



## Article

# Fast and Efficient Drone Path Planning Using Riemannian Manifold in Indoor Environment

Rohit Dujari <sup>1</sup>, Brijesh Patel <sup>1,2,\*</sup>  and Bhumeshwar K. Patle <sup>3</sup> 

<sup>1</sup> Department of Mechanical Engineering, School of Engineering and Information Technology, MATS University, Raipur 493441, Chhattisgarh, India; rohitdujari@gmail.com

<sup>2</sup> Department of Mechanical Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan

<sup>3</sup> Department of Mechanical Engineering, School of Engineering and Sciences, MIT Art, Design, and Technology University, Pune 412201, Maharashtra, India; balu\_patle@rediffmail.com

\* Correspondence: aero.brijesh@gmail.com; Tel.: +91-9926119575

**Abstract:** This paper introduces an innovative dual-path planning algorithm rooted in a topological three-dimensional Riemannian manifold (T3DRM) to optimize drone navigation in complex environments. It seamlessly integrates strategies for both discrete and continuous obstacles, employing spherical navigation for the former and hyperbolic paths for the latter. Serving as a transformative tool, the T3DRM facilitates efficient path planning by transitioning between discrete and continuous domains. In uncertain environments with unpredictable obstacle positions, our methodology categorizes these positions as discrete or continuous based on their distribution patterns. Discrete obstacles exhibit random distributions, while continuous obstacles display symmetrical patterns with continuity. Leveraging topological metrics, the T3DRM efficiently classifies these patterns for effective path planning. The findings of this research demonstrate the efficiency of path planning based on classified obstacle positions, enabling swift and efficient drone navigation. This research introduces a pioneering application of a T3DRM, accelerating drone navigation in uncertain environments through a dual approach that simultaneously transforms navigation in primal and dual domains. By enabling spherical and hyperbolic navigation concurrently, the T3DRM offers a comprehensive solution to discrete and continuous path planning challenges. The proposed approach can be used for various indoor applications, especially for warehouse management, surveillance and security, navigation in complex structures, indoor farming, site inspection, healthcare facilities, etc.

**Keywords:** drone navigation; path planning; Riemannian manifold; topology



**Citation:** Dujari, R.; Patel, B.; Patle, B.K. Fast and Efficient Drone Path Planning Using Riemannian Manifold in Indoor Environment. *Automation* **2024**, *5*, 450–466. <https://doi.org/10.3390/automation5030026>

Academic Editor: Eyad H. Abed

Received: 31 May 2024

Revised: 11 September 2024

Accepted: 12 September 2024

Published: 15 September 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

A drone, also known as an unmanned aerial vehicle (UAV), is an aircraft that can fly without the assistance of a human pilot. Drones can be remotely controlled by a pilot on the ground or programmed to fly autonomously using pre-programmed waypoints and instructions. Drones range in size and shape from small handheld models to large commercial drones used for industrial purposes. They are outfitted with sensors and cameras that allow them to capture high-quality photos and videos from the air, and they can also be used for a variety of other tasks, such as land surveying, infrastructure inspection, package delivery, and even emergency medical services. Drones have grown in popularity in recent years due to their versatility, ease of use, and low cost. They have the potential to revolutionize many industries and are expected to play a growing role in transportation, logistics, and many other fields in the future. Drone path planning is the act of identifying the best route or path for a drone to take to reach its destination or fulfill a specified mission. Path planning employs algorithms and sensors to maneuver the drone around the surroundings while avoiding obstacles and keeping on course. The path planning procedure normally begins with identifying the drone's beginning and ending

points, as well as any intermediate waypoints that it must reach. The planner then develops a path that minimizes the time, energy, or distance required to complete the objective while considering any environmental limits or safety concerns. Drone path planning must take various things into account, such as the drone's capabilities, the area in which it will operate, and the mission objectives. For example, if the drone is surveying a big region, the path planner may need to optimize the path to cover the entire area effectively while avoiding obstacles such as buildings, trees, or power lines. Overall, path planning is an important component of drone operation since it ensures that the drone can navigate safely and effectively to its intended goals [1].

Because of the nature of their operation in airspace, drone navigation involves several challenges [2], including object avoidance, localization, GPS interference, weather conditions, and hardware interfacing. Drone navigation involves careful planning, reliable sensors, and sophisticated algorithms to overcome the challenges of flying in the sky. As technology progresses, we may see more advanced navigation systems that can overcome these issues and allow drones to operate safely and efficiently in a range of environments [3]. There are numerous methods available for drone navigation, including Dijkstra's Algorithm [4], the A\* Algorithm [5], a Rapidly Exploring Random Tree Algorithm [6,7], Graph-based Algorithms [8], Artificial Potential Fields [9], Genetic Algorithms [10], the Dynamic-Window Approach [11], Ant Colony Optimization [12], Particle Swarm Optimization [13], Simulated Annealing [14], the Harmony Search Algorithm [15], a Probabilistic Roadmap [16], the Visibility Graph Algorithm [17], Model Predictive Control [18], Gradient-Based Motion Planning [19], Differential Evolution [20], Artificial Bee Colony [21], the Cuckoo search Algorithm [22], the Firefly Algorithm [23], and the Mayfly Algorithm. Different bio-inspired algorithms, such as the Mayfly and Firefly algorithms, have primarily focused on 2D navigation problems. These navigation algorithms are computer programs that enable drones to navigate independently. However, with the increasing complexity of real-world environments, there has been a growing demand for 3D navigation approaches to allow drones to operate more efficiently in three-dimensional spaces, such as urban areas, indoor environments, and densely populated regions.

The use of Riemannian manifolds [24] in the field of robotics has become an area of increasing interest in recent years [25]. Riemannian manifolds provide a powerful mathematical framework for analyzing and controlling the motion of drones, which can enable them to perform complex tasks such as surveillance, delivery, and search and rescue operations. Firstly, Riemannian manifolds are a type of geometric space used to describe curved surfaces. In drone navigation, Riemannian manifolds are often used to represent the configuration space of the drone, which is the set of all possible configurations that the drone can reach. By using Riemannian manifolds, we can analyze the motion of the drone as it moves through this configuration space and determine the optimal path for the drone to reach a desired goal [26]. One of the key applications of Riemannian manifolds in drone navigation is path planning. Path planning is the process of determining a sequence of motions that will enable a drone to move from its current state to a desired goal state while avoiding obstacles and minimizing the cost of the motion. Riemannian manifolds provide a powerful framework for path planning by enabling us to define a cost function on the configuration space and optimize this cost function using Riemannian optimization techniques [27]. Several recent research papers have explored the use of Riemannian manifolds in the field of aerial navigation. For example, Chahrour et al. [28] contributed significantly to the field of signal processing by exploring the application of Riemannian geometry in the context of drone detection. By demonstrating the superiority of Riemannian-based methods over traditional techniques, the study opens new possibilities for enhancing drone surveillance, security, and remote sensing systems. Beik-Mohammadi et al. [29] presented an innovative approach to equip robots with the ability to learn and adapt motion skills using a Riemannian manifold. The utilization of geodesics as natural motion skills enables efficient planning and obstacle avoidance while also empowering the robot to generate novel movements in unexplored scenarios. This advancement holds promise for enhancing

the capabilities of robots in real-world settings, paving the way for more seamless human–robot collaboration in diverse and dynamic environments. Biess et al. [30] presented a robotics system focusing on anthropomorphic hands. The researchers transformed the dimension and basis of the anthropomorphic hand structure into the Riemannian metric. This transformation allowed them to apply manifold classification to the symmetric and asymmetric patterns of particle positions. The use of the Riemannian metric in the context of anthropomorphic hands provided a novel approach to capturing the intricate movements and configurations of the hand. By leveraging the manifold application, the system was able to effectively classify and analyze both symmetric and asymmetric patterns of particle positions within the hand structure. Jaquier et al. [31] developed geometry-aware Bayesian optimization for robotics. The researchers introduced techniques to compute geometry-aware kernels on different Riemannian manifolds commonly found in robotic applications. An examination of the simplification of general expressions for Matérn kernels using stochastic partial differential equations on these manifolds resulted in explicit formulas for geometry-aware kernels. The derived formulas provide practitioners with practical tools to apply geometry-aware Bayesian optimization in novel and diverse settings. This enables roboticists to consider the underlying geometry of the robotic system, leading to more efficient and effective optimization strategies.

