A Multi-metric Benchmarking Study of Hybrid Path-planning Algorithms under Different Grid Resolutions

Rubel Ahmed

Graduate School of Science and Engineering, Iwate University, Morioka, Japan
Email: jxrubelahmed@gmail.com

***Abstract*—This study systematically evaluates the performance of four hybrid path-planning algorithm combinations formed in two global planners (A* and Dijkstra) with two local planners, the Dynamic Window Approach (DWA) and the Time Elastic Band (TEB). Four combinations of the hybrid planner were conducted by autonomous differential-drive robots navigating in simulated environments. The proposed study tested across three grid map resolutions (0.025 m, 0.10 m, and 0.15 m) to understand the impact of environmental granularity on various environmental settings with variation of the velocity range (0.1 m/s–1 m/s). A comprehensive quantitative evaluation framework was implemented to justify the performance of the planner incorporating some key metrics such as the Spatial Coefficient (SPC), Temporal Coefficient (TEC), Smoothness Coefficient (SMC), Success Rate (SR), Failure Rate (FR), path length, execution time and Goal Convergence (GC). The collected experimental data were evaluated through a normalization process with the execution of some statistical analyses, such as Analysis of Variance (ANOVA), T-test, and Bonferroni correction, which revealed differences between planner performance and the impact of grid resolutions. Results revealed that Dijkstra + TEB consistently outperformed the other combinations and demonstrated superior SPC, TEC, and SMC values, particularly in complex environments. Conversely, the DWA-based planners suffered from low success rates and parameter sensitivities. This work offers a systematic benchmarking framework for evaluating path-planning strategies and serves as a valuable guide for deploying reliable autonomous navigation in real-world scenarios.**

***Keywords*—autonomous navigation, global-local planner integration, grid resolution, A* + Time Elastic Band (TEB), Dijkstra + TEB**

I. Introduction

Autonomous Mobile Robots (AMRs) are becoming inevitable in modern industrial, logistics, and service domains due to their adaptive navigation and safe and efficient system, which is crucial in the complex warehouse automation, smart manufacturing, and transportation systems. On account of achieving the smooth trajectory and trustworthy path planning methods is a significant issue up to now [1]. Conventional global planners such as the A* [2] and Dijkstra algorithms [3] are well-known for their optimal search capabilities and have provided the framework for many years of research in robot navigation systems. Whereas, these methods mostly produce piecewise-linear paths with sudden turns, leading to suboptimal execution in real-world environments that require kinematic feasibility and dynamic adaptability. Present research has deeply focused on global and local path planning strategies to confront the issues. Two well-known local planners, such as the Dynamic Window Approach (DWA) [4] and Timed Elastic Band (TEB) [5], enable the robot to execute smooth motions and a shorter trajectory by avoiding the obstacles in the presence of a dynamic environment. Nevertheless, in dynamic situations with high map resolution and obstacles, it's challenging to generate a smooth trajectory by coupling global and local planners, in sight of the conflicting objectives, it can lead to degraded smoothness or unstable trajectories [6, 7].

The development of the modular navigation architecture by provenance of open-source frameworks such as the Robot Operating System (ROS) [8] and its extensions like Autoware [9] and Move_Base Flex. With this architecture, researchers ameliorate a hybrid integration of global-local planners combined with multiple planning modules, which enables reproducible testing and an established ecosystem [10]. ROS-based research platforms, such as iCab and TurtleBot variants, have demonstrated the robustness of hybrid systems in both simulated and real environments [9, 11]. Regardless, the performance relies on a few factors, such as task-dependent, influenced by grid resolution, obstacle density, velocity, and environment dynamics [12].

Recent studies highlight the multi-metric evaluation frameworks that move beyond single-criterion inspection, such as trajectory path length or robot execution time [13–15]. The investigation spread on various parameter evaluations, such as path smoothness, temporal consistency, and goal convergence, must also be considered to capture realistic navigation performance in the autonomous mobile robotics. In addition to that, few of the studies investigate the robot goal point localization by

Manuscript received December 12, 2026; revised January 29, 2026; accepted March 6, 2026; July 10, 2026.

doi: 10.18178/ijmerr.15.4.362-378

the combination of the A* and Dijkstra global planners with the DWA and TEB local planners [16, 17]. Also, some researches have modified the grid cell traverse methods and traditional algorithms to minimize the travel time of the robot [18, 19]. However, most of the research has focused on the fixed velocity and grid resolution size. Presented research has evaluated multi-metric criteria in the presence of static and dynamic environments with a constant velocity and grid size [20].

To proceed with this action, the researcher addressed a few hybrid algorithms, such as hybrid particle swarm optimization [21], improved Rapidly-exploring Random Tree (RRT) [22], multi-objective-based genetic algorithm [23], coupling genetic algorithm with ant colony [24], and the E*-based optimal search algorithm [25]. These approaches leverage heuristic optimization to ensure the trajectories are smooth and dynamically feasible in simulation and real-world environments. Furthermore, to improve the accuracy and computational efficiency, the authors presented grid-based hybrid map techniques in large-scale dynamic environments [26–28]. In addition to improving the navigation quality, established intelligent planning algorithms underscore the demand for benchmarking frameworks [29]. This study presents a comprehensive evaluation process to handle the deficiencies of the planner evaluation methods and their practical application. The main aim of this research is to thoroughly evaluate the effectiveness of the global-local planner combinations in the simulation environment. Mainly focus on four combinations: A* + DWA, A* + TEB, Dijkstra + DWA, and Dijkstra + TEB.

Present research makes a variant key contribution to the evaluation and touchstone hybrid path-planning algorithms for autonomous mobile robots.

- Observe the overall performance of the global and local planner trajectories by offering multi-resolution benchmarking with three different grid resolution sizes (0.025 m, 0.10 m, and 0.15 m). Notably, the research systematically quantifies the influence of map granularity on the planner's success rate and overall performance.
- The proposed study identifies the most robust and reliable planner combination by evaluating 3d multi-metric parameters analysis, namely Spatial Coefficient (SPC), Temporal Coefficient (TEC), Smoothness Coefficient (SMC) and quantitative evaluation of the path execution time, path length, and success rate of the planner.
- Also, investigate the effectiveness of the velocity on local planner combinations under various grid resolution sizes and robot trajectories. In addition, the research addresses three different environments to identify the optimal combination of the global-local planner.

Despite extensive studies comparing global and local planners, most existing works focus commonly on several metrics such as path length, computational time, or success rate. Although hybrid combinations such as A* + DWA, A* + TEB, A*, and Dijkstra-based integrations have been explored in various research articles. Most of the research work on the evaluation methodologies remains largely fragmented. Many studies lack a unified multi-metric-based analysis that simultaneously considers spatial optimality, temporal efficiency, smoothness, goal localization accuracy, and statistical validation under varying map grid size resolutions and various velocity range constraints. Furthermore, limited attention has been given to understanding how environmental discretization (grid size variation) systematically influences hybrid planner behavior across different environment complexities with variation of the velocity range. In the existing studies there is no comprehensive and statistically grounded benchmarking framework that evaluates hybrid global–local planner combinations across multiple grid resolutions, environmental structures, and multi-metric cluster evaluations. Overall, this study bridges the gap between theory and practice, providing a systematic way to evaluate the best global-local planner combination for robot navigation system. The combination analyzed by the multi-metric normalization methods with statistical evaluation sets a new standard for benchmarking intelligent robotic systems in diverse environments.

The rest of the article is organized as follows. Section II presents the theoretical background of the global and local planners used in this study. Section III recounts the proposed hybrid framework of the planner, the experimental setup, and the formation of the evaluation metrics. Section IV provides the experimental analysis of the results and presents a comparative analysis across different grid resolutions based on various environment setups. At the end of this section discusses the implications of the findings and highlights key performance trade-offs. Finally, Section V concludes the paper and outlines directions for future research.

II. Literature Review

In the AMR navigation system, the global path planning algorithms have provided a structural backbone for a long time. The A* algorithm [30], since its inception, has been recognized for its heuristic-guided optimality and completeness, whereas Dijkstra's approach provides guaranteed shortest paths through exhaustive exploration [13, 18]. Comparative analyses, the A* algorithm generally converges faster due to its heuristic method; on the other hand, Dijkstra remains advantageous in complex terrains where uniform cost expansion ensconces magnificent exploration [14, 15]. To handle the dynamic constraints, the research has extended to classical methods such as D* and E* [25, 31].

However, due to the discrete cell representation by the grid-based method, the global planners intrinsically suffer from path inefficiencies, often producing less smooth and more jagged trajectories [20, 32]. Concerning issues, the researchers have proposed several methods, including path-counting optimization [32], grid tie-breaking strategies [33], and hybrid map representations [26, 29]. Moreover, intelligent techniques combining A* with adaptive or learning-based heuristics have shown improved efficiency in both simulation and field

applications [6, 33, 34]. Conversely, local planning approaches, such as the DWA local planner, are one of the most widely used reactive planners for differential-drive robots, which efficiently navigate the environment while avoiding obstacles [4]. In addition to that, the TEB planner, with its special characteristics, on-time parameterized optimization framework, enhances trajectory elasticity, ensuring smoother and dynamically feasible paths even in cluttered spaces [5]. Integrations of both planners within ROS architectures have been evaluated across various mobile robot navigation platforms [8, 17]. Yet, empirical results indicate that local planners are highly sensitive to configuration parameters, especially under variable velocities or changing map granularities [6, 7, 16].

Further research has addressed hybrid global-local frameworks that have emerged to tackle the expletive strengths of these algorithms. At that place, to improve the trajectory smoothness, adaptability, and goal point localizations accuracy the global and local planner combining strategies such as Dijkstra with DWA [16], A* with TEB [35], or RRT-based methods with adaptive control [22, 27, 28]. Studies on hybrid genetic and ant colony optimization [24, 36], or particle swarm-based metaheuristics [21], highlight the growing role of hybridization in achieving optimal performance in dynamic environments. Besides, the hybrid map depictions integrating several navigation techniques such as LiDAR-based mapping [27], multi-layered grids [26], and semantic features [37] have improved path quality for autonomous ground, aerial, and marine vehicles [28, 38].

Despite these advances, most of the evaluations still rely on single metric evaluations such as path length or completion time [14, 18, 19]. Few studies visualize multi-dimensional trade-offs among spatial, temporal, and smoothness coefficient parameters [17, 39]. This gap underscores the necessity of multi-criteria benchmarking frameworks to capture real-world navigation behavior effectively. Researcher have emphasized the need to integrate statistical validation, normalization, and visual analysis techniques into algorithmic evaluations for broader generalization [1, 12, 23].

In this context, the present research addresses a unified experimental evaluation process in the presence of various environments with different grid resolutions and velocity ranges for robot movement. The quantitative research provides an evidence-based assessment of global-local hybrid planners within ROS. The resulting insights aim to establish standardized evaluation procedures that enhance the interpretability, reproducibility, and practical deployment of hybrid path-planning systems in autonomous mobile robotics.

