Real Time Obstacle Avoidance and Path Planning of Drones with Kinodynamic Constraints in Unknown Environments.

Dhruv Agrawal
Robotics Engineering Department
Worcester Polytechnic Institute
Worcester, MA
dagrawal@wpi.edu

Hrishikesh Pawar
Robotics Engineering Department
Worcester Polytechnic Institute
Worcester, MA
hpawar@wpi.edu

Shreyas Khobragade
Robotics Engineering Department
Worcester Polytechnic Institute
Worcester, MA
skhobragade@wpi.edu

I. Introduction

This paper proposes a real-time trajectory planner for drones operating in unknown environments under kinodynamic constraints. The primary aim is to develop an intelligent navigation system for drones that can operate autonomously in environments that are not known beforehand. The research specifically addresses the challenge of incorporating kinodynamic constraints considering both kinematic limitations and dynamic constraints. The emphasis on real-time computation is crucial because decisions must be made instantaneously during flight and path adjustments need to occur immediately when new obstacles are detected. The primary research question to be investigated in this study is: **How can drones effectively navigate and avoid obstacles in real-time while operating in completely unknown environments?**

The Application Impacts of this study include areas such as:

- Emergency Response
  - Enhanced search and rescue operations in situations of a calamity
  - More efficient disaster area assessment and victim location

  This autonomous drone framework revolutionizes rescue operations by enabling remote environmental assessment and data collection, thereby minimizing human exposure to hazardous conditions while providing rescue teams with crucial real-time information for informed decision-making.
- Industrial Applications
  - Infrastructure inspection in complex environments
  - Agricultural monitoring and precision farming
  - Package delivery in urban environments

  This autonomous navigation framework transforms multiple sectors: enabling comprehensive infrastructure assessment in complex environments, enhancing agricultural productivity through precision aerial monitoring and real-time crop analysis, and optimizing urban logistics through efficient autonomous delivery systems. These capabilities directly translate to improved safety protocols, increased crop yields, and streamlined delivery operations.
- Safety and Reliability
  - More reliable autonomous navigation in challenging conditions

  The framework guarantees reliable autonomous navigation through a multi-layered safety approach: incorporating extensive simulation validation, implementing robust perception algorithms, deploying adaptive planning and control strategies, and utilizing real-time decision-making systems. This comprehensive architecture ensures operational safety even in the most challenging environmental conditions.

II. Related Work

The field of autonomous drone navigation and obstacle avoidance in unknown environments has seen significant advancements in recent years. Numerous studies have focused on developing efficient mapping, collision detection and planning to enable drones to operate autonomously in complex and cluttered environments.

Several studies have also focused on the specific challenges of trajectory generation for quadrotors. Chen et al. [4] proposed a real-time safe trajectory generation method for quadrotors operating in cluttered environments. Their approach emphasizes safety by ensuring that generated trajectories avoid obstacles while considering the drone's dynamic constraints. Similarly, Gao and Shen [5] introduced an online trajectory generation technique that leverages point cloud data to navigate complex environments autonomously.

Further advancements were made by Ding et al. [6], who developed a kinodynamic search-based approach combined with elastic optimization for trajectory replanning. This method allows quadrotors to adapt their flight paths dynamically in response to changes in the environment, ensuring both safety and efficiency during flight.

Another notable contribution is the work by Han et al. [8], who introduced FIESTA, a fast incremental Euclidean distance field method designed for real-time motion planning of aerial robots. FIESTA enables rapid updates to the environment map, facilitating quick adjustments to planned trajectories when new obstacles are detected.

Finally, Oleynikova et al. [9] presented Voxblox, an incremental 3D Euclidean Signed Distance Field (ESDF) framework that is particularly well-suited for onboard UAV planning. Voxblox allows drones to efficiently compute collision-free paths by maintaining an up-to-date ESDF representation of the environment, making it ideal for real-time applications where computational efficiency is critical.

These works form the foundation upon which our proposed system builds. By integrating these established methodologies with kinodynamic planning, our system aims to enhance the performance of autonomous drones in unknown environments while ensuring real-time computational efficiency and safety.

