Multi-UAV Path Planning and Conflict-Free Navigation in 3D Space

report on the Winter Project

Bachelor of Technology
in
Electrical Engineering

Submitted by

Roll No	Names of Students
U23EE036	Sujay Bhati

Under the guidance of
Dr. Sharad Kumar Singh

Department of Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY INDORE
M.P., Indore, India – 453 552

Department of Electrical Engineering

Indian Institute of Technology Indore

Certificate

This is to certify that Mr. Sujay Bhati, B.Tech., Electrical Engineering, 3rd Year student of Sardar Vallabhbhai National Institute of Technology, Surat has undergone the Winter Internship Program in our Institute, Indian Institute of Technology Indore, from 7 Dec-2025 to 7-Jan-2026. He completed the internship program under the guidance of Dr. Sharad Kumar Singh in the area of "Multi-UAV Path Planning and Conflict-Free Navigation in 3D Space". His conduct and performance during the period were found to be satisfactory.

Dr. Sharad Kumar Singh
(Project Guide)

Date:07 Jan 2026

Abstract

Autonomous path planning is a critical requirement for Unmanned Aerial Vehicles (UAVs) operating in environments with obstacles and multiple agents. This project presents the design and simulation of a path-finding approach for single and multiple drones in a three-dimensional grid-based environment. The path planning problem is formulated with constraints to ensure obstacle avoidance and prevent collisions between drones by enforcing spatial and temporal separation. An optimization-based algorithm is implemented in Python to generate feasible and efficient waypoints from given start and goal positions. These waypoints are then used in a MATLAB simulation to visualize drone motion and validate the effectiveness of the proposed approach. Simulation results demonstrate successful collision-free navigation and obstacle avoidance for different test scenarios. The project provides a practical framework for multi-UAV path planning and serves as a basis for future enhancements such as dynamic obstacle handling and real-time path replanning.

Contents

List of Figures

Chapter 1

Problem Definition

The increasing use of Unmanned Aerial Vehicles (UAVs) in autonomous applications requires reliable path planning techniques that ensure safe and efficient navigation in constrained environments. When multiple drones operate simultaneously within the same airspace, the complexity of navigation increases due to the presence of static obstacles and the risk of inter-drone collisions. Traditional single-agent path planning methods are insufficient to address these challenges, as they do not account for conflicts arising from shared space and time.

The problem addressed in this project is the development of a path-finding framework for single and multiple UAVs that generates collision-free trajectories from specified start positions to desired goal locations within a three-dimensional environment containing obstacles. The solution must ensure effective obstacle avoidance and safe coordination among multiple drones by preventing spatial and temporal conflicts, while maintaining feasible and efficient paths.

Chapter 2

Introduction

2.1 Background and Motivation

Unmanned Aerial Vehicles (UAVs) are increasingly deployed in applications such as surveillance, inspection, disaster response, and autonomous delivery, where multiple drones are required to operate simultaneously within a shared three-dimensional environment. As the number of UAVs increases, ensuring safe, efficient, and coordinated navigation becomes a critical challenge due to the risk of inter-drone collisions and interactions with static obstacles. Traditional single-UAV path planning methods are insufficient in such scenarios, as they fail to account for spatiotemporal conflicts that arise when multiple agents occupy the same airspace. This has motivated the development of multi-UAV path planning approaches that explicitly consider collision avoidance while optimizing flight paths. In this project, conflict-free navigation in 3D space is addressed through a centralized planning framework that models both spatial and temporal constraints, enabling drones to avoid being at the same location at the same time. By leveraging algorithmic strategies such as conflict detection and constraint-based resolution, and validating the approach through simulation using Python-based planning and MATLAB-based visualization, the project aims to demonstrate a scalable and systematic solution for coordinated multi-UAV navigation in complex environments.

2.2 Objectives of the Project

The primary objective of this project is to develop and evaluate a simulation-based framework for conflict-free path planning of multiple UAVs in three-dimensional space. The project aims to design a centralized path planning approach that generates feasible and collision-free trajectories for multiple drones while considering both spatial and temporal constraints. Specific objectives include modeling the 3D environment with static obstacles, defining inter-drone conflict conditions, and implementing constraint-handling mechanisms to prevent drones from occupying the same position at the same time. Additionally, the project seeks to implement the path planning logic using Python and validate the generated trajectories through MATLAB-based simulation and visualization. Performance evaluation is carried out by analyzing path optimality, conflict resolution effectiveness, and scalability with respect to the number of drones. Overall, the project aims to provide a structured and extensible solution for coordinated multi-UAV navigation suitable for simulation and future real-world extension.