Consequently, although A*/Dijkstra is well-established for global planning and DWA/TEB for local control, a more comprehensive benchmarking framework is still needed. Such frameworks should integrate SPC, TEC, and SMC elements while providing enhanced visual representations with different grid resolution sizes and velocity profiles of the robot manoeuvring system. To identify the proper planner combination, the research needs to consider empirical analysis of the data to realize the planner strength in navigation on real-world autonomous systems.

III. Resources and Approaches

A. Robot Model and Control System

This section describes an indoor ground robot denoted SMART AMR in the Gazebo simulation environment. The architecture of the proposed system is influenced by the Dobot AI-Starter kit [40]. The rear part of the robot is comprised of two DC gear motors, which mainly function to maneuver the robot, but the front section features a freewheel. It is designated as a differential wheeled robot. A lidar sensor, inspired by the Intel RealSense L515 Lidar camera, which is affixed on the robot [41]. The camera is affixed to the 26 cm tall robotic vehicle. The Lidar sensor can generate lidar rays up to a 70-degree field of view and can identify objects at up to 9 m. The robot's body construction is 20 cm in length and 18 cm in width.

Fig. 1 demonstrates the visual representation of the SMART AMR system, which is preliminarily designed in the SolidWorks platform. Afterward, through the URDF file, it is executed in the Gazebo simulation framework. The primary operating system is ROS [42]. The research was performed on a Core i7 PC equipped with 32 GB of RAM. The robot initially maps the entire surroundings environment, and the target point is established by the stop position of the robot while mapping. The robot does remote mapping of the area and identifies the target location by using slam mapping [43]. Subsequently, the constructed map is stored in the system. Subsequently, the ROS system produces a local occupancy grid map. The path planning algorithm operates to propel the robot ahead according to the spatial constraints of the surroundings. The path planning algorithms seek to expedite the robot's arrival at the target location. The path-planning nodes transmitted a message to the robot's motor controller by the combination of the global-local path planning strategies. In accordance with the directives of the path planning algorithms, the robot subscribes to the message and navigates to the target point by adhering to the designated route.

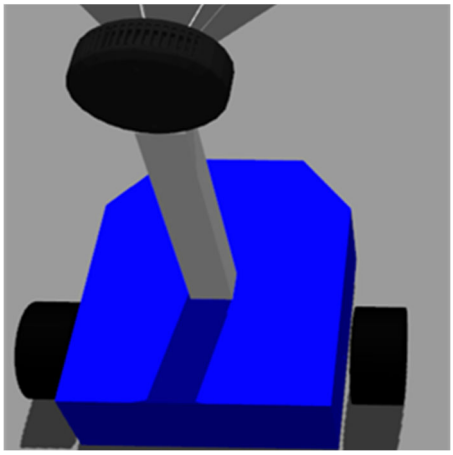


Fig. 1. Simulation model in gazebo environment.

B. Environment Overview

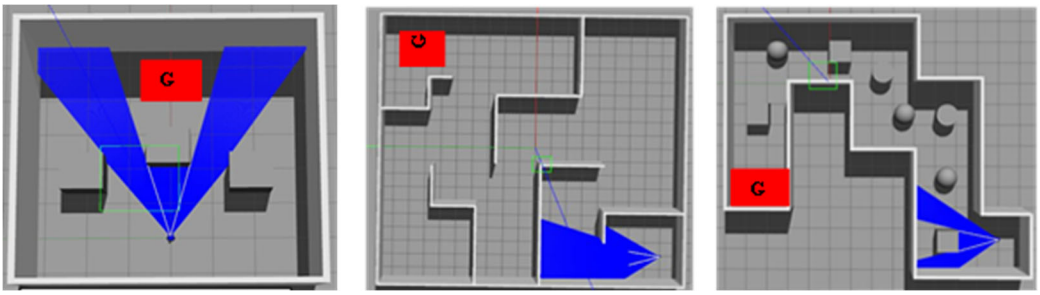
Three different kinds of environments were constructed in this work inside the Gazebo simulation platform to assess the navigation system of the robot. Every environment offers different difficulties that let one evaluate the path-planning and collision-avoidance techniques of the robot.

(1) Simple Environment: The first scene has three rectangular boxes set in an open area with a simple arrangement. A simple layout helps with basic navigation chores, including path planning and obstacle detection. It provides a basis for assessing the robot's minimum obstacle navigating capacity. Fig. 2(a) depicts the visual image of the environment.

(2) Room_Type: The second environment resembles a room with walls and open parts, therefore reflecting a more ordered scene. This arrangement limits the robot's mobility inside specified borders, therefore adding more complexity. It is therefore appropriate for testing navigation in limited areas as well as for the robot's capacity to effectively traverse walls. Fig. 2(b) displays a visual representation of the environment.

(3) Maze Environment: The final environment is combines several forms of obstacles, a narrow navigation space, and walls to anticipate a more laborious navigation situation created as a maze environment. This environment evaluates the robot's path-planning capacity in high-level constrained environments and its adaptation to increasingly complicated configurations with different obstacle shapes and layouts. Fig. 2(c) shows a visual representation of the environment.

From the basic obstacle avoidance in the simple type environment to sophisticated pathfinding methods and proper execution of the planner path in the presented three environments, each environment was carefully chosen to assess several facets of the robot's navigation system.



(a) (b) (c)

Fig. 2. Models of simulated test conditions: (a) a simple environment with an open area and three boxes; (b) room-type environment with wall partitions; (c) maze environment with mixed obstacles and a sophisticated layout. G is the defined goal point.

C. Planner Combinations

Four generally utilized global-local planner combinations were investigated in this work to grasp their behavioral variations in various contexts and both local planner's basic parameters are enlisted in Appendix A:

- A* + DWA: A* is the global planner which follow the heuristic search process and produces an initial shortest path. The DWA manages real-time obstacle avoidance depending on velocity sampling [32]. The discrete character of A* causes this mix between path optimality and dynamic feasibility to yield quite inflexible trajectories.
- A* + TEB: A* offers a grid-based global path with the TEB local planner. Particularly good for congested surroundings, TEB maximizes the trajectory in both space and time to produce smoother and more natural motion [32].
- Dijkstra + DWA: which generates the lowest pathways on the grid by computing a complete-cost global path free of heuristic method-based path planning. Then DWA runs the path reactively. Although this combination provides strong resilience. It can be computationally heavier and less flexible in confined areas than heuristic-guided approaches [18].
- Dijkstra + TEB: Dijkstra's exhaustive path exploration in concert with TEB's trajectory to smoothing robot maneuvering with high level of the optimality. It is particularly useful in situations when Safe and predictable travel is essential, even if the lack of heuristic direction could slow down the global planning phase in intricate maps [10].

D. Grid Resolutions

The proposed navigation algorithms are evaluated in terms of adaptability and robustness in environmental layouts. In this study, three different environmental layouts, namely, simple, room type, and maze, are used. Three grid resolutions were selected for each environment: 0.025 m, 0.10 m, and 0.15 m. The selection of these resolutions was made to balance the details and performance of environmental representation. The smaller grid size of 0.025 m captured finer environment. environment details and provided obstacle information to the planning, thus allowing the planner navigation. e more complex situations. However, is required more computing power. On the contrary, lower resolution (0.10 m and 0.15 m) could render a more general map with lower computational requirements at the expense of some relevant details for accurate navigation. This study will systematically evaluate the algorithm's performance across these resolutions to obtain a detailed analysis of the algorithm's adaptability and scalability. In real-world robotics applications, it often varies the complexity of the environment and the constraints.

E. Performance Metrics

A set of performance criteria has been developed to objectively assess the efficiency of any global-local planner combination. These metrics evaluate several aspects of navigation quality. Each of them is briefly explained in the following sections.

(1) Success Rates (SR): This metric counts the number of successful navigation tryouts out of all the trials shows in Eq. (1). If the SR is higher, it means that the planner combination can get the robot to its goal without any problems. Compare each planner's success rate when navigating through different settings or surroundings.

$$\text{SR} = \frac{Successfully\ Completed}{Maximum\ No.\ of\ Experiment} \times 100 \quad (1)$$

(2) Goal Convergence (GC): Identify how closely the robot's trajectory matches the global-local planner paths. Its aim is to estimate goal convergence between global and local planner paths. The Euclidean distance between the global path's goal point and the local path's endpoint provides the most straightforward indication of goal convergence shows in Eq. (2).

$$\text{GC} = \sqrt{\left(X_{local}^{end} - X_{global}^{goal}\right)^2 + \left(Y_{local}^{end} - Y_{global}^{goal}\right)^2} \quad (2)$$

(3) Spatial Coefficient (SPC): This term emphasizes that the proximity of the distance traversed by the robot and the generated global planner to the intended navigation distance, as specified by

$$\text{SPC} = 1 - \frac{S_r - S_{min}}{S_{max} - S_{min}} \quad (3)$$

In Eq. (3), Sr represents the distance traveled by the robot, S_{min} denotes the straight-line distance between the start point and target point, and S_{max} signifies the greatest distance the robot may maneuver in this experiment. The result value ranges from 0 to 1, with 1 indicating that the executed path aligns with the planned path.

(4) Temporal Coefficient (TMC): This statistic correlation between the robot's execution time and the projected navigation time, derived in Eq. (4) as follows:

$$\text{TEC} = 1 - \frac{t_r - t_{min}}{t_{max} - t_{min}} \quad (4)$$

- Let t_r denote the duration expended by the robot throughout the experiment.
- Define t_{min} as the minimal duration necessary for the robot with maximum velocity to traverse a linear path from the initial point to the goal point.
- t_{max} be the longest duration the robot is permitted to navigate in this experiment, defined as 10 times of t_{min}

The resultant number spans from 0 to 1, with a value of 1 signifying that the elapsed time corresponds to the intended period.

(5) Smoothness Coefficient (SMC): The smoothness of the trajectory the local planner generates is a key indication of the robot's natural mobility. The average angle variations between every trajectory segment help assess this smoothness.

$$\text{SMC} = 1 - \frac{\sum_{i=1}^{n} (\arctan(y_i - y_{i-1}, x_i - x_{i-1})/\pi)}{n-1} \quad (5)$$

In Eq. (5), n denotes the total number of points along the carried-out path; x and y are the coordinates of these points. The resultant smoothness score runs from 0 to 1; the calculation value is nearer to 1, which defines the best possible smoothness path navigation.

SMC is directly connected to trajectory curvature characteristics and dynamic motion quality. Lower smoothness of the curve typically implies the sharp turns, sudden angular velocity changes, high lateral accelerations and rise the mechanical stress, which are strongly related to jerk and mechanical stress. Besides TEC for time performance and velocity stability which accounting the velocity variation and motions interruption of the robot maneuvering time. In addition, SPC for path optimality and energy-related distance efficiency, which measure the deviation from the optimal trajectories and obstacle clearance consistency. The suggested framework combines these measures to provide a single evaluation structure that accounts for temporal efficacy, spatial efficiency, and dynamic feasibility.

