Simulation-Based Analysis of Guidance, Navigation, and Scheduling for Advanced Air Mobility in Northern California

Alexander Aghili Zoë La Clair Andre Aledia Christine Perez Eric Vin
Daniel J. Fremont

April 23, 2025

Contents

Abstract

Urban transportation systems are increasingly strained, with traffic congestion costing billions in time, fuel, and economic losses. This proposal introduces a comprehensive evaluation framework for Advanced Air Mobility systems, using simulation to assess safety, efficiency, and feasibility across core AAM components. Our system supports evaluation of routing strategies, terminal procedures, and scheduling systems. Integrated tools provide detailed analyses of terrain hazards, glide performance, obstacle clearance, traffic conflicts, noise exposure, and energy consumption. A discrete-event scheduling simulator, coupled with an economic analysis engine, quantifies the viability of various operational models, including on-demand air taxi services. This framework enables stakeholders to rigorously test any AAM configuration using realistic flight and traffic data, ensuring informed decisions that balance safety, community impact, and long-term profitability. Designed for flexibility and extensibility, our system accelerates innovation and supports the responsible integration of AAM into the National Airspace System.

1 Introduction

1.1 Problem

Transportation in urban environments is being stretched to its limit. More than half of the world's 8 billion people live in urban areas, leading to crippling congestion that cost the U.S. economy over $190 billion in 2019 and wastes billions of hours and gallons of fuel each year [1]. The prohibitive expense of expanding ground-based infrastructure, which can exceed $1 billion per mile for new subway lines, highlights the urgent need for a new, sustainable mobility solution.

1.2 Goals of Project

This proposal presents a comprehensive system for evaluating the critical components of a proposed Advanced Air Mobility (AAM) system. By analyzing navigation routes, terminal procedures, scheduling structures, and additional operational factors, we ensure a robust, data-driven foundation for making informed decisions about AAM design and deployment. This system aims to concretely show errors in safety, efficiency, economic, and regulatory areas to improve proposed plans.

1.3 Our Approach

We designed our evaluation system to be adaptable to any proposed AAM configuration, even those beyond the scope of this proposal, by requiring only the essential design information and eliminating the need for code restructuring. Each evaluation model is organized into self-contained modules that can be easily updated, ensuring maximum flexibility and extensibility. Using Microsoft Flight Simulator 2024 for high-fidelity simulations, our framework encompasses three core evaluation systems. First, the routing system analyzes AAM routes for safety, efficiency, and how they integrate with the National Airspace System (NAS). Second, the terminal procedures system assesses operations in close-proximity airspace, ensuring safe and orderly terminal procedures. Finally, a discrete-event simulation evaluates scheduling algorithms and provides economic analysis, giving stakeholders the proposed AAM design's feasibility and profitability. This framework's adaptability will accelerate innovation for future AAM designs, enabling more safe, efficient, and economically viable solutions for advanced air mobility systems.

2 System Evaluation Overview

2.1 Components of AAM System

2.1.1 Vehicles

The choice of vehicle is critical in any AAM system. Each vehicle has different specifications, determining its range, load capacity, flight dynamics, and more. Differing dynamics dictate what procedures would be within the operating limits of the vehicle. Since MSFS 24 already has two vehicles, the Archer Midnight [Figure 1] and Joby S4 [Figure 3], included in the simulator with realistic dynamics and configurations, we used those to evaluate our system with. If one wanted to evaluate a system with a different vehicle, creating an MSFS model is possible. Any vehicle can be modeled in MSFS using manufacturer specifications and creating the model according to the documentation.

The example configuration data for the Archer Midnight is provided, which is used to inform the simulation of the aerodynamics, configurations, and systems within the vehicle. This includes exterior design, navigation systems, vehicle modes, and more [Figure 2]. The external conditions, such as weather, also realistically impact the aircraft, which is key for evaluating our system under differing conditions.

2.1.2 Vertiport Design

Vertiports form the backbone of any AAM system, ensuring seamless, convenient travel for passengers. By strategically positioning these facilities in areas that balance accessibility with environmental sensitivity, we can maximize the utility of AAM networks while minimizing ecological impact. This thoughtful approach to vertiport location is critical to delivering both operational efficiency and long-term sustainability. The placement of vertiports is determined by the designer, and, if desired, realistic scenery can be integrated into MSFS. Such additions are highly beneficial for illustrating the vertiport's impact to stakeholders and for evaluating terminal procedures. While a detailed guide to creating scenery is omitted here, you can find one at [19].

As an example, we designed a vertiport on the UC Santa Cruz campus, following our 2024 proposal guidelines, which incorporate official FAA recommendations on vertiport design [Figure 4]. This is useful for outline certain procedures but is especially useful for visual renditions [Figure 5] for important stakeholders to understand the expected impact may be in highly realistic detail.

2.1.3 Terminal Procedures

Terminal procedures will be essential for any AAM system, especially as scale increases. In traditional uncontrolled environments, aircraft avoid collision using standard best practices (eg. tear-drop entry into the pattern) and verbal communication between pilots using VHF radio. However, miscommunications, confusions, and faulty equipment mean that the rate of general aviation accidents in the terminal environment is significantly higher than in commercial operations. In instrument environments, a one-in-one-out policy is used to ensure separation in uncontrolled environments, but this significantly reduces aircraft throughput. To alleviate this pressure, standard instrument procedures and systems can be established that allow for increased throughput of aircraft while providing instrument-based separation services. Additionally, terminal procedures must be evaluated to determine the community impact and efficiency on aircraft performing these operations.

2.1.4 Enroute Procedures

Enroute is defined as the routing between any two terminal environments. For AAM system, enroute may only be 10s of miles and exist within the terminal environment of international airports. Enroute procedures may be the complex aspect of any AAM since incorporating high-density traffic density in urban areas while integrating with an already stressed air traffic terminal environment is a significant challenge. Due to the airspace constraints, dynamic routing does not make sense. Thus, all routes will be pre-established between vertiports (or general vertiport regions, such as the San Francisco Downtown region). Section 3 covers routing in AAM.

2.1.5 Business Models and Scheduling Systems

Beneath the technical complexities of implementing AAM, it's crucial to remember that long-term profitability underpins any viable solution. Although many business models exist for AAM, our work emphasizes an on-demand "Uber-style" approach, where passengers request and book flights as needed. To support this, we developed a discrete-event simulation system to test various scheduling algorithms, accompanied by an analysis tool that evaluates the profitability of each simulated scenario.

2.2 General Considerations

Several key factors must be taken into account when evaluating an AAM system, including safety, regulatory requirements, infrastructure needs, technological feasibility, and community acceptance.

A key decision is whether to operate in Visual Flight Rules (VFR), Instrument Flight Rules (IFR), or to establish to new Flight Rules system [3]. VFR is much simpler for operators as compared to IFR since the separation requirements for VFR are significantly reduced, air traffic procedures are (generally) simpler, and air traffic congestion won't cause as many delays. However, VFR systems must operate within strict weather minima, may be denied air traffic services or clearance into essential airspace, and are less protected from other traffic (thereby increasing risk). A new Flight Rules system may have to be established to accommodate the conflicting needs to decreased separation with continued operation in adverse weather.

For the initial deployment of AAM operations, VFR will likely serve as the primary regulatory framework. However, as traffic density increases, relying solely on VFR introduces significant operational challenges. The recent mid-air collision near Washington, D.C. highlights the limitations of visual separation. These concerns will only intensify as eVTOL fleets grow, making it harder for pilots to identify and maintain safe distances among numerous aircraft. Consequently, dedicated routes for low-flying traffic, especially where they intersect with existing commercial corridors, are expected to be tightly regulated and limited to areas offering greater safety margins and spacing.