2.3 Initial Setup

2.3.1 Waypoint and Trajectory Generation Module

The first module of the proposed framework is responsible for waypoint generation and trajectory computation. In this stage, a sequence of predefined waypoints with associated time stamps is used to define the desired UAV path in three-dimensional space. Based on these waypoints, reference position commands along the X, Y, and Z axes are computed by evaluating segment-wise velocities between consecutive waypoints. The code generates smooth position command signals over time by superposing individual motion segments, including take-off, level flight, and hover phases. These reference trajectories serve as high-level navigation inputs and form the basis for subsequent control and simulation stages.

2.3.2 Animation and Visualization Module

The second module is dedicated to 3D animation and visualization of the UAV motion. Using the simulated state outputs such as position, velocity, and Euler angles, this code renders the UAV trajectory and orientation in a three-dimensional environment. The animation provides real-time visualization of the UAV body frame, direction of motion, and target tracking behavior, enabling qualitative assessment of path smoothness and navigation accuracy. This module does not influence the control logic but plays a crucial role in validating the correctness of the generated waypoints and the resulting drone motion.

2.3.3 Dynamic Modeling and Control Evaluation

The Simulink model represents the dynamic and control evaluation layer of the system. It takes the reference waypoints and velocity commands generated by the first code as inputs and simulates the UAV's response based on its physical parameters, including mass, inertia, and gravitational effects. The model computes the required forces, torques, and thrust outputs necessary for the UAV to follow the specified trajectory. By incorporating drone dynamics and control loops, the Simulink environment enables quantitative evaluation of flight performance, stability, and tracking accuracy. This model bridges the gap between high-level path planning and low-level drone actuation within a unified simulation framework.

