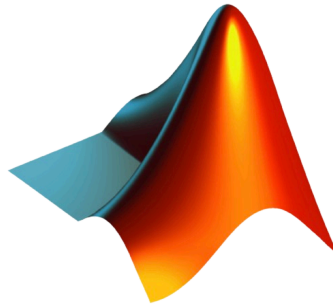


Senior Design Laboratory: Final Report

UAV Path-Planning with MathWorks



Version 1.0 - May 7, 2024

Team Members:

Jason Alvarez

Jonathan Dang

Lara de Jesus

Abraham Diaz

Marcos Olvera

Bryan Segovia

Prashant Tewary

Juan Tiguila

Kevin Velez

Erick Vergara

Faculty Advisor:

Dr. Manveen Kaur

Liaison:

Dr. Michael Thorburn

Table of Contents

1. Introduction
 - 1.1. Background
 - 1.2. Design Principles
 - 1.3. Design Benefits
 - 1.4. Achievements
2. Related Technologies
 - 2.1. Existing Solutions
 - 2.2. Reused Products
3. System Architecture
 - 3.1. Overview
 - 3.2. Data Flow
 - 3.3. Implementation
4. Conclusions
 - 4.1. Results
 - 4.2. Future
5. References

1. Introduction

1.1. Background

MathWorks is our Sponsor for this project.

Path-planning for Urban Air Mobility (UAM), such as drone deliveries, poses a critical challenge in transportation. As UAM continues to rapidly expand, there is a pressing need for advanced path-planning algorithms to optimize operations. This project aims to develop a cutting-edge algorithm, utilizing the A* (A-star) [2] algorithm, for coordinating collision-free paths for multiple drones. A* is a widely used search algorithm that efficiently finds the shortest path between nodes in a graph. The goal is to enhance UAM efficiency and sustainability by leveraging the innovative capabilities of the A* algorithm for path planning in complex urban environments.

1.2. Requirements

1. Design and optimize an efficient drone path planning algorithm.
2. Implement collision avoidance strategies to ensure safe drone navigation.
3. Validate the algorithm's effectiveness through comprehensive simulation tests.
4. Demonstrate the algorithm's capabilities in a simulated environment.

1.3. Motivation

Path planning within the realm of Urban Air Mobility (UAM), encompassing air taxis and drone deliveries, presents a pivotal challenge for the transportation sector. With the exponential growth of UAM, the need for highly efficient and optimized path-planning algorithms is poised to surge. According to the Grand View Research Report, the drone delivery market is projected to reach an estimated \$583.51 billion by 2023, while Morgan Stanley predicts the air taxi market will soar to \$1.5 trillion by 2040. The development of an efficient path-planning system for UAM has the potential to revolutionize urban transportation, rendering cities more livable and sustainable. Such an algorithm will be instrumental in orchestrating collision-free paths for multiple drones operating within the same environment, all while minimizing time and cost. This project represents a distinctive opportunity to wield cutting-edge technology in addressing the intricate challenges of path planning in UAM, thereby leaving an indelible mark on the future of transportation and logistics.

1.4. Member Contributions and Subteams

Fall Semester 2023		
Algorithm Team	Landscape Team	UAV Team
Jason Alvarez	Lara de Jesus	Abraham Diaz
Jonathan Dang	Marcos Olvera	Kevin Velez
Bryan Segovia	Prashant Tewary	Erick Vergara
	Juan Tiguila	

Spring Semester 2024			
Algorithm Team	Landscape Team	UAV Team	Testing Team
Jason Alvarez	Lara de Jesus	Abraham Diaz	Lara de Jesus
Jonathan Dang	Marcos Olvera	Kevin Velez	Abraham Diaz
Bryan Segovia	Prashant Tewary	Erick Vergara	Bryan Segovia
Juan Tiguila	Juan Tiguila		Juan Tiguila

1.5 Contrasting Our Solution with Previous Solutions

1.5.1. Optimized Efficiency with A* Algorithm

We chose to utilize the A* algorithm for its proficiency in efficiently navigating complex environments. By focusing on finding the shortest path from point A to point B, our solution maximizes time and cost efficiency for Urban Air Mobility operations. This approach directly addresses the critical need to optimize route planning in dynamic urban settings.