2.3 Analysis Programs

2.3.1 Overview

In order to understand components of our system, we created analysis tools that provide data and visualizations from an experiment. Our analysis process is a three-step system. First, in our high-fidelity simulator, we perform an experiment such as flying a route or terminal procedure. While this is occurring, we developed a data extraction program that collects all of the relevant MSFS simulation variables[1] and stores them in a MySQL database. Then, once the experiment has been completed, we perform transformations on the data to make it useful for our analysis programs. Finally, we developed analysis programs in Matlab to provide data visualization and report generation.

2.3.2 Aircraft Data

Aircraft parameters and in-flight data are collected via Simconnect [20] from within MSFS. While there is a large amount of raw data available from MSFS, the interactions/simulation of the environment using this data is lower in accuracy compared to a full-fledged physics simulation. However, for the purposes of evaluating a proposed route under a set of configurable and realistic parameters [24], it is more than sufficient. Relevant data includes location[Figure 6], speed [Figure 7], altitude [Figure 8], attitude [Figure 9] [Figure 10], control surfaces [Figure 11], aircraft performance [Figure 12], and more [Figure 13]. While the example figures show data collected for a full route, this information is critical for other procedures to understand the expected requirements and behaviors of the vehicles during execution.

2.3.3 Terrain

With AAMs being proposed for short distance passenger transport, it is natural that the typical operating altitude be closer in proximity to the ground. To assess the risks of certain route characteristics such as cruising altitude, we have created a tool that projects an overlay onto a terrain map to indicate potentially hazardous terrain features.

Our flight simulations evaluated terrain based on elevation relative to the aircraft at a given point in time, at all times over the course of the flight. Terrain is classified into three regions. Green represents terrain more than 1000 ft. below the aircraft at the given time, yellow is between 1000 and 100 ft below, and red is any terrain feature higher than 100 feet below the aircraft. The visual representation can be seen in Figure 14, Figure 15, and Figure 16; each of which represent a different leg of this particular flight. All zone classifications are fully configurable to address different safety. The data here was acquired by comparing our flight simulation data against terrain data from the USGS[2].

2.3.4 Glide

In the event of malfunctions resulting in complete engine cutoff, we have designed a program to calculate the maximum glide distance of the aircraft relative to the surrounding terrain. The glide analysis program uses the technical details of given aircraft parameters [Figure 17] to compute the best glide speed and angle. It then uses that output information and the position and altitude of the aircraft to propagate a glide estimate, accounting for intersecting terrain. Then, given a set of aircraft positions (ϕ_i, λ_i, h_i), where:

- ϕ_i: latitude of aircraft i,
- λ_i: longitude of aircraft i,
- h_i: altitude above mean sea level (MSL) in meters.

We let

- $R(\theta)$: effective glide radius in the direction $\theta \in [0^\circ, 360^\circ]$,
- γ: glide angle (from `GlideCalculation`),
- d_{max}: maximum glide distance from `GlideCalculation`,
- $\alpha = \frac{h_i}{d_{\text{max}}}$: descent rate (m/m),
- $\theta_k \in [0^\circ, 360^\circ]$ in increments, e.g., 1°.

For each direction θ_k, compute the distance d_k where the glide path intersects terrain

$$\text{glideAltitude}(d) = h_i - \alpha \cdot d$$

$$(\phi_k, \lambda_k) = \texttt{reckon}(\phi_i, \lambda_i, d, \theta_k)$$

$$h_{\text{terrain}}(\phi_k, \lambda_k) = \texttt{getTerrainElevation}(\phi_k, \lambda_k)$$

The maximum effective distance is

$$R(\theta_k) = \max\left\{d \le d_{\text{max}} \mid h_i - \alpha \cdot d \ge h_{\text{terrain}}(\phi_k, \lambda_k)\right\}$$

For each θ_k, the boundary point is computed as:

$$(\phi_k, \lambda_k) = \texttt{reckon}(\phi_i, \lambda_i, R(\theta_k), \theta_k)$$

These boundary points form the polygon representing the reachable glide area for aircraft i. The union of all glide polygons is thus

$$\mathcal{P}_{\text{union}} = \bigcup_i \mathcal{P}_i$$

[1] A list of collected simulation variables is defined here

[2] Collected via Opentopography

where $\mathcal{P}_i$ is the polyshape for aircraft i. The union of the boundaries[Figure 18][Figure 19] creates a bubble around the aircraft which represents the reachable distance in the event of an emergency that renders the aircraft unable to sustain powered flight. Designing safe routes depends on considering safe and legal altitudes for which recovery is possible and that is what we enable with this tool.

2.3.5 Obstacle

Ground obstacles are a hazard that must be considered alongside terrain when evaluating aircraft routes. In contrast to natural terrain, obstacles are classified by the FAA as man-made structures on the ground that can represent hazards to aerial operations.The FAA classifies these structures and aggregates them into a Digital Obstacle File (DOF)[3]. This file consists of structures deemed hazardous to aerial operations. We can easily evaluate the presence of such obstacles along a supposed flight path with this data. By extending a circle out horizontally from an aircraft's latitudinal and longitudinal position, we can determine if there exists any number of obstacles within a given radius. By extending the circle downwards by a given height and also knowing the height of the structure, the obstacle in question can be determined a potential hazard. Depending on the decided flight rules and margins of safety, obstacles deemed hazards can be dealt with pilot maneuvers or adjusting proposed flight paths.

The sample vehicle flight path is shown in Figure 20 with green with objects within the 6 nautical mile zone of assessment. In our simulations, we have decided on a minimal height clearance of 500 feet vertical and a hazard radius of 0.1 latitude, equivalent to 6 nautical miles to demonstrate a zone of potential man-made hazards throughout a given flight route.

2.3.6 Energy

Energy consumption for an eVTOL is a function of throttle input and several environmental and vehicular factors. Primary factors that were evaluated include, but are not limited to, engine throttle, engine rpm, ambient air pressure, ambient air temperature, and vehicle velocity. In our testing, we saw across several tests that energy consumption was highest during ascent, almost constant during level flight, and lowest during descent. Energy consumption through simulation was therefore, as expected, proportional to the throttle input to the vehicle. Environmental factors such as air density and temperature reflect a vertical shift in the energy consumption of the eVTOL platform across the full duration of the flight.

Figure 21 detail energy statistics. The red line plots raw power consumption at discrete points over the simulation. The green line follows an energy consumption rate calculated from the average over the span of a minute. We believe the oscillations in energy consumption to be due to the vertical motors on the Midnight Archer, for which the simulation used, to be intermittently spinning up and down as needed.

2.3.7 Noise

Noise is a critical component of the analysis as the community impact is essential to consider. Unlike traditional airports, which usually reside in industrial areas and have mitigating strategies for noise impact, vertiports are likely to be placed directly next to residential communities with strict noise ordinances.

To show what the noise impacts may be in a given configuration, we developed a noise analysis tool which creates a contour plot of noise overlayed on the geography [Figure 22]. This way, the decibel noise impacts are clear when examining a procedure. Similar to previous systems, initial configurations are defined and then a mathematical approximation is performed [Figure 23]. This is done by first generating a 2D cartesian grid

$$(x_m, y_m) \in \{-r, -r+\Delta, \cdots, r-\Delta, r\}^2$$

with r being the simulation radius and Δ the grid resolution. The slant distance to the ground is then calculated for each grid

$$d(x,y) = \sqrt{x^2 + y^2 + (h_{\text{aircraft}} - h_{\text{terrain}}(x,y))^2}$$

A simple linear attenuation model is used to simulate atmospheric absorption:

$$A_{\text{atm}}(x,y) = \alpha \cdot d(x,y)$$

where $\alpha = 0.1$ dB/m is the atmospheric attenuation coefficient. The received sound pressure level at ground points is given by:

$$L_{p'}(x,y) = L_w - 20\log_{10}(d(x,y)) - A_{\text{atm}}(x,y)$$

and

$$L_p(x,y) = \max\left(L_{p'}(x,y), 30\right)$$

to avoid negative infinity values. The source sound power level L_w is estimated as:

$$\begin{aligned} L_w =& L_{\text{base}} + A + B\log_{10}(M_{\text{tip}}) \\ &+ C\log_{10}(Re) + 10\log_{10}(N_{\text{rotors}}) \end{aligned}$$

where $A = 80$, $B = 15$, $C = -10$ are approximated constants [16], M_{tip} is the tip Mach number, Re the Reynolds number, and N_{rotors} the number of rotors.