III. Proposed Methods

The proposed autonomous navigation framework builds upon established methodologies while introducing novel integrations for enhanced performance. At its core, the system comprises three tightly coupled modules that work in synchronized harmony: global environmental mapping using OctoMap [1];local real-time 3D obstacle detection and collision checking using EDT3D library and point cloud processing; sophisticated path planning that incorporates kinodynamic constraints and sampling-based methods for safe waypoint and trajectory generation using OMPL [2]. These interconnected modules form a comprehensive pipeline that enables autonomous navigation in complex, unknown environments while maintaining computational efficiency and operational safety through continuous feedback loops and real-time updates.

IV. Mapping

The mapping framework in our system is divided into two components: **OctoMap** for global mapping and **EDT3D** for efficient collision detection in the local vicinity of the drone. These components work together to enable precise navigation and obstacle avoidance.

A. OctoMap

The system leverages OctoMap's hierarchical tree structure to generate and maintain an efficient 3D occupancy grid representation from depth sensor point cloud data. The hierarchical nature of OctoMap allows for scalable storage and computation by partitioning the 3D space into voxels with probabilistic occupancy values. This facilitates a compact representation of large environments while maintaining the fidelity required for precise obstacle mapping.

Grid resolution represents a key parameter that balances computational overhead with the accuracy of obstacle representation. Finer resolutions provide more detailed mappings but increase computational demands, whereas coarser resolutions are computationally cheaper but may compromise accuracy. OctoMap's flexibility allows for efficient navigation in complex environments by dynamically balancing these trade-offs.

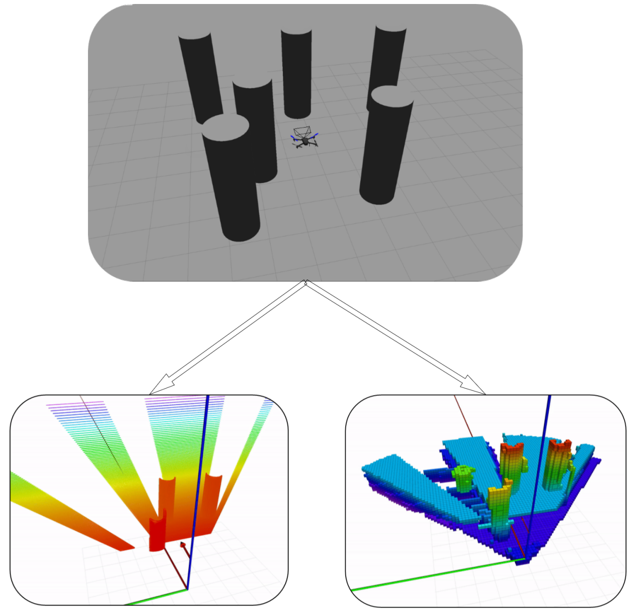

Fig. 1: Top: Gazebo envronment, Bottom Left: Point-Cloud Representation, Bottom Right: Octomap Representation

B. EDT3D

To complement the global mapping provided by OctoMap, the system incorporates the **Euclidean Distance Transform (EDT)** using the EDT3D library for local collision detection.

- **Why EDT3D is Used:** Occupancy grid-based methods assign probabilities to voxels but fail to account for proximity to obstacles. A voxel may be marked as free but still be very close to an obstacle, posing a collision risk. EDT solves this issue by computing the exact distance of each free voxel to the nearest obstacle, ensuring safer navigation.
- **What is EDT3D:** The EDT in 3D computes the Euclidean distance for each free voxel in a grid to the nearest occupied voxel. The Euclidean distance is given by:

  $$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

  This distance provides precise proximity information that occupancy grids lack.
- **Optimized Local Window Approach:** Computing EDT for the entire map is computationally expensive, particularly in real-time scenarios. Instead, the system maintains a *local window* around the drone, defined by a fixed bounding box. The size of this bounding box depends on the drone's sensing range. In our case:
  - An 8-meter sensing range, mimicking the Intel RealSense depth camera, is used.
  - A 4-meter range on each side of the drone is maintained for collision checking.

  This localized computation reduces processing demands while ensuring accurate and reliable collision detection.

By combining OctoMap's global mapping capabilities with EDT3D's precise local collision detection, the system achieves a robust mapping framework optimized for real-time operation in unknown environments.