1.5.2. Pre-Flight Environment Scanning

Prior to drone deployment, our project employs comprehensive simulation and environment scanning techniques. This step ensures that the path planned by the A* algorithm is free from unexpected errors. By integrating environmental data into our planning process, we enhanced the reliability and safety of the drone's navigation.

1.5.3. Integration of P.I.D. Controller for Enhanced Control

To further augment the effectiveness of our drone operations, we conducted thorough testing with a PID (Proportional-Integral-Derivative) controller. This controller adjusts motor speeds based on real-time sensor data, enabling precise course maintenance. By fine-tuning the drone's responsiveness to environmental changes, our solution achieves a higher degree of accuracy and safety during flight.

1.5.4. Meticulous Decision-Making

Our project's success stems from meticulous testing and intentional decision-making at every stage. Each component, from algorithm selection to controller implementation, was carefully chosen to ensure optimal performance and reliability in real-world scenarios. This rigorous approach underscores our commitment to developing a robust solution for path planning in Urban Air Mobility.

a. Design Principles

UAV Survival: The primary objective of our project is not to ensure that the UAV strictly adheres to the path but rather to prevent it from simulating a crash scenario. This objective ensures that with each change and development decision, the UAV in the simulation will always survive by any means necessary.

UAV Path Following: The project design also guarantees that changes to the project retains the UAVs ability to follow any path given to it.

b. Design Benefits

The primary benefits of our design is that it ensures the UAVs consistency within simulation time. It also grants the ability to swap components out within the project to fit any future works applied directly to the project.

c. Experiment and Testing Results

Currently, more experiments must be conducted on the project as it assumes the UAV has only itself and a LIDAR sensor and does not account for additional weight, weather, or dynamic objects. However, it does include all necessary testing for optimal PID value configurations for the solo UAV.

2. Related Works and Technologies

2.1. Existing Solutions

Based on our thorough research, there are no existing works that tackle the same issue - doing path planning in a simulation. However, there was a deprecated model that used Unreal Engine as a simulation ground instead of the usual UAV Scenario object provided by MATLAB. We decided to move away from Unreal Engine (UE) as it was deprecated in a previous edition of MATLAB and required additional setup to even run properly. We then opted to use MATLAB and Simulink [4] exclusively as the support for their communication was consistently updated.

2.2. Techniques Used for Implementation

We chose the development techniques due to the consistency and the speed of developing our solution. It provided much-needed structure in order to provide our targeted goals for the project. The scope of this project is to simulate a UAV delivering a package from one point to another following the shortest path and avoiding any obstacles. MATLAB can do just that. Using UE or any other powerful game engine for the simulation is out of scope and requires more resources (i.e., storage). This is due to Unreal Engine being a powerful game engine. Its main purpose includes creating high-quality 3D games across various platforms, including PC, consoles, and more. It provides a comprehensive suite of tools for game development, including a visual scripting system, Blueprints, and a powerful rendering engine among others. UE offers more tools than needed for this project. Thus, MATLAB was the best option because it is resource efficient unlike UE for the purposes of this project.

2.3. Justifying Our Process

Unreal Engine had to be set to a previous edition, any edition in 4.2, as well as a previous edition of MATLAB that supported the deprecated plugins. It would have required us to forcibly create a connection between the two softwares at the same time while both systems are active, which required a powerful enough machine to even run. We chose then to use MATLAB and Simulink as it was not deprecated, and we were able to use more supported editions such as 2023B which housed a simulation speed up module which assisted us in testing.