[3] FAA Digital Obstacle File

While our initial noise tool is sufficient for our analysis, much more could be done with it, as estimated noise impact is critical for any regulatory approval. The model should be improved by refining it to create better source noise estimates based on propeller angle, vertical rate, and more. The propagation system should account for monopole, dipole, and quadrupole waves and the associated frequencies to understand the expected types of disturbances.

2.3.8 Traffic

To ensure that AAM vehicles have minimal impact on the existing NAS, we developed a tool to assess how proposed procedures interact with current air traffic. Using publicly available historical ADS-B data[4], we capture all recorded air traffic in a time-window and filter it by region. For instance, Figure 24 and Figure 25 represent all air traffic in the NorCal region on December 30, 2024. The tool allows users to define a conflict zone, for example, a radius of 3 nautical miles and 1000 feet in altitude. It then evaluates a given MSFS flight path against the ADS-B dataset to identify any conflicting traffic. To measure distance between two geographic points using latitude and longitude, we approximate

$$d(\phi_1, \lambda_1, \phi_2, \lambda_2) \approx$$
$$69 \cdot \sqrt{(\phi_2 - \phi_1)^2 + \left[(\lambda_2 - \lambda_1) \cos\left(\frac{\phi_1 + \phi_2}{2}\right)\right]^2}$$

where ϕ is latitude in degrees, λ is longitude in degrees, and distance is in miles (using average of 69 miles per degree of latitude). Let ξ be the cutoff radial distance and ψ be the cutoff altitude. For each VTOL aircraft position (ϕ_v, λ_v, h_v), a set of external flight tracks (ϕ_t, λ_t, h_t) is tested. A track is matched if:

$$d(\phi_v, \lambda_v, \phi_t, \lambda_t) < \xi \text{ miles}$$
$$|h_v - h_t| < \psi \text{ feet}$$

If both conditions hold for any point in a trace, the entire trace is added to the set of matched points P. Finally,

$$P = \bigcup_{\substack{i \\ \exists t \in T_i \\ d(\phi_v, \lambda_v, \phi_t, \lambda_t) < \xi \wedge |h_v - h_t| < \psi}} T_i$$

where T_i is a flight track (a sequence of lat-lon-alt points). The specific points that are within the parameters are marked as red dots in the visualization. Users can choose to view only the conflict points or include the full flight path of any aircraft involved in a conflict. This helps provide deeper insight into the nature of the conflicting traffic and the types of operations affected.

As an example usage, we have this route from Berkeley to Merced [Figure 26]. As shown, there are certain conflict areas highlighted in red [Figure 27]. We can see the full trajectories of the conflicts [Figure 28]. In this example, this route seems to conflict heavily with Livermore departures and arrivals [Figure 29]. Thus, adjusting the route to avoid this area would be recommended. Using this methodology, one can design routes that reduce conflicts with existing traffic.

3 Routing

3.1 Routing in AAM Systems

The National Airspace System (NAS) is a complicated, highly regulated system. Integrating new eVTOL traffic is challenging due to the constraints of urban terminal environments, particularly in Northern California, which encompasses one Class Bravo airspace, two Class Charlies, and several Class Deltas [Figure 30]. Dynamic routing would intersect too frequently with existing air traffic [Figure 31]. Consequently, carefully designing routes manually is optimal for minimizing conflicts. Nevertheless, even manually created routes should be evaluated according to safety and efficiency metrics to understand their performance and compare them with alternative proposed routes.

The criteria in selecting VFR and IFR routes differs immensely. VFR aircraft have greater barriers to enter airspace such as Bravo Clearances and Two-Way Radio Communication [14]. ATC may also not prioritize VFR traffic during times of heavy workload [13]. As a result, where possible, VFR routes should remain clear from class B, C, and D airspace. IFR routes, on the other hand, require more separation (3 miles and 1000 ft usually), thereby restricting the possible routes that meet the proper separation from existing traffic. However, these routes will also allow operations in adverse weather conditions and provide access to tightly controlled airspace, making them more reliable but less flexible than VFR routes.

All of these considerations play a key role in route development. For our example evaluation, we designed flight paths guided by these criteria, using the 2024 proposal as a framework. The following section outlines how these paths are assessed.

3.2 Routing Evaluation

Although a static, no-simulation evaluation of the routing system is possible, we chose to fly each route once[5] to capture detailed performance data.

[4] Collected via readsb and downloaded from adsb.lol

[5] At standard pressure (29.92 inHg) and standard temperature (15°C).

Once a route has been created, it must be encoded to be flown in MSFS. MSFS uses a .PLN [17] format whose formal specification is defined here. Alternatively, a flight plan can be created using the MSFS flight planner tool [18] [Figure 32]. Routes are loaded into the aircraft and flown using the autopilot system for deterministic results.

After a route has been created and tested, it is analyzed using the analysis programs. We examine the capabilities of testing and comparing two routes through a study of a Santa Cruz to Berkeley routing.

A route from Santa Cruz to Berkeley is an excellent case study to examine routing structures. A straight path [Figure 33] between the two locations goes straight through the San Francisco final approach path and directly over Oakland Airport. Thus, careful route and altitude constructions must be considered to avoid existing traffic.

To navigate this area, a route that flies over the midspan of the San Mateo bridge has been proposed [Figure 34] [Figure 35]. This route is typically flown by VFR aircraft attempting to cross the bay while avoiding commercial traffic from San Francisco and Oakland. When examining a route, we will look at its energy usage [Figure 36], its glide distances [Figure 37], and more. Taking a close look at the glide range [Figure 38], it does show that when flying over the San Mateo bridge at 1400 feet, there may be few options to glide to when an engine failure occurs. Thus, altering the route to remain more aligned with landing options could increase the safety of the operations. A key point, in this case, is to ensure that proposed routes either have proper separation from other aircraft already or can be realistically turned into IFR routes in the future. As shown [Figure 39], we see that with IFR separation, the proposed route has several major conflicts [Figure 40] with SFO arrivals to 28L/R and OAK 30 + 28L/R arrivals and departures. Reducing the separation to 2 miles and <500 feet shows [Figure 41] that the potential conflicts are greatly reduced[6], though still existent, raising the potential for TA/RAs.

Additionally, a conflict with OAK go-arounds on Runway 30 presents a challenge, as the eVTOL paths cross directly over the OAK runways at 1,200 feet. One approach to incorporating safety in this portion of the route is to require the eVTOL to hover-wait for traffic on final to OAK. Using Scenic, we generated various flight configurations and wait times to assess when the procedure could be performed safely. To do this, we wrote a Scenic[22] program [Figure 42] which had an Archer Midnight start over the San Mateo bridge and, after a random time, would begin flight to cross over the OAK 30 runway numbers at 1200 feet [Figure 43]. The other vehicle in the simulation is a Boeing 737 Max 8 on the ILS 30 at OAK [Figure 44]. Then, at a random time along its final path, it would go around at a randomized climb rate. Running this over many iterations yielded many results where the vehicles would get too close, within 1 mile and 500 ft [Figure 45]. After many iterations, it was determined that the earliest the eVTOL can safely begin moving toward the OAK Runway 30 threshold is when the 737 is 1.75 miles from the touchdown point.