(6) Statistical Analysis: To assess the impact of velocity variations and the grid resolution size on local planner performance across different map environments, present study employed a multi-level quantitative analysis framework. The goal is to identify the impact of velocity and grid resolution variations on the robot trajectory quality based on SPC, TEC, and SMC metrics under hybrid planner combination strategies. Primarily, a single-factor Analysis of Variance (ANOVA) was conducted for each environment type (Simple, Room type, Maze) and map resolution (0.025 m, 0.10 m, and 0.15 m) to identify whether the differences in velocity groups (0.5, 0.8, 1.0 m/s). Furthermore, Identify the significant influence of each performance metric (SPC, TEC, SMC) on every single planner combination. ANOVA is crucial in this context as it determines whether the mean differences observed among velocity groups are statistically significant or simply due to random variance. This justifies that the impact of velocity variations plays an essential role in trajectory behavior across various planning configurations.

For performance metrics that showed significant ANOVA results, post-hoc pairwise t-tests with Bonferroni correction were applied to pinpoint specification in between the similar groups of data to identify the group differences. The Bonferroni adjustment plays a crucial role here to control the Type I error rate that can arise from conducting multiple comparisons and ensuring statistical reliability. The current section revealed which specific velocity pairs (e.g., 0.5 m/s, 0.8 m/s, 0.8 & 1.0 m/s) had a significant influence on robot trajectory performance at the experiment time. These findings highlight the sensitivity of local planners to velocity tuning, especially under

specific map complexities and grid resolution sizes. This analysis ensures a transparent and repeatable strengthening of the specific group's differences rather than just general patterns.

By executing this quantitative analysis on the result verifications, the proposed study not only quantifies the velocity effects on the local planning behavior in resolution sensitive environments but also provides a robust statistical foundation for future comparative studies in autonomous navigation research.

IV. Result and Discussion

Several distinct test scenarios were used to evaluate four planner combinations: A* + DWA, A* + TEB, Dijkstra + DWA, and Dijkstra + TEB. These settings paired three sorts of environments such as simple, room-type, and maze with three maps resolutions (0.025 m, 0.10 m, and 0.15 m). Moreover, 2000 simulations in all were run, with velocities ranging from 0.2 m/s to 1 m/s. Plotting the average controller metrics SPC, TEC, and SMC.

A. Success Rate Evaluation

This chapter compares two local planners, TEB and DWA, by performing in different experimental environments, namely: Simple, Room Type, and Maze. while focusing particularly on their success rates. Results have shown that, particularly in complicated situations, the TEB planner performs better than DWA. Furthermore, the limitations of DWA that necessitated its exclusion from more comprehensive analysis are discussed.

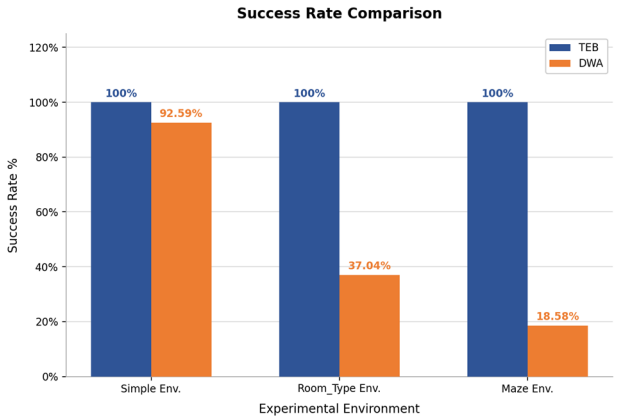


Fig. 3. Planner success rate comparison.

As depicted in Fig. 3, the TEB and DWA planners had high success rates in experimental environments. The TEB planner completed every trajectory in all environments (100% success), demonstrating its robustness even in very cluttered settings. Unlike DWA, planners could reach a goal in the Simple Environment 92.59% of the time. 37.04% for Room Type Environment and 18.58% for Maze Environment.

The underperformance of the DWA planner is due to several factors related to its algorithmic design and parameter-tuning requirements. DWA planners typically need a higher inflation radius than 0.35 m to operate correctly and not collide with the obstacles since they navigate in inflation layers. But such an inflated safety margin can severely hamper the planner's navigability in narrow or cluttered areas, causing the complexity to increase.

DWA uses a method that relies on a velocity option process that evaluates a discrete velocity command.

Though this method is computationally effective, it often faces challenges such as getting trapped in local minima and opting for less-than-ideal trajectories as conditions become increasingly complex and dynamic. For instance, when doing a sharp turn frequently or when there are too many obstacles, sampling-based trajectory generation will be problematic. Furthermore, because the DWA local planner does not have trajectory refinement based on optimization available in TEB, its flexibility in complex situations is further restricted.

The DWA using some key weighted factors like goal distance bias, path distance bias, and obstacle cost scaling to control the velocity command at the navigation process. With the robot faster moves the dynamic window gets smaller because of acceleration and deceleration limits. In cluttered or maze-like environments, this makes it harder to find safe velocity options, which can lead to oscillations, getting stuck in local minima, or stopping too soon.

Grid resolution makes this problem worse. With coarse grids, like 0.15 m, obstacle edges are less accurate, so obstacles look bigger or blockier. Since DWA checks for nearby obstacles at every control step, it might see these larger boundaries as collision risks, leading to overly cautious or stopped paths. On the other hand, very fine grids, such as 0.025 m, give more detail but create sharper cost changes. If the obstacle cost weights are not set well, this can make the robot switch speeds unpredictably.

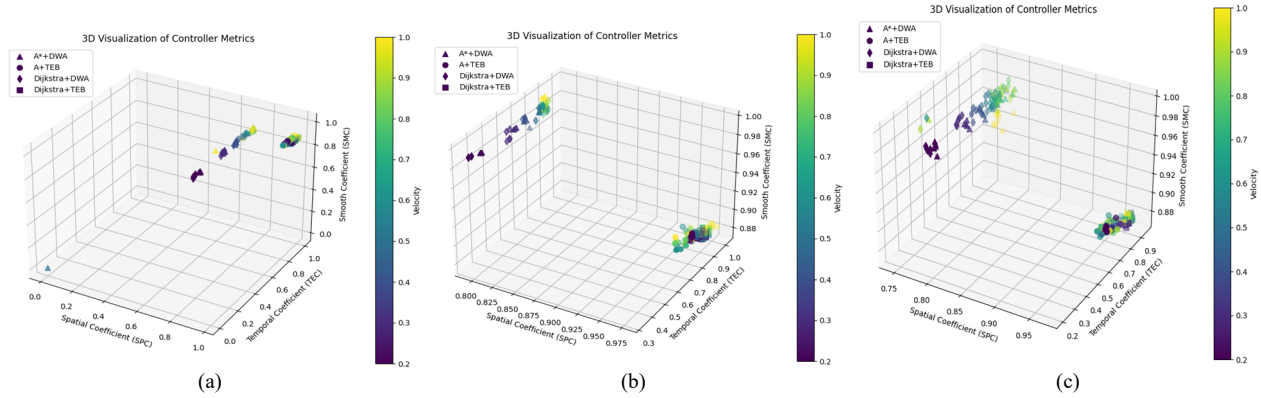
Because of these restrictions and the consistently low success rates of DWA planners, all subsequent analyses in this study only focus on the TEB planner. This focus ensures performance evaluations and optimization analyses yield the robust and reliable results that exhibit the characteristics of autonomy and dynamism.

B. 3D multi-Metrics Cluster Analysis

Table I depicts the performance evaluation of A* + TEB and Dijkstra + TEB local planners, which was conducted across three environments—Simple, Room type, and Maze—at three different grid resolutions (0.025 m, 0.1 m, and 0.15 m), using SPC, TEC, and SMC as key metrics. The presented results indicate that Dijkstra + TEB consistently outperforms A* + TEB, particularly in SPC and TEC, with the highest TEC (0.95) and SPC (0.98) observed in the Simple environment at 0.025m grid resolution. On the other hand, both algorithms displayed similar SMC values. However, the Dijkstra + TEB combination maintained slightly better or equal values across all the conditions with minimal variance, signifying stable and reliable behavior.

TABLE I. COMPARATIVE EVALUATION OF THE A* + TEB AND DIJKSTRA + TEB HYBRID LOCAL PLANNERS BASED ON THEIR TRAJECTORY PERFORMANCE

Environment	Resolution (m)	(Mean & SD)	A* + TEB			Dijkstra + TEB		
			SPC	TEC	SMC	SPC	TEC	SMC
Simple	0.025	Mean	0.97	0.91	0.89	0.98	0.95	0.89
		Std.	0.00	0.04	0.01	0.00	0.03	0.00
	0.1	Mean	0.96	0.93	0.88	0.98	0.93	0.89
		Std.	0.01	0.05	0.00	0.00	0.03	0.00
	0.15	Mean	0.96	0.80	0.88	0.97	0.89	0.88
		Std.	0.01	0.05	0.00	0.00	0.06	0.00
Room_type	0.025	Mean	0.79	0.87	0.74	0.81	0.91	0.74
		Std.	0.00	0.09	0.01	0.00	0.10	0.01
	0.1	Mean	0.79	0.87	0.74	0.80	0.93	0.74
		Std.	0.00	0.09	0.00	0.00	0.10	0.01
	0.15	Mean	0.79	0.85	0.74	0.80	0.91	0.74
		Std.	0.00	0.09	0.01	0.00	0.11	0.01
Maze	0.025	Mean	0.60	0.86	0.78	0.61	0.93	0.77
		Std.	0.00	0.06	0.01	0.00	0.08	0.01
	0.1	Mean	0.60	0.82	0.77	0.61	0.91	0.77
		Std.	0.00	0.09	0.00	0.00	0.12	0.01
	0.15	Mean	0.60	0.90	0.77	0.61	0.94	0.77
		Std.	0.00	0.07	0.01	0.00	0.08	0.01



(a) (b) (c)

Fig. 4. 3D visualization illustrates how planner efficiency and smoothness vary with resolution granularity and environmental complexity. Each subplot represents a unique environment-resolution pair and plots the SPC, TEC, and SMC in a 3D metric space in simple environment. (a) Resolution Size 0.025 m; (b) Resolution Size 0.10 m; (c) Resolution Size 0.15 m.

In the other two cases, Room type and Maze, Dijkstra + TEB continuously demonstrates superior performance, especially on TEC and SPC metrics. This highlights the better adaptability to environmental complexity with a minimum variance of the values. Based on the combined evaluation of mean values and low standard deviations across all metrics, Dijkstra + TEB is identified as the optimal local planner for generating efficient and smooth trajectories across varying resolutions and environmental structures.