3. System Architecture

The system architecture for this project is built off one another in order for it to fully function and come together. As a basis, the Landscape or 3D environment is needed to demonstrate the goal of this project (UAV pathfinding). With this 3D environment, we are able to add in the UAV model with LiDAR sensor. Now that we have both the UAV and landscape we can implement the A* algorithm to the UAV which will determine how we can find the most optimal route in our environment from point A to B.

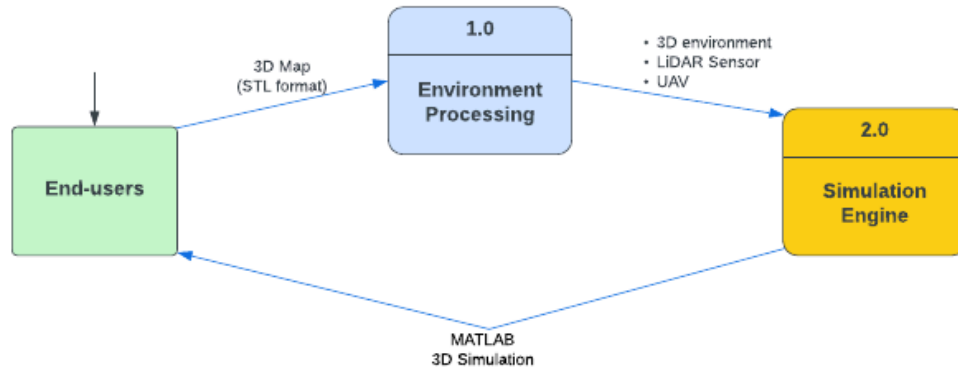


Figure 3.1. Level 0 data flow diagram.

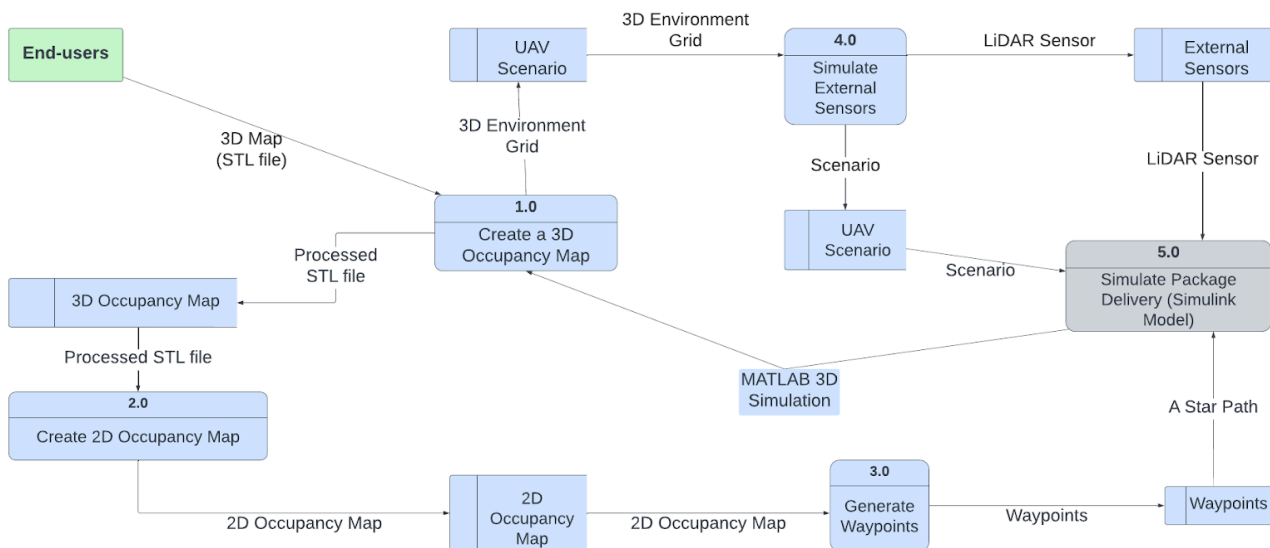


Figure 3.2. Level 1 data flow diagram.