There are several papers in the existing literature on the path planning of drones using Riemannian manifolds for indoor environments, but these works are computationally intensive and may not be scalable for large indoor environments with complex layouts. The previous technique must be updated to handle larger situations with an acceptable level of accuracy. Many existing works are intended for offline planning and may not be appropriate for real-time path planning. Algorithms that can generate pathways in real time must be modified. Most of the current literature on Riemannian manifolds deals with static obstacles; hence, the performance of Riemannian manifolds need to be tested in indoor environment for dynamic obstacles. It has also been found that many Riemannian manifold-based studies are not compared to a pre-existing algorithm to compare performance. Hence, this study presents the fastest and most efficient navigating approach for drones in an interior setting with static and dynamic environments by exploiting the Riemannian manifold. In addition, this work performs real-time path planning of drones in an uncertain environment by comparing the existing relevant literature. Einstein's general theory of relativity is based on Riemannian geometry. The theory of gravity is simplified with the Riemannian metric by transforming  $n$ -dimensional parameters into two-dimensional ones. The use of Riemannian manifolds makes it feasible to measure space–time curvature and massive objects. Riemannian geometry serves as a revolutionary tool for designing efficient and compact path-planning algorithms, utilizing the characteristics of curvature and manifolds. Key advantages in robot navigation, such as fast obstacle avoidance and shortest path selection, are achieved through the Riemannian metric approach [32]. Other applications of Riemannian geometry include statistical data-based robot navigation, optimal path planning algorithms, and machine learning-based robot navigation [33].

This paper is structured as follows: Section 2 presents the mathematical framework of Riemannian manifolds as it applies to drone path planning. Section 3 details the simulation and experimental evaluation of the proposed method within an indoor setting, with findings and discussion provided in Section 4. Lastly, Section 5 concludes this study and outlines potential future research directions.

## 2. Riemannian Manifolds for Path Planning of Drone

This paper presents a novel approach to achieve the fastest and most efficient navigating strategy for drones in indoor environments, leveraging the power of Riemannian manifolds. Riemannian manifolds are utilized in robotics and control theory to model robot configuration spaces, which are often non-Euclidean due to motion constraints. By defining a metric tensor on the configuration space, essential geometric properties like distance, curvature, and geodesics can be established. These notions play a crucial role in robot

motion planning and control on curved surfaces. The use of Riemannian manifolds offers several advantages for drone navigation. Firstly, they can capture the inherent curvature and topology of complex environments, making them ideal for modeling intricate settings. Moreover, the geometry of Riemannian manifolds remains invariant under coordinate transformations, ensuring robustness in path-planning algorithms, even when the drone's orientation changes during the mission. Another significant benefit is the ability to express optimization problems with geometric constraints, facilitating the development of path-planning algorithms that consider complex environments. Riemannian manifolds also introduce the concept of parallel transport, which enables vectors or frames to be carried along a path while preserving their direction. This feature proves valuable in developing obstacle avoidance algorithms, where the drone needs to maintain a specific orientation while avoiding obstacles. Furthermore, Riemannian manifolds provide a strong foundation for constructing feedback control algorithms based on environment geometry. This is essential in scenarios where precise path-following or orientation control is required, such as navigating along a desired path while avoiding obstacles. The technical introduction to path planning of drones using a Riemannian manifold is given as follows.

The proposed drone is three-dimensional (3D), and it is designed to achieve various objectives in the path planning of the drone. As a 3D study, it is closely related to topology, which provides the necessary 3D variants. The drone's navigation system incorporates the characteristics of topological data analysis on Riemannian manifolds. It is a subdomain of topology, which exhibits dual characteristics over manifolds, namely spherical and hyperbolic. These dual characteristics are effectively applied in the proposed drone's navigation system. The drone's path planning algorithm is designed to interact with uncertain environments where obstacles are present in different positions. These positions can either be randomly distributed or follow specific patterns, leading to two distinct classes, i.e., discrete for random positions and continuous for patterns. The drone's navigation strategy is driven by the classification of obstacle positions. By following the spherical and hyperbolic duality-based path planning algorithm, the drone efficiently navigates through uncertain environments. This duality decision allows for fast navigation, as the path planning is based on classified positions of obstacles, enabling compact decision making instead of individual vertex-by-vertex decisions. Furthermore, the efficiency advantage is achieved through the utilization of Ricci flow curvature, which is a defining characteristic of T3DRMs. Some applications of T3DRMs include studying space through its structure, dimensions, and gravitation, designing spacecraft-robot hybrids for generating automatic decisions, creating compact satellites using topological transformation, developing drones for medical surgery that work with 3D data based on manifolds, and producing micro-macro drones that utilize Riemannian metrics for diagnosis. The drone's navigation is transformed into a topological data analysis of the Riemannian manifolds' curvatures, and as a result, the drone automatically accelerates without encountering any obstacle challenges. The acceleration controller relies on the spherical and hyperbolic curvature over the Ricci flow curvature, ensuring the drone's efficient navigation by following the curvature. The terms required in the mathematical modeling are explained below.

1. Topology: A topological space is an ordered pair  $(X, T)$ , where  $X$  is a non-empty set and  $T$  is a subset of  $X$  that satisfies the following properties:
  - I.  $\emptyset, X \in T$ ;
  - II. If  $A, B \in T$  then  $A \cap B \in T$ ;
  - III. If  $T_1, T_2, \dots, T_n \in T$  then  $T_1 \cup T_2 \dots \cup T_n \in T$ .

The collection  $T$  is referred to as a topology acting on  $X$ . The elements of  $T$  are called the open sets.

- 2 Topological Data Analysis on Riemannian Manifolds:  $(M, g_p)$ , where  $M$  is a real and smooth manifold, and  $g_p$  is the positive definite inner product on the tangent space  $T_pM$  at each point  $p$ .

- 3 Manifold: A topological space is said to be a manifold if there exists a locally resembling Euclidean space at each point, e.g., circle, sphere, parabola, hyperbola, or other curves.
- 4 Ricci Flow Equation:  $\frac{d}{dt}g_{ij}(t) = -2R_{ij}$ , where  $R$  is a Riemannian metric defined by  $g_{ij}(t)$ .
- 5 Riemannian Matrix: A Riemannian matrix  $g$  on the smooth manifold  $M$  is a smooth family of inner products on the tangent spaces of  $M$ . Its bilinear form on  $T_pM$ ,  $p \in M$  is defined by

$$g_p : T_pM \times T_pM \rightarrow R$$

We next consider the function

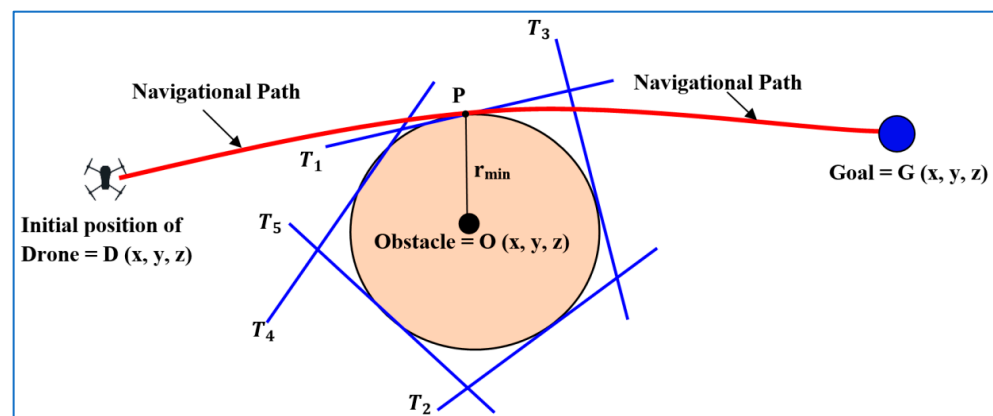
$$M \rightarrow g_p(x_p, y_p) \in R,$$

where the vector fields  $(x, y) \in M$  are given by  $M$ , also Riemannian manifold is defined as a pair  $(M, g)$ , where  $M$  is a differentiable manifold and  $g$  is a Riemannian metric on  $M$ .