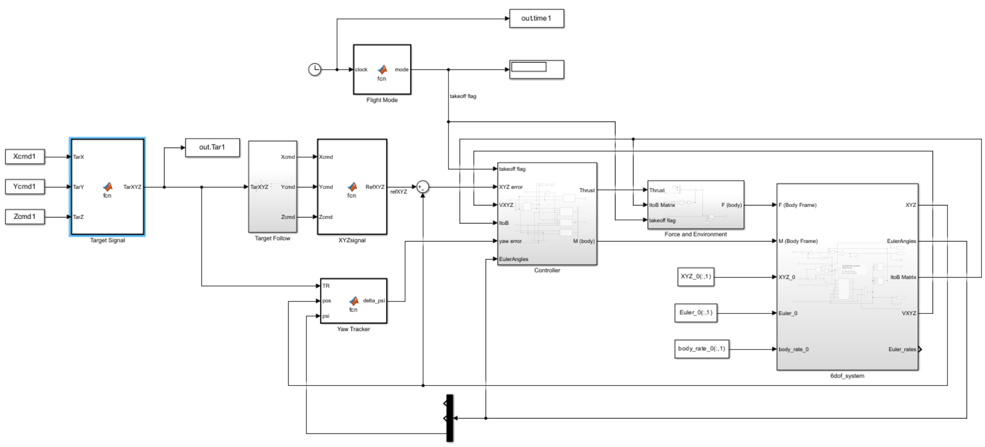

Figure 2.1: Drone Control Simulink Model

Chapter 3

Work Done

3.1 Trajectory Generation Using Waypoints

3.1.1 Hexagonal Trajectory

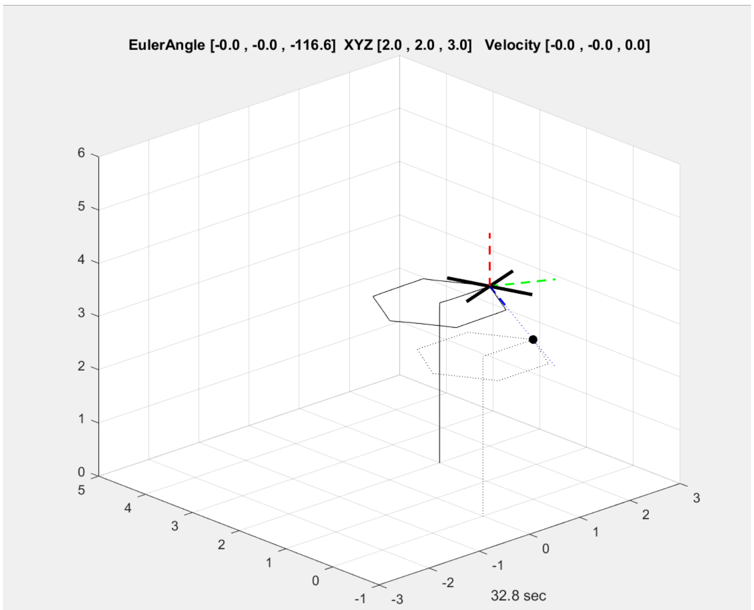

Figure 3.1: `Hexagonal Trajectory`

To validate structured waypoint-based trajectory planning, a hexagonal flight path was implemented by explicitly defining the vertices of a hexagon through discrete position commands in the X, Y, and Z directions. The waypoint coordinates were carefully selected such that each vertex of the hexagon corresponded to a specific spatial location, while the associated time stamps were manipulated to control the duration of travel between consecutive vertices.Additional dwell time was introduced at each vertex to allow the UAV to stabilize and reorient its heading before proceeding to the next segment. This deliberate pause enabled the drone to perform sharp directional changes, ensuring that the executed trajectory closely followed a geometrically distinct hexagonal shape rather than a smooth, rounded approximation. This approach demonstrated precise control over trajectory shape through waypoint timing and highlighted the importance of temporal parameterization in achieving accurate geometric path representation.

3.1.2 Circular Trajectory

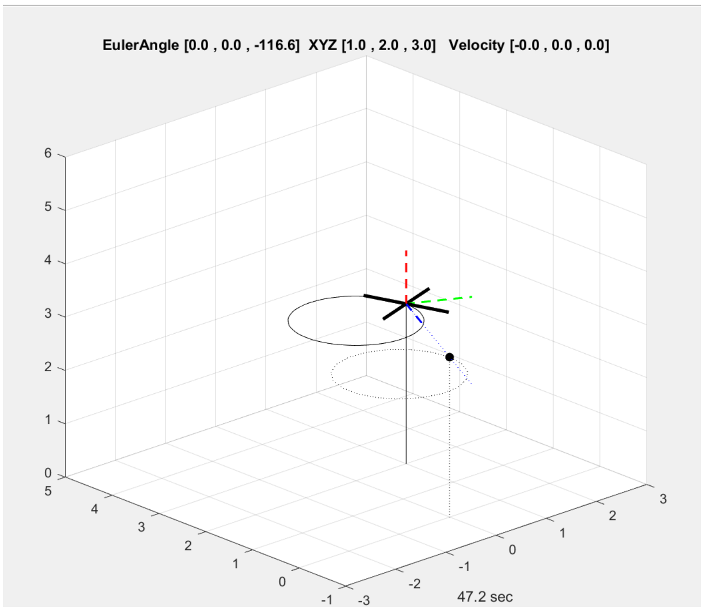

Figure 3.2: `Circular Trajectory`

After successfully implementing the hexagonal path, a circular trajectory was designed to enable smooth, continuous motion in 3D space. The circle was defined mathematically by specifying its center coordinates (Xc, Yc, Zc) and a radius of 1 unit. Waypoints along the circle were generated using the standard parametric equations of a circle, with angular positions incremented uniformly over time to maintain consistent velocity. This allowed the UAV to follow a smooth curved path, in contrast to the previously executed hexagonal trajectory where sharp turns at vertices were intentional to maintain the geometric shape. Time assignments for each waypoint ensured proper pacing along the circular path, demonstrating precise control over both spatial and temporal aspects of UAV motion.

