

Exploring RRT and BiRRT Algorithms: A Review and Simulation-Based Comparison for Fixed-Wing UAV Path Planning

Gilang Nugraha Putu Pratama^{1,2}, Oktaf Agni Dhewa¹, Mentari Putri Jati³, Indra Hidayatulloh⁴,
Teddy Surya Gunawan⁵, Syaiful Ardy Gunawan⁶

¹ Department of Electrical and Electronics Engineering, Universitas Negeri Yogyakarta, Indonesia

² Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada, Indonesia

³ Department of Electro-Optical Engineering, National Taipei University of Technology, Taiwan

⁴ School of Computing Engineering and Built Environment, Glasgow Caledonian University, United Kingdom

⁵ Department of Electrical and Computer Engineering, International Islamic University Malaysia, Malaysia

⁶ Seiko Epson Corporation, Japan

ARTICLE INFORMATION

Article History:

Received 21 August 2025

Revised 07 April 2026

Accepted 23 April 2026

Keywords:

RRT;

BiRRT;

Path Planning;

Fixed-Wing;

UAV

Corresponding Author:

Gilang Nugraha Putu Pratama,
Department of Electrical and
Electronics Engineering,
Universitas Negeri Yogyakarta,
Indonesia.

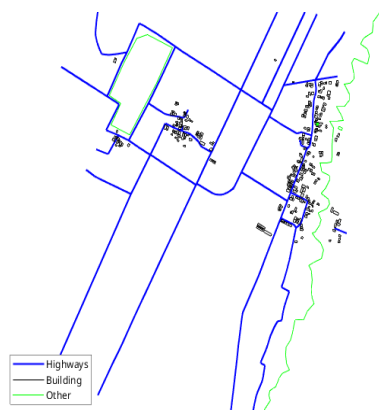
Email:

gilang.n.p.pratama@uny.ac.id

This work is open access under a
[Creative Commons Attribution-Share
Alike 4.0](https://creativecommons.org/licenses/by-sa/4.0/)



ABSTRACT



Path planning plays a vital role in ensuring the safe and efficient navigation of fixed-wing unmanned aerial vehicles (UAVs), particularly in cluttered and complex environments. The increasing demand for autonomous UAV operations highlights the need for reliable algorithms capable of generating optimal and collision-free trajectories. This study addresses the challenge by reviewing recent uses of the Rapidly-exploring Random Tree (RRT) algorithm in various robotic platforms and navigation tasks. The research contribution of this paper is a comparative analysis of RRT and BiRRT for fixed-wing UAV path planning, quantifying trade-offs between path length, computation time, and obstacle clearance using a real-world 2D urban map. This addresses a gap in the literature, as few studies have directly compared these algorithms specifically for fixed-wing UAV surveillance missions. The methods involve implementing both RRT and BiRRT in a simulated environment where each algorithm is evaluated over 100 runs to measure performance metrics such as path length, computation time, and obstacle clearance. A realistic urban map is used to test the algorithms under consistent starting and goal positions. The results show that both RRT and BiRRT achieve a 100% success rate in finding collision-free paths. BiRRT consistently generates shorter paths and requires less computation time, making it more suitable for time-sensitive missions. However, RRT produces safer trajectories with greater average clearance from obstacles, which is advantageous in environments with high collision risk. The findings demonstrate a clear trade-off between safety and efficiency. In conclusion, BiRRT is recommended for missions where speed and efficiency are prioritized, while RRT is better suited for operations emphasizing safety and obstacle avoidance.

Document Citation:

G. N. P. Pratama, O. A. Dhewa, M. P. Jati, I. Hidayatulloh, T. S. Gunawan, and S. A. Gunawan, "Exploring RRT and BiRRT Algorithms: A Review and Simulation-Based Comparison for Fixed-Wing UAV Path Planning," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 8, no. 2, pp. 476-487, 2026, DOI: 10.12928/biste.v8i2.14511.

1. INTRODUCTION

In recent years, unmanned aerial vehicles (UAVs) have experienced a surge in popularity, owing to their expanding utility across a broad spectrum of applications. While UAVs were initially developed and deployed primarily for research initiatives and military operations [1]-[3], advancements in technology and reductions in production costs have facilitated their adoption in various civilian sectors. Today, UAVs play a pivotal role in tasks such as security surveillance, infrastructure inspection, precision agriculture for pest control, and aerial imaging for commercial or recreational purposes [4]-[7]. These platforms offer numerous advantages, including enhanced maneuverability, cost-efficiency, and the ability to access areas that may be dangerous or impractical for human operators.

When it comes to long-range surveillance missions, the selection of UAV platforms that combine energy efficiency with flight stability becomes particularly important. Among the available configurations, fixed-wing UAVs stand out due to their aerodynamic design, which enables them to fly longer distances using less energy compared to multicopter counterparts. The integration of vertical take-off and landing (VTOL) capability into fixed-wing UAVs further enhances their operational flexibility. This hybrid approach allows the aircraft to take off and land vertically in confined spaces (such as urban environments or rugged terrain) while still benefiting from the efficient cruise performance of fixed-wing flight [8]-[11]. As a result, fixed-wing VTOL UAVs are especially well-suited for extended surveillance operations in complex and diverse environments.

However, navigating such environments presents numerous challenges for path planning systems. In order to achieve mission success while ensuring safety, UAV flight paths must be carefully optimized to satisfy multiple, often competing, criteria. These include minimizing energy consumption, adhering to mission objectives, and ensuring the safety of both the UAV and its surroundings [12]-[14]. A critical requirement is the avoidance of populated or heavily built-up areas, even if this necessitates a longer flight path. Furthermore, UAVs must maintain clear lines of sight for data acquisition and surveillance, while avoiding collisions with static and dynamic obstacles, such as buildings, trees, or other aerial vehicles. The complexity is compounded by environmental factors such as weather conditions, topographical variations, and airspace restrictions, all of which demand a robust and adaptive path planning strategy that can operate effectively from the moment of take-off to the point of landing [15].

One of the most promising solutions to these planning challenges lies in the use of the Rapidly-exploring Random Tree (RRT) algorithm. Originally introduced by LaValle [16][17], RRT is a sampling-based motion planning algorithm designed to efficiently explore high-dimensional configuration spaces. By incrementally building a tree that spans the space, RRT can quickly discover feasible paths even in environments with complex constraints. Its computational efficiency and ability to handle dynamic and high-dimensional spaces have led to its adoption in a wide range of robotic systems beyond UAVs. These include autonomous ground vehicles, robotic manipulators, surgical robots for precise needle steering, and robotic systems used in automated logistics and warehouse management [18]-[22].

In an effort to overcome the limitations of the original RRT algorithm (such as suboptimal path quality and long planning times) various improved versions have been proposed. Among them, Bidirectional RRT (BiRRT) is one of the most effective. BiRRT enhances planning efficiency by simultaneously growing two trees: one from the starting point and another from the goal. These trees attempt to connect to each other, often resulting in faster convergence and better path solutions, especially in environments with tight constraints or limited connectivity [23]-[25].