To identify the most reliable global-local planner combinations a comprehensive 3D multi-cluster analysis was performed across all environments, grid resolutions, and velocity settings. In 3D plane Visualizing the three most important parameters, metrics SPC, TEC and SMC with velocity encoded as a continuous color gradient, clear performance patterns emerged which is depicted in Figs. 4–6 for each different environment grid resolution. The TEB-based hybrid planners (A* + TEB and Dijkstra + TEB) consistently formed dense and compact clusters in the high-performance region, characterized by SPC and TEC values above 0.85 and SMC values approaching 0.95. These clusters remained stable across all the grid resolutions, indicating that TEB is highly robust to map granularity and velocity variation. In contrast, the DWA-based planner combinations (A* + DWA and Dijkstra + DWA) exhibited noticeably lower SPC and SMC values at wider dispersion and shifted, particularly at high grid resolution sizes, which reflects reduced accuracy of spatial coefficient and increased trajectory irregularity. The cluster analysis across all the attached figures on Fig. 4 reveals that TEB planner combination maintains superior smoothness and temporal consistency even under high-velocity conditions. On the other hand, the DWA performance deteriorates as velocity and grid resolution size increase. Overall, considering the delivery of accurate, time-efficient, and trajectories smoothness parameters 3D cluster distribution demonstrates that the TEB-based planners outperform their DWA counterparts. A* + TEB and Dijkstra + TEB establishing as the most optimal and robust hybrid planning strategies under all tested conditions.

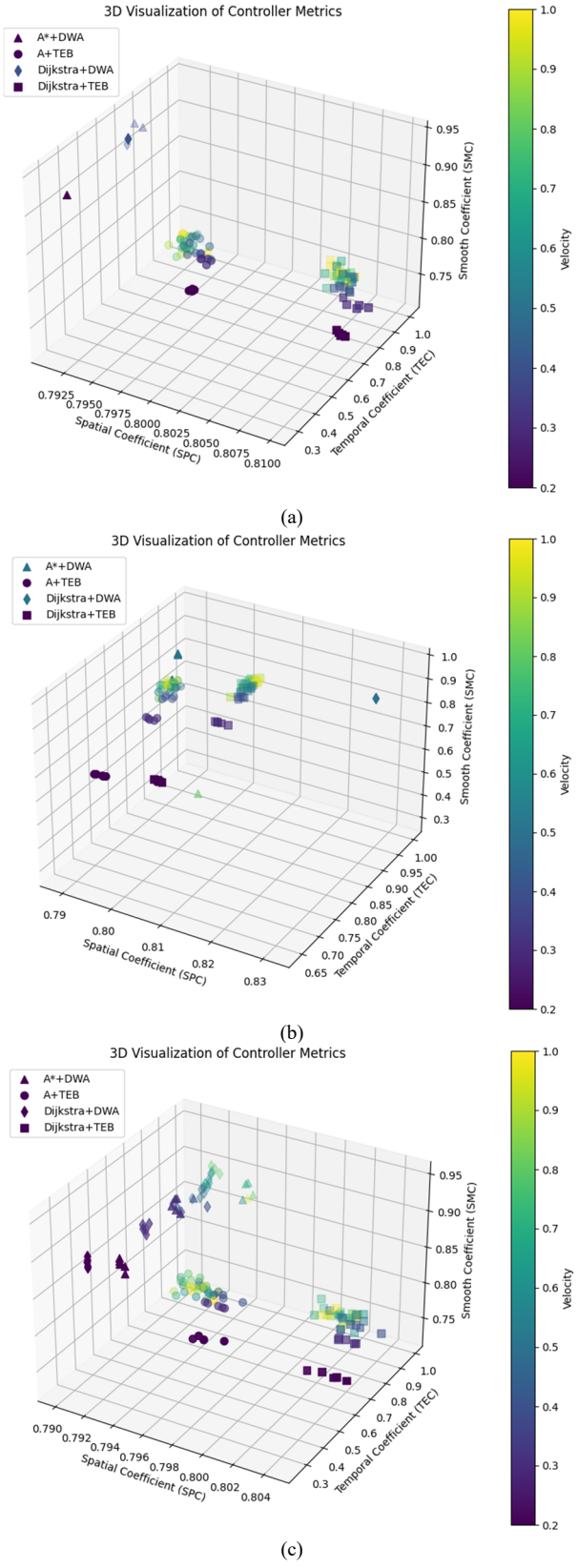


Fig. 5. 3D visualization illustrates how planner efficiency and smoothness vary with resolution granularity and environmental complexity. Each subplot represents a unique environment-resolution pair and plots the SPC, TEC, and SMC in a 3D metric space in Room_Type environment. (a) Resolution size 0.025 m; (b) Resolution size 0.10 m; (c) Resolution size 0.15 m.

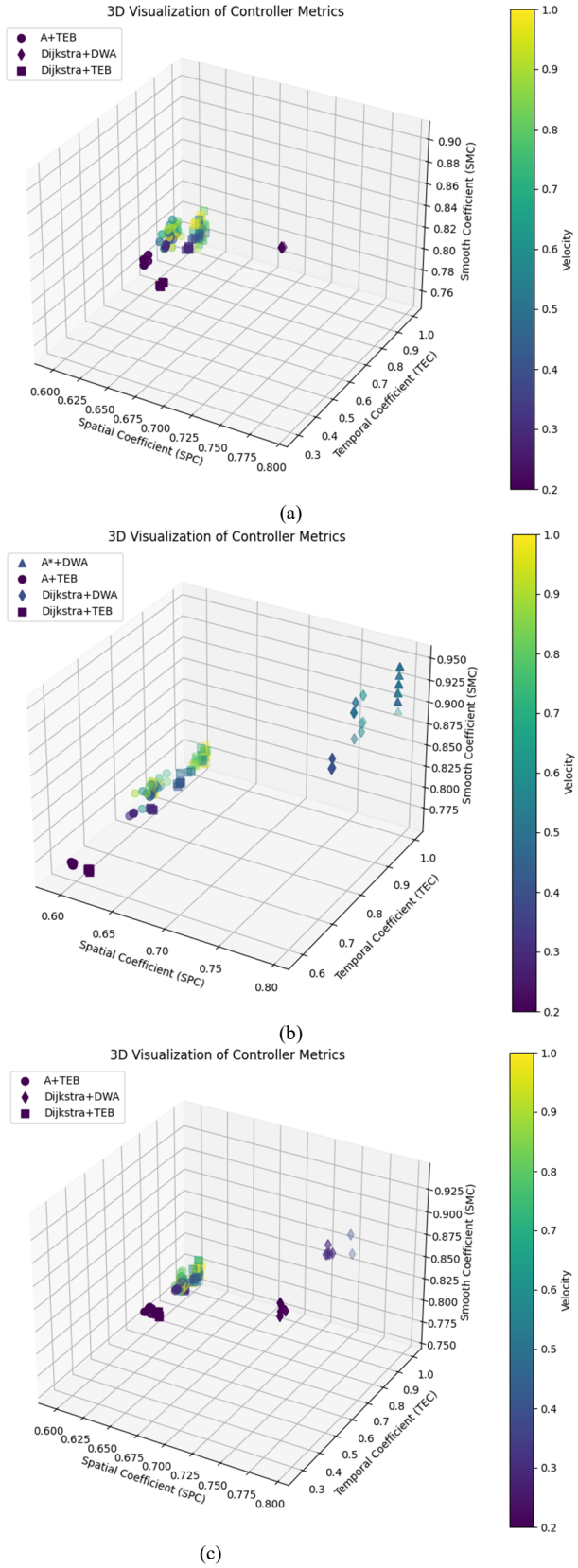


Fig. 6. 3D visualization illustrates how planner efficiency and smoothness vary with resolution of granularity and environmental complexity. Each subplot represents a unique environment-resolution pair and plots the SPC, TEC, and SMC in a 3D metric space in Maze environment. (a) Resolution size 0.025 m; (b) Resolution size 0.10 m; (c) Resolution size 0.15 m.

C. Evaluation of Velocity Effects on Trajectory

To evaluate the influence of velocity on the robot's trajectory performance, statistical analyses like ANOVA and post hoc t-test were conducted for three environment types. The ANOVA analysis revealed that velocity has an overall effect on one or more performance parameters in most conditions. Subsequently, post hoc t-tests with Bonferroni correction were employed to detect which specific velocity pairs exhibited statistically significant differences. Table II shows that significant velocity effects were observed in very few cases in the Simple environment, particularly in TEC and SMC for both A* + TEB and Dijkstra + TEB planner cases, due to the simplicity of the environment and less turning track.

TABLE II. SIMPLE ENVIRONMENT VELOCITY EFFECTS ON PARAMETERS OF THE ROBOT TRAJECTORY

Resolutions (m)	Parameters	Velocity Groups (m/s)	A* + TEB		Dijkstra + TEB	
			p-value (T-test)	Significant?	*p*-value (T-test)	Significant?
0.025	SPC	0.5 vs. 0.8	0.09182949	NO	0.239356324	NO
		0.8 vs. 1.0	0.933174843	NO	0.002651321	YES
		1.0 vs 0.5	0.047747841	NO	0.232262197	NO
	TEC	0.5 vs. 0.8	0.001987849	YES	0.001987849	YES
		0.8 vs. 1.0	0.277611339	NO	0.277611339	NO
		1.0 vs. 0.5	0.000539392	YES	0.000539392	YES
	SMC	0.5 vs. 0.8	0.261877385	NO	0.261877385	NO
		0.8 vs. 1.0	0.319858296	NO	0.319858296	NO
		1.0 vs 0.5	0.782652361	NO	0.782652361	NO
0.1	SPC	0.5 vs. 0.8	0.355328844	NO	0.278647708	NO
		0.8 vs. 1.0	0.840899141	NO	6.24825E−06	YES
		1.0 vs. 0.5	0.413689203	NO	0.001117813	YES
	TEC	0.5 vs. 0.8	0.221410965	NO	0.221996935	NO
		0.8 vs. 1.0	0.375601342	NO	0.005114602	YES
		1.0 vs. 0.5	0.790339468	NO	0.342778845	NO
	SMC	0.5 vs. 0.8	0.043877045	NO	0.605898277	NO
		0.8 vs. 1.0	0.006023225	YES	0.75218181	NO
		1.0 vs. 0.5	0.274809601	NO	0.951833835	NO
0.15	SPC	0.5 vs. 0.8	0.776690604	NO	0.204406434	NO
		0.8 vs. 1.0	0.319194098	NO	0.533456362	NO
		1.0 vs. 0.5	0.333158159	NO	0.066205328	NO
	TEC	0.5 vs. 0.8	0.181332954	NO	0.094317267	NO
		0.8 vs. 1.0	0.994880601	NO	0.000307157	YES
		1.0 vs. 0.5	0.189531753	NO	0.29312511	NO
	SMC	0.5 vs. 0.8	0.562624729	NO	0.003314208	YES
		0.8 vs. 1.0	0.368591954	NO	0.225933828	NO
		1.0 vs. 0.5	0.655853034	NO	0.004059531	YES

In the Room type environment (Table III), the results further confirmed that velocity variations significantly affected the performance of TEC metrics. Because the TEC metrics completely depend on the robot execution time. Each resolution variation of the TEC parameters shows a significant impact in the room type environment scenario. In addition to that, the SPC metric also shows a significant effect at the presence of 0.025 and 0.1 m grid resolution sizes.