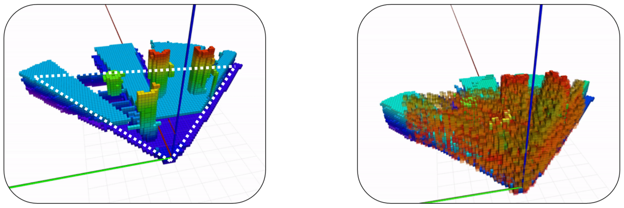

Fig. 2: Left: Octomap with a visible range, Right: Euclidean Representation within the visible range

C. Planning

Our framework implements a kinodynamically-constrained RRT planner that synthesizes feasible trajectories by incorporating the drone's dynamic and kinematic constraints during path computation from start to goal states. While operating in the exploration phase, our system employs the tree-based RRT planner since it provides efficient sampling-based exploration of configuration space, enables real-time path computation, and naturally incorporates both kinodynamic constraints and information gain metrics for unknown environment navigation.

Our implementation of the planner consists of an addition to the original RRT planner. Since we can only see the obstacles upto a certain range from the drone, we cannot plan the complete trajectory in one shot. To achieve a trajectory from start to goal, we compute the sub-paths till the visible region and append these to create the complete path. At the initial configuration, the drone plans the complete path from start to goal. However, since it can only see upto the visible region, we extract the sub-path from the planned trajectory and prune the rest of the path. The last node in the visible region now becomes our new start position and we again plan the trajectory from the new start position to the goal position. We repeat the pruning process and re-assigning of the start position until our goal is inside the visible region. Every iteration of this adds a sub-path to our global path of the initial start and goal conditions.

To start with the implementation on the drone, we consider a 2D environment as the workspace with pre-defined rectangular obstacles and consider the drone as a square. Then we do a collision check for each random sampled state and apply our kinodynamic constraints on the drone. If the sampled node and the path to it from the nearest node in the tree are collision free then we move towards it for a maximum of ten steps and add that node to the tree. We verify our implementation for different start and goal locations. We also build upon our planner so that it works in the 3D environment. For the collision checking, the drone is assumed as a cube and we deploy a smart naive collision checking mechanism where if the base of the cube is higher than the obstacle then there is no collision check and if the base is within the obstacle height then the cube is assumed as a square and obstacle is assumed as a rectangle and we can do the collision check in the same way as we did for the 2D case.