Recent studies have explored the use of RRT and its variants to further enhance the navigational capabilities of UAVs. For instance, Iwashina *et al.* [26] integrated vehicle-to-everything (V2X) communication data into the RRT framework to construct dynamic risk maps, thereby improving the safety of UAV flight operations in urban environments. Pohan *et al.* [27] proposed a stable and robust RRT optimization specifically adapted to the needs of UAVs and autonomous ground vehicles, demonstrating improvements in both path quality and computational performance. Kelner *et al.* [28] developed an RRT variant tailored for use in UAV swarms operating in flying ad-hoc networks (FANETs), focusing on real-time collision avoidance and communication reliability. Similarly, Lindqvist *et al.* [29] extended RRT by integrating real-time exploration and path planning capabilities into unknown environments, implemented through the Robot Operating System (ROS) to support autonomous operation in dynamic and unstructured settings.

Despite these advancements and the widespread application of RRT-based algorithms in UAV navigation, there remains a limited body of research focused on the comparative performance of RRT and BiRRT in the specific context of fixed-wing UAV path planning for surveillance missions. This study addresses that gap by presenting a detailed comparative analysis of RRT and BiRRT algorithms applied to fixed-wing UAVs equipped with VTOL capabilities. The evaluation is based on critical performance metrics, including the total path length, computation or planning time, and the ability to effectively avoid obstacles.

To enhance flight safety and reflect realistic operational conditions, a protective safety buffer is implemented around structures in the simulation environment. Specifically, a circular safe zone with a radius of 10 meters is established around the geometric center of each building. This measure ensures that UAV flight paths maintain a safe clearance from potential collision hazards and supports compliance with real-world operational safety standards.

The remainder of this paper is organized as follows. The next section provides a detailed explanation of the RRT and BiRRT algorithms utilized in this study, including their implementation principles and operational differences. Following that, a review of relevant literature is presented alongside simulation results comparing the performance of the two algorithms in fixed-wing UAV path planning scenarios. The final section summarizes the findings of the study, discusses their implications for UAV mission planning, and offers recommendations for future research to further improve the autonomy and safety of UAV systems in complex environments.

2. BASIC THEORY

Here, we discuss two algorithms central to these works. First, we briefly introduce the concept of the RRT algorithm. Then, we provide a detailed explanation of the path planning process using BiRRT. Notably, the pseudo-code for both algorithms is also included.

2.1. RRT Algorithm

RRT was introduced in 1998 by Steven M. LaValle and James J. Kuffner as a solution to the problem of path planning in high-dimensional spaces [30]–[32]. RRT works by iteratively selecting a random point in the configuration space and expanding the tree by moving from the closest node in the tree toward this random point [33][34]. If this new point is collision-free, it is added to the tree. The process continues until a path from the start to the goal is found, or a maximum number of iterations is reached. The Algorithm 1 of RRT is as follows.

Algorithm 1 RRT Algorithm
1: $Q_{goal}, stepSize, maxIter$
2: $counter = 0, G(V,E) \leftarrow \emptyset$
3: add X_{start} to V
4: while $counter < maxIter$ do
5: $X_{rand} \leftarrow RandomPosition()$
6: if $IsInObstacle(X_{rand})$ then <i>continue</i>
7: $X_{nearest} \leftarrow Nearest(G, X_{rand})$
8: $X_{new} \leftarrow Steer(X_{nearest}, X_{rand}, stepSize)$
9: $Link \leftarrow Edge(X_{nearest}, X_{new})$
10: if $IsCollisionFree(Link)$ then
11: $G \leftarrow G \cup \{X_{new}, Link\}$
12: if $X_{new} \in Q_{goal}$ then return G
13: $counter \leftarrow counter + 1$
14: return <i>Failure</i>

The algorithm takes the goal region Q_{goal} , a step size, and a maximum iteration limit. It starts with an empty graph G and adds the start position X_{start} to it. A counter keeps track of how many iterations have run. In each iteration, the algorithm picks a random position X_{rand} . If X_{rand} lies inside an obstacle, the algorithm skips to the next iteration. Otherwise, it finds the nearest node $X_{nearest}$ already in the graph. It then creates a new node X_{new} by moving from $X_{nearest}$ toward X_{rand} by the step size. The algorithm checks whether the straight line from $X_{nearest}$ to X_{new} hits any obstacle. If the line is clear, it adds X_{new} and the edge to the graph. If X_{new} falls inside the goal region Q_{goal} , the algorithm stops and returns the graph. If the loop reaches the iteration limit without reaching the goal, the algorithm returns the graph as it is.

2.2. BiRRT Algorithm

It was developed as an extension of the RRT algorithm to improve the efficiency of pathfinding in high-dimensional spaces [32]. The BiRRT approach involves growing two RRTs simultaneously, one from the start and the other from the goal [35]–[38]. The Algorithm 2 for BiRRT is as follows.

Algorithm 2 BiRRT Algorithm

```

1:  $Q_{goal}$ 
2:  $counter = 0, lim = n, G_{start}(V_{start}, E_{start}) \leftarrow \emptyset, G_{goal}(V_{goal}, E_{goal}) \leftarrow \emptyset$ 
3: add  $X_{start}$  to  $V_{start}$ , add  $X_{goal}$  to  $V_{goal}$ 
4: while  $counter < lim$  do
5:    $X_{new} \leftarrow RandomPosition()$ 
6:   if  $IsInObstacle(X_{new})$  then
7:     continue
8:    $X_{nearest} \leftarrow Nearest(G_{start}(V_{start}, E_{start}), X_{new})$ 
9:    $Link_{start} \leftarrow Chain(X_{new}, X_{nearest})$ 
10:  if  $IsCollisionFree(Link_{start})$  then
11:     $G_{start} \leftarrow G_{start} \cup \{Link_{start}\}$ 
12:    if  $ConnectGraphs(G_{start}, G_{goal})$  then
13:      return  $Merge(G_{start}, G_{goal})$ 
14:     $X_{nearest} \leftarrow Nearest(G_{goal}(V_{goal}, E_{goal}), X_{new})$ 
15:     $Link_{goal} \leftarrow Chain(X_{new}, X_{nearest})$ 
16:    if  $IsCollisionFree(Link_{goal})$  then
17:       $G_{goal} \leftarrow G_{goal} \cup \{Link_{goal}\}$ 
18:      if  $ConnectGraphs(G_{start}, G_{goal})$  then
19:        return  $Merge(G_{start}, G_{goal})$ 
20: return Failure

```

Similar to RRT, the BiRRT algorithm begins by defining the goal region Q_{goal} and the iteration limit lim . Two graphs, $G_{start}(V_{start}, E_{start})$ and $G_{goal}(V_{goal}, E_{goal})$, are initialized as empty to represent exploration from the start and goal points, respectively. The start position X_{start} is added to V_{start} , and the goal position X_{goal} is added to V_{goal} . In each iteration, a random position X_{new} is sampled from the search space. If X_{new} lies within an obstacle, as determined by $IsInObstacle()$, the algorithm skips the current iteration and moves to the next. Otherwise, the nearest vertex $X_{nearest}$ in G_{start} is identified using $Nearest()$. A potential connection, $Link_{start}$, is then created between X_{new} and $X_{nearest}$. If $Link_{start}$ is collision-free, it is added to G_{start} . The algorithm then checks if G_{start} and G_{goal} can be connected using $ConnectGraphs()$. If a connection is successful, the two graphs are merged with $Merge()$, and the algorithm terminates with a solution. If no connection is found, the algorithm proceeds to explore G_{goal} . For this graph, the nearest vertex $X_{nearest}$ to X_{new} is identified, and a potential connection $Link_{goal}$ is created between X_{new} and $X_{nearest}$. If $Link_{goal}$ is collision-free, it is added to G_{goal} . The algorithm again checks for a connection between G_{start} and G_{goal} . If successful, the two graphs are merged and returned as the solution. This process continues until either a valid connection is found or the iteration limit lim is reached. If no solution is found within the limit, the algorithm terminates and returns failure, indicating that the goal was not reached.