3.2 Drone Navigation with Static Obstacle

For point to point navigation in the presence of a static obstacle, a hybrid trajectory strategy was implemented by combining straight line motion with a spherical avoidance path. The obstacle was modeled using a spherical region centered at the obstacle location, with a radius chosen larger than the physical size of the obstacle to ensure a safe clearance margin. Under normal conditions, the UAV follows a straight line equation from the start position to the target position. When the UAV approaches the obstacle and enters a predefined proximity threshold, the trajectory generation logic switches from the straight line equation to the spherical equation, causing the drone to move along the surface of the sphere and bypass the obstacle. Once the obstacle is safely cleared, the trajectory switches back to the straight line equation and the UAV continues toward its target. This approach enabled deterministic and smooth obstacle avoidance while preserving overall point to point navigation efficiency.

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 = R^2 \tag{3.1}$$

$$\mathbf{p}(t) = \mathbf{c} + R\left(\mathbf{u}\cos(t) + \mathbf{v}\sin(t)\right) \tag{3.2}$$

where $\mathbf{c} = [x_c\ y_c\ z_c]^T$ is the center of the sphere, R is the radius, and $\mathbf{u}$ and $\mathbf{v}$ are orthonormal vectors lying in the plane of motion.
Alternatively, using spherical coordinates, a point on the spherical surface can be expressed as:

$$z = z_c \pm \sqrt{R^2 - (x - x_c)^2 - (y - y_c)^2} \tag{3.3}$$

When the drone is to close to the obstacle one of the coordinates out of the three is chnaged making the drone go around the obstacle
The trajectory switching condition between straight-line motion and spherical avoidance can be defined as:

$$\|\mathbf{p}_{uav}(t) - \mathbf{c}\| \leq R \tag{3.4}$$

We later switched to another method in which we calculated the plane in which drone is travelling then finds the shortest path out of the two path in that plane to find the shortest possible path for drone to go around the obstacle

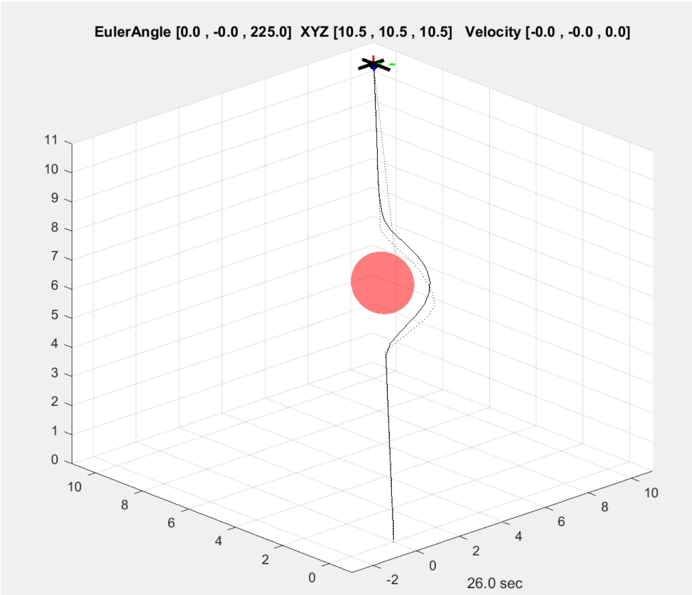

Figure 3.3: `Point-to-Point Navigation with Static Obstacle`

3.3 Multi-UAV Interaction and Dynamic Obstacle Modeling

To extend the obstacle avoidance strategy to a multi UAV scenario, each UAV was modeled as a dynamic obstacle for the others operating in the same environment. The spherical avoidance formulation used for static obstacles was retained; however, the center of the avoidance sphere was made time varying and defined by the instantaneous position of the neighboring UAV. This allows the avoidance region to move along the trajectory of the other drone, enabling continuous and adaptive collision avoidance.

When two UAVs approach within a predefined safety distance, the lower priority UAV switches from its nominal trajectory to a spherical avoidance path around the higher priority UAV. The higher priority UAV continues to follow the shortest feasible path toward its target without deviation. Once sufficient separation is reestablished, the avoiding UAV transitions back to its original trajectory.

A priority based conflict resolution strategy was implemented to ensure deterministic behavior. Each UAV is assigned a fixed priority level, and avoidance responsibility is determined accordingly. This framework is scalable, as the same logic can be extended to multiple UAVs by assigning priorities from first to nth, enabling orderly and conflict free navigation in multi UAV environments.