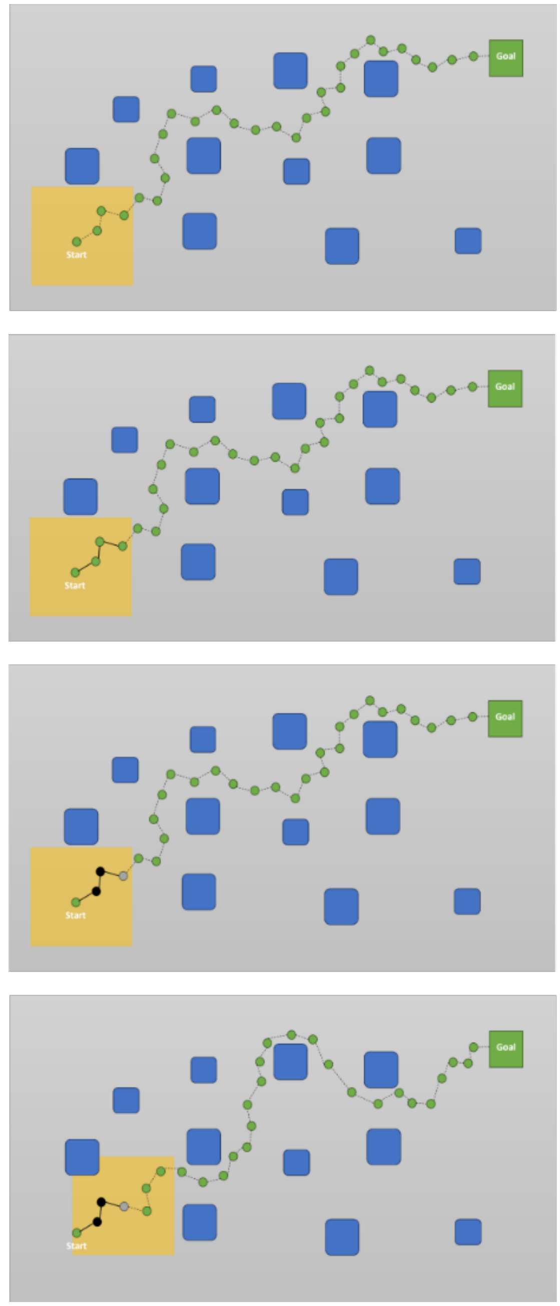

Fig. 3: Panning mechanism for visible region

The proposed planner was implemented and tested in two distinct environments to evaluate its performance and adaptability. These environments are illustrated in Fig. 4 and Fig. 5. The first environment was developed using only the Open Motion Planning Library (OMPL), while the second environment incorporated both OctoMap for global mapping and the EDT3D library for precise local collision detection.

1) Environment and Pipeline Configuration: The pipeline was configured to work offline, primarily due to computational constraints. Running the OctoMap library, EDT3D library, and the planner in real-time proved to be challenging on the testing platform, which was limited to 16 GB of RAM. To address

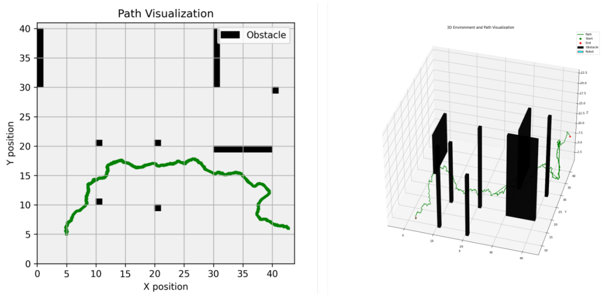

Fig. 4: OMPL Environment

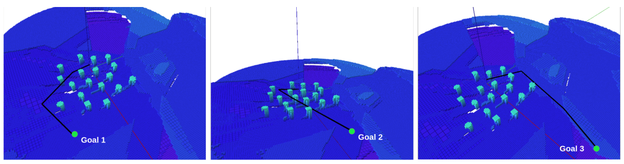

Fig. 5: Octomap Environment

these constraints, a static map of the environment was pre-generated. This allowed the planner to operate independently of real-time mapping updates while still leveraging the benefits of OctoMap and EDT3D for accurate representation and collision detection.

2) Description of the Environments:

- **OMPL-Only Environment:** This environment was designed as a simple workspace where planning tasks relied solely on the OMPL framework. It served as a baseline to evaluate the planner's ability to generate kinodynamically feasible trajectories without the added complexity of integrating mapping and collision detection libraries.
- **Integrated OctoMap and EDT3D Environment:** The second environment was more complex, featuring a group of pillars represented within an OctoMap framework. These pillars were modeled as obstacles, providing a realistic and cluttered environment for testing. The OctoMap provided a hierarchical representation of the 3D space, while the EDT3D library enabled precise local collision detection by computing the distances to obstacles within a defined local window around the drone.

3) Evaluation and Testing: The planner was evaluated in the OctoMap-integrated environment for three distinct cases, each involving different start and goal configurations. For every case:

- The planner successfully generated collision-free trajectories that adhered to the drone's kinodynamic constraints.
- Sub-paths were iteratively generated for segments within the drone's sensing range, and these were seamlessly appended to form a complete path to the goal.
- Evaluations were performed on the 2D planner, taking three different step sizes, where the planner was run 5 times for each condition.
- It was observed that lower step sizes resulted in higher probability of finding the path, where the success rate was defined on number of sub-paths planned for the complete path.
- If the number of sub-paths crossed 20 and the planner failed to reach the goal region then the result of the planner was indicated as FAIL else it was indicated as SUCCESS.

STEPSIZE = 0.5		STEPSIZE = 1.0		STEPSIZE = 0.2	
No. of Subpaths	Result	No. of Subpaths	Result	No. of Subpaths	Result
20	FAIL	9	SUCCESS	13	SUCCESS
8	SUCCESS	20	FAIL	8	SUCCESS
20	FAIL	20	FAIL	20	SUCCESS
17	SUCCESS	17	SUCCESS	9	SUCCESS
11	SUCCESS	11	SUCCESS	20	FAIL

Fig. 6: Metrics

4) Challenges and Results: The integration of OctoMap and EDT3D posed additional computational challenges due to the high demands of processing large maps and performing real-time collision detection. However, the offline setup allowed the planner to utilize the pre-generated map efficiently. The results demonstrated:

- **Accuracy:** The planner effectively avoided all obstacles in the environment in its successful run, showcasing its robust collision avoidance capabilities.
- **Scalability:** The modular design of the planner enabled it to handle both simpler OMPL-only scenarios and more complex OctoMap-based environments with ease.
- **Flexibility:** The planner adapted seamlessly to varied start and goal configurations while maintaining smooth, kinodynamically feasible trajectories.