In the Maze environment, which represents the most complex layout in Table IV, A* + TEB demonstrated significant velocity dependent behavior, particularly significant *p*-values were found for SPC and TEC (e.g., SPC at 0.1 m with $p = 0.002129207$; SMC at 0.15 m with $p = 0.005402834$) at A* + TEB planner combinations, besides (e.g., SPC at 0.1 m with $p = 0.000124914$; SMC with $p = 0.00069246$) at Dijkstra + TEB planner combinations. Both planners showed a significant impact on the robot trajectories in a different range of velocities. This indicates that changes in velocity resulted in meaningful differences in trajectory performance.

The data table represented the significant difference among the grid sizes on various parameters and the environment is clearly noticeable. Especially in the case of the A* + TEB planner, it has shown a significant impact at every grid resolution variation. Conversely, the Dijkstra + TEB showed a significant difference in SPC and SMC metrics. There has been no impactful difference shown at the TEC planner in the three environmental variations.

To clearly visualize the difference in the data, the table has attached a significant command row that identifies the difference by viewing "YES" or "NO" command based on the t-test *p*-value.

To analyze the effect of the grid resolution on various metrics, a statistical analysis has been performed in this section, as shown in Table VI. The Eta-squared (η^2) values were collected by using the one-way ANOVA method, which indicates that grid resolution had a substantial influence on navigation metrics, particularly in the simple and maze environments. For the A* + TEB planner, large effect sizes were observed in the simple environment ($\eta^2 = 0.38–0.64$), clearly showing that discretization significantly impacts path smoothness and trajectory efficiency. On the other side, the room-type environment exhibited minimal effects on TEC and SMC ($\eta^2 \leq 0.05$), indicating that geometric constraints dominate performance more than grid resolution.

TABLE III. ROOM TYPE ENVIRONMENT VELOCITY EFFECTS ON PARAMETERS OF THE ROBOT TRAJECTORY

Resolutions (m)	Parameters	Velocity Groups (m/s)	A* + TEB		Dijkstra + TEB	
			p-value (T-test)	Significant?	*p*-value (T-test)	Significant?
0.025	SPC	0.5 vs. 0.8	0.102977445	NO	0.290427697	NO
		0.8 vs. 1.0	1	NO	0.12587244	NO
		1.0 vs. 0.5	0.102977445	NO	0.04597542	NO
	TEC	0.5 vs. 0.8	0.011384434	YES	0.058956324	NO
		0.8 vs. 1.0	0.0002217	YES	0.98143223	NO
		1.0 vs. 0.5	4.70968E−06	YES	0.024562359	NO
	SMC	0.5 vs. 0.8	0.128983838	NO	0.84945521	NO
		0.8 vs. 1.0	0.067987038	NO	0.719471069	NO
		1.0 vs. 0.5	0.50098082	NO	0.532813725	NO
0.1	SPC	0.5 vs. 0.8	0.018752193	NO	3.7859E−11	YES
		0.8 vs. 1.0	0.316290408	NO	0.369538434	NO
		1.0 vs. 0.5	0.005972923	YES	0.003817777	YES
	TEC	0.5 vs. 0.8	0.003144461	YES	2.87594E−06	YES
		0.8 vs. 1.0	0.012759449	YES	7.34131E-06	YES
		1.0 vs. 0.5	0.657304436	NO	0.003784527	YES
	SMC	0.5 vs. 0.8	0.430952333	NO	0.239838139	NO
		0.8 vs. 1.0	0.004150194	YES	0.421904362	NO
		1.0 vs. 0.5	2.159E−11	YES	0.023970775	NO
0.15	SPC	0.5 vs. 0.8	0.142636994	NO	0.172393092	NO
		0.8 vs. 1.0	0.986338978	NO	0.809574049	NO
		1.0 vs. 0.5	0.386798478	NO	0.056265369	NO
	TEC	0.5 vs. 0.8	0.015790275	YES	0.786595547	NO
		0.8 vs. 1.0	0.165538823	NO	0.433941474	NO
		1.0 vs. 0.5	0.962455527	NO	0.400446103	NO
	SMC	0.5 vs. 0.8	0.886403572	NO	0.627384713	NO
		0.8 vs. 1.0	0.483175394	NO	0.545054204	NO
		1.0 vs. 0.5	0.076808301	NO	0.367959875	NO

D. Influence of Grid Resolution

In this section, investigate the impact of map grid resolution on robot trajectory performance among resolution groups (0.025 m, 0.10 m, and 0.15 m). A single-factor ANOVA was first conducted to assess whether significant differences existed. The analysis was performed separately for each environment with a constant velocity of 0.5 m/s.

The post hoc t-tests with Bonferroni correction were applied to identify specific pairwise differences among resolution levels as presented in Table V.

For the Dijkstra + TEB planner, the influence of grid resolution on SPC was especially pronounced, with very large effect sizes observed across environments (η^2 = 0.69–0.92). Nevertheless, on-grid changes in a structured environment, the metrics TEC and SMC showed small to moderate sensitivity. These results show that the grid resolution plays a dominant role in path complexity metrics, while its influence on energy and motion smoothness depends strongly on environmental structure and planner characteristics.

E. Comparative Analysis of Path Lengths

In this section, the path length difference is demonstrated, which is produced by the global and local planners. The bar chart in Fig. 7 illustrates a comparative analysis of path lengths produced by four planning strategies, such as A*, A + TEB*, Dijkstra, and Dijkstra + TEB. In the Simple environment, all the planners produced similarly optimal paths due to the many open spaces and the less complex environment. The length of the start to end point is nearly (~5 m).

In the Room type and maze environment, these two environments were developed with the highest density and complexity of structured obstacles. Among the four path planning methods, the Dijkstra + TEB consistently produced the optimal path lengths, slightly outperforming A* + TEB, to reveal that Dijkstra-based planners may handle room navigation more efficiently when supported by the TEB local planner. Due to the greedy nature, A* and A* + TEB tend to generate marginally longer paths in these conditions.

TABLE IV. MAZE ENVIRONMENT VELOCITY EFFECTS ON PARAMETERS OF THE ROBOT TRAJECTORY

Resolutions (m)	Parameters	Velocity Groups (m/s)	A* + TEB		Dijkstra + TEB	
			p-value (T-test)	Significant?	*p*-value (T-test)	Significant?
0.025	SPC	0.5 vs. 0.8	0.119570484	NO	0.022497533	NO
		0.8 vs. 1.0	0.608898201	NO	0.396649774	NO
		1.0 vs. 0.5	0.010817329	YES	0.197596502	NO
	TEC	0.5 vs. 0.8	0.102367661	NO	0.661706445	NO
		0.8 vs. 1.0	0.938514813	NO	0.078401516	NO
		1.0 vs. 0.5	0.064556381	NO	0.023279592	NO
	SMC	0.5 vs. 0.8	0.887588285	NO	0.001715749	YES
		0.8 vs. 1.0	0.567174919	NO	0.908534078	NO
		1.0 vs. 0.5	0.507397117	NO	0.004545634	YES
0.1	SPC	0.5 vs. 0.8	0.002129207	YES	0.462158986	NO
		0.8 vs. 1.0	0.5876197	NO	0.000124914	YES
		1.0 vs 0.5	0.005130497	YES	4.19045E−05	YES
	TEC	0.5 vs. 0.8	0.00014629	YES	0.204488734	NO
		0.8 vs. 1.0	0.031706189	NO	0.00069246	YES
		1.0 vs. 0.5	0.000369296	YES	0.00133512	YES
	SMC	0.5 vs. 0.8	0.12317563	NO	0.031261732	NO
		0.8 vs. 1.0	0.498938477	NO	0.745121194	NO
		1.0 vs. 0.5	0.219245251	NO	0.003713939	YES
0.15	SPC	0.5 vs. 0.8	0.252373407	NO	0.440732148	NO
		0.8 vs. 1.0	0.592527856	NO	0.262735295	NO
		1.0 vs. 0.5	0.553289845	NO	0.63930984	NO
	TEC	0.5 vs. 0.8	0.258612546	NO	0.849293327	NO
		0.8 vs. 1.0	0.518260834	NO	0.228396205	NO
		1.0 vs. 0.5	0.078687705	NO	0.028114336	NO
	SMC	0.5 vs. 0.8	0.396515777	NO	0.825562499	NO
		0.8 vs. 1.0	0.279865759	NO	0.368404397	NO
		1.0 vs. 0.5	0.005402834	YES	0.2540096	NO

TABLE V. STATISTICAL EVALUATION OF GRID RESOLUTION EFFECTS ON TRAJECTORY PERFORMANCE

Environment	Parameters	Resolution Groups (m)	A* + TEB		Dijkstra + TEB	
			p-value (T-test)	Significant?	*p*-value (T-test)	Significant?
Simple	SPC	0.025 vs. 0.1	2.11328E−82	YES	2.95528E−91	YES
		0.1 vs. 0.15	4.09088E−21	YES	3.21687E−43	YES
		0.15 vs. 0.025	1.02476E−38	YES	5.54206E−74	YES
	TEC	0.025 vs. 0.1	0.009536326	YES	0.216547686	NO
		0.1 vs. 0.15	0.002410465	YES	0.355663016	NO
		0.15 vs. 0.025	1.68958E−06	YES	0.02933321	NO
	SMC	0.025 vs. 0.1	4.46403E−27	YES	4.75618E−19	YES
		0.1 vs. 0.15	2.46699E−05	YES	3.49994E−20	YES
		0.15 vs. 0.025	2.1139E−30	YES	3.06395E−36	YES
Room_type	SPC	0.025 vs. 0.1	2.722E−142	YES	4.8101E−151	YES
		0.1 vs. 0.15	5.3405E−143	YES	2.3096E−153	YES
		0.15 vs. 0.025	0.00332126	YES	0.00020476	YES
	TEC	0.025 vs. 0.1	0.000528627	YES	0.471843129	NO
		0.1 vs. 0.15	0.959873158	NO	0.135793982	NO
		0.15 vs. 0.025	0.003701102	YES	0.394005344	NO
	SMC	0.025 vs. 0.1	0.360814838	NO	0.000260075	YES
		0.1 vs. 0.15	6.21632E−10	YES	4.54657E−15	YES
		0.15 vs. 0.025	5.00758E−08	YES	0.000744777	YES
Maze	SPC	0.025 vs. 0.1	1.40254E−53	YES	2.39157E−57	YES
		0.1 vs. 0.15	2.01805E−50	YES	9.43557E−71	YES
		0.15 vs. 0.025	0.032955466	NO	2.72726E−24	YES
	TEC	0.025 vs. 0.1	0.090051295	NO	0.63091935	NO
		0.1 vs. 0.15	0.267838306	NO	0.749292709	NO
		0.15 vs. 0.025	0.002455457	YES	0.410002786	NO
	SMC	0.025 vs. 0.1	2.26956E−16	YES	4.11447E−10	YES
		0.1 vs. 0.15	1.13927E−07	YES	0.000260743	YES
		0.15 vs. 0.025	1.49636E−05	YES	4.44246E−05	YES