In the case that IFR separation is required, the above route will not work by our analysis. In order to alleviate this, an IFR route [Figure 46] [Figure 47] is proposed which bypasses the dense traffic by routing around the dense zones, thereby adding additional time to the route. Indeed, conflicts are reduced [Figure 48] except for SFO 28L/R departures [Figure 49] (the route should be altered to avoid this) and for VFR tours above San Francisco (which can also be altered). Thus, this route has much greater separation and is a candidate for a future IFR route. However, it does come at the cost of 5 additional minutes on a previously 30 minute route, a 16% increase in travel time. This route also incurs at 10% increase in energy consumption [Figure 50] due to the additional time enroute, though it remains at higher altitudes throughout the flight, thereby being more efficient enroute. Finally, this route overflies the ocean [Figure 51][Figure 52], far from many suitable landing spots, and therefore introduces more risk. This should be another factor considered when choosing the final routes.

This routing analysis provides a framework for how data can inform design decisions. Of course, this is just one example—similar analyses should be conducted for a variety of proposed routes. Another important factor is the operational ‘plan’ in use at major airports. For instance, SFO may operate under East Plan (departing from Runway 10 and landing on Runway 19), which affects safety margins. While a thorough analysis should account for such variations, we do not include it here, as East Plan accounts for less than 20% of SFO operations.

4 Terminal Procedures

4.1 Terminal Procedures in AAM Systems

Terminal procedures are a critical component of any AAM system. Given that the terminal area typically experiences the highest density of traffic, the risk of collision is significantly elevated. Identifying throughput limitations and delays within the terminal environment is essential; not only for ensuring safe operations, but also for optimizing energy efficiency and minimizing passenger delays. AAM systems are expected to differ from traditional terminal procedures due to VTOL requirements and the design of vertiports [11]. Instead of long approach procedures with

[6] The red conflicts following the route along the bay is likely existing VFR traffic performing the same operation, which can be safely ignored since its a small number of conflict aircraft (2) and visual separation would be established.

aircraft following an extended runway centerline, VTOL aircraft are expected to approach in a tiered ring structure [Figure 53] [Figure 54] designed to provide coordinate flow to the Takeoff and Landing Pad (TOLP).

MSFS uses XML [21] to create terminal procedures which can be loaded onto aircraft and flown (via autopilot or manually). These procedures can use radial distance-to-fix as defined here or any other valid routing structure. This allows the procedures to be defined exactly as it could be for real aircraft and flown as such.

4.2 Terminal Procedure Evaluation

To evaluate terminal procedures, we implemented a multi-pronged approach that combines high-fidelity simulation with scalable traffic assessments. The core of our methodology involves encoding individual terminal procedures, such as Standard Instrument Departures (SIDs) and Standard Terminal Arrival Routes (STARs), into MSFS allowing us to assess both procedural performance and operational feasibility in a controlled environment.

Once encoded into MSFS, each procedure is flown using the simulator's autopilot and flight management systems. This enables detailed data collection, including the actual flight path, adherence to altitude and speed constraints, turn performance, and automation behavior. Additionally, the simulation allows for monitoring of fuel burn, energy profiles, and time-to-climb or descent metrics, which are critical for evaluating efficiency and environmental impact.

For instance, this [Figure 55] Joby defined departure procedure [12] was manually transcribed into MSFS's custom approach editor. The aircraft was then flown under fully automated conditions to mimic realistic pilot and system responses. The trajectory closely follows the published chart [Figure 56], with precise lateral and vertical navigation observed throughout the departure phase. We were then able to collect relevant data about the departure [Figure 57][Figure 58].

Beyond single-aircraft assessments, we also conducted multi-aircraft evaluations using Scenic. In this setup, several aircraft were initialized from distinct starting points around the terminal area, each assigned a pre-defined trajectory that intersects or merges into a shared airspace, such as a final approach path to a single runway.

These traffic scenarios allowed us to evaluate airspace capacity, sequencing strategies, and conflict resolution effectiveness. By varying initial spacing intervals and route convergence points, we empirically determine the minimum required longitudinal and lateral separation to maintain safe and efficient flow during high-density operations. This analysis provides insights into optimal scheduling windows, route deconfliction strategies, and the robustness of proposed procedures under stress-tested conditions.

5 Scheduling System

5.1 Scheduling System Overview

While procedures and routes are critical to any UAM system, efficient flight scheduling is essential to prevent vertiport congestion, minimize passenger delays, and maintain economic viability [7]. In order to assess scheduling, we built a discrete-event simulator which acts as a scheduling evaluator. Our simulation takes in a pre-defined state (defined in both environmental and system configurations) and evaluates a given scheduler. We then use an analysis program to view various metrics from the simulator and provide an economic analysis to determine the profitability of a given system and scheduler [Figure 59].

5.2 Configuration

5.2.1 System Configuration

In the scheduling simulator, all system configurations are defined using TXT or CSV files containing data about the system. These include:

1. vertiport.txt: The possible vertiports in the system. [Figure 60][Figure 61]

2. starting_state.txt: The starting location of each eVTOL in the system. [Figure 62][Figure 63]

3. transport_time.txt: The flight time between each vertiport (in minutes). [Figure 64]

4. passenger_demand.csv: The number of passengers that show up in a defined time-interval seeking to fly from a source to a destination. [Figure 65][Figure 66]

5. ground_transport.csv: The associated public transport times at each location. [Figure 67]

Exact specifications and examples for each file is detailed in the figure attachments for each file.

5.2.2 Environmental Configuration

Environmental configurations are configurations of the system that can be gathered using outside data, such as from MSFS or public data. In the current scheduling simulator, environmental configuration is the transportation times between the different vertiports. For our system, we flew each of the routes [Figure 68] in MSFS to get an expected transport time, instead of manually calculating appropriate values. In future iterations, more environmental conditions such as wind patterns could be included. The resultant transportation times can be found at Figure 64 .

5.3 Scheduling Simulation

Our scheduling simulation was designed for easy extendability and analysis. While this section will outline how our current simulation works, we emphasize that additional analysis and complexity can be added with relative ease due to the simple design.

The system revolves around a time-based priority queue, the event queue, which holds the events for the simulation ordered by lowest time.

There are three main entities in the simulation: Vertiports, Passengers, and Aircraft. Vertiports do not represent discrete events; rather, they maintain an informed state of the system. This helps collect vertiport specific statistics, manage passengers and aircraft, and inform the scheduler of the state of the Vertiports within the system.

Passengers [Figure 69] have three events which represent their state. When a passenger first arrives at a vertiport, they register a PassengerBook event, entering the system with an origin and destination vertiport. A passenger is then loaded onto an aircraft by the scheduler and is officially departed when the aircraft departs which is registered as a PassengerDepart event (although this isn't an independent event on the queue). Finally, upon arrival at the destination, a PassengerArrive event is collected.

Aircraft [Figure 70] begin in the idle state after initialization. The scheduler can load passengers onto the aircraft (which are automatically deboarded after arrival to destination), depart an aircraft, or charge the aircraft. The events are registered as appropriate with ChargeStart, ChargeEnd, AircraftDepart, and AircraftArrive.

At initialization, the simulation populates Vertiports with Aircraft as specified from system configuration. PassengerBook events are populated within the event queue according to the passenger demand graphs, plus some randomization within each time interval. Once the simulation is initialized from the system configuration, the simulation is run.