3. REVIEW AND SIMULATION

In this section, we present two main subsections that serve distinct but complementary purposes. Subsection 3.1. provides a comprehensive review of recent developments and implementations of the Rapidly-exploring Random Tree (RRT) algorithm across a variety of applications, with a particular emphasis on its use in unmanned aerial vehicles (UAVs). This review highlights how the RRT algorithm has been adapted and extended to meet the diverse demands of real-world environments.

Following that, Subsection 3.2. focuses on a practical comparison between the RRT and BiRRT algorithms. We present simulation results based on a realistic scenario involving fixed-wing UAVs navigating an actual map. The objective of this comparison is to evaluate the performance of both algorithms in terms of path length, computational efficiency, and obstacle avoidance under constraints typically encountered in real-world missions. Together, these subsections aim to bridge theoretical advancements with practical UAV path planning applications.

3.1. Review

This review draws on indexed articles from Scopus, covering the period from 2020 to 2025. It highlights the extensive use of RRT in various contexts—including 2D and 3D path planning, cluttered and urban

environments, ground vehicles, UAVs, and medical robotics—underscoring its adaptability and widespread appeal. For clarity and ease of comparison, the studies discussed here are summarized in a dedicated [Table 1](#).

Table 1. Summary of recent works using RRT and BiRRT.

Author	Year	Note
Jhang <i>et al.</i> [39]	2020	Autonomous parking using BiRRT with model predictive control.
Khan <i>et al.</i> [40]	2020	Trajectory planning for soft robotic manipulators using RRT.
Pohan <i>et al.</i> [18]	2021	RRT combined with Ant Colony System for improved path planning.
Mthabela <i>et al.</i> [41]	2021	BiRRT for ground robot navigation over 3D mesh surfaces.
Wang <i>et al.</i> [42]	2022	RRT for visual servoing under FOV constraints.
Lathrop <i>et al.</i> [43]	2022	Distributionally robust path planning using Wasserstein Safe RRT.
Chang <i>et al.</i> [23]	2022	UAV trajectory planning using skeleton extraction and RRT.
Ma <i>et al.</i> [44]	2022	Efficient BiRRT for ground vehicle motion planning.
Li and Chen [45]	2022	Adaptive RRT strategy based on environmental complexity.
Zhang <i>et al.</i> [46]	2023	Improved RRT for UAV navigation in complex environments.
Gai <i>et al.</i> [47]	2023	Memory-efficient RRT with adaptive thresholding.
Al-Ansarry <i>et al.</i> [48]	2023	BiRRT with segmented interpolation for dynamic environments.
Pohan and Utama [49]	2023	Efficient RRT for UAV navigation in dynamic environments.
Liu <i>et al.</i> [50]	2024	APF-based UAV simulation enhanced with RRT.
Sun <i>et al.</i> [51]	2024	Multi-Risk-RRT for autonomous airport luggage trolleys.
Yao <i>et al.</i> [52]	2025	RRT combined with artificial potential fields for path planning.
Fu <i>et al.</i> [53]	2025	Fixed-wing UAV trajectory planning using RRT.
William <i>et al.</i> [24]	2025	RRT-based safe path planning for urban air mobility.
Umer <i>et al.</i> [25]	2025	RRT-based energy-efficient and safe UAV trajectory optimization.
Li <i>et al.</i> [54]	2025	BiRRT for UAVs in 3D obstacle environments.
Zhao <i>et al.</i> [55]	2025	RRT navigation under challenging visual conditions.
Wang <i>et al.</i> [56]	2025	RRT with differential flatness-based trajectory planning for UAVs.
Qian and Wang [22]	2025	RRT for trajectory optimization in coal mining masonry robots.
Zhang <i>et al.</i> [57]	2025	Deep learning-based RRT for UAVs in dynamic environments.
Halder and Orlando [21]	2025	3D RRT path planning for bevel-tip needle steering.
Yu <i>et al.</i> [58]	2025	RRT-based homing for parafoils under wind disturbances.
Asimakopoulos and Chatzilygeroudis [59]	2025	RRT in quality-diversity search for kinodynamic optimization.

Over the years, the Rapidly-Exploring Random Tree (RRT) algorithm has undergone significant development and application across diverse fields, highlighting its popularity and flexibility in path planning. A foundational review by Elbanhawi *et al.* recognized RRT as one of the most widely used sampling-based motion planning algorithms in robotics [60]. Since then, researchers have applied and enhanced RRT in increasingly complex and varied environments. Later, Belter explored efficient modeling and evaluation of constraints in path planning for multi-legged walking robots using RRT [61].

The progress continued in 2020, when Jhang *et al.* proposed an autonomous parking system by integrating BiRRT with parking-oriented model predictive control [39], while Khan *et al.* applied RRT in a trajectory planning framework for soft robotic manipulators [40]. In 2021, Pohan *et al.* introduced a hybrid path planning approach that combined RRT with Ant Colony Systems to improve search efficiency [18]. Concurrently, Mthabela *et al.* extended BiRRT to support ground robot navigation over 3D mesh surfaces using localized region decomposition [41]. The year 2022 brought several notable contributions. Wang *et al.* developed a Virtual-Goal-Guided RRT for mobile robot visual servoing under field-of-view constraints [42]. Lathrop *et al.* presented Wasserstein Safe RRT for distributionally robust path planning [43]. Chang *et al.* combined skeleton extraction and greedy algorithms with RRT for UAV trajectory tracking [23]. Ma *et al.* proposed a BiRRT framework for efficient motion planning in autonomous ground vehicles [44], while Li and Chen developed an adaptive RRT that adjusts its exploration strategy based on environmental complexity [45].

In 2023, research interest in RRT intensified. Zhang *et al.* developed an improved RRT for UAV trajectory planning in complex open storage environments [46]. Gai *et al.* proposed a memory-efficient RRT using adaptive thresholding [47], and Al-Ansarry *et al.* introduced a bi-directional probabilistic approach with triangular segmented interpolation for dynamic robot path planning [48]. Meanwhile, Pohan and Utama designed an efficient RRT-based planner for UAVs in dynamic environments [49]. In 2024, Liu *et al.* refined the artificial potential field model for UAV simulations by incorporating RRT [50]. In another work, Muling Liu *et al.* applied RRT alongside Ant Colony Optimization in MATLAB for 3D UAV path planning [62]. Sun *et al.* proposed Multi-Risk-RRT for motion planning of autonomous airport luggage trolleys [51].

By 2025, RRT had become a centerpiece in numerous research initiatives. Yao *et al.* integrated the artificial potential field with RRT to enhance path planning [52]. Fu *et al.* utilized RRT for fixed-wing UAV trajectory planning [53], while William *et al.* focused on safe navigation for urban air mobility [24]. Umer *et al.* emphasized safety and energy efficiency in UAV trajectory optimization using RRT [25]. Li *et al.* employed

a BiRRT for UAV navigation in complex 3D environments [54], and Zhao *et al.* addressed navigation challenges under varying visual and illumination conditions using RRT [55]. Wang *et al.* proposed a differential flatness-based trajectory planner using RRT for small-scale fixed-wing UAVs [56], while Qian and Wang explored RRT for optimizing trajectories in coal mining masonry robots [22].