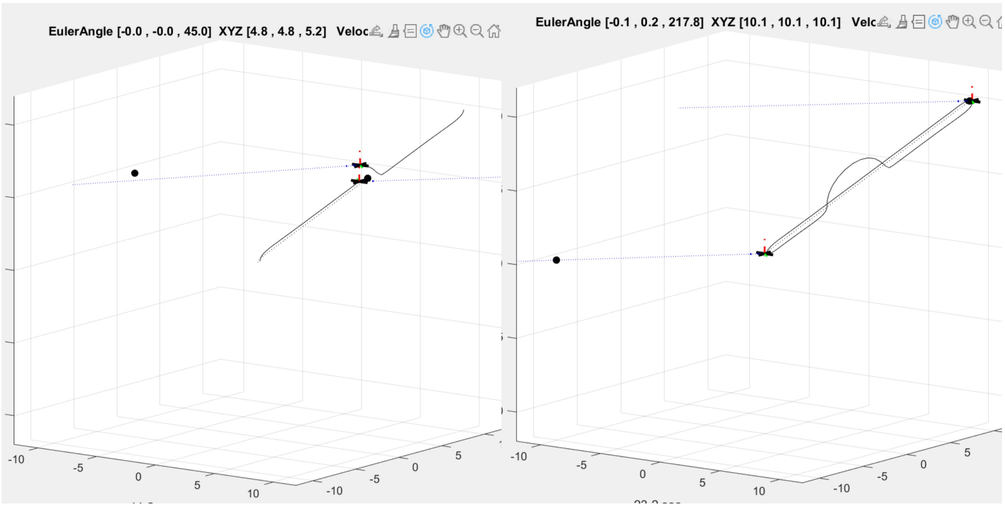

Figure 3.4: `Multi-UAV Interaction and Dynamic Obstacle Avoidance`

3.4 Grid-Based Conflict-Free Planning

Grid-based conflict-free planning refers to the problem of computing collision-free paths for multiple agents operating in a discretized environment. The continuous 3D workspace is represented as a finite grid, where each cell corresponds to a valid spatial location. UAV motion is restricted to adjacent grid cells, and the objective is to determine optimal paths from start to goal positions while avoiding static obstacles and preventing inter-agent conflicts in space and time.

In multi-UAV systems, conflict-free planning ensures that no two drones occupy the same grid cell or traverse conflicting edges at the same time step, thereby guaranteeing safe and coordinated navigation.

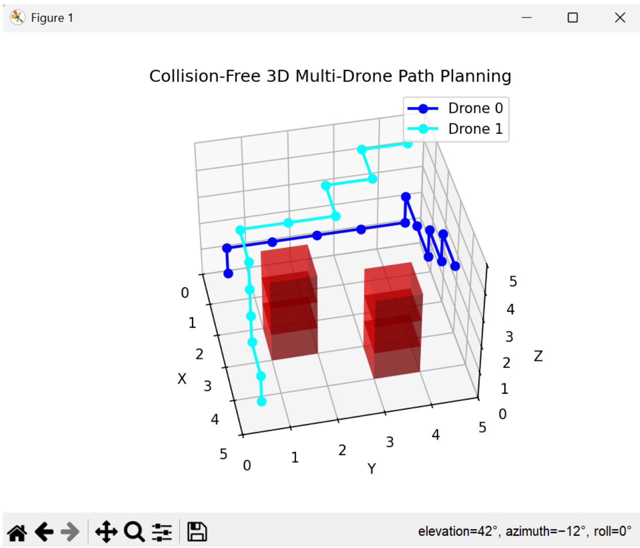

Figure 3.5: `Multi-UAV Path Planning`

At this stage of the project, multi-UAV path planning has been successfully implemented in the Python framework. The developed model supports multiple drones operating simultaneously within a three-dimensional grid environment. Each drone is assigned a distinct starting position and target destination, and optimal paths are computed such that all static obstacles are avoided. In addition, inter-drone conflicts are fully resolved, ensuring that no two drones occupy the same grid cell or traverse conflicting paths during execution. The resulting paths are globally conflict-free and allow all drones to reach their respective destinations safely.

Chapter 4

Future Work

4.1 Transition from Path-Level to Execution-Level Avoidance

The current implementation plans complete paths in advance while enforcing collision avoidance constraints throughout the entire planning horizon. Although this guarantees safety, it introduces conservative restrictions that can lead to unnecessarily long paths and increased computational complexity.

