024.e26403.

[49] Z. Ou, A. Xiong, J. Chen, and J. Wei, "Path Planning of Mobile Robot Based on the Improved A∗ Algorithm," *2024 4th International Conference on Electronic Information Engineering and Computer Technology, EIECT 2024*, vol. 14, no. 1, pp. 99–104, 2024, doi: 10.1109/EIECT64462.2024.10866740.

[50] J. Wang, T. Li, B. Li, and M. Q. H. Meng, "GMR-RRT∗: Sampling-Based Path Planning Using Gaussian Mixture Regression," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 690–700, 2022, doi: 10.1109/TIV.2022.3150748.

[51] W. Wang, H. Gao, Q. Yi, K. Zheng, and T. Gu, "An Improved RRT∗ Path Planning Algorithm for Service Robot," in *Proceedings of 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2020*, vol. 1, pp. 1824–1828, 2020, doi: 10.1109/ITNEC48623.2020.9085226.

[52] D. Wang, S. Zheng, Y. Ren, and D. Du, "Path planning based on the improved RRT∗ algorithm for themining truck," *Computers, Materials and Continua*, vol. 71, no. 2, pp. 3571–3587, 2022, doi: 10.32604/cmc.2022.022183.

[53] S. Ganesan, M. Thangamuthu, B. Ramalingam, M. M. Rayguru, and S. N. Tamilselvan, "Accelerating RRT* convergence with novel nonuniform and uniform sampling approach," *Scientific Reports*, vol. 15, no. 1, p. 28342, 2025, doi: 10.1038/s41598-025-09992-y.

[54] Z. Tian, "RRT-connect path planning algorithm based on gravitational field guidance," in *2025 IEEE 5th International Conference on Electronic Technology, Communication and Information (ICETCI)*, pp. 1480–1485, 2025, doi: 10.1109/icetci64844.2025.11084244.

[55] O. Gervasi, B. Murgante, E. Pardede, and B. O. Apduhan. *Computational Science and Its Applications-ICCSA 2021*. Springer International Publishing, 2021.

[56] J. Yan, Z. Huang, and Z. An, "Mobile Robot Path planning based on RRT-Reward Algorithm," in *ACM International Conference Proceeding Series*, pp. 420–425, 2024, doi: 10.1145/3679409.3679487.

[57] M. J. Rice, "Autonomous Underwater Vehicle Planning Using Hybrid D* Lite with PPO and TD3: Experimental Design and Performance Analysis," *Undergraduate Theses*, 2025.

[58] S. Shi, Y. Zuo, and T. Li, "Path Planning of Unmanned Surface Vessel Based on Improved RRT," *IECON Proceedings (Industrial Electronics Conference)*, vol. 266, p. 112873, 2023, doi: 10.1109/IECON51785.2023.10312195.

[59] J. Jin, Y. Zhang, Z. Zhou, M. Jin, X. Yang, and F. Hu, "Conflict-based search with D* lite algorithm for robot path planning in unknown dynamic environments," *Computers and Electrical Engineering*, vol. 105, p. 108473, 2023, doi: 10.1016/j.compeleceng.2022.108473.

[60] Y. Li, F. Yang, X. Zhang, D. Yu, and X. Yang, "Improved D* Lite Algorithm for Ship Route Planning," *Journal of Marine Science and Engineering*, vol. 12, no. 9, p. 1554, 2024, doi: 10.3390/jmse12091554.

[61] L. Yang *et al.*, "Path planning technique for mobile robots: A review," *Machines*, vol. 11, no. 10, p. 980, 2023.

[62] W. Zhang, "A Path Planning Algorithm for Unmanned Overhead Cranes Integrating Improved A* and D* Lite with Kinematic Optimization," *Computer Life*, vol. 13, no. 2, pp. 15–19, 2025, doi: 10.54097/0ews8r57.

[63] W. Liangwen, L. Yanzi, F. Cheng, and S. Midori, "Application of Path Planning Algorithms in Dynamic Nursing Home Environments for Autonomous Patrol Robots," in *Proceedings of Asia Pacific Conference on Robot IoT System Development and Platform*, vol. 2024, pp. 1–6, 2024.

[64] Y. Li *et al.*, "AD-RRT*: An RRT*-based global path planning approach for underwater gliders with alpha shapes and DBSCAN," *Expert Systems with Applications*, vol. 291, p. 128219, 2025, doi: 10.1016/j.eswa.2025.128219.

[65] B. Y. Suprapto, S. Dwijayanti, D. Windisari, and G. A. Pratama, "Optimizing Route Planning for Autonomous Electric Vehicles Using the D-Star Lite Algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 1, pp. 1069–1078, 2025, doi: 10.14569/IJACSA.2025.01601103.