Further contributions in 2025 included a hybrid deep learning model by Zhang *et al.* integrating RRT for UAVs in dynamic environments [57]. Halder and Orlando applied RRT in 3D path planning of steerable bevel-tip needles [21], and Yu *et al.* developed an RRT-based planner for parafoil homing under wind disturbances [58]. Lastly, Asimakopoulos and Chatzilygeroudis embedded RRT within a quality-diversity search framework to support kinodynamic trajectory optimization [59].

3.2. Simulation

The simulation ran in MATLAB R2024a on an Ubuntu 24.04 laptop with an AMD Ryzen 7 5800H and 8 GB RAM. We set the step size to 5 meters, the maximum iterations to 5000 per tree, the safety buffer radius to 10 meters, and the goal tolerance to 10 meters. Each algorithm ran 100 times from the same start point at 400,100 meters to the same goal at 100,400 meters. The map was a 500 by 500 meter occupancy grid built from real satellite data. The latitude ranged from -7.748382 to -7.740600 and the longitude from 110.237010 to 110.247996. We chose this region because it contains both buildings and open spaces, making it a realistic test environment for UAV path planning.

The map features several types of elements relevant to UAV navigation. As illustrated in Figure 1(a), highways are shown using blue lines, buildings are outlined in black, and other features such as vegetation or pathways are depicted in green. Among these, buildings are of primary importance due to their potential to obstruct flight paths and cause collisions. Therefore, they serve as the main constraint in our path planning objectives. To focus the analysis and simplify the computational process, we remove all non-essential elements from the map, such as highways and miscellaneous features. The resulting simplified map includes only the building outlines, as shown in Figure 1(b). This filtered version allows the UAV path planning algorithm to prioritize structural avoidance, improving the safety and efficiency of the planned routes.

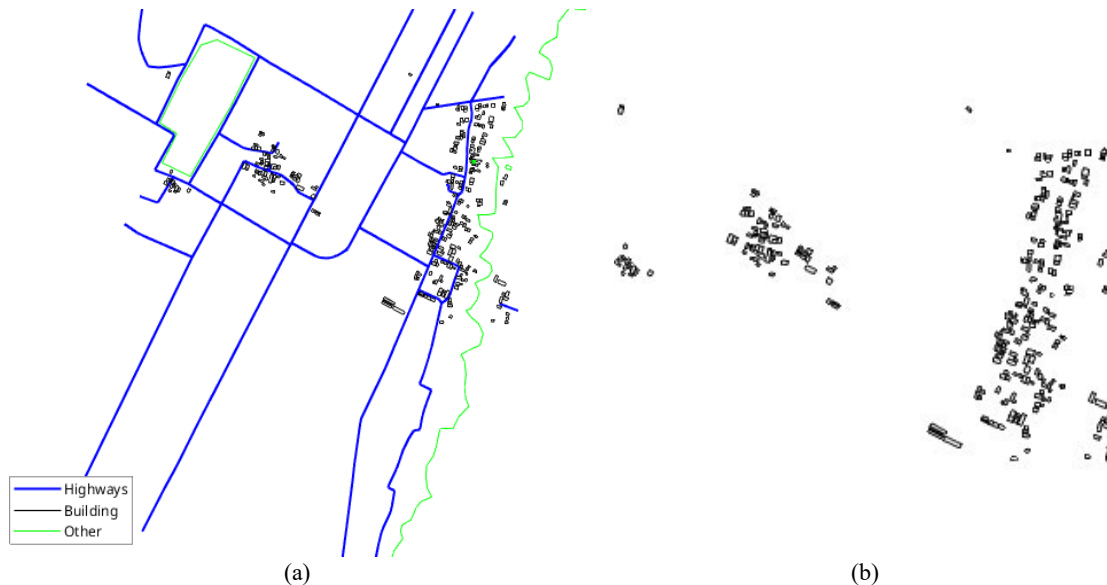


Figure 1. Real-world map for the test flight: (a) includes details on highways and other features, while (b) focuses solely on the buildings

After isolating the buildings, we expand the radius around each one by 10 meters to include an additional safety margin. Additionally, we add x -axes and y -axes, labeled in meters, to illustrate the distances between objects. The expanded area is shown in Figure 2. This updated occupancy grid will serve as the map for generating UAV waypoints. Since the occupancy grid map is already in Cartesian coordinates, we can directly initialize the start and goal positions as points on the x - and y -axes, such as:

$$\mathbf{start} = [400, 100]^T, \mathbf{goal} = [100, 400]^T$$

Here, Figure 3 illustrates the map with the start and goal coordinates. Next, the simulation will be conducted in MATLAB with 100 runs for both algorithms. The waypoints generated by each algorithm are shown in Figure 4. Figure 4 reveals a noticeable difference in waypoint distribution between the two algorithms. The waypoints generated by BiRRT tend to be more concentrated, whereas those from RRT are more dispersed, covering a wider area. While these observations are based on visual analysis, we will evaluate their performance statistically using parameters such as the success rate of reaching the goal, path length, clearance, and execution time. Based on 100 simulation runs, both algorithms successfully reached the goal with a 100% success rate. This indicates their potential for real-world applications, where they can reliably reach the goal while avoiding restricted regions with a safety radius of 10 meters. It is also worth noting that, despite the expansion of restricted areas, the map remains sufficiently sparse, making it easier for the algorithms to find waypoints. Although both algorithms achieve the same success rate in reaching the goal, their other parameters differ significantly, as illustrated in Figure 5.