We implemented the A* algorithm to generate an optimal path from the starting point to the destination. A* path is generated to go from point A to point B by utilizing MATLAB's built-in Hybrid A* algorithm known as *plannerHybridAStar*.

The implementation steps are as follows, see Figure 3.1:

1. Obtain STL Mapping:
Obtain an STL mapping of the desired region from open-source media or create one using 3D rendering software such as Blender.
2. 3D to 2D Occupancy Module:
Create a 2D occupancy map. This 2D occupancy map is used to run A* algorithm to generate waypoints by finding an optimal path from a starting point to a destination.
3. Configuration Setup:
Input desired coordinates, orientation, and other required variables into the Variable Config file.
4. System Execution:
Run the system.
5. Live Simulation Monitoring:
Observe the simulation live.
6. Data Capture and Analysis:
Save the spawned figure depicting the STL map, projected path, and traversed path.
Run SaveData to obtain a .csv file configuration based on current PID values.
Process data further using Testing.py.
7. PID Tuning:
Repeat steps 3 to 6, adjusting PID values, until the full PID data spectrum is obtained.

4. Results and Conclusions

4.1 Results and Findings

The results varied based on the pre-existing values of the UAVs PID controls. Essentially changing the values controlled how the UAV moved when encountering additional stimuli in the form of obstacles. Alone, having the PID set to its maximum settings appeared to have the best results when observing the UAVs movement during simulation. However, the testing did not extend to packages and weather conditions.

4.2 What Succeeded and Failed

Having the PID values at the absolute minimum, as well as within the first 25% of the lower bounds would cause a great amount of percent error when attempting to follow the path, as well as increasing likelihood of unwanted collisions with the map and obstacles. Past the 25% threshold, it appears to stabilize, decreasing deviation from the path and increasing likelihood of obstacle avoidance.

4.3 Follow-Up Research and Future Work

A few follow-up research developments required to successfully implement this project and move on to a more practical use case for this project include the following.

Account for change in weight of the UAV during flight and adjust the PID values accordingly. To successfully solve this issue, you may want to record your results and the specific PID values with varying weights of the UAV. After you have done *enough* tests, adjust the PID values during flight - increase/decrease in weight -, choose the optimal PID setting that yields the least percent error. A 4% percent error or lower is the best case. Any percent error between 4% ~ 11.5% is considered acceptable. Any percent error higher than 11.5% is not acceptable as the UAV is not following the expected path and may be deviating from this path.

Moreover, use a higher fidelity environment to simulate a more realist use case for this UAV Path Planning simulation. Use Unreal Engine (UE) from Epic Games to render the simulated environment. UE is designed to simulate complex physics (collision, humans) interaction between objects and the environment and create realistic *actors*.. Thus, a far better approach to using the simulation from MATLAB, a lower fidelity environment where the objects in the environment are not as realistic as they can be.

Lastly, move on to implementing this simulation on *hardware*. This scope is a more multidisciplinary approach that integrates aspects of aerospace engineering, electrical engineering, computer science, and systems engineering all in one.

5. References

[1] Multi-UAV Path Planning for Urban Air Mobility

<https://github.com/mathworks/MATLAB-Simulink-Challenge-Project-Hub/tree/main/projects/Multi-UAV%20Path%20Planning%20for%20Urban%20Air%20Mobility>

[2] Hybrid A* Path Planner

<https://www.mathworks.com/help/nav/ref/plannerhybridastar.html>

[3] UAV Package Delivery

<https://www.mathworks.com/help/uav/ug/uav-package-delivery.html>

[4] UAV Obstacle Avoidance in Simulink

<https://www.mathworks.com/help/uav/ug/uav-obstacle-avoidance-in-simulink.html>

[5] Lidar and Radar Fusion in UAV Scenario

<https://www.mathworks.com/help/fusion/ug/lidar-and-radar-fusion-in-an-urban-air-mobility-scenario.html>