[66] W. Wattanapornprom, C. Wipachainun, T. Lertpinitamonkul, W. Suksai, Y. Rodkaew, and W. Susutti, "Procedural Content Generation for 2.5D Rogue-Lite Games: An Evolutionary Algorithm Approach," in *International Computer Science and Engineering Conference*, no. 2024, pp. 1–6, 2024, doi: 10.1109/ICSEC62781.2024.10770719.

[67] Z. Luo *et al.*, "A UAV Path Planning Algorithm Based on an Improved D∗ Lite Algorithm for Forest Firefighting," *Proceedings - 2020 Chinese Automation Congress, CAC 2020*, vol. 15, no. 15, pp. 4233–4237, 2020, doi: 10.1109/CAC51589.2020.9327111.

[68] Y. Zhang *et al.*, "Multi-Strategy Fusion RRT-Based Algorithm for Optimizing Path Planning in Continuous Cherry Picking," *Agriculture*

(Switzerland), vol. 15, no. 15, p. 1699, 2025, doi: 10.3390/agriculture15151699.

[69] Z. Yu and L. Xiang, "NPQ-RRT ∗: An Improved RRT ∗ Approach to Hybrid Path Planning," *Complexity*, vol. 2021, no. 1, p. 6633878, 2021, doi: 10.1155/2021/6633878.

[70] F. A. Raheem, And, and I. I. Gorial, "Comparative Study between Joint Space and Cartesian Space Path Planning for Two-Link Robot Manipulator using Fuzzy Logic," *RAQI J Comput Commun Control Syst Eng*, vol. 13, no. 2, pp. 1–10, 2013.

[71] S. Bouraine, Y. Bellalia, I. Chaabeni, and D. Naceur, "When robots learn from nature: GLWOA-RRT*, a nature-inspired motion planning approach," *Swarm and Evolutionary Computation*, vol. 98, p. 102062, 2025, doi: 10.1016/j.swevo.2025.102062.

[72] J. O. Adibeli, Y. K. Liu, N. Chao, and N. J. Awodi, "Modified bidirectional rapidly exploring random tree star (Bi-RRT*) algorithm with variable node parameter for optimized path planning in nuclear decommissioning environment," *Nuclear Engineering and Design*, vol. 433, p. 113876, 2025, doi: 10.1016/j.nucengdes.2025.113876.

[73] L. Petrovic, I. Markovic, and I. Petrovic, "Mixtures of Gaussian Processes for Robot Motion Planning Using Stochastic Trajectory Optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 12, pp. 7378–7390, 2022, doi: 10.1109/TSMC.2022.3155378.

[74] J. Meng, Y. Liu, R. Bucknall, W. Guo, and Z. Ji, "Anisotropic GPMP2: A Fast Continuous-Time Gaussian Processes Based Motion Planner for Unmanned Surface Vehicles in Environments With Ocean Currents," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3914–3931, 2022, doi: 10.1109/TASE.2021.3139163.

[75] H. Ali, G. Xiong, H. Wu, B. Hu, Z. Shen, and H. Bai, "Multi-robot Path Planning and Trajectory Smoothing," in *IEEE International Conference on Automation Science and Engineering*, vol. 2020, pp. 685–690, 2020, doi: 10.1109/CASE48305.2020.9216972.

[76] F. F. Li, Y. Du, and K. J. Jia, "Path planning and smoothing of mobile robot based on improved artificial fish swarm algorithm," *Scientific Reports*, vol. 12, no. 1, p. 659, 2022, doi: 10.1038/s41598-021-04506-y.

[77] T. Kim, G. Park, K. Kwak, J. Bae, and W. Lee, "Smooth Model Predictive Path Integral Control Without Smoothing," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10406–10413, 2022, doi: 10.1109/LRA.2022.3192800.

[78] L. Cosier *et al.*, "A Unifying Variational Framework for Gaussian Process Motion Planning," in *Proceedings of Machine Learning Research*, vol. 238, pp. 1315–1323, 2024.

[79] Z. Zhang *et al.*, "Investigation into the Efficient Cooperative Planning Approach for Dual-Arm Picking Sequences of Dwarf, High-Density Safflowers," *Sensors*, vol. 25, no. 14, p. 4459, 2025, doi: 10.3390/s25144459.

[80] J. Fan, X. Chen, Y. Wang, and X. Chen, "UAV trajectory planning in cluttered environments based on PF-RRT* algorithm with goal-biased strategy," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105182, 2022, doi: 10.1016/j.engappai.2022.105182.