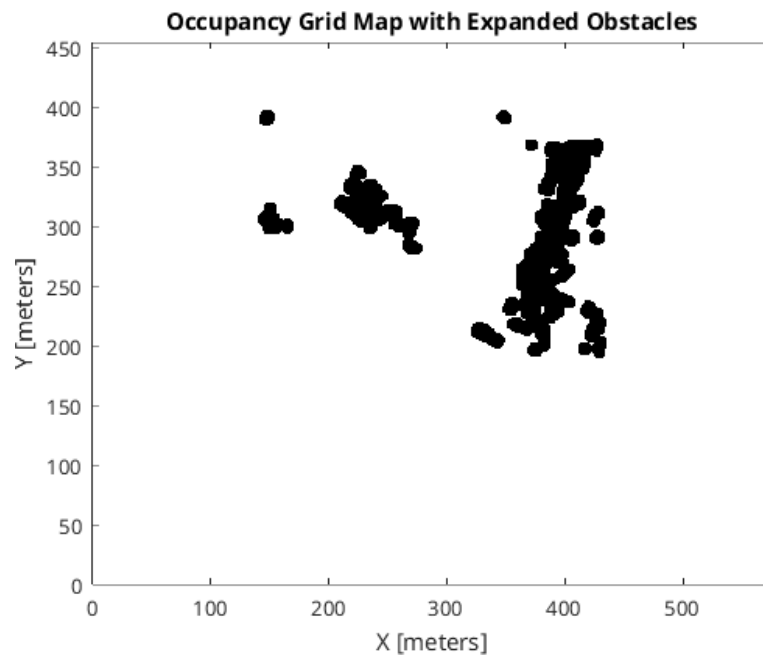


Figure 2. The occupancy grid map with expanded obstacles

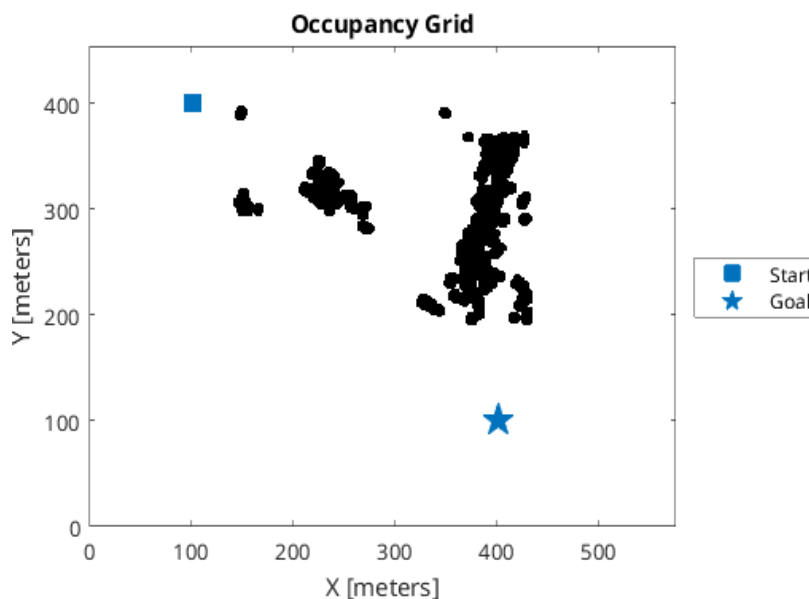


Figure 3. The occupancy grid map with expanded obstacles and start and goal coordinates

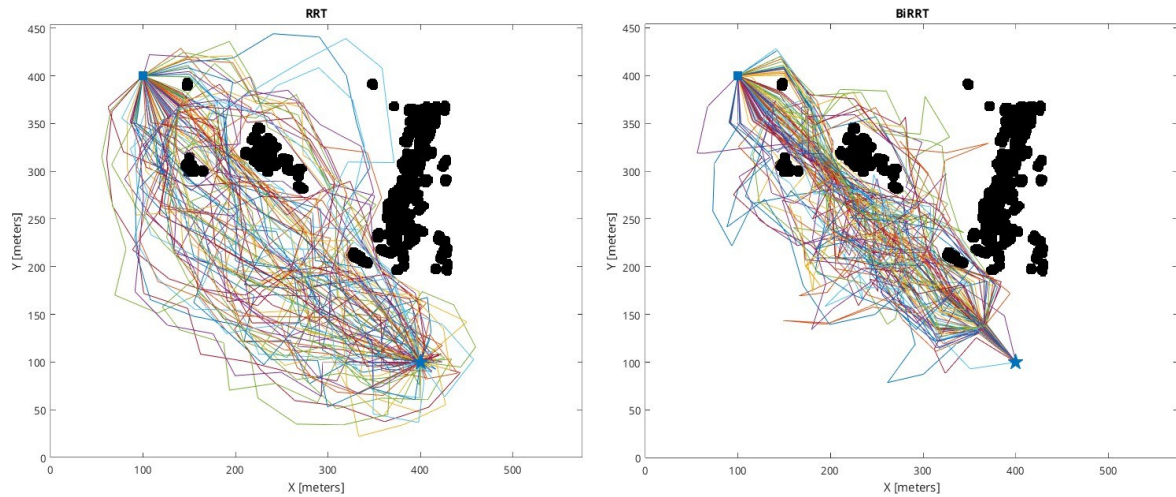


Figure 4. The waypoints generated by each algorithm

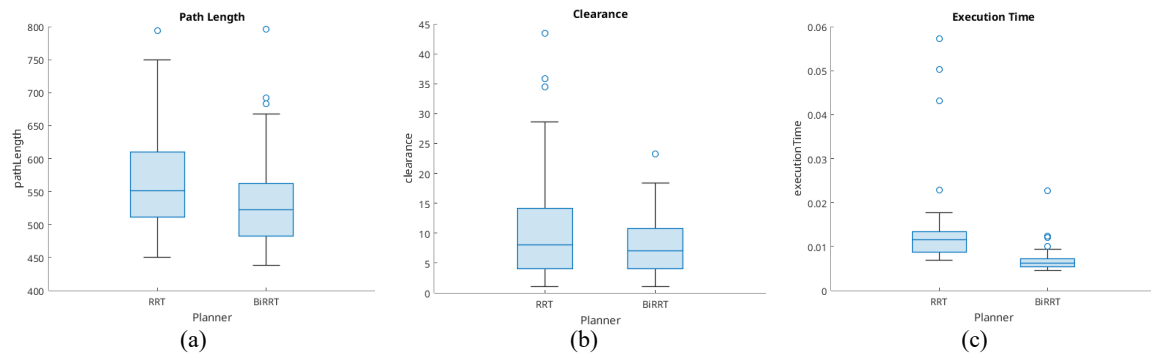


Figure 5. Box plot diagrams for: (a) path length, (b) clearance, (c) execution time

From Figure 5(a), it can be observed that the median and quartiles of the path length for BiRRT are shorter than those of RRT. However, BiRRT has more outliers (three) compared to RRT, which only has one. Next, in Figure 5(b), the clearance distance for RRT is higher than for BiRRT. This is consistent with Figure 4, which shows that the waypoints generated by RRT tend to be more widely spaced than those from BiRRT. This greater spacing results in larger clearance distances between waypoints and obstacles, ultimately leading to a longer generated path. Finally, in Figure 5(c), it can be seen that the execution time for RRT is longer than for BiRRT. Since BiRRT generates trees from both the start and goal simultaneously, it tends to complete the process more quickly than RRT. After visually analyzing the results, we can consider the statistics such as presented on Table 2 and Table 3.

Based on Table 2, the mean path length of the RRT algorithm is 561.1 meters, with a median of 550.78 meters and a standard deviation of 70.754 meters, indicating moderate variability in path distances. The clearance values have a mean of 11.114 meters, a median of 8.0311 meters, and a standard deviation of 9.5357 meters, reflecting variability in the distance between waypoints and obstacles. The execution time is consistent, with a mean of 0.012621 seconds, a median of 0.01161 seconds, and a standard deviation of 0.0073436 seconds. Next, we examine the statistics of the BiRRT algorithm, as shown in Table 3.

Comparing the data from Table 2 and Table 3 shows differences in the raw measurements. BiRRT has a mean path length of 534.47 meters, a median of 521.96 meters, and a standard deviation of 66.10 meters. RRT has a mean of 561.11 meters, a median of 550.78 meters, and a standard deviation of 70.75 meters. BiRRT produced shorter paths on average with slightly less spread in the data. For clearance, BiRRT has a mean of 7.69 meters, a median of 7.07 meters, and a standard deviation of 4.76 meters. RRT has a mean of 11.11 meters, a median of 8.03 meters, and a standard deviation of 9.54 meters. RRT keeps more distance from obstacles but shows wider variation. For execution time, BiRRT has a mean of 0.00667 seconds, a median of 0.00625 seconds, and a standard deviation of 0.00221 seconds. RRT has a mean of 0.01262 seconds, a median of 0.01161 seconds, and a standard deviation of 0.00734 seconds. BiRRT finished faster in every run. We report these numbers as measured from 100 runs per algorithm. We do not perform statistical significance tests here. The goal is to document the observed performance, not to make claims about population parameters.