Drone navigation follows the manifold characteristics classified into linear, spherical, and hyperbolic. Linear navigation refers to 2D; thus, its manifolds have a constant value. In the linear trajectory, the drone projects the goal using the Euclidean metric. Three-dimensional navigation exhibits both spherical and hyperbolic characteristics depending on the axioms of obstacle environments (discrete, continuous, and mixed), which is demonstrated as follows:

**Spherical Navigation:** The property of obstacles to be discretely mapped within the domain of a sphere. Thus, the spherical manifolds avoid the obstacles by utilizing the discrete distribution function.

Let  $D \equiv D(x, y, z)$  be the drone,  $G \equiv G(x, y, z)$  be the goal and  $O(x, y, z) \equiv \{o_1(x, y, z), \dots, o_n(x, y, z)\}$  be the finite set of obstacles in the three-dimensional space.  $D$  has a sensor for identifying the obstacles and measuring the distance from the drone to the obstacle. Thus, the challenge exists to reach a goal through optimal path planning. The drone's spherical navigation helps avoid discrete obstacles. Consider a sphere with a center ( $o$ ) and radius ( $r$ ) defined by  $S(o, r)$ . There is a set of tangents of a sphere. The drone selects these tangents as the navigation path  $DP$ . At point  $P$ , the drone measures the distance using sensor  $PG$  and reaches goal  $G$ . The drone follows Equation (1) for this optimal navigation. Figure 1 represents the trajectory planning through the spherical navigation:



**Figure 1.** Spherical navigation by avoiding discrete obstacles.

The optimal path planning comprises the speed of the drone and the fast avoidance of obstacles. This objective is achieved by Hamilton's Ricci flow (HRF) [34]. HRF is an observation of the 3D transformation, e.g., from a sphere into a cylinder and vice versa. This provides a set of options to select the optimal structure, and its application appears in drone navigation. Next, Hamilton's Ricci flow is defined.

Hamilton's Ricci Flow: Let  $M$  be a Riemannian Ricci flow with metric  $g_0$ . The Ricci flow is a partial differential equation that evolves the metric tensor:

$$\frac{\partial}{\partial t} g(t) = -2Rc g(t); g(0) = g_0 \quad (1)$$

where a Ricci curvature of the metric with respect to time  $t$  is denoted as  $Rc g(t)$ , and its solution is

$$r(t) = \sqrt{r_0^2 - 2(n-1)t} \quad (2)$$

where  $r_0$  is the initial radius of the sphere's center ' $r$ '. For the hyperbolic  $n$ -space  $H^n$ , the Ricci flow can be simplified to the ordinary differential equation (ODE)

$$\frac{d}{dt} (r^2) = 2(n-1) \quad (3)$$

Then, its solution is given as

$$r(t) = \sqrt{r_0^2 + 2(n-1)t} \quad (4)$$

The general solution is  $u = \{x_u, y_u, z_u\}$

$$\frac{\partial}{\partial t} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{x_u^3 + y_u^3 + z_u^3} \begin{bmatrix} z_u^3 & & \\ & y_u^3 & \\ & & x_u^3 \end{bmatrix} \begin{bmatrix} x_{uuu} \\ y_{uu\mu} \\ z_{uuu} \end{bmatrix} \quad (5)$$

where  $x_u = \frac{\partial x}{\partial u}$ ,  $y_u = \frac{\partial y}{\partial u}$ , and  $z_u = \frac{\partial z}{\partial u}$ . The parameter of the left-hand side defines the velocity of the drone, and the right-hand side defines the three-dimensional manifolds. The drone ( $D$ ) navigates towards the goal ( $G$ ), and if there is no obstacle, then  $D$  reaches  $G$ . Mathematically, it can be said that  $D$  and  $R$  are simultaneous. If there is any obstacle, then  $D$  avoids it by measuring the distance by metric  $g(t)$ . It treats the obstacle as the center and the radius  $r$  of the sphere. On the sphere, there exists a finite set of tangents  $\{T_1, \dots, T_n\}$ . The drone selects the tangent with minimum distance. By following this tangent, the drone reaches the surface of the sphere, and, hence, obstacles are avoided. If there are more such obstacles, then the drone follows the same process to avoid these. The topological navigation is presented in Figure 2 based on hyperbolic geometry. There are a finite number of obstacles referred to as continuous obstacles. When applying the spherical navigation method,  $n$  spheres and  $n$  tangents will result from the  $n$  obstacles, making efficient navigation impossible. The selection of optimal tangents is challenging. Hyperbolic navigation will be feasible and efficient in this case. The drone selects point  $P$  in the hyperbola and follows the path  $DP$  for optimal navigation.

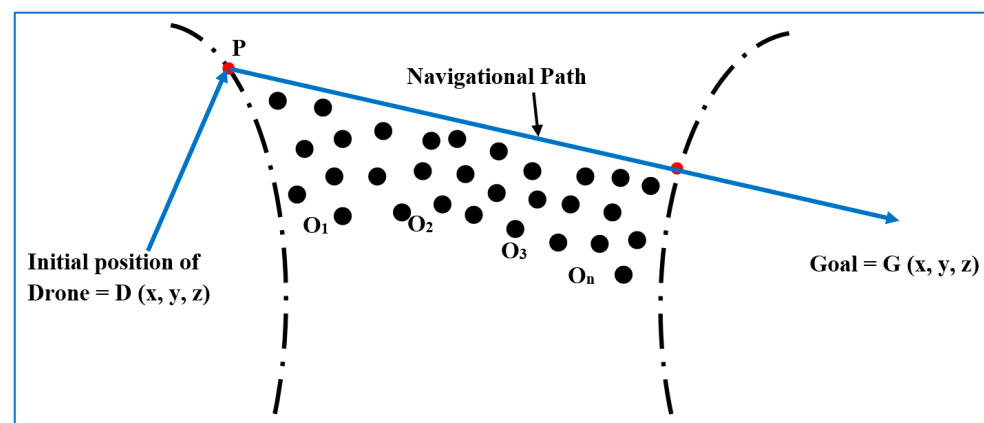


Figure 2. Hyperbolic navigation by avoiding continuous obstacles.

The next navigation will be based on the path PG. The curvature exists in the environment contained with discrete and continuous obstacles. This is called the hyperbolic navigation. This hyperbolic navigation is described as follows:

Hyperbolic Navigation: If the distribution of obstacles is continuous, then the continuous distribution function-based hyperbolic manifolds exist. There is no obstacle in the path PG. Thus, the drone reaches goal G. The transformation from a sphere into a hyperbola and vice versa by HRF follows the following hyperbolic navigation:

The existing obstacles are classified into two classes: single obstacles and multiple obstacles, with both having discrete, continuous, and mixed distribution functions. The discrete distribution of obstacles is observed in a spherical trajectory, and the continuous distribution of obstacles follows the principles of hyperbolic manifolds. However, the mixed (discrete and continuous) distribution of obstacles follows the tangent line axioms, which implies that a tangent line simultaneously separates the trajectory for spherical and hyperbolic navigation. The single obstacle is avoided by tangent sequences. If multiple obstacles exist, then by following the hyperbolic navigation with Equations (3) and (4), optimal path planning is generated. Let the displacement of the drone be  $x$ ; then, the velocity will be  $\frac{dx}{dt}$  and the acceleration will be  $\frac{d^2x}{dt^2}$ . The direction of the drone follows the tangent line, but the displacement towards the tangent line follows the following rules as shown in Table 1.

**Table 1.** Rules for displacement.

S.N.	Direction	$(x_0)$	$(y_0)$	$(x_0, y_0)$
1	Front	$+R^*$	$+R^*$	$(+R^*, +R^*)$
2	Back	$-R^*$	$-R^*$	$(-R^*, -R^*)$
3	Right	$+R^*$	$-R^*$	$(+R^*, -R^*)$
4	Left	$-R^*$	$+R^*$	$(-R^*, +R^*)$
5	Up	$+R^*$	0	$(0, +R^*)$
6	Down	0	$-R^*$	$(0, -R^*)$

Where  $+R^*$  denotes a set of non-zero real positive numbers, and  $-R^*$  denotes a set of non-zero negative real numbers. The drone follows the direction obtained in the pair. If the pair is  $(+R^*, +R^*)$ , then the drone will move in a front direction because this basis provides a linear combination with coefficients expressed as positive real numbers. If the pair is  $(-R^*, -R^*)$ , then the drone will move in a backward direction because this provides a linear combination with coefficients expressed as negative real numbers. If the first coordinate is a positive real number and the second one is a negative real number  $(-R^*, +R^*)$ , then the drone will move towards the left. The drone's upward direction is represented by the coordinates  $(-R^*, +R^*)$ ; the second coordinate is zero, resulting in linear independence with the non-zero positive real number coefficient in the first coordinate. If the first coordinate is zero, then linear independence also exists with the second coordinate's non-zero coefficient. This will ensure that the drone moves in a downward direction.