TABLE VI. ETA-SQUARE EVALUATION OF GRID RESOLUTION EFFECTS ON PERFORMANCE METRICS

Environment	Metrics	A* + TEB		Dijkstra + TEB	
		η^2	Interpretation	η^2	Interpretation
Simple	SPC	0.38	Large	0.92	Very Large
	TEC	0.56	Very Large	0.27	Medium
	SMC	0.64	Very Large	0.48	Very Large
Room_Type	SPC	0.20	Large	0.69	Very Large
	TEC	0.01	Small	0.01	Small
	SMC	0.05	Small	0.02	Small
Maze	SPC	0.37	Large	0.70	Very Large
	TEC	0.17	Large	0.02	Small
	SMC	0.28	Large	0.05	Small

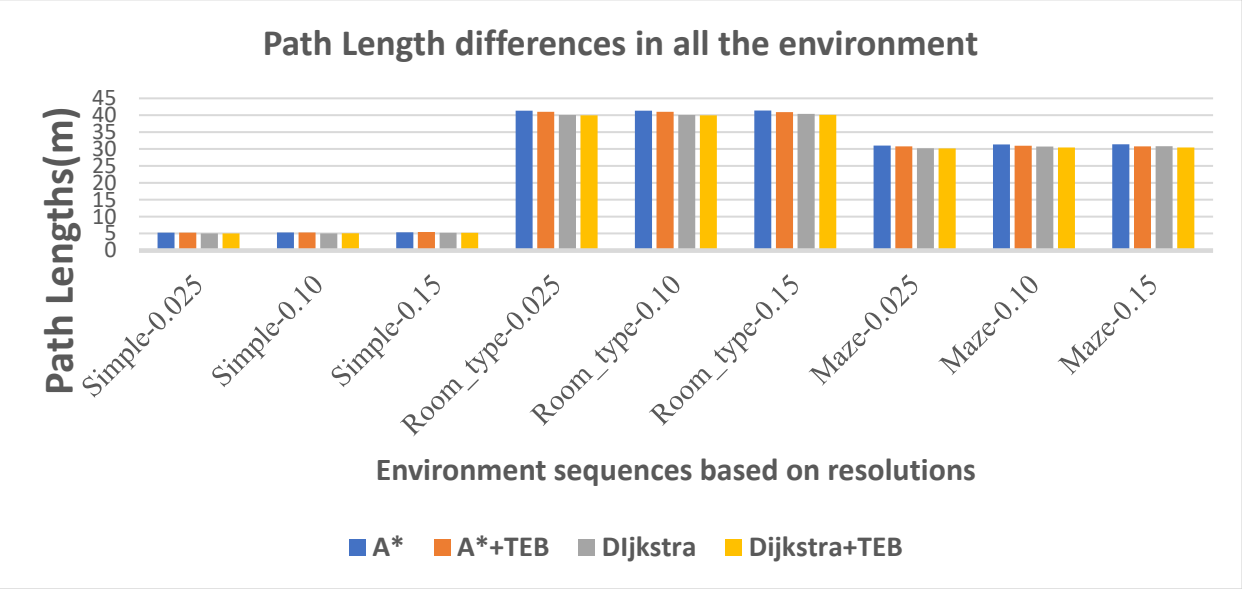


Fig. 7. Path length comparison between global and local planner.

F. Goal Convergence (GC)

In this section, describe the goal convergence performance among the A* + TEB and Dijkstra + TEB path planner combinations. The box plots in Fig.6 illustrate the Goal Convergence (GC) accuracy of various planning algorithms across Simple, Room type, and Maze environments. Fig. 8(a) shows that the A* + TEB planner has a high value fluctuation compared to Dijkstra + TEB. Only in the 0.10 grid resolution case does the A* + TEB present fewer variations of the GC parameters.

Fig. 8(b) and (c) depict the room type and maze environment case GC performance. In both environment cases, the Dijkstra + TEB shows outperform compared to the A* + TEB planner combination. The Dijkstra + TEB planner shows less value fluctuations with a minimal error rate. Which clearly defines the Dijkstra + TEB planner combination to localize the robot most effectively to the destination point.

Fig. 9(a), (c), and (e) show the performance of the A* global planner across three grid resolutions in the mentioned environmental setup. In a simple environment, with a finer resolution of 0.025 m, the A* planner captures fine details of obstacle boundaries, leading to more precise but longer trajectories due to dense node expansion. This enhances spatial adherence of the path but increases computational time. A good balance is maintained by maintaining efficient curvature while reducing unnecessary turns, allowing for a smoother path planning strategy by the global planner in a medium grid resolution of 0.10 m. At a coarser resolution of 0.15 m, paths become more angular and less adaptable, especially in the maze's narrow corridors, as shown in Fig. 9(e). Overall, these findings suggest that A* is versatile across different resolutions, with finer grids preferred for prioritizing precision and collision avoidance. The results show that grid resolution significantly influences path smoothness and fidelity.

Fig. 9(b), (d), and (f) illustrate how the Dijkstra–TEB planner operates in different environments. At a grid resolution of 0.025 m, Dijkstra creates the most detailed paths, keeping a safe distance from obstacles and resulting in smooth trajectories. Overall, the Dijkstra global planner oversimplifies path structure, missing small waypoints, which can lead to shifts and sharper turns at every turning track.

In all, Dijkstra generally produces a more robust path than A*. However, finer grids are better for detailed environments. Both global planners behave similarly regarding grid resolution, but A* is more sensitive to changes, leading to larger differences in paths at various resolutions. In open or structured spaces, paths are nearly identical, but in cluttered areas like mazes, finer grids greatly enhance path realism. The 0.10 m resolution remains the best choice, balancing smooth paths with reduced computational effort.

Fig. 10(a)–(c) show the trajectories of two hybrid planner combinations, A* + TEB and Dijkstra + TEB planners, which are tested in simple, room-type, and maze environments at grid resolutions of 0.025 m, 0.10 m, and 0.15 m. Both planners, when showing localization to the goal point, perform well across three environments. However, Dijkstra + TEB provided smoother and more aligned trajectories compared to the A* + TEB planner combination. The A* + TEB showed minor lateral deviations, particularly at finer resolutions. A* + TEB produced overfitted paths around corners in the room-type environment navigation time. In the maze environment, Dijkstra + TEB achieved stable and collision-free trajectories with smoother transitions, while A* + TEB displayed sharper turns.

Considering all the aspects of the planner's path, Dijkstra + TEB proved to be the most accurate and smooth across all scenarios, especially in the maze environment, which benefits from uniform cost exploration that minimizes oscillations and enhances convergence. On the other hand, A* + TEB remains effective in more open spaces where heuristic bias aids in faster pathfinding.

G. Result Discussion

This study presented a comprehensive evaluation of four combinations of path planning strategies, particularly A + TEB* and Dijkstra + TEB were tested under varying environmental complexities, map resolutions, and velocity conditions. The primary objective was to find out the optimal path planning combinations across various environmental settings to assess trajectory quality, execution efficiency, and success reliability using a wide array of performance indicators, including SPC, TEC, SMC, GC, path length, execution time, and success rate.

The findings consistently highlight that Dijkstra + TEB outperforms A + TEB* across almost all tested conditions. Both algorithms achieved almost similar path lengths and convergence accuracy. However, The Dijkstra + TEB offered slightly smoother trajectories and faster execution times in the simple structured environment. However, in more structured and complex environments such as Room type and Maze type conditions the differences became more pronounced.

Dijkstra + TEB planner shows outperform which maintained significantly lower path lengths, tighter goal convergence, and reduced computational costs compared to A* + TEB. At the higher velocity profile, the Dijkstra + TEB combinations exhibited stable and efficient motion, while A* + TEB exhibited more fluctuation and increased execution time.

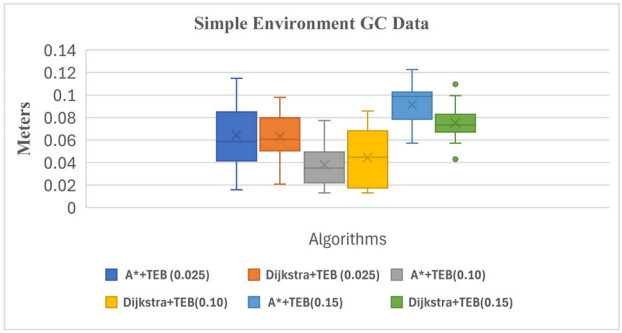
The experimental results indicate that at the finer grid size of 0.025 m, the planners can capture more accurate details of the environment to execute precise and collision-free paths. The quantitative data analysis asserted that the grid resolution size significantly affects the performance and trajectory quality of hybrid planner combinations. However, this increased spatial fidelity leads to a higher computational load and longer planning times. On the other hand, a medium grid size of 0.10 m strikes the best balance between computational efficiency and trajectory smoothness.

The research investigated at the grid size increases beyond 0.15 m, which reveals that the planner begins to lose the environment details, resulting in the path of the planner being more angular, less smooth, and occasionally inaccurate in goal localization.

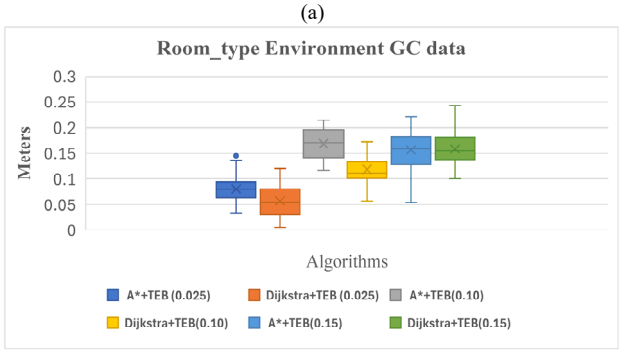
In this study the experimental data of the three important metrics were tested by placing 3D multi-metric cluster analysis to visualize the optimality of every single test planner under various conditions. Furthermore, some renowned statistical analyses were performed to identify the impact of the various grid resolutions and velocities on the trajectory performance. Dijkstra + TEB demonstrated greater resilience to resolution and velocity changes, particularly in SPC and SMC metrics.

In contrast, A* + TEB showed more sensitivity and suggested its dependency on fine-tuned parameters especially in TEC and SMC parameters. The GC box plots depicted Dijkstra + TEB's accuracy with achieving the lowest and most consistent errors across all scenarios. Moreover, the DWA performance is drastically declining in the complex environment (18.58% in maze) which is shown in the success rate plot. On the other hand, TEB-based planners consistently achieved 100% success across all environments, proving their robustness.

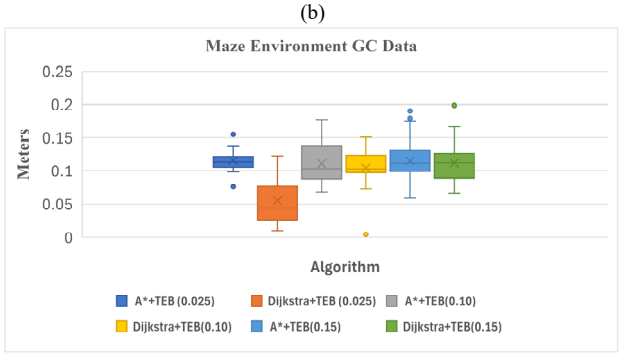
#



(a)



(b)



(c)

Fig. 8. Goal Convergence data comparison in each resolution of the grid map. (a) Simple Environment; (b) Room_Type Environment; (c) Maze Environment. Trajectory visualization.