V. Platform and Evaluation

Tools and Platforms to be used in the project are as follows:

- ROS Melodic LTS release
- Octomap for Global Mapping
- EDT3D library for local distance measurement and collision check
- OMPL for planner creation
- PX4 for flight control
- Gazebo Classic [3] for Simulation

VI. Limitations of the Project

- Since the main focus of the project is planning of the motion of the drone, the project would be tested and targeted for a simulation environment and not a real world environment.
- Our system would not be as fast as RL based approaches since the sequential nature of our mapping-then-planning architecture introduces additional processing time compared to RL-based methods which learn to map and plan concurrently.
- The approach followed in this paper is meant for static environments and is not very suitable for dynamic scenes.

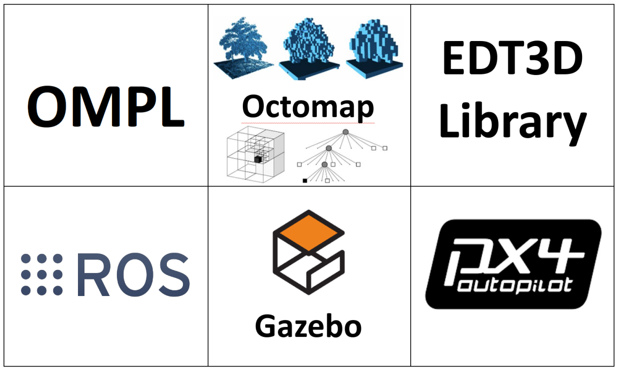

Fig. 7: Tools and Platforms

- Quality of the mapping and collision check highly depends on the sensor accuracy since sensor data is quite noisy.

VII. Break-down contribution of the team members

Task	People	Deadline
Project Idea & Research Work	Hrishikesh	Nov 18
Presentation & Interim report	Dhruv, Shreyas	Nov 18
Environment Setup	Dhruv, Hrishikesh	Nov 18
Kinodynamic Path Planner	Dhruv, Shreyas	Nov 27
Trajectory Optimization	Hrishikesh, Shreyas	Nov 27
Integration, Simulation and testing	Everyone	Dec 5

TABLE I: Task Delegation

References

[1] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees," Autonomous Robots, vol. 34, no. 3, pp. 189–206, April 2013.

[2] I. A. Sucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," IEEE Robotics & Automation Magazine, vol. 19, no. 4, pp. 72–82, December 2012.

[3] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, pp. 2149–2154, September 2004.

[4] J. Chen, K. Su, and S. Shen, "Real-time safe trajectory generation for quadrotor flight in cluttered environments," in IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1678–1685, December 2015.

[5] F. Gao and S. Shen, "Online quadrotor trajectory generation and autonomous navigation on point clouds," in IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), pp. 139–146, October 2016.

[6] W. Ding, W. Gao, K. Wang, and S. Shen, "Trajectory Replanning for Quadrotors Using Kinodynamic Search and Elastic Optimization," in IEEE International Conference on Robotics and Automation (ICRA), pp. 7595–7602, May 2018.

[7] F. Gao, Y. Lin, and S. Shen, "Gradient-based online safe trajectory generation for quadrotor flight in complex environments," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3681–3688, September 2017.

[8] L. Han, F. Gao, B. Zhou, and S. Shen, "FIESTA: Fast Incremental Euclidean Distance Fields for Online Motion Planning of Aerial Robots," in IEEE Robotics and Automation Letters (RA-L), vol. 4, no. 2, pp. 1343–1350, April 2019.

[9] H. Oleynikova, Z. Taylor, M. Fehr, R. Y. Siegwart, and J. I. Nieto, "Voxblox: Incremental 3D Euclidean Signed Distance Fields for on-board MAV planning," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1366–1373, September 2017.