The volume of the sphere plays a crucial role in controlling the speed of the drone as it is directly proportional to the volume of the sphere. The volume of the sphere is  $V$ , and its normalized (volume-preserving) Ricci flow equation is presented by Equation (6).

$$\frac{\partial}{\partial t}g(t) = -2Rc + \frac{2}{n} \frac{\int_{M^n} R dV}{\int_{M^n} dV} g, \quad (6)$$

where  $M^n$  is a topological space with  $n$ -manifolds,  $g$  is a Riemannian metric,  $Rc$  is the Ricci Tensor, and  $R$  is the scalar curvature. For three-manifolds,  $n = 3$ , then (6) becomes

$$\frac{\partial}{\partial t}g(t) = -2Rc + \frac{2}{n} \frac{\int_{M^3} R dV}{\int_{M^3} dV} g \quad (7)$$

In Table 2, the first row displays the maximum speed of the drone using the second-order partial derivative and exhibits the negative values that occur when the specified input of the first-order partial derivative is applied. The drone defines its minimum speed as the point at which the first-order partial derivative is zero, and the second-order partial derivative is positive. The speed of the drone will be considered constant when the speed command is received. Otherwise, the speed is not specified by either zero or alternate values.

**Table 2.** Speed controller decision.

$(V, dV)$	$\frac{\partial}{\partial t}g = -2Rc + \frac{2}{n} \frac{\int_{M^3} R dV}{\int_{M^3} dV} g$	Speed
$(V_1, +R^*)$	$(-R^*, -R^*)$	Maximum
$(V_1, -R^*)$	$(+R^*, +R^*)$	Minimum
$(V_1, 0)$ and otherwise	$(0, 0)$	0

In this work, minimizing the computational time is the prime objective concerning path planning from the initial position to the final position, taking into account various constraints and environmental factors. Factors such as the complexity of the environment, the number of obstacles, the position of an obstacle, the dimensionality of search space, and computational power are considered for obtaining an optimal path with less computational time as described in Algorithm 1. By balancing the trade-off between speed and path optimality, effective path planning algorithms aim to minimize this time while maintaining safety and adherence to mission criteria, as illustrated in the flow chart in Figure 3. The complete pseudocode is described in Algorithm 2.

---

**Algorithm 1:** Riemannian manifolds computational time algorithm

---

1. Drone starts navigating in the environment and time starts.

2.  $D \rightarrow \{O_1, O_2, O_3, \dots, O_n\} \rightarrow G$ .

If

$$D \rightarrow \{O_1\} \rightarrow G.$$

Then,

$$\frac{\partial}{\partial t} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \lim_{t \rightarrow 1^+} r(t) = \lim_{t \rightarrow 1^+} \sqrt{R_0^2 - 2(n-1)t} = \sqrt{R_0^2 - 2n + 2}. \quad (8)$$

If

$$D \rightarrow \{O_1, O_2, O_3, \dots, O_n\} \rightarrow G.$$

Then,

$$\frac{\partial}{\partial t} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \lim_{t \rightarrow n^+} r(t) = \lim_{t \rightarrow n^+} \sqrt{R_0^2 - 2(n-1)t} = \sqrt{R_0^2 - 2n^2 + 2n}. \quad (9)$$

3. Total computational time = Equation (7) + . . . . . + Equation (7).

4. Optimal time =  $\min \left\{ \frac{\partial}{\partial t} g(t) \right\} = \min \{-2R_c g(t)\}$ .

5. Path completed and complete time is recorded.

---



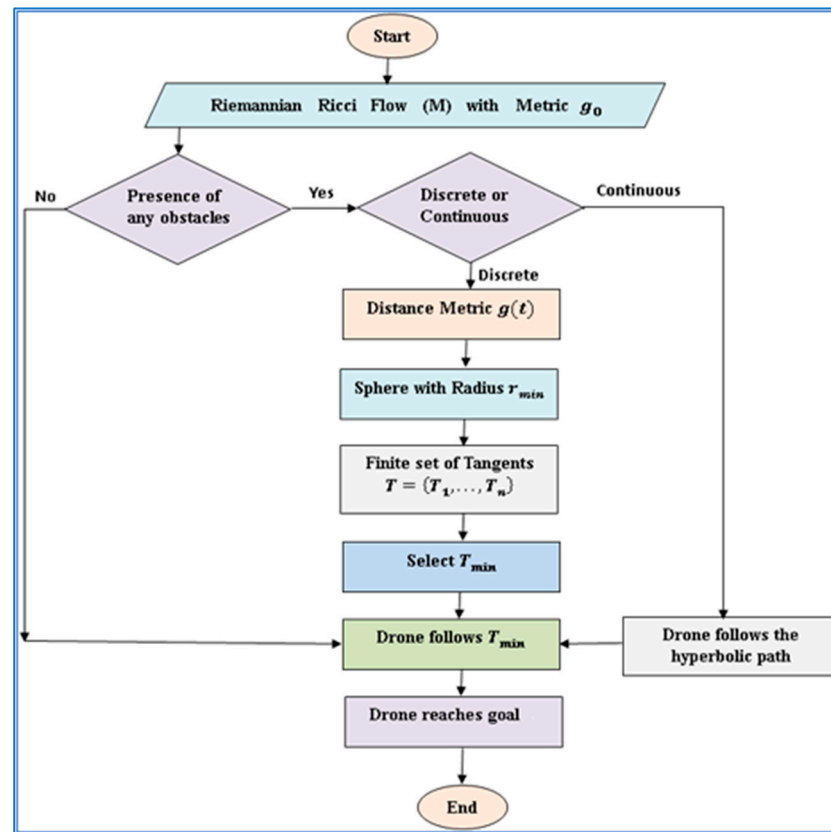


Figure 3. Flow chart of Riemannian manifold method.

#### Algorithm 2: Pseudocode

1. Drone starts.
2. Riemannian Ricci flow  $M$  with metric  $g_0$ .
3. If any obstacle:  
No; go to (12).  
Yes;  
if discrete, then go to (4).  
If continuous, then go to (10).
4. Distance metric  $g(t)$ .
5. Consider a sphere with center (obstacle) and radius ( $r_{min}$ ).
6. Finite set of tangents  $T = \{T_1, \dots, T_n\}$ .
7. Select ( $T_{min}$ ).
8. Drone follows ( $T_{min}$ , go to (3)).
9. An obstacle is avoided.
10. If multiple objects exist,  
then, drone follows the hyperbolic path through  $\frac{\partial}{\partial t}g(t) = -2Rc g(t); g(0) = g_0$  and its velocity is set by  $\frac{d}{dt}(r^2) = 2(n - 1)$ .
11. Go to (3).
12. Drone reaches goal.
13. Drone stops.