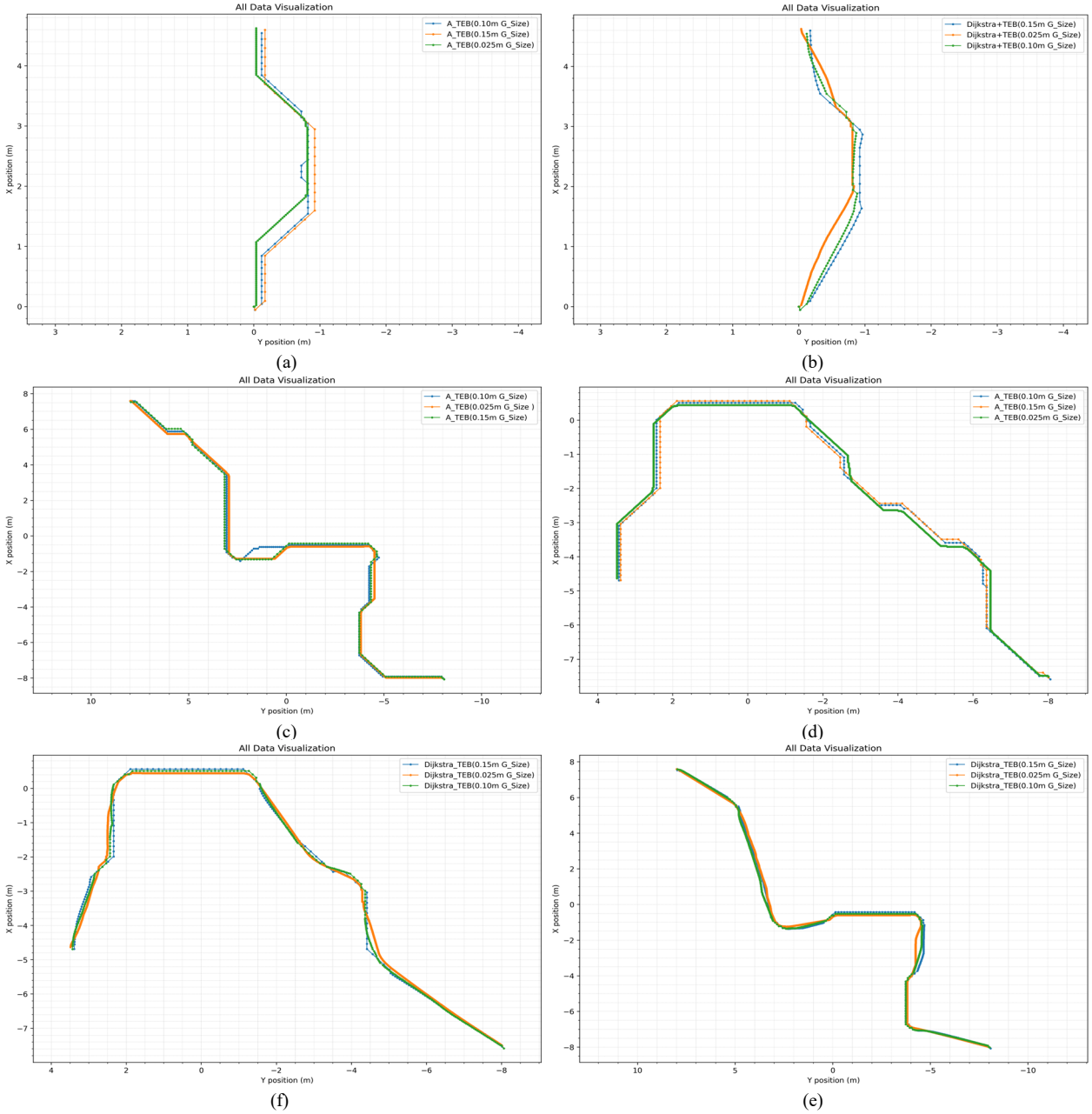


Fig. 9. Evaluate the performance of two global planner strategies—(a) Simple Env. A* with 0.025 m grid resolution; (b) Simple Env. Dijkstra with 0.025 m grid resolution; (c) Room type Env. A* with 0.10 m grid resolution; (d) Room type Env. Dijkstra with 0.10 m grid resolution; (e) Maze Env. A* with 0.15 m grid resolution; (f) Maze Env. Dijkstra with 0.15 m grid resolution.

H. Generalizability and Real-World Considerations

The proposed framework was tested in a controlled simulation environment. However, the various issues arise when transferring the approach to physical robotic platforms. Simulation-based evaluation provides consistency, repeatability, and controlled parameter tuning. however, real-world deployment introduces additional uncertainties that may influence planner performance. Such as real robots are subject to sensor noise, localization drift, actuator delays, and wheel slippage, which are typically simplified or idealized in simulation. These factors may affect trajectory tracking accuracy and goal localization precision, particularly for planners that rely heavily on fine spatial discretization.

Secondly In the real-world mapping the occupancy grids generated from LiDAR or vision sensors often contain dynamic artifacts, partial occlusions, and mapping inconsistencies which may impact on the grid-based planners' environmental representation.

Third, dynamic obstacles and human interaction introduce non-deterministic behavior that were not fully modelled in the simulation environments. such kind situations inherently may control by the support of the TEB planner. on the other hand, the real-time perception latency and computational constraints may reduce responsiveness compared to simulated execution.

Despite these limitations, the relative performance trends observed in this study particularly the trade-offs between grid resolution, smoothness, and convergence

accuracy are expected to remain consistent in real-world scenarios. However, absolute metric values such as execution time, path smoothness, and success rate may vary depending on hardware characteristics and environmental uncertainty.

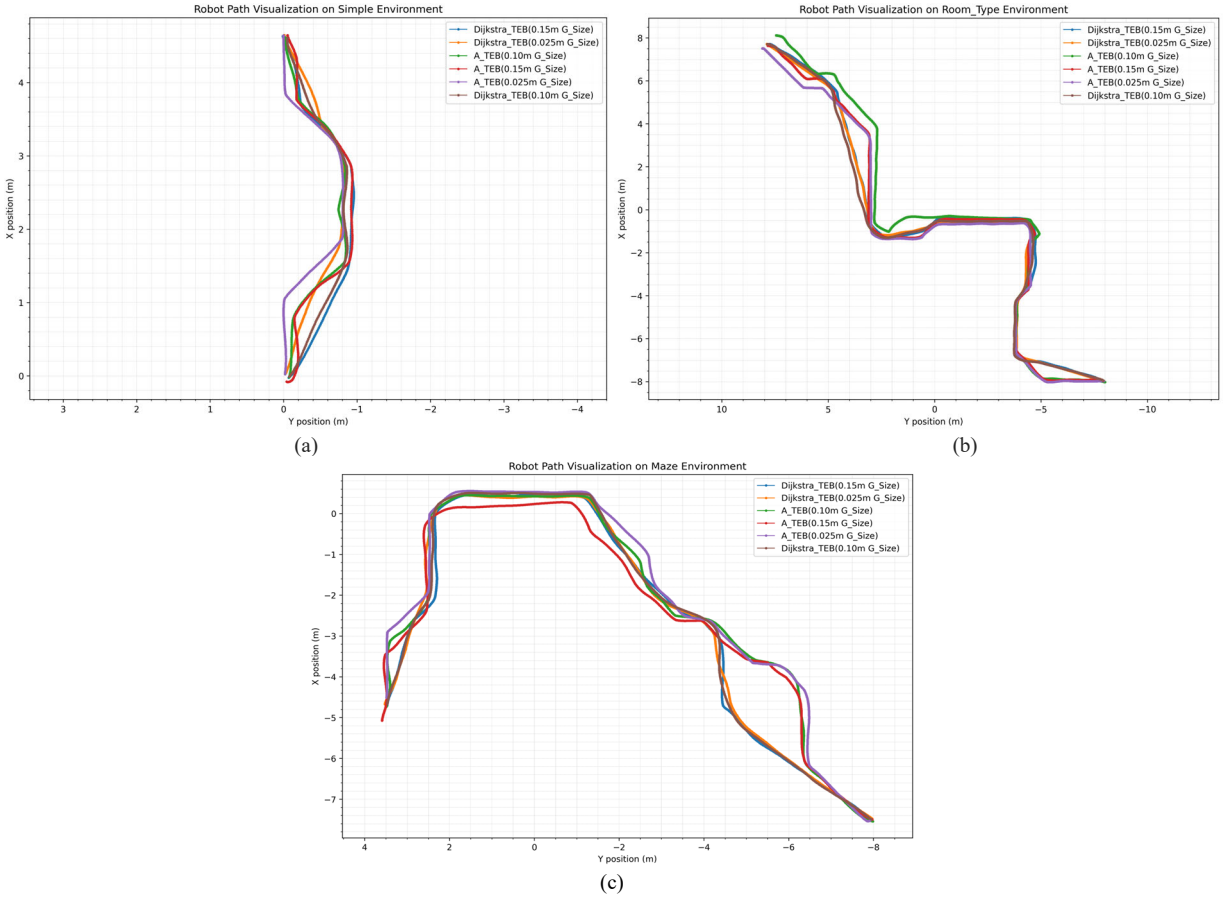


Fig. 10. Evaluate the performance of two hybrid planning strategies—(a) simple Env. planner trajectories; (b) Room_type Env. planner trajectories; (c) Maze Env. planner trajectories.

V. Conclusion

This research provides a detailed comparative framework for evaluating path planning algorithm combinations across four types of hybrid path planning combinations in autonomous mobile robots. The experimental data have been analyzed by various quantitative formulas and 3D cluster-based analysis, along with extensive trajectory visualization, statistical validation, and success rate assessment. Among the four planner combinations, Dijkstra + TEB emerges as the most optimal hybrid planning approach, which offers superior performance in terms of trajectory efficiency, accuracy, and computational cost. Especially in semi-structured and complex environments, which consistently perform well in the variations of the grid resolutions and velocity settings, it's where robustness and reliability are critical.

In contrast, the DWA local planner-based path planning methods show a poor performance in a structured environment. The DWA local planner needs an open space to generate a safe trajectory by avoiding obstacles. In a complex structure with a high density of obstacles, the DWA planner often failed to produce a path and reach the destination. On the other hand, A* + TEB performed reasonably well in simpler scenarios, which create a higher path length and execution time under challenging conditions, making it less favorable in dynamic or densely populated spaces. The presented research addresses a crucial fact of the path planning algorithm, which plays a vital role in the autonomous mobile robot navigation system. It helps the users to select a proper path planning combination based on the demand of the structure to maneuver the robots to the destination point safely.