These results show that BiRRT produced shorter path lengths and faster execution times than RRT in our 100 runs. If the priority is keeping distance from obstacles, RRT gave more clearance than BiRRT.

Table 2. The performance of RRT algorithm

Parameters	Path Length (meters)	Clearance (meters)	Time Execution (seconds)
Mean	561.1	11.114	0.012621
Median	550.78	8.0311	0.01161
Standard Deviation	70.754	9.5357	0.0073436

Table 3. The performance of BiRRT algorithm

Parameters	Path Length (meters)	Clearance (meters)	Time Execution (seconds)
Mean	534.47	7.6885	0.0066704
Median	521.96	7.0711	0.006248
Standard Deviation	66.102	4.761	0.0022105

4. CONCLUSIONS

This research article has presented a comparative analysis of the RRT and BiRRT algorithms for fixed-wing UAV path planning in urban environments. To enhance safety, the occupancy map used in the simulations incorporates an expanded safety margin by defining a 10-meter radius around each building, ensuring that the UAV avoids potential collisions with obstacles below. Both algorithms successfully generated collision-free paths in 100 out of 100 simulation runs. In our simulations, BiRRT produced shorter path lengths and faster execution times than RRT. RRT generated paths that maintained greater distances from obstacles. Therefore, BiRRT may be more suitable for missions where speed and shorter paths are priorities, while RRT may be the preferred choice when staying far from obstacles matters more.

In addition to the simulation study, this article has reviewed recent implementations of the RRT algorithm across various domains. These include applications in urban environments, in 2D and 3D space, as well as in cluttered areas for ground and aerial vehicles. The breadth of these applications shows that RRT-based methods are used widely in path planning tasks. This combination of a literature review and a simulation comparison provides material for researchers choosing between RRT and BiRRT for fixed-wing UAV operations.

REFERENCES

- [1] A. Cahyadi, A. P. Sandiwan, and S. Herdjunto, "An extended kalman filter for nonsmooth attitude control design of quadrotors using quaternion representation," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 10, no. 2, pp. 480–488, 2022, <https://doi.org/10.52549/ijeei.v10i2.3532>.
- [2] N. P. Santos, V. B. Rodrigues, A. B. Pinto, and B. Damas, "Automatic detection of civilian and military personnel in reconnaissance missions using a uav," in *2023 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 157–162, 2023, <https://doi.org/10.1109/ICARSC58346.2023.10129575>.
- [3] G. N. P. Pratama, A. Maarif, I. Iswanto, and E. W. Pratiwi, "Addressing model errors in uav altitude control using compensator," *Franklin Open*, vol. 11, 2025, <https://doi.org/10.1016/j.fraope.2025.100279>.
- [4] S. Herdjunto, "Actuator fault signal isolation of an unmanned aerial vehicle (uav) quadrotor based on detection filter," in *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pp. 1–6, 2016, <https://doi.org/10.1109/ICITEE.2016.7863232>.
- [5] L. Lasmadi, A. I. Cahyadi, S. Herdjunto, and R. Hidayat, "Inertial navigation for quadrotor using kalman filter with drift compensation," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 5, pp. 2596–2604, 2017, <https://doi.org/10.11591/ijece.v7i5.pp2596-2604>.
- [6] S. Herdjunto, A. Cahyadi, and B. R. Dewangga, "Actuator fault decoupled residual generation on lateral moving aircraft," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 16, no. 4, pp. 1886–1893, 2018, <https://doi.org/10.12928/TELKOMNIKA.v16i4.9055>.
- [7] K. Yamaguchi and S. Hara, "On structural parameter optimization method for quad tilt-wing uav based on indirect size estimation of domain of attraction," *IEEE Access*, vol. 10, pp. 1678–1687, 2022, <https://doi.org/10.1109/ACCESS.2021.3139156>.
- [8] D. Wada, S. Araujo-Estrada, and S. Windsor, "Sim-to-real transfer for fixed-wing uncrewed aerial vehicle: Pitch control by high-fidelity modelling and domain randomization," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 735–11 742, 2022, <https://doi.org/10.1109/LRA.2022.3205442>.
- [9] C. Kikumoto, T. Urakubo, K. Sabe, and Y. Hazama, "Back-transition control with large deceleration for a dual propulsion vtol uav based on its maneuverability," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 697–11 704, 2022, <https://doi.org/10.1109/LRA.2022.3205450>.
- [10] T. K. Priyambodo, A. Majid, and Z. S. S. Shouran, "Validation of quad tail-sitter vtol uav model in fixed wing mode," *Journal of Robotics and Control (JRC)*, vol. 4, no. 2, pp. 179–191, 2023, <https://doi.org/10.18196/jrc.v4i2.17253>.
- [11] K. A. Tsintotas, L. Bampis, A. Taitzoglou, I. Kansizoglou, P. Kaparos, C. Bliamis, K. Yakinthos, and A. Gasteratos, "The mpu rx-4 project: Design, electronics, and software development of a geofence protection system for a fixed-