### 3. Simulation and Experimental Analysis

In this simulation study, we evaluated the performance of a T3DRM to solve path planning optimization problems for drones in static environments. The path planning approach was designed with the main objective of achieving optimal path planning from the drone's initial position to a target destination while also ensuring path safety through obstacle avoidance. Visualization and statistical analysis of the simulation results were

tested using MATLAB R2022b software. The programming for drone navigation was conducted in a 2D rectangular space measuring 600 cm by 600 cm while considering multiple static obstacles with different shapes and orientations. An assessment of drone control parameters when encountering obstacles was conducted through simulation analysis. By simulating various obstacle avoidance scenarios, researchers can assess how different control parameters impact the drone's performance, and this allows for informed decision making when selecting optimal parameter values for real-time experimentations. Altering the parameters of the T3DRM provides valuable insights into its effectiveness in path planning and obstacle avoidance. The drone parameters, such as position control, altitude control, and velocity control, are shown in Table 3, with specified ranges for  $k_p$  (Proportional Gain),  $k_d$  (Derivative Gain), and  $k_i$  (Integral Gain). In this simulation study, these control parameters were carefully adjusted to ensure stable and responsive flight dynamics. Position control used  $k_p$  for quick adjustments,  $k_i$  for correcting accumulated errors, and  $k_d$  for minimizing oscillations. Altitude control relied on  $k_p$  to maintain the desired height,  $k_i$  to correct long-term deviations, and  $k_d$  to smooth vertical transitions. Velocity control involved  $k_p$  for rapid speed adjustments,  $k_i$  for consistent velocity tracking, and  $k_d$  for dampening abrupt changes. By varying these gains within their specified ranges, the simulation identified optimal settings that achieved precise and reliable control, mirroring real-world flight conditions and enhancing the drone's overall performance.

**Table 3.** Drone parameters and their ranges.

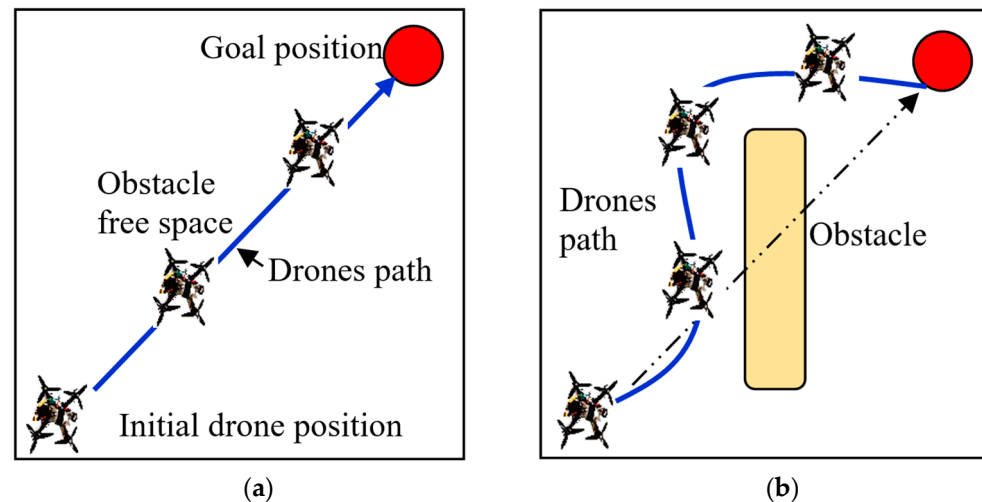
Drone Parameters	$k_p$ (Range)	$k_i$ (Range)	$k_d$ (Range)
Position Control	0–100	0–1	0–10
Altitude Control	0–10	0–1	0–10
Velocity Control	0–10	0–1	0–10

A specially designed drone with six well-placed infrared sensors was used for the real-time analysis as shown in Appendix A. This allowed for obstacle recognition and avoidance within a 2 to 100 cm range. A GPS sensor also made sure the drone stayed stable and oriented while in flight. Four blades on the drone provided thrust for flying and maneuverability. An onboard minicomputer handled operations, ran flight control algorithms, and processed sensor data. Pitch, roll, throttle, and yaw were basic motions that allowed the drone to perform a variety of orientations and maneuvers. The drone's navigation in a real-time interior environment with well-placed obstacles is shown graphically in Figure 4a,b. When there were no obstacles in the way, the drone moved on its own and did not need the T3DRM-based decision system to guide it toward its target, as shown in Figure 4a. On the other hand, this system was activated when obstacles were identified, ensuring that the drone's navigation was effectively and safely planned around them. This demo took place in a small area with obstacles placed to meet certain needs, which effectively highlighted the drone's versatility in real-world situations. The drone's navigation path from its original starting point to its intended destination is shown in Figure 4. This image demonstrates how the drone navigated its surroundings on its own, intelligently avoiding obstacles and precisely aiming for its target.

The drone's navigation programming used Equation (10), the Euclidean formula, to determine the shortest path between its starting position and the destination. The drone used this formula to find the navigation angle ( $\theta$ ) and then followed a straight line to the goal position, as shown in Figure 4. When there were no obstacles in the area, as shown in Figure 4a, the drone easily reached the desired position without having to activate the suggested approach. However, when more obstacles were present, as shown in Figure 4b, the drone used the same navigational strategy detailed in Figure 4a. However, the drone stopped moving forward when it sensed an obstacle. It then initiated the T3DRM (three-dimensional route mapping) mechanism to circumvent the obstacle and continue traveling in the direction of the target place, creating an optimized path in the process. While there

were many possible ways to reach the destination, the main goal of this study was to find the optimal path in order to cut traversal time. This emphasis highlights how crucial efficacy and efficiency are in real-world drone navigation situations. Here,  $(X_D, Y_D)$  and  $(X_G, Y_G)$  were the drone's initial positions and the destination, respectively.

$$\text{Euclidean distance} = \sqrt{(X_G^2 - X_D^2) + (Y_G^2 - Y_D^2)} \quad (10)$$



**Figure 4.** Drone navigation in (a) in the absence of an obstacle and (b) in the presence of an obstacle.

#### 4. Results and Discussion

Over twenty trials with a maximum velocity of 0.2 m/s were carried out on the drone in both simulated and real-time scenarios for this study. The drone's real-time navigation and its navigation in the 2D simulation environment is shown in Figures 5–8. Scenario 1 involves comparing the virtual navigation of the drone in simulation (Figure 6) with the real-time environment (Figure 5). In Scenario 2, Figures 7 and 8 are compared in obstacle-contained environments. The drone's starting location, the goal position, the number of obstacles, and the obstacle's position are all similar in the environments taken into consideration for the simulation analysis and the experimental setup. The goal of the analysis is to compare the path's length and the amount of time needed for navigation when the drone plans its route from the starting point to the destination. It is evident from the results in Tables 4 and 5 (Scenarios 1 and 2) that the path length and navigational time needed in simulation results are significantly shorter than in experiment results. Our primary goal is to minimize the percentage of the difference between path length and navigational time when comparing the outcomes of simulations and experiments. Based on the collected data, the deviation percentage in both cases is less than 5.0%, indicating that the suggested T3DRM-based controller is quite effective.

The methodology offered by other drone navigation researchers is compared with the suggested T3DRM-based strategy in Scenarios 3, 4, and 5. A potential field approach is employed in one such study [35] to guide drones in indoor spaces, especially when they come across unknown objects. Figure 9a,c show how the drone navigates using this technique. As seen in Figure 9b,d, a fresh simulation setup is made with the same environmental parameters as those considered in the cited work to conduct a comparative analysis. This covers elements like the quantity, size, and locations of obstacles in addition to the drone's starting and finishing places. Similarly, the proposed approach is also compared with Ant Colony Optimization (ACO) [36] in a similar simulation environment that includes multiple obstacles (Figure 9e,f). It is clear from the numerical results for Scenarios 3, 4, and 5 (Tables 6 and 7) that using the T3DRM technique leads to notable gains. In Scenarios 3, 4, and 5, the T3DRM technique delivers path length reductions of up

to 7.52%, 9.98%, and 2.0, respectively. Furthermore, using the T3DRM technique results in navigational time savings of 5.4%, 8.6%, and 7.04 in Scenarios 3, 4, and 5, respectively.