On each step, the simulation retrieves the next event from the event queue and alters the system state as necessary. The system state is then sent to the scheduler which, with a decision policy, will make choices to add events (eg. load passengers onto a flight and depart it), which it executes using the action commands. Events the scheduler creates are put onto the event queue and the simulation continues to the next step [Figure 71]. The simulation can continue until a pre-defined end time (eg. 8 hours of simulation to simulate a day of opertions) or until all passengers have reached their destination.

When handling events, the simulation will write events to an event log[Figure 72]. This event log is a reflection of all the events in the simulator, acting as a recreation of the simulation performed. This log is used by the analysis program to collect statistics about the simulation run, which is described in detail in the analysis section.

5.4 Scheduling Analysis

5.4.1 Analysis Outputs

To evaluate the scheduling system's performance, we collect data from the event log [Figure 72] and then aggregate and transform it into key performance metrics.

First, we determine the wait times for passengers. To display wait times, we show each passengers wait time[Figure 73], the average wait times across all passengers, and a histogram[Figure 74] of wait times. We evaluate throughput (arrived passengers per hour) using a histogram [Figure 75] and average throughput per hour. We also evaluate the average flight load [Figure 76] [Figure 76]. The wait times, throughput, and flight load provide information on how efficient the scheduler is[Figure 78]. The passenger book rate [Figure 79] is also provided as a means of comparing the throughput of the system against the input into the system. If the throughput is less than the input, delays will accumulate and the system is likely overcapacity (for the given scheduler). We also want to evaluate the cycles. A cycle is the time between two arrivals of aircraft. Since aircraft are in one of three states, idle, in-flight, or charging, we can determine the amount of charging, idle, and flying time per cycle [Figure 80]. This shows how much time is needed for charging, how much time is spent in idle, and average flight lengths per cycle. This provides further insight into how efficiently aircraft are being utilized by the scheduler.

Finally, we provide a scheduling timeline [Figure 81] [Figure 82] to show visually how aircraft are scheduled throughout the simulation.

5.4.2 Cost and Economics

To complement our operational performance evaluation, we conduct a financial analysis [4] that estimates the economic viability of the scheduling system under specific configurations. This analysis is based on both static input data (such as capital expenditures, operational costs, and revenue models) and dynamic outputs from the discrete-event simulation (i.e., aircraft movements, passenger volumes, and resource utilization).

We begin by estimating the capital expenditures (CAPEX), which include infrastructure investments like aircraft acquisition, charging stations, and other ground facilities. These assets are depreciated annually based on their expected useful life, giving us an annualized depreciation cost that contributes to the total overhead.

Operating expenditures (OPEX) are broken down into several components:

- Fixed Annual Costs: These include recurring expenses such as pilot training, administrative overhead, software systems, marketing, and compliance.
- Labor Costs: The number of pilots, flight operators, and ground staff are estimated based on the fleet size, and their corresponding salaries are accounted for annually.
- Per-Aircraft Costs: These include routine maintenance and insurance, scaled with the number of aircraft in operation.
- Variable Costs: These are directly tied to operational intensity and include landing fees (per flight) and energy costs (calculated from estimated energy consumption per mile).

Revenue projections are derived from the simulation's flight logs. For each route taken during the simulation, ticket sales are calculated based on the number of passengers and pre-defined ticket prices per origin-destination pair. This gives us a granular view of revenue generation at the flight level, which is then scaled to project annual revenue.

By aggregating costs and revenues, we estimate key financial indicators:

- Operating costs before depreciation: A baseline for assessing cost efficiency.
- EBIT (Earnings Before Interest and Taxes): Operational profitability before taxes and capital costs.
- Tax obligations: Estimated based on EBIT and an applicable tax rate.
- Net income: Final indicator of system profitability under the simulated scenario.

To display the financial models capabilities, we examine differing ticket price estimates on a given simulation and cost structure to determine the finances. We also show how changing cost inputs alters the cost structure.

For this, we ran a simulation with 15 aircraft, 30 pilots, and several ground and ops crew. OPEX [Figure 83] and CAPEX [Figure 84]information is estimated and run through the financial model with the log data from the simulation. We evaluate three ticket pricing models, without due regard for the demand outcomes. While in reality, demand and pricing will have significant effect on each other, our financial model is designed to show financial outcomes assuming that demand from the passenger demand would be met with the given financial data. This model will help give rough starting estimates for which more advanced models can be refined from.

We examine cheap, moderate, and expensive ticket prices. With cheap ticket prices [Figure 85] [Figure 86], each flight will cost more than the revenue it brings in. With moderate prices [Figure 87] [Figure 88], while the cost structure remains the same, the revenue increases. Thus, net income increases into the black. Similarly, with expensive ticket prices [Figure 89] [Figure 90], a full 10x the cheap prices, the net income is substantial with profit margins of about 60%.

However, in the previous system, the scheduler booked flights that may be less than 100% capacity [Figure 91], leading to less wait times but flights that occur with less revenue. Thus, we provide an example were the scheduler restricts departing flights to 100% capacity [Figure 92]. As a result [Figure 93], operating costs are lower as flights don't get booked under capacity and revenue per flight is increased. Final annual net income estimates are $2M higher than previously (though still negative).

This financial modeling is designed to be flexible, allowing for rapid sensitivity testing of different configurations, such as increased fleet sizes, altered scheduling systems, or changes in pricing strategies. This cost and economic analysis provides a critical lens for evaluating whether a particular scheduler and operational setup can sustain long-term financial viability within a AAM framework.

5.5 Scheduling Algorithms

The purpose of the scheduling evaluator is to, given a scheduling algorithm, evaluate its performance. Thus, some time should be spent on the scheduler itself. In order to test our evaluator, we developed a couple scheduling algorithms as a baseline for which much improvement can stem. In this section, we show how the scheduler interfaces with our simulation and some of the theoretical foundations of UAM scheduling systems that were considered.

There have been many proposed scheduling algorithms, including Reinforcement Learning [5], Heuristic-Based[6], and Time-Based Flow Management [10]. For example purposes, we wrote a naive scheduler with a simple, deterministic rule set, and a reward-based scheduler which uses a reward function to determine next actions.

5.6 Naive Scheduler

The `NaiveScheduler` implements a straightforward rule-based strategy:

1. For each vertiport, identify all idle aircraft.
2. Group waiting passengers by their destination.
3. Select the destination with the largest passenger group.

4. Load as many passengers as possible onto the aircraft.

5. Schedule a flight to the chosen destination if the aircraft has sufficient battery.

If no suitable trip is found due to battery constraints, the aircraft is scheduled for a default 30-minute charging session. This scheduler prioritizes throughput by always serving the most popular routes, but does not account for passenger waiting times or future demand forecasting.

5.7 Reward-Based Scheduler

To address the shortcomings of the naive approach, the `RewardScheduler` employs a heuristic reward function to evaluate the utility of potential flights. This scheduler ranks destination groups described as follows:

Each feature contributes to a composite reward score using a customizable weight function:

Table 1: Scheduler Definitions

Symbol	**Definition**
V_i	Vertiport i
r_{ij}	Route from source V_i to destination V_j
$\text{ETA}(\text{dep}_t, r_{ij})$	Estimated Time of Arrival leaving at time dep_t on route r_{ij}
$\text{NGT}(t, V_i)$	Next Public Transportation departure time after time t at Vertiport V_i
$\text{TP}(t, r_{ij})$	Total passengers waiting at V_i for route r_{ij} at time t
$\text{HL}(t, r_{ij})$	Longest currently waiting (passenger) time at V_i for route r_{ij} at time t
$\text{AL}(t, r_{ij})$	Average current waiting (passenger) time at V_i for route r_{ij} at time t
$\text{TP}(t, V_i)$	Total passengers waiting at V_i (for any route) at time t.
$f_k(x)$	Weight function f (version k) for reward function.

$$\begin{aligned} R(t, r_{ij}) = {} & f_1(\text{TP}(t, r_{ij})) + f_2(\text{HL}(t, r_{ij})) \\ & + f_3(\text{AL}(t, r_{ij})) + f_4(\text{NGT}(\text{ETA}(t, r_{ij}))) \\ & + f_5(\text{TP}(t, V_j)) \end{aligned}$$

The functions $f_k(x)$ can be designed to prioritize a metric for the scheduler, such as the wait times or the flight load.