- wing vtol uav,” *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–13, 2023, <https://doi.org/10.1109/TIM.2022.3225020>.
- [12] A. S. Prasetya, R.-J. Wai, Y.-L. Wen, and Y.-K. Wang, “Mission-based energy consumption prediction of multirotor uav,” *IEEE Access*, vol. 7, pp. 33 055–33 063, 2019, <https://doi.org/10.1109/ACCESS.2019.2903644>.
- [13] B. Talbot, F. Dayoub, P. Corke, and G. Wyeth, “Robot navigation in unseen spaces using an abstract map,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 4, pp. 791–805, 2021, <https://doi.org/10.1109/TCDS.2020.2993855>.
- [14] P. Corke, W. Jachimczyk, and R. Pillat, “Navigation,” *Springer Tracts in Advanced Robotics*, vol. 147, pp. 161–212, 2023, <https://doi.org/10.1007/978-3-031-07262-85>.
- [15] G. Airlangga, J. Bata, O. Nugroho, L. Sugianto, P. Saputro, S. Makin, and A. Alamsyah, “Enhanced advanced multi-objective path planning (eamopp) for uav navigation in complex dynamic 3d environments,” *International Journal of Robotics and Control Systems*, vol. 5, no. 2, pp. 713–729, 2025, <https://doi.org/10.31763/ijrcs.v5i2.1759>.
- [16] S. LaValle and J. Kuffner Jr., “Randomized kinodynamic planning,” *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001, <https://doi.org/10.1177/02783640122067453>.
- [17] S. M. LaValle, “Motion planning,” *IEEE Robotics & Automation Magazine*, vol. 18, no. 1, pp. 79–89, 2011, <https://doi.org/10.1109/MRA.2011.940276>.
- [18] M. A. R. Pohan, B. R. Trilaksono, S. P. Santosa, and A. S. Rohman, “Path planning algorithm using the hybridization of the rapidly-exploring random tree and ant colony systems,” *IEEE Access*, vol. 9, pp. 153 599–153 615, 2021, <https://doi.org/10.1109/ACCESS.2021.3127635>.
- [19] A. R. Pedram, R. Funada, and T. Tanaka, “Gaussian belief space path planning for minimum sensing navigation,” *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2040–2059, 2023, <https://doi.org/10.1109/TRO.2022.3228128>.
- [20] Y. Satake and H. Ishii, “Path planning method with constant bending angle constraint for soft growing robot using heat welding mechanism,” *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2836–2843, 2023, <https://doi.org/10.1109/LRA.2023.3260705>.
- [21] K. Halder and M. F. Orlando, “Path planning for robot assisted steerable bevel-tip needle in 3d dynamic environment,” *IEEE Robotics and Automation Letters*, vol. 10, no. 6, pp. 5887–5894, 2025, <https://doi.org/10.1109/LRA.2025.3562010>.
- [22] X. Qian and Y. Wang, “Path planning for coal mining masonry robots combined with trajectory optimization,” *IEEE Access*, vol. 13, pp. 24 197–24 206, 2025, <https://doi.org/10.1109/ACCESS.2025.3539023>.
- [23] J. Chang, N. Dong, D. Li, W. H. Ip, and K. L. Yung, “Skeleton extraction and greedy-algorithm-based path planning and its application in uav trajectory tracking,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 6, pp. 4953–4964, 2022, <https://doi.org/10.1109/TAES.2022.3198925>.
- [24] B. A. William, D. Udayachandran, S. Gowripriya, and S. H. Shivakumar, “Path planning and safe navigation strategy for urban air mobility,” vol. 3259, no. 1, 2025, <https://doi.org/10.1063/5.0263842>.
- [25] M. M. Umer, H. Venter, O. Muhammad, T. Shafique, F. A. Awwad, and E. A. Ismail, “Cognitive strategies for uav trajectory optimization: Ensuring safety and energy efficiency in real-world scenarios,” *Ain Shams Engineering Journal*, vol. 16, no. 3, 2025, <https://doi.org/10.1016/j.asej.2025.103301>.
- [26] Y. Iwashina, M. Kunibe, S. Kato, and H. Shigeno, “Uav path planning in urban environments with dynamic risk-map generation by vehicle and pedestrian perception,” in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, pp. 1–5, 2022, <https://doi.org/10.1109/VTC2022-Fall57202.2022.10012880>.
- [27] M. A. R. Pohan, B. R. Trilaksono, S. P. Santosa, and A. S. Rohman, “Path planning using combined informed rapidly-exploring random tree star and particle swarm optimization algorithms,” *IEEE Access*, vol. 12, pp. 56582–56608, 2024, <https://doi.org/10.1109/ACCESS.2024.3389152>.
- [28] J. M. Kelner, W. Burzynski, and W. Stecz, “Modeling uav swarm flight trajectories using rapidly-exploring random tree algorithm,” *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 1, 2024, <https://doi.org/10.1016/j.jksuci.2023.101909>.
- [29] B. Lindqvist, A. Patel, K. Lo’fgren, and G. Nikolakopoulos, “A tree-based next-best-trajectory method for 3-d uav exploration,” *IEEE Transactions on Robotics*, vol. 40, pp. 3496–3513, 2024, <https://doi.org/10.1109/TRO.2024.3422052>.
- [30] S. LaValle and J. Kuffner, “Randomized kinodynamic planning,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, 1999, pp. 473–479, <https://doi.org/10.1109/ROBOT.1999.770022>.
- [31] J. Kuffner and S. LaValle, “Rrt-connect: An efficient approach to single-query path planning,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, pp. 995–1001, 2000, <https://doi.org/10.1109/ROBOT.2000.844730>.
- [32] S. M. LaValle. *Planning algorithms*. Cambridge University Press. 2006. <https://doi.org/10.1017/CBO9780511546877>.
- [33] H. Mawatari, T. Uchiya, and I. Takumi, “An exploration algorithm suitable for indoor mapping based on rrt-exploration,” in *2023 IEEE 12th Global Conference on Consumer Electronics (GCCE)*, pp. 316–317, 2023, <https://doi.org/10.1109/GCCE59613.2023.10315525>.
- [34] H. Suwoyo, A. Adriansyah, J. Andika, A. U. Shamsudin, and Y. Tian, “An effective way for repositioning the beacon nodes of fast rrt results utilizing grey wolf optimization,” *Journal of Robotics and Control (JRC)*, vol. 6, no. 1, pp. 272–284, 2025, <https://doi.org/10.18196/jrc.v6i1.22062>.