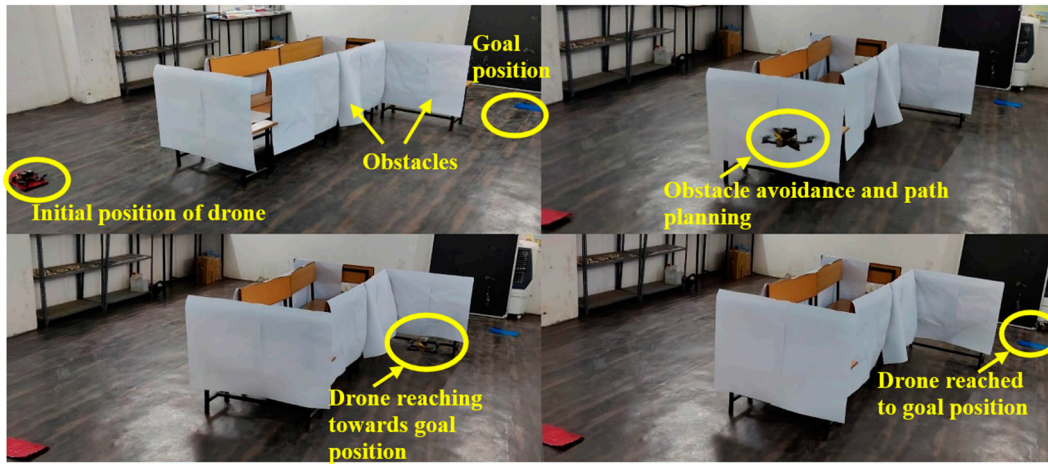


Figure 5. Real-time navigation of drone in the presence of obstacles(Scenario-1).

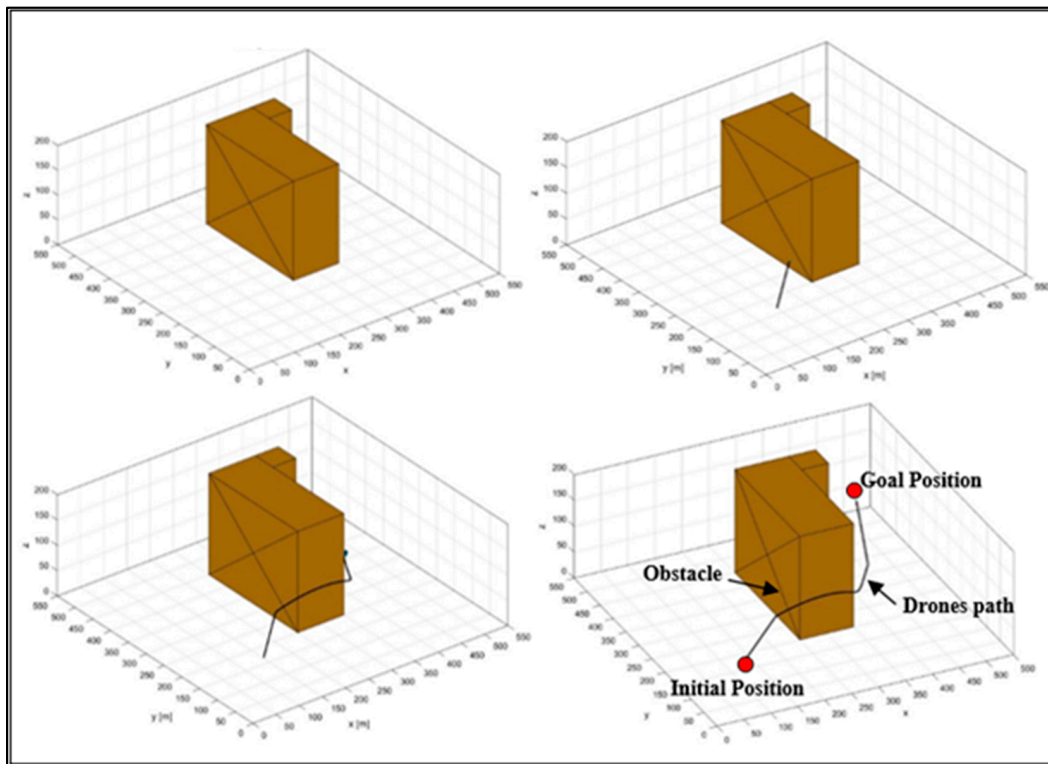


Figure 6. Navigation of drone in a simulation environment (Scenario-1).

Table 4. Drone’s experimental path length vs. simulation path length.

S. N.	Drone’s Path Length in Simulation (cm)	Drone’s Path Length in Real-Time (cm)	% of Deviation
Scenario 1	613.89 (Figure 6)	644.14 (Figure 5)	4.69
Scenario 2	761.32 (Figure 8)	799.57 (Figure 7)	4.78

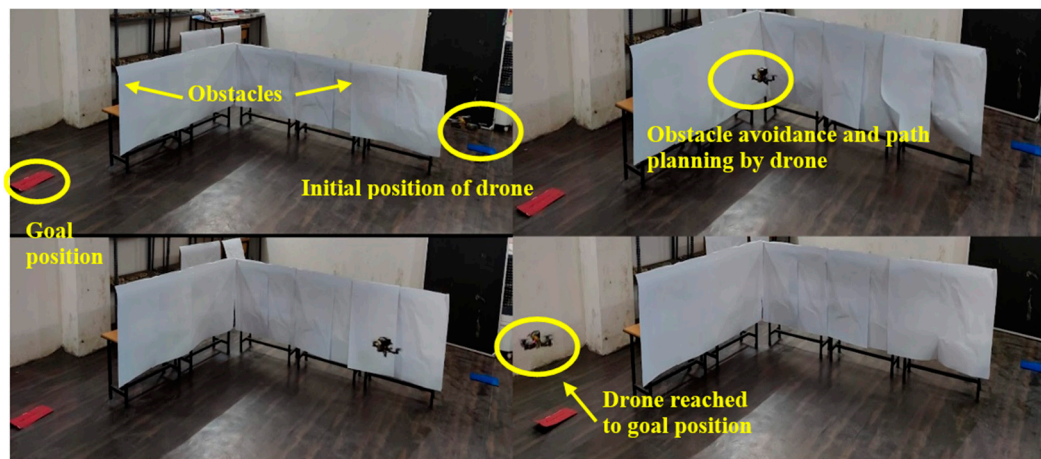


Figure 7. Real-time navigation of drone in the presence of obstacles (Scenario-2).