The destination with the highest reward is selected, subject to aircraft battery constraints. If no feasible destination exists, the aircraft is scheduled to charge for a dynamically calculated duration based on the average future transport need, capped between 15 and 90 minutes.

While the NaiveScheduler performed better than expected, the configurability of the Reward scheduler made testing different scenarios much easier and thus more optimal schedules were designed with Reward scheduler.

6 Conclusion

In this proposal, we have outlined a modular framework for evaluating AAM systems using MSFS as a high-fidelity simulation environment. Our system enables detailed analysis of aircraft behavior, route viability, terminal procedures, scheduling efficiency, and broader operational and environmental considerations such as noise, energy usage, and traffic conflicts. Each tool was designed to be extensible, adjustable, and data-driven, ensuring flexibility across various AAM designs and configurations.

From terrain clearance and glide range assessments to noise modeling and discrete-event scheduling simulation, each component provides critical insights necessary for stakeholders to make informed decisions. Moreover, by enabling integration of real-world data and modeling complex interactions with the NAS, our framework ensures proposed systems are not only technically feasible but also operationally sound and economically viable.

There are several areas for improvement to enhance model accuracy, accommodate more complex systems, and incorporate varying rules. Nevertheless, the work completed in this proposal provides a strong foundation for developing a comprehensive system that simplifies AAM deployment.

As urban air mobility moves closer to mainstream deployment, tools like ours will be essential for ensuring safety, efficiency, and community acceptance. We hope that this framework will accelerate the research, development, and responsible deployment of AAM solutions that are sustainable, scalable, and ready to meet the mobility demands of the future.