Appendix A: DWA and TEB Parameters List

TABLE AI. Key Parameters of the Local Planner

DWA Parameters	TEB Parameters
A) Sampling Parameters	**A) Optimization Weights**
vx_samples: 80	weight_max_vel_x: 2

vy_samples: 0 vth_samples: 50 sim_time: 2.0 sim_granularity: 1.0 angular_sim_granularity: 0.8	weight_max_vel_theta: 1 weight_acc_lim_x: 1 weight_acc_lim_theta: 1 weight_kinematics_nh: 1000 weight_kinematics_forward_drive: 2000 weight_optimaltime: 0.1 weight_obstacle: 20.0 weight_viapoint: 20.0 weight_inflation: 0.1
B) Trajectory Scoring Weights path_distance_bias: 1000 goal_distance_bias: 200 occdist_scale: 0.05 forward_point_distance: 0.325 stop_time_buffer: 0.5	**B) Trajectory Optimization Parameters** dt_ref → 0.2 dt_hysteresis → 0.15 teb_autosize → True global_plan_overwrite_orientation: True
C) Goal Tolerance xy_goal_tolerance: 0.1 2.14 latch_xy_goal_tolerance: false	**C) Goal Tolerance** xy_goal_tolerance: 0.1 yaw_goal_tolerance: 2.14 free_goal_vel: false
D) Oscillation Prevention: oscillation_reset_dist: 0.01 oscillation_reset_angle: 0.05	**D) Homotopy Class** Planning enable_homotopy_class_planning: True enable_multithreading: max_number_classes: 3 ue max_number_classes:3 h_signature_threshold: 0.1 h_signature_prescaler: 1.0
-	**E) Obstacle** parameters: min_obstacle_dist: 0.1 **rameters** min_obstacle_dist: inflation_dist: 0.1 0.1 include_costmap costmap_obstacles_behind_robot_dist: 0.1 true costmap_obstacles_behind_robot_dist: obstacle_poses_affected: 40

CONFLICT OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] G. Fragapane, R. de Koster, F. Sgarbossa *et al.,* "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," *European Journalof Operational Research*, vol. 294, no. 2, 2021, pp. 405–426, 2021. doi: 10.1016/j.ejor.2021.01.019

[2] J. Liang, "A path planning algorithm of mobile robots in a known 3D environment," *Procedia Engineering*, vol. 15, no. 157–162, 2011.

[3] S. A. M. Abgenah, A. A. Jamal, and S. A. Fadzli, "Multi-robot path planning using Dijkstra's algorithm with multi-layer dictionaries," *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, vol. 10, no. 1, pp. 52–57, 2021.

[4] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[5] N. Persson, M. C. Ekström, M. Ekström *et al.,* "On the Initialization Problem for Timed-Elastic Bands," *IFAC-Papers On Line*, vol. 56, no. 2, pp. 11802–11807, 2023. doi: 10.1016/j.ifacol.2023.10.574

[6] S. Xing, P. Fan, X. Ma *et al.,* "Research on robot path planning by integrating state-based decision-making a* algorithm and inertial dynamic window approach," *Intelligent Service Robotics*, vol. 17, no. 4, pp. 901–914, 2024. doi: 10.1007/s11370-024-00547-0

[7] Z. Zhou, E. Abdi, C. P. Chea *et al.,* "Global path planning for autonomous construction vehicles in building construction: A comparative study with a focus on vehicle kinematic characteristics," *Journal of Building Engineering*, vol. 93, 109837, 2024. doi: 10.1016/j.jobe.2024.109837

[8] M. Quigley, K. Conley, B. Gerkey *et al.,* "ROS: An open-source robot operating system," in *Proc. of ICRA Workshop on Open Source Software*, vol. 3, 5. 2009.

[9] S. Kato, S. Tokunaga, Y. Maruyama *et al.,* "Autoware on board: Enabling autonomous vehicles with embedded systems," in *Proc. 2018 ACM/IEEE 9th International Conf. on Cyber-Physical Systems (ICCPS)*, 2018, pp. 287–296.

[10] P. Marin-Plaza, A. Hussein, D. Martin *et al.,* "Global and local path planning study in a ROS-based research platform for autonomous vehicles," *Journal of Advanced Transportation*, vol. 2018, no. 1, 6392697, 2018. doi: 10.1155/2018/6392697

[11] H. Martínez-Barberá and D. Herrero-Pérez, "Autonomous navigation of an automated guided vehicle in industrial environments," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 3, pp. 296–311, 2010. doi: 10.1016/j.rcim.2009.10.003

[12] M. Reda, A. Onsy, A. Y. Haikal *et al.,* "Path planning algorithms in the autonomous driving system: A comprehensive review," *Robotics and Autonomous Systems*, vol. 174, 104630, 2024.

[13] M. E. Miyombo, Y. Liu, C. M. Mulenga *et al.,* "Optimal path planning in a real- world radioactive environment: A comparative study of A-star and Dijkstra algorithms," *Nuclear Engineering and Design*, vol. 420, 113039, 2024.

[14] C. Saranya, K. K. Rao, M. Unnikrishnan *et al.,* "Real-time evaluation of grid-based path planning algorithms: A comparative study," *IFAC Proceedings Volumes*, vol. 47, no. 1, pp. 766–772, 2014.

[15] J. P. Bailey, A. Nash, C. A. Tovey *et al.,* "Path-length analysis for grid-based path planning," *Artificial Intelligence,* vol. 301, 103560, 2021.

[16] L. Liu, J. Lin, J. Yao *et al.,* "Path planning for smart car based on Dijkstra algorithm and dynamic window approach," *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, 8881684, 2021.

[17] C. Z. Looi and D. W. K. Ng, "A study on the effect of parameters for ROS motion planner and navigation system for indoor robot," *Journal of Electrical and Computer Engineering Research*, vol. 1, no. 1, pp. 29–36, 2021.

[18] I. Chaari, A. Koubaa, H. Bennaceur *et al.,* "Design and performance analysis of global path planning techniques for autonomous mobile robots in grid environments," *International Journal of Advanced Robotic Systems*, vol. 14, no. 2, 2017.

[19] D. H. Han, Y. D. Kim, and J. Y. Lee, "Multiple-criterion shortest path algorithms for global path planning of unmanned combat vehicles," *Computers & Industrial Engineering*, vol. 71, pp. 57–69, 2014.

[20] F. A. M. Pimentel and P. T. Aquino-Jr, "Evaluation of ROS navigation stack for social navigation in simulated environments," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 4, 87, 2021.

[21] Z. Yu, Z. Si, X. Li *et al.,* "A novel hybrid particle swarm optimization algorithm for path planning of UAVs," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22547–22558, 2022. doi: 10.1109/JIOT.2022.3182798

[22] R. Zhang, H. Guo, D.Andriukaitis *et al.,* "Intelligent path planning by an improved RRT algorithm with dual grid map," *Alexandria Engineering Journal*, vol. 88, pp. 91–104, 2024. doi: 10.1016/j.aej.2023.12.044

[23] M. Nazarahari, E. Khanmirza, and S. Doostie, "Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm," *Expert Systems with Applications*, vol. 115, pp. 106–120, 2019. doi: 10.1016/j.eswa.2018.08.008
[24] K. Shi, L. Huang, D. Jiang *et al.,* "Path planning optimization of intelligent vehicle based on improved genetic and ant colony hybrid algorithm," *Frontiers in Bioengineering and Biotechnology*, vol. 10, 905983, 2022. doi: 10.3389/fbioe.2022.905983
[25] G. Klančar, M. Seder, S. Blažič *et al.,* "Drivable path planning using hybrid search algorithm based on E* and Bernstein–Bézier motion primitives," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 8, pp. 4868–4882, 2021. doi: 10.1109/TSMC.2019.2945110
[26] Y. Baek and J. K. Park, "Fast path generation algorithm for mobile robot navigation using hybrid map," *Applied Sciences*, vol. 15, no. 5, 2414, 2025. doi: 10.3390/app15052414
[27] H. Wen, Z. Song, S. Liu *et al.,* "A hybrid LiDAR-based mapping framework for efficient path planning of AGVs in a massive indoor environment," *IEEE Transactions on Transportation Electrification*, vol. 10, no. 2, pp. 3504–3517, 2024. doi: 10.1109/TTE.2023.3280738
[28] H. Gao, T. Zhang, Z. Zuo *et al.,* "USV path planning in a hybrid map using a genetic algorithm with a feedback mechanism," *Journal of Marine Science and Engineering*, vol. 12, no. 6, 939, 2024. doi: 10.3390/jmse12060939
[29] J. Liu, X. Chen, J. Xiao *et al.,* "Hybrid map-based path planning for robot navigation in unstructured environments," in *Proc. the 2023 IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS)*, 2023, pp. 2216–2223.
[30] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
[31] J. Gu and Q. Cao, "Path planning using hybrid grid representation on rough terrain," *Industrial Robot*, vol. 36, no. 5, pp. 497–502, 2009. doi: 10.1108/01439910910980222
[32] R. Goldstein, K. Walmsley, J. Bibliowicz *et al.,* "Path counting for grid-based navigation," *Journal of Artificial Intelligence Research*, vol. 74, pp. 917–955, 2022.
[33] Y. Ou, Y. Fan, X. Zhang *et al.,* "Improved A* path planning method based on the grid map," *Sensors*, vol. 22, no. 16, 6198, 2022. doi: 10.3390/s22166198
[34] T. Mohanraj, T. Dinesh, and B. Guruchandhramavli, "Mobile robot path planning and obstacle avoidance using hybrid algorithm," *International Journal of Information Technology*, vol. 15, no. 8, pp. 4481–4490, 2023. doi: 10.1007/s41870-023-01475-5
[35] X. Zhong, J. Tian, and H. Hu, "Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment," *Journal of Intelligent & Robotic Systems*, vol. 99, no. 1, pp. 65–77, 2020. doi: 10.1007/s10846-019-01112-z
[36] J. Zhu and D. Pan, "Improved genetic algorithm for solving robot path planning based on grid maps," *Mathematics*, vol. 12, no. 24, 4017, 2024. doi: 10.3390/math12244017
[37] B. Han, T. Qu, X. Tong *et al.,* "Grid-optimized UAV indoor path planning algorithms in a complex environment," *International Journal of Applied Earth Observation and Geoinformation*, vol. 111, 102857, 2022. doi: 10.1016/j.jag.2022.102857
[38] Z. Wang, G. Li, and J. Ren, "Dynamic path planning for unmanned surface vehicle in complex offshore areas based on hybrid algorithm," *Computer Communications*, vol. 166, pp. 49–56, 2021. doi: 10.1016/j.comcom.2020.11.012
[39] X. Fan, Y. Wang, and Z. Zhang, "An evaluation of Lidar-based 2D SLAM techniques with an exploration mode," *Journal of Physics: Conference Series*, vol. 1905, no. 1, 012021, 2021.
[40] DOBOT AI-starter. [Online]. Available: https://www.dobot.nu/en/product/dobot-ai-starter/
[41] LiDAR Camera L515 Datasheet. [Online]. Available: https://dev.intelrealsense. com/docs/lidar-camera-l515-datasheet
[42] ROS Navigation Stack. [Online]. Available: https://wiki.ros.org/navigation
[43] Slam_karto. [Online]. Available: https://wiki.ros.org/slam_karto/