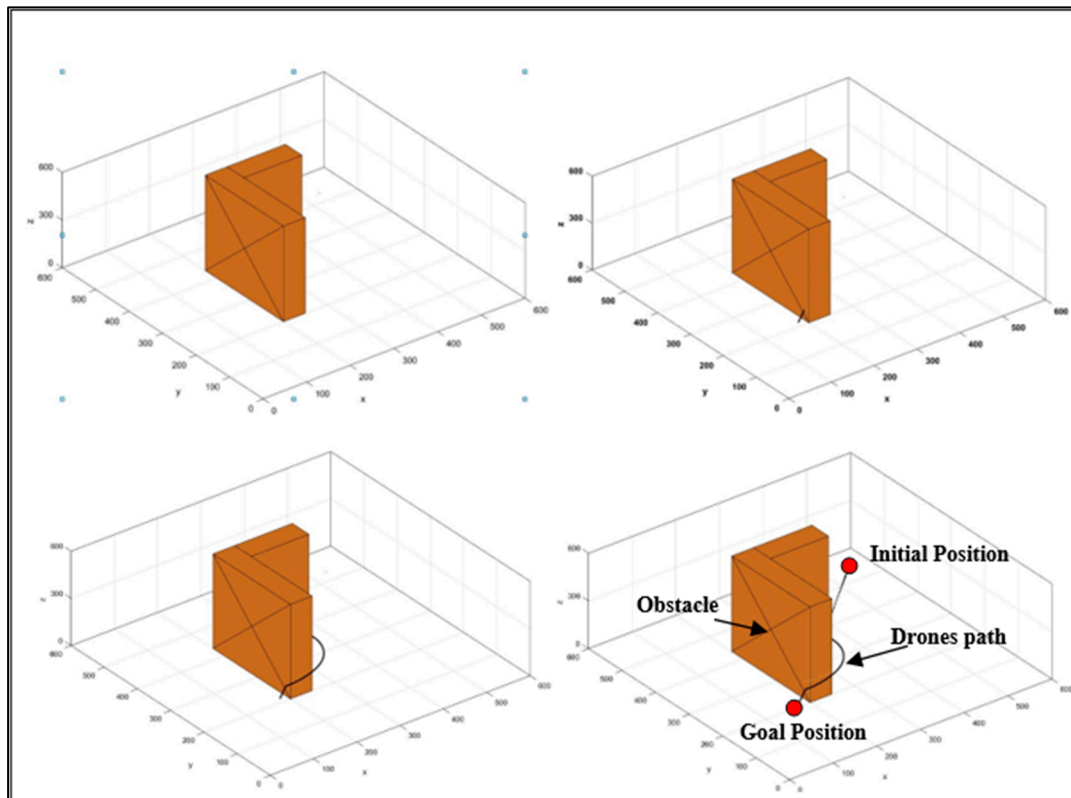
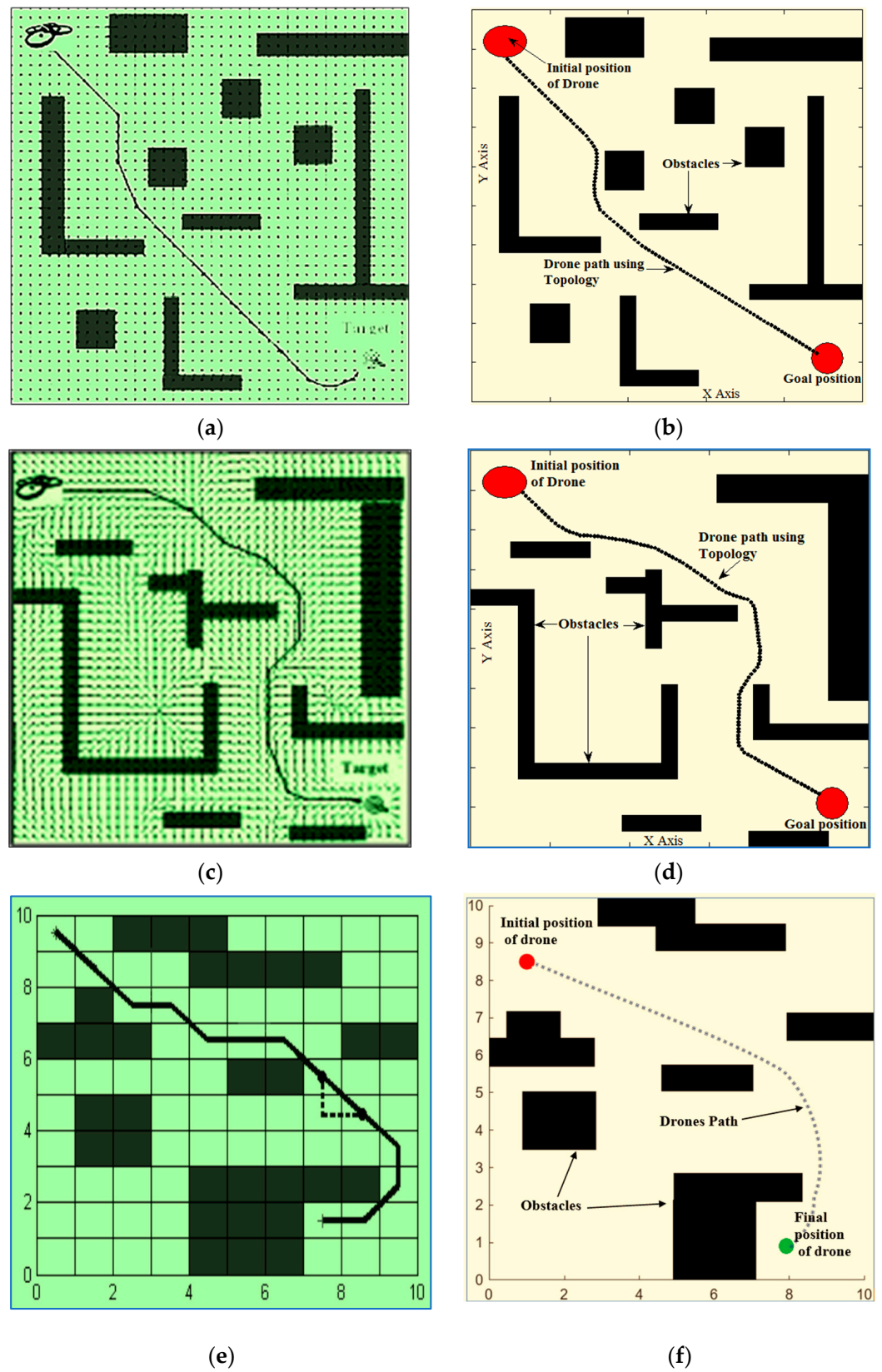


Figure 8. Navigation of drone in a simulation environment (Scenario-2).

Table 5. Drones experimental time vs. simulation time.

S. N.	Navigational Time in Simulation (s)	Navigational Time in a Real-Time Experiment (s)	% of Deviation
Scenario 1	31.33 (Figure 6)	32.9 (Figure 5)	4.77
Scenario 2	35.63 (Figure 8)	37.5 (Figure 7)	4.98



**Figure 9.** Comparative analysis (a,b) T3DRM technique vs. potential field method for scenario 3, (c,d) T3DRM technique vs. potential field method for scenario 4, (e,f) T3DRM technique vs. ACO for scenario 5.

**Table 6.** Path length comparison with other researchers' work.

S.N.	Initial Position	Final Position	Path Length Using the T3DRM Approach (cm)	Path Length Using Other Approaches (cm)	% of Path Length Saved by the T3DRM Approach
Scenario 3	(8, 88)	(87, 7)	130.25 (Figure 9b)	140.04 (Figure 9a)	7.52
Scenario 4	(12, 90)	(87, 7)	139.5 (Figure 9d)	153.42 (Figure 9c)	9.98
Scenario 5	(1, 9)	(7.5, 1.5)	14.6 (Figure 9f)	15.47 (Figure 9e)	2.0

**Table 7.** Navigational time comparison with other researchers' work.

S.N.	Initial Position	Final Position	Navigational Time Using the T3DRM Approach	Navigational Time Using Other Approaches	% of Time Saved by the T3DRM Approach
Scenario 3	(8, 88)	(87, 7)	7.3 (Figure 9b)	7.7 (Figure 9a)	5.4
Scenario 4	(12, 90)	(87, 7)	8.1 (Figure 9d)	8.8 (Figure 9c)	8.6
Scenario 5	(9, 9)	(7.5, 1.5)	7.9 (Figure 9f)	7.04 (Figure 9e)	6

## 5. Conclusions

In conclusion, the use of Riemannian manifolds in drone navigation has opened up a new avenue for analyzing and controlling the motion of drones. By using Riemannian manifolds, we can develop more efficient and robust path planning and control algorithms that can enable drones to perform complex tasks in a wide range of environments. The literature available on Riemannian manifolds in drone navigation is growing rapidly, and it is an exciting area of research with many promising applications. In conclusion, the integration of Riemannian manifold theory into drone navigation systems offers significant benefits, including accurate modeling of complex environments, robustness against orientation changes, and efficient obstacle avoidance capabilities. The proposed approach presents a promising avenue for optimizing drone navigation in uncertain indoor environments, paving the way for enhanced autonomy and reliability in drone operations. The effectiveness of the T3DRM-based strategy for optimal path planning in the presence of maximal uncertainty is confirmed by the real-time results obtained from the drone used in-house and the simulation results obtained using MATLAB. The findings obtained in both the simulation and real-time environments demonstrate a high degree of adherence to the navigational parameters with the lowest possible percentage of deviation (less than 5%). Additionally, a performance gain over comparable environmental conditions is revealed when comparing the suggested controller with the other intelligent controllers, such as the potential field approach and Ant Colony Optimization. This study might be expanded in the future to include multi-drone path planning over uncertain terrain. The proposed algorithm can be used for submersible, humanoid, and wheeled robot navigation.