7 Figures

7.1 Section 2: System Evaluation Overview

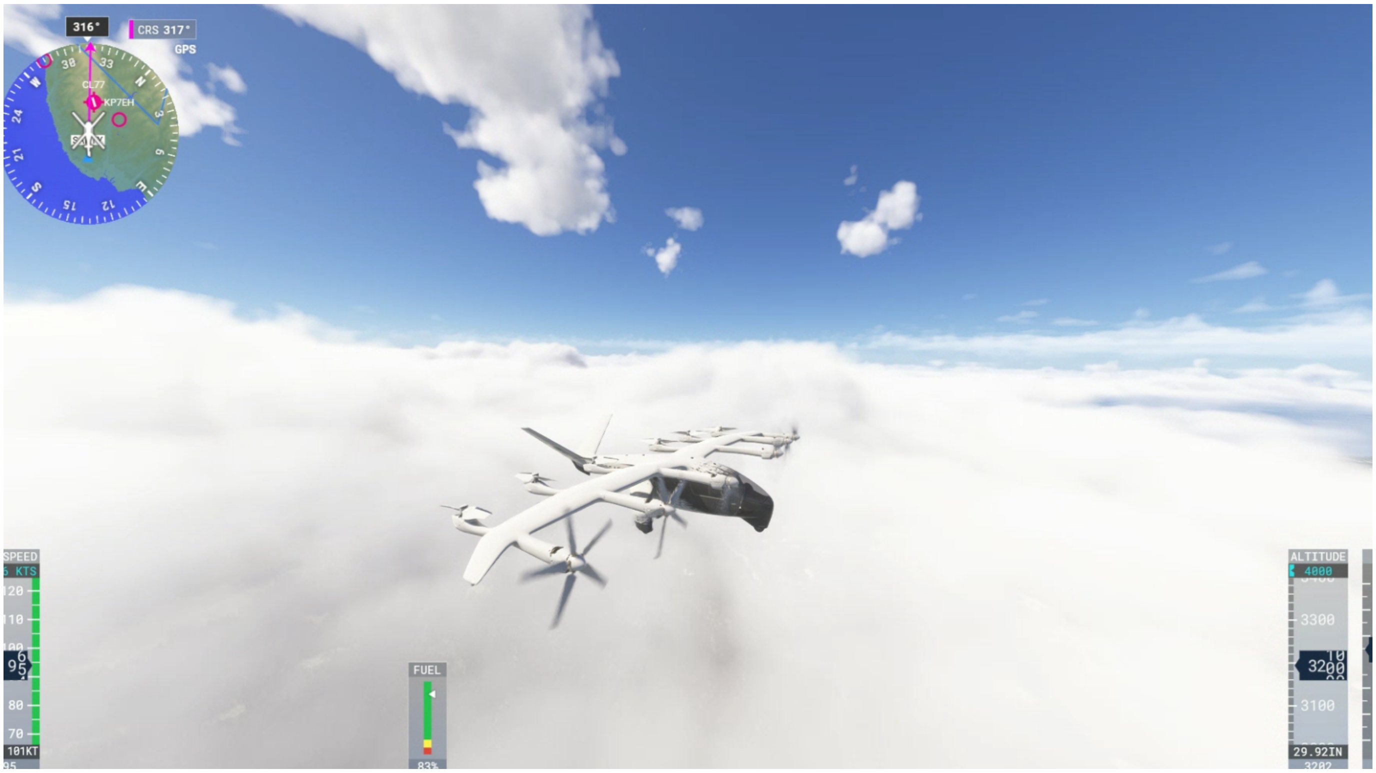


Figure 1: Archer Midnight in MSFS

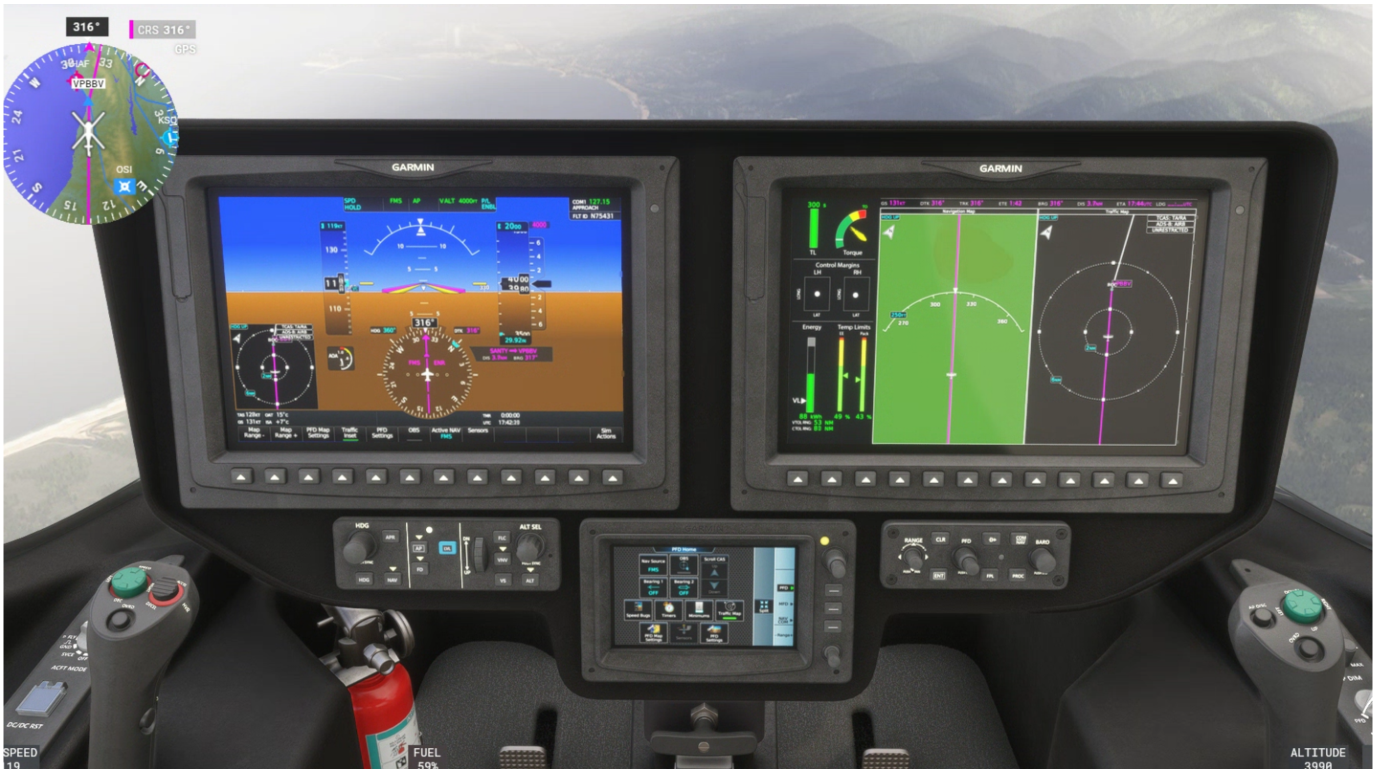


Figure 2: Archer Midnight Cockpit in MSFS

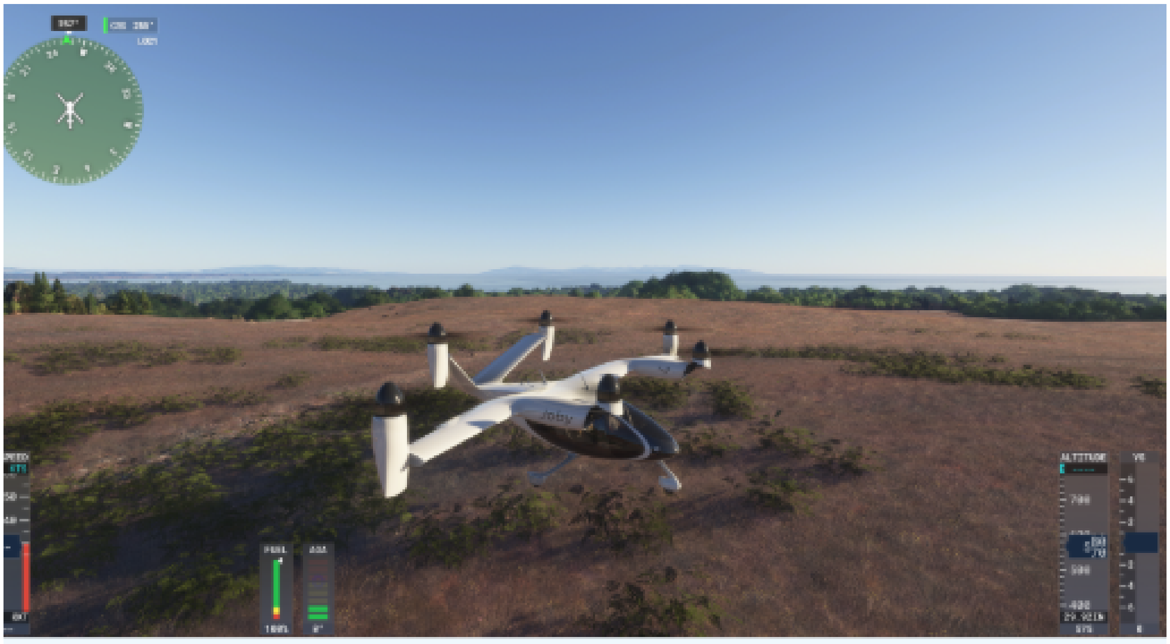