- [35] A. H. Qureshi and Y. Ayaz, "Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments," *Robotics and Autonomous Systems*, vol. 68, pp. 1–11, 2015, <https://doi.org/10.1016/j.robot.2015.02.007>.
- [36] E. Dönmez, A. F. Kocamaz and M. Dirik, "Bi-rrt path extraction and curve fitting smooth with visual based configuration space mapping," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1–5, 2017, <https://doi.org/10.1109/IDAP.2017.8090214>.
- [37] T. Rybus, J. Prokopczuk, M. Wojtunik, K. Aleksiejuk, and J. Musiał, "Application of bidirectional rapidly exploring random trees (birrt) algorithm for collision-free trajectory planning of free-floating space manipulator," *Robotica*, vol. 40, no. 12, pp. 4326–4357, 2022, <https://doi.org/10.1017/S0263574722000935>.
- [38] I. A. Hassan, I. A. Abed, and W. A. Al-Hussaibi, "Path planning and trajectory tracking control for two-wheel mobile robot," *Journal of Robotics and Control (JRC)*, vol. 5, no. 1, pp. 1–15, 2024, <https://doi.org/10.18196/jrc.v5i1.20489>.
- [39] J.-H. Jhang and F.-L. Lian, "An autonomous parking system of optimally integrating bidirectional rapidly-exploring random trees and parking-oriented model predictive control," *IEEE Access*, vol. 8, pp. 163 502–163 523, 2020, <https://doi.org/10.1109/ACCESS.2020.3020859>.
- [40] A. T. Khan, S. Li, S. Kadry, and Y. Nam, "Control framework for trajectory planning of soft manipulator using optimized rrt algorithm," *IEEE Access*, vol. 8, pp. 171 730–171 743, 2020, <https://doi.org/10.1109/ACCESS.2020.3024630>.
- [41] C. Mthabela, D. Withey and C. Kuchwa-Dube, "Ground Robot Path Planning on 3D Mesh Surfaces Using Bi-directional RRT based on Local Regions," in *2021 Rapid Product Development Association of South Africa - Robotics and Mechatronics - Pattern Recognition Association of South Africa (RAPDASA-RobMech-PRASA)*, pp. 01-06, 2021, <https://doi.org/10.1109/RAPDASA-RobMech-PRAS53819.2021.9829138>.
- [42] R. Wang, X. Zhang, Y. Fang, and B. Li, "Virtual-goal-guided rrt for visual servoing of mobile robots with fov constraint," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 4, pp. 2073–2083, 2022, <https://doi.org/10.1109/TSMC.2020.3044347>.
- [43] P. Lathrop, B. Boardman and S. Martínez, "Distributionally safe path planning: Wasserstein safe rrt," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 430–437, 2022, <https://doi.org/10.1109/LRA.2021.3128696>.
- [44] H. Ma, F. Meng, C. Ye, J. Wang, and M. Q.-H. Meng, "Bi-risk-rrt based efficient motion planning for autonomous ground vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 722–733, 2022, <https://doi.org/10.1109/TIV.2022.3152740>.
- [45] 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, <https://doi.org/10.1109/JAS.2021.1004252>.
- [46] J. Zhang, Y. An, J. Cao, S. Ouyang, and L. Wang, "Uav trajectory planning for complex open storage environments based on an improved rrt algorithm," *IEEE Access*, vol. 11, pp. 23 189–23 204, 2023, <https://doi.org/10.1109/ACCESS.2023.3252018>.
- [47] R. Gai, X. Wang, and K. Wang, "Efficient-low memory path planning algorithm based on adaptive thresholding," *IEEE Access*, vol. 11, pp. 81 378–81 388, 2023, <https://doi.org/10.1109/ACCESS.2023.3300244>.
- [48] S. Al-Ansary, S. Al-Darraj, A. Shareef, D. G. Honi, and F. Fallucchi, "Bi-directional adaptive probabilistic method with a triangular segmented interpolation for robot path planning in complex dynamic-environments," *IEEE Access*, vol. 11, pp. 87 747–87 759, 2023, <https://doi.org/10.1109/ACCESS.2023.3290897>.
- [49] M. A. Rajasa Pohan and J. Utama, "Efficient path planning algorithm for unmanned aerial vehicle in dynamic environment," in *2023 International Conference on Informatics Engineering, Science & Technology (INCITEST)*, pp. 1–7, 2023, <https://doi.org/10.1109/INCITEST59455.2023.10397062>.
- [50] J. Liu, Y. Yan, Y. Yang, and J. Li, "An improved artificial potential field uav path planning algorithm guided by rrt under environment-aware modeling: Theory and simulation," *IEEE Access*, vol. 12, pp. 12 080–12 097, 2024, <https://doi.org/10.1109/ACCESS.2024.3355275>.
- [51] Z. Sun, B. Lei, P. Xie, F. Liu, J. Gao, Y. Zhang, and J. Wang, "Multi-risk-rrt: An efficient motion planning algorithm for robotic autonomous luggage trolley collection at airports," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 2, pp. 3450–3463, 2024, <https://doi.org/10.1109/TIV.2023.3349171>.
- [52] Z. Yao, Z. Liu, and C. Han, "The improved rrt integrated with the artificial potential field path planning algorithm," *IEEE Access*, vol. 13, pp. 68 398–68 409, 2025, <https://doi.org/10.1109/ACCESS.2025.3561348>.
- [53] Q. Fu, J. Li, and W. Zhao, "Motion primitives and sample-based trajectory planning for fixed-wing unmanned aircraft," *Transactions of the Institute of Measurement and Control*, vol. 47, no. 9, pp. 1873–1882, 2025, <https://doi.org/10.1177/01423312241264873>.
- [54] N. Li and S. I. Han, "Adaptive bi-directional rrt algorithm for three-dimensional path planning of unmanned aerial vehicles in complex environments," *IEEE Access*, vol. 13, pp. 23 748–23 767, 2025, <https://doi.org/10.1109/ACCESS.2025.3537697>.
- [55] L. Zhao, W. Wang, Q. He, L. Yan, and X. Li, "Visual-inertial autonomous uav navigation in complex illumination and highly cluttered under-canopy environments," *Drones*, vol. 9, no. 1, 2025, <https://doi.org/10.3390/drones9010027>.
- [56] Y. Wang, W. Zeng, Y. Peng, Q. Yang, and J. Zhou, "Differential flatness-based trajectory planning for small fixed-wing uavs," *Lecture Notes in Electrical Engineering*, vol. 1377 LNEE, pp. 360–369, 2025, https://doi.org/10.1007/978-981-96-3568-9_34.

- [57] J. Zhang, Y. Xian, X. Zhu, and H. Deng, "A hybrid deep learning model for uav path planning in dynamic environments," *IEEE Access*, vol. 13, pp. 67 459–67 475, 2025, <https://doi.org/10.1109/ACCESS.2025.3557394>.
- [58] Z. Yu, H. Sun, Q. Sun, P. Tan, Z. Chen, and J. Qu, "Dynamic path planning for parafoil homing in surface wind disturbance environment," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 15 045–15 059, 2025, <https://doi.org/10.1109/TASE.2025.3563071>.
- [59] K. A. Asimakopoulos and K. I. Chatzilygeroudis, "Integrating trajectory optimization in quality-diversity for kinodynamic motion planning," in *2025 11th International Conference on Automation, Robotics, and Applications (ICARA)*, 2025, pp. 388–393, <https://doi.org/10.1109/ICARA64554.2025.10977687>.
- [60] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014, <https://doi.org/10.1109/ACCESS.2014.2302442>.
- [61] D. Belter, "Efficient modeling and evaluation of constraints in path planning for multi-legged walking robots," *IEEE Access*, vol. 7, pp. 107 845–107 862, 2019, <https://doi.org/10.1109/ACCESS.2019.2933178>.
- [62] M. Liu, J. Liu, Q. Lan, Z. Lu, W. Zhang, and G. Zou, "Matlab simulation of uav 3d path planning research based on aco algorithm, astar algorithm and rrt algorithm," in *2024 3rd International Symposium on Semiconductor and Electronic Technology (ISSET)*, pp. 634–637, 2024, <https://doi.org/10.1109/ISSET62871.2024.10779618>.

AUTHOR BIOGRAPHY

Gilang Nugraha Putu Pratama is a robotics enthusiast who previously worked as an assistant researcher at the Department of Electrical Engineering and Information Technology at Universitas Gadjah Mada, and as a control engineer at Beehive Drones. During his career, he has published more than 20 conference papers and 6 journal articles, primarily focused on robotics and control systems. Can be reached at email: gilang.n.p.pratama@uny.ac.id.

Oktaf Agni Dhewa holds a Master's degree in Computer Science from Universitas Gadjah Mada. His research focuses on UAVs (Unmanned Aerial Vehicles) and control systems. During his research, he has published more than five articles indexed in Scopus.

Mentari Putri Jati received the B.S. and M.S. degrees in electrical engineering from Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia, in 2018 and 2020, respectively. She is currently pursuing the Ph.D. degree in Electrical Engineering and Computer Science with the National Taipei University of Technology, Taipei, Taiwan. Her research interests include electric drives, electrical machines, power electronics, and optical fiber sensors.

Indra Hidayatulloh is currently pursuing the Ph.D. degree in School of Computing Engineering and Built Environment, Glasgow Caledonian University. His research interests include A.I especially machine learning.

Teddy Surya Gunawan received his B.Eng degree in Electrical Engineering with cum laude award from Institut Teknologi Bandung (ITB), Indonesia in 1998. He obtained his M.Eng degree in 2001 from the School of Computer Engineering at Nanyang Technological University, Singapore, and PhD degree in 2007 from the School of Electrical Engineering and Telecommunications, The University of New South Wales, Australia. His research interests are in speech and audio processing, biomedical signal processing and instrumentation, image and video processing, and parallel computing. He is currently an IEEE Senior Member (since 2012), was chairman of IEEE Instrumentation and Measurement Society – Malaysia Section (2013 and 2014), Professor (since 2019), Head of Department (2015-2016) at Department of Electrical and Computer Engineering, and Head of Programme Accreditation and Quality Assurance for Faculty of Engineering (2017-2018), International Islamic University Malaysia. He is Chartered Engineer (IET, UK) and Insinyur Profesional Madya (PII, Indonesia) since 2016, and registered ASEAN engineer since 2018.

Syaiful Ardy Gunawan holds a Master's degree in Electrical Engineering from Universitas Gadjah Mada. His research focuses on path planning for robotics applications, with publications indexed in Scopus. He is currently an engineer at Epson Seiko Corporation.