**Author Contributions:** Conceptualization and supervision, B.P. and B.K.P.; methodology and validation, B.P. and R.D.; software and investigation, R.D.; writing—original draft preparation, B.P. and R.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A



**Figure A1.** Drone's sensor placements, (1–6) are representing the IR sensor placement.

## References

1. Suzuki, S. Recent researches on innovative drone technologies in robotics field. *Adv. Robot.* **2018**, *32*, 1008–1022. [\[CrossRef\]](#)
2. Gagan, G.; Haque, A. Path planning for autonomous drones: Challenges and future directions. *Drones* **2023**, *7*, 169. [\[CrossRef\]](#)
3. Aggarwal, S.; Kumar, N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Comput. Commun.* **2020**, *149*, 270–299. [\[CrossRef\]](#)
4. Wang, J.; Li, Y.; Li, R.; Chen, H.; Chu, K. Trajectory planning for UAV navigation in dynamic environments with matrix alignment Dijkstra. *Soft Comput.* **2022**, *26*, 12599–12610. [\[CrossRef\]](#)
5. Mandloi, D.; Arya, R.; Verma, A.K. Unmanned aerial vehicle path planning based on A\* algorithm and its variants in 3d environment. *Int. J. Syst. Assur. Eng. Manag.* **2021**, *12*, 990–1000. [\[CrossRef\]](#)
6. Yin, C.; Lin, M.; Liu, Q.; Zhu, H. An Efficient Heuristic Rapidly-Exploring Random Tree for Unmanned Aerial Vehicle. In Proceedings of the The International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, Fuzhou, China, 30 July–1 August 2022; pp. 764–772.
7. Yin, X.; Dong, W.; Wang, X.; Yu, Y.; Yao, D. Route planning of mobile robot based on improved RRT star and TEB algorithm. *Sci. Rep.* **2024**, *14*, 8942. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Lei, T.; Sellers, T.; Luo, C.; Carruth, D.W.; Bi, Z. Graph-based robot optimal path planning with bio-inspired algorithms. *Biomim. Intell. Robot.* **2023**, *3*, 100119. [\[CrossRef\]](#)
9. Wu, Z.; Dong, S.; Yuan, M.; Cui, J.; Zhao, L.; Tong, C. Rotate artificial potential field algorithm toward 3D real-time path planning for unmanned aerial vehicle. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2023**, *237*, 940–955. [\[CrossRef\]](#)
10. Pichot, S.; Khalifé, H.; Conan, V.; Jakllari, G. ColorUAV: A genetic algorithm for dynamic drone path planning under multiple constraints. In Proceedings of the 2022 International Wireless Communications and Mobile Computing (IWCMC), Dubrovnik, Croatia, 30 May–3 June 2022; pp. 623–628.
11. Chang, L.; Shan, L.; Jiang, C.; Dai, Y. Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment. *Auton. Robot.* **2021**, *45*, 51–76. [\[CrossRef\]](#)
12. Tian, H.; Mo, Z.; Ma, C.; Xiao, J.; Jia, R.; Lan, Y.; Zhang, Y. Design and validation of a multi-objective waypoint planning algorithm for UAV spraying in orchards based on improved ant colony algorithm. *Front. Plant Sci.* **2023**, *14*, 1101828. [\[CrossRef\]](#)
13. Huang, C.; Zhou, X.; Ran, X.; Wang, J.; Chen, H.; Deng, W. Adaptive cylinder vector particle swarm optimization with differential evolution for UAV path planning. *Eng. Appl. Artif. Intell.* **2023**, *121*, 105942. [\[CrossRef\]](#)
14. Huo, L.; Zhu, J.; Wu, G.; Li, Z. A novel simulated annealing based strategy for balanced UAV task assignment and path planning. *Sensors* **2020**, *20*, 4769. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Xing, P.; Zhang, H.; Ghoneim, M.E.; Shutaywi, M. UAV flight path design using multi-objective grasshopper with harmony search for cluster head selection in wireless sensor networks. *Wirel. Netw.* **2023**, *29*, 955–967. [\[CrossRef\]](#)
16. Yan, F.; Liu, Y.-S.; Xiao, J.-Z. Path planning in complex 3D environments using a probabilistic roadmap method. *Int. J. Autom. Comput.* **2013**, *10*, 525–533. [\[CrossRef\]](#)
17. Majeed, A.; Lee, S. A fast global flight path planning algorithm based on space circumscription and sparse visibility graph for unmanned aerial vehicle. *Electronics* **2018**, *7*, 375. [\[CrossRef\]](#)
18. Feng, Y.; Zhang, C.; Baek, S.; Rawashdeh, S.; Mohammadi, A. Autonomous landing of a UAV on a moving platform using model predictive control. *Drones* **2018**, *2*, 34. [\[CrossRef\]](#)
19. Zhou, X.; Wang, Z.; Ye, H.; Xu, C.; Gao, F. Ego-planner: An esdf-free gradient-based local planner for quadrotors. *IEEE Robot. Autom. Lett.* **2020**, *6*, 478–485. [\[CrossRef\]](#)
20. Yu, X.; Li, C.; Zhou, J. A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios. *Knowl.-Based Syst.* **2020**, *204*, 106209. [\[CrossRef\]](#)



21. Han, Z.; Chen, M.; Shao, S.; Wu, Q. Improved artificial bee colony algorithm-based path planning of unmanned autonomous helicopter using multi-strategy evolutionary learning. *Aerosp. Sci. Technol.* **2022**, *122*, 107374. [[CrossRef](#)]
22. Xie, C.; Zheng, H. Application of improved Cuckoo search algorithm to path planning unmanned aerial vehicle. In Proceedings of the Intelligent Computing Theories and Application: 12th International Conference, ICIC 2016, Lanzhou, China, 2–5 August 2016; Proceedings, Part I 12, 2016. pp. 722–729.
23. Patel, B.; Patle, B. Analysis of firefly–fuzzy hybrid algorithm for navigation of quad-rotor unmanned aerial vehicle. *Inventions* **2020**, *5*, 48. [[CrossRef](#)]
24. Lee, J.M. *Introduction to Riemannian Manifolds*; Springer: Cham, Switzerland, 2018; Volume 2.
25. Bhattacharya, S.; Ghrist, R.; Kumar, V. Multi-robot coverage and exploration on Riemannian manifolds with boundaries. *Int. J. Robot. Res.* **2014**, *33*, 113–137. [[CrossRef](#)]
26. Calinon, S. Gaussians on Riemannian manifolds: Applications for robot learning and adaptive control. *IEEE Robot. Autom. Mag.* **2020**, *27*, 33–45. [[CrossRef](#)]
27. Smith, S.T. Optimization techniques on Riemannian manifolds. *arXiv* **2014**, arXiv:1407.5965.
28. Chahrour, H.M. Signal Processing Methods in Riemannian Geometry with Application to Drone Detection. Ph.D. Thesis, Carleton University, Ottawa, ON, Canada, 2021.
29. Beik-Mohammadi, H.; Hauberg, S.; Arvanitidis, G.; Neumann, G.; Rozo, L. Learning riemannian manifolds for geodesic motion skills. *arXiv* **2021**, arXiv:2106.04315.
30. Biess, A.; Liebermann, D.G.; Flash, T. A computational model for redundant human three-dimensional pointing movements: Integration of independent spatial and temporal motor plans simplifies movement dynamics. *J. Neurosci.* **2007**, *27*, 13045–13064. [[CrossRef](#)]
31. Jaquier, N.; Borovitskiy, V.; Smolensky, A.; Terenin, A.; Asfour, T.; Rozo, L. Geometry-aware Bayesian optimization in robotics using Riemannian Matérn kernels. In Proceedings of the Conference on Robot Learning, Auckland, New Zealand, 14–18 December 2022; pp. 794–805.
32. Jaquier, N.; Asfour, T. Riemannian geometry as a unifying theory for robot motion learning and control. In Proceedings of the The International Symposium of Robotics Research, Geneva, Switzerland, 25–30 September 2022; pp. 395–403.
33. Minh, H.Q.; Murino, V.; Minh, H.Q. *Algorithmic Advances in Riemannian Geometry and Applications*; Springer: Berlin/Heidelberg, Germany, 2016.
34. Chow, B.; Lu, P.; Ni, L. *Hamilton’s Ricci Flow*; American Mathematical Society, Science Providence Press: Providence, RI, USA, 2023; Volume 77.
35. Mac, T.T.; Copot, C.; Hernandez, A.; De Keyser, R. Improved potential field method for unknown obstacle avoidance using UAV in indoor environment. In Proceedings of the 2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMII), Herl’any, Slovakia, 21–23 January 2016; pp. 345–350.
36. He, Y.; Zeng, Q.; Liu, J.; Xu, G.; Deng, X. Path planning for indoor UAV based on Ant Colony Optimization. In Proceedings of the 2013 25th Chinese Control and Decision Conference, CCDC2013, Guiyang, China, 25–27 May 2013; pp. 2919–2923.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.