Figure 3: Joby S4 in MSFS

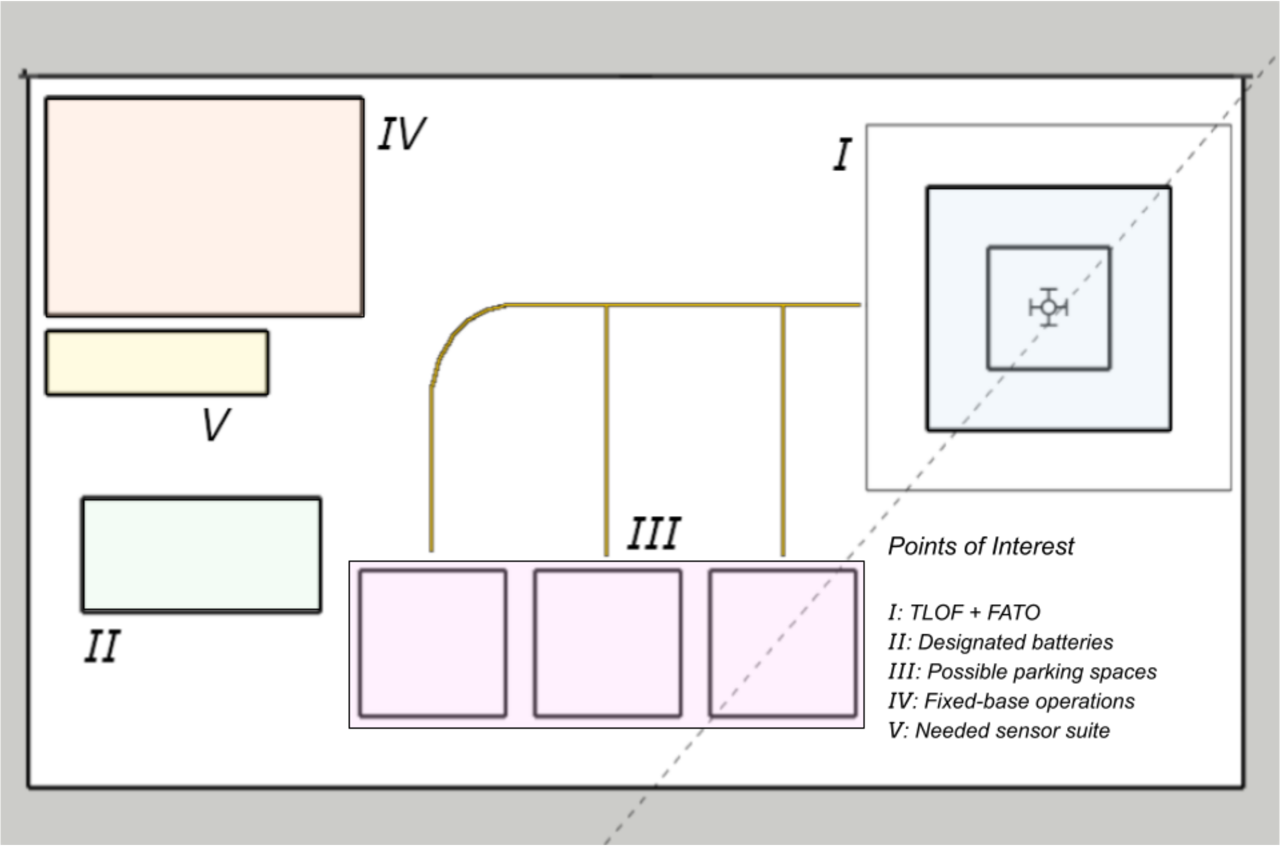


Figure 4: Santa Cruz Vertiport Design

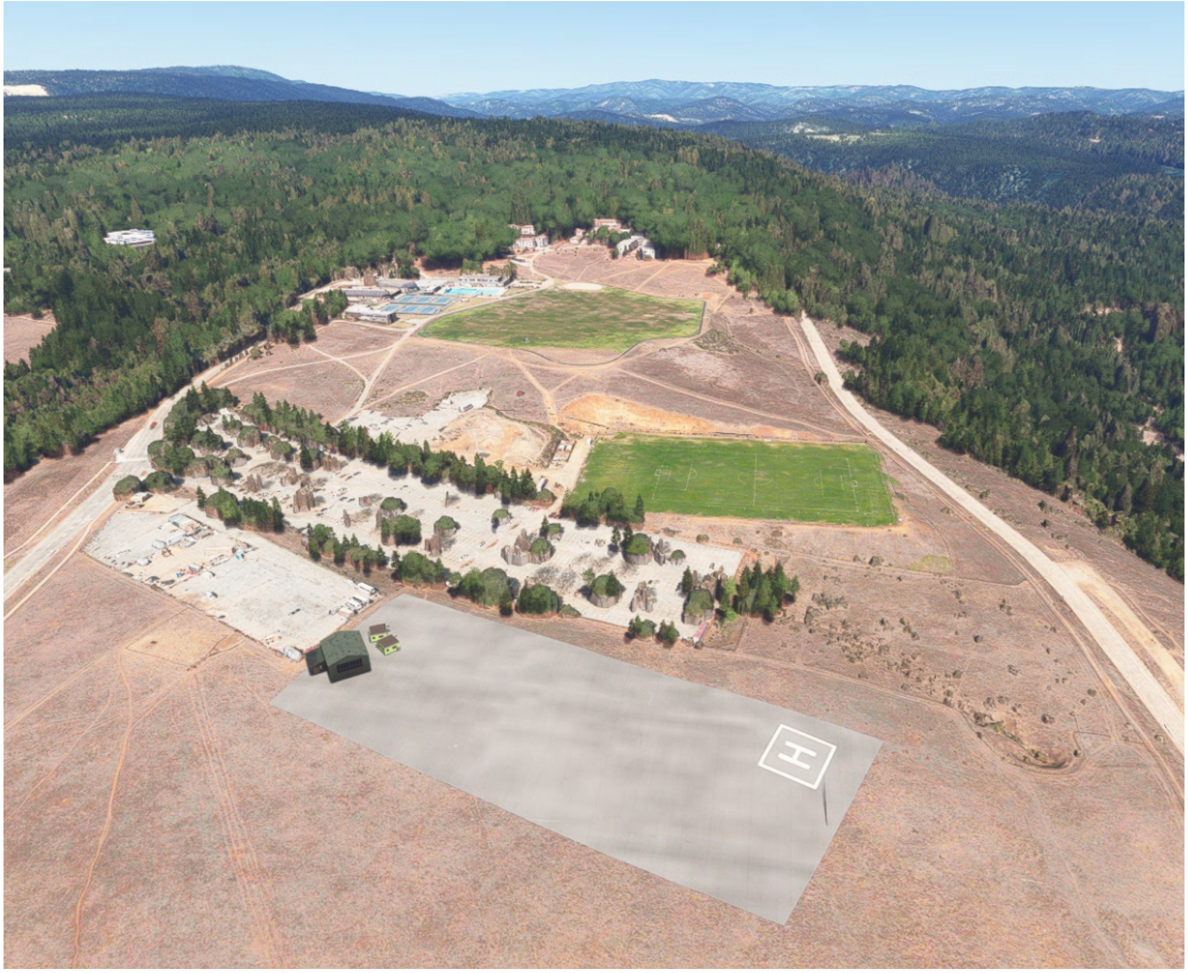


Figure 5: MSFS Scenery for Santa Cruz Vertiport

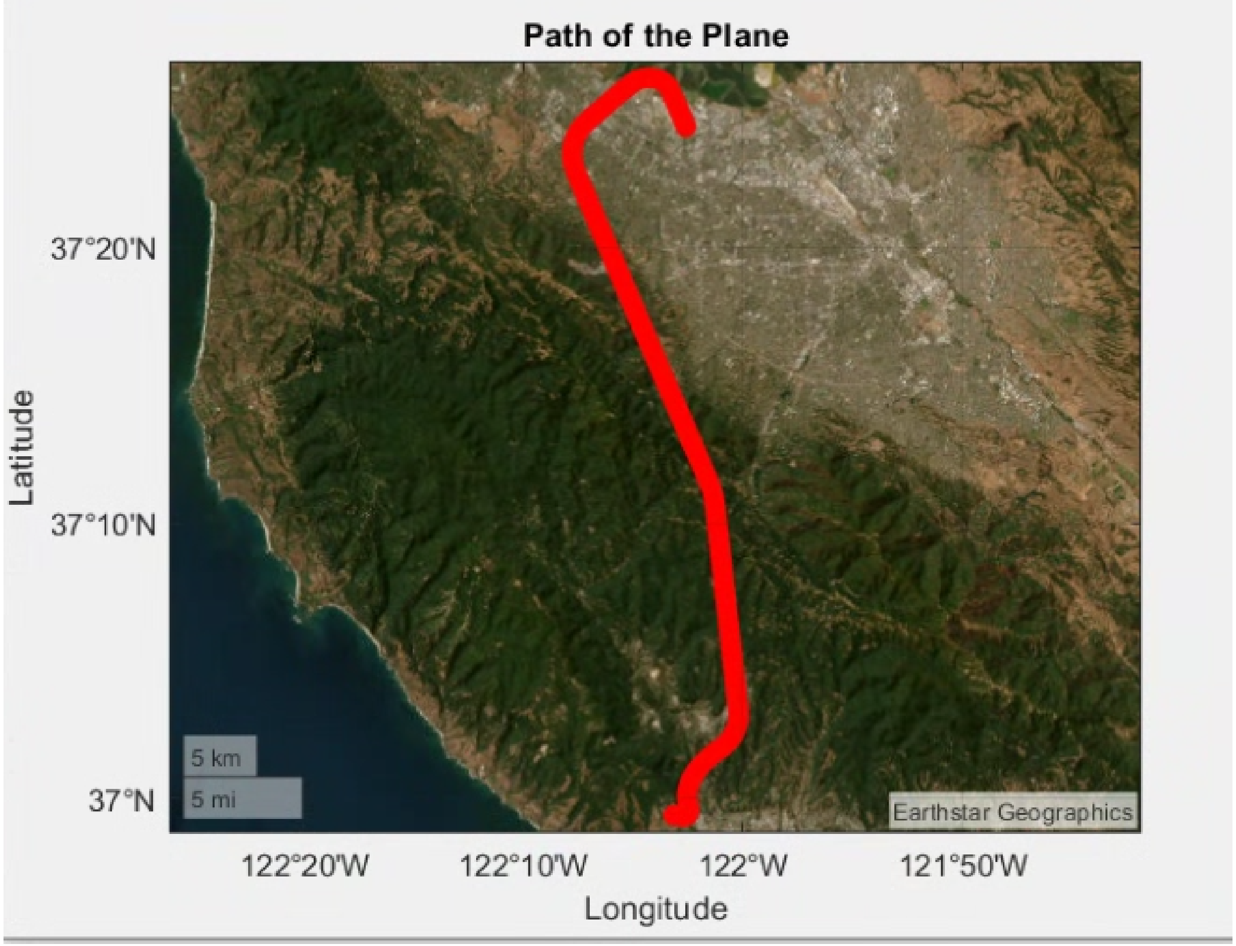


Figure 6: Example Aircraft Path