As a direction for future work, collision avoidance can be shifted from *path-level planning* to *execution-level coordination*. In this approach, drones initially compute independent shortest paths without considering other agents. During execution, collision checks are performed only at the current step, and coordination is enforced locally when a conflict is detected.

Specifically, this strategy involves:

- Planning independent shortest paths for each drone
- Detecting conflicts only at the current execution step
- Performing local re-planning when a collision risk is identified

By limiting collision avoidance to the current positions of drones rather than the entire planned trajectory, the overall constraint tightness is reduced. This results in shorter paths, lower computational overhead, and improved scalability, particularly for larger grid environments and higher numbers of

4.2 Execution of CBS and PBS

In the current approach, collision avoidance is enforced over the entire planned path, which can lead to conservative solutions and unnecessary detours. As future work, collision avoidance can be applied only at the current positions of drones during execution rather than across the full trajectory. This local coordination allows drones to reuse grid cells at different stages, resulting in shorter paths and improved planning efficiency

4.2.1 Conflict-Based Search (CBS)

Conflict-Based Search is a multi-agent path planning algorithm that resolves collisions by introducing constraints only when conflicts occur. Each drone initially computes an independent shortest path. When a conflict between drones is detected, CBS branches the solution space by adding constraints that prevent the conflicting behavior, replanning paths only for the affected drones. This selective constraint generation reduces unnecessary restrictions and often leads to shorter and more optimal paths compared to global constraint-based methods.

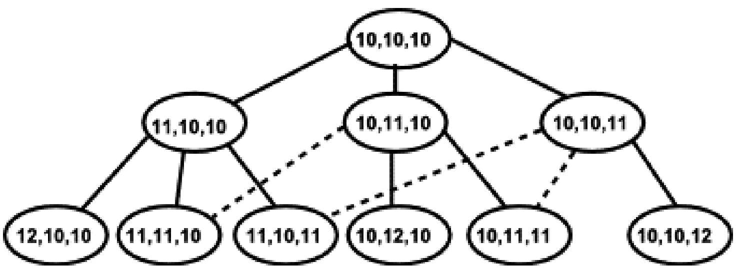

Figure 4.1: `Multi-UAV Interaction and Dynamic Obstacle Avoidance`

4.2.2 Priority-Based Search (PBS)

Priority-Based Search assigns priorities to drones and resolves conflicts by replanning paths for lower-priority agents while higher-priority agents remain fixed. When a conflict is detected, the priority ordering is adjusted, and only the necessary drones are replanned. PBS reduces computational complexity by avoiding full replanning and is particularly effective in large-scale environments with many agents, making it a suitable future extension for efficient multi-drone path planning.

Chapter 5

Conclusion

This project focused on the development of a multi-drone path planning framework in a three-dimensional grid environment with static obstacles. A mixed-integer linear programming (MILP) formulation was designed to generate feasible paths for multiple drones, ensuring valid movement between adjacent grid cells while avoiding obstacles and satisfying specified start and goal conditions.

The proposed approach successfully enabled multiple drones to navigate the environment simultaneously while maintaining collision-free operation through appropriate constraint design. The results demonstrate that the formulated model can effectively compute feasible and structured paths in a constrained 3D space. Overall, the work establishes a solid baseline for centralized multi-drone path planning and highlights the effectiveness of optimization-based methods for coordinated autonomous navigation.

Acknowledgments

I would like to express my sincere gratitude to my project advisor for their continuous guidance, constructive feedback, and encouragement throughout the course of this online project. Despite the challenges of remote communication, their support and insights were instrumental in shaping the direction of my work and overcoming difficulties encountered during the project.

I would also like to thank my peers for their discussions and suggestions, which provided valuable perspectives and contributed positively to the progress of the project.

Finally, I acknowledge the motivating academic environment provided by the department, which fostered independent learning and research, enabling me to successfully complete this project in an online setting.

Sujay Bhati

January 2026
National Institute of Technology Surat

References

[1] Multi-Agent Path Finding in Unmanned Aircraft System Traffic Management With Scheduling and Speed Variation, https://ieeexplore.ieee.org/document/9516995

[2] Efficient Path Planning in Multi-Agent Environment of AAVs With Payloads, https://ieeexplore.ieee.org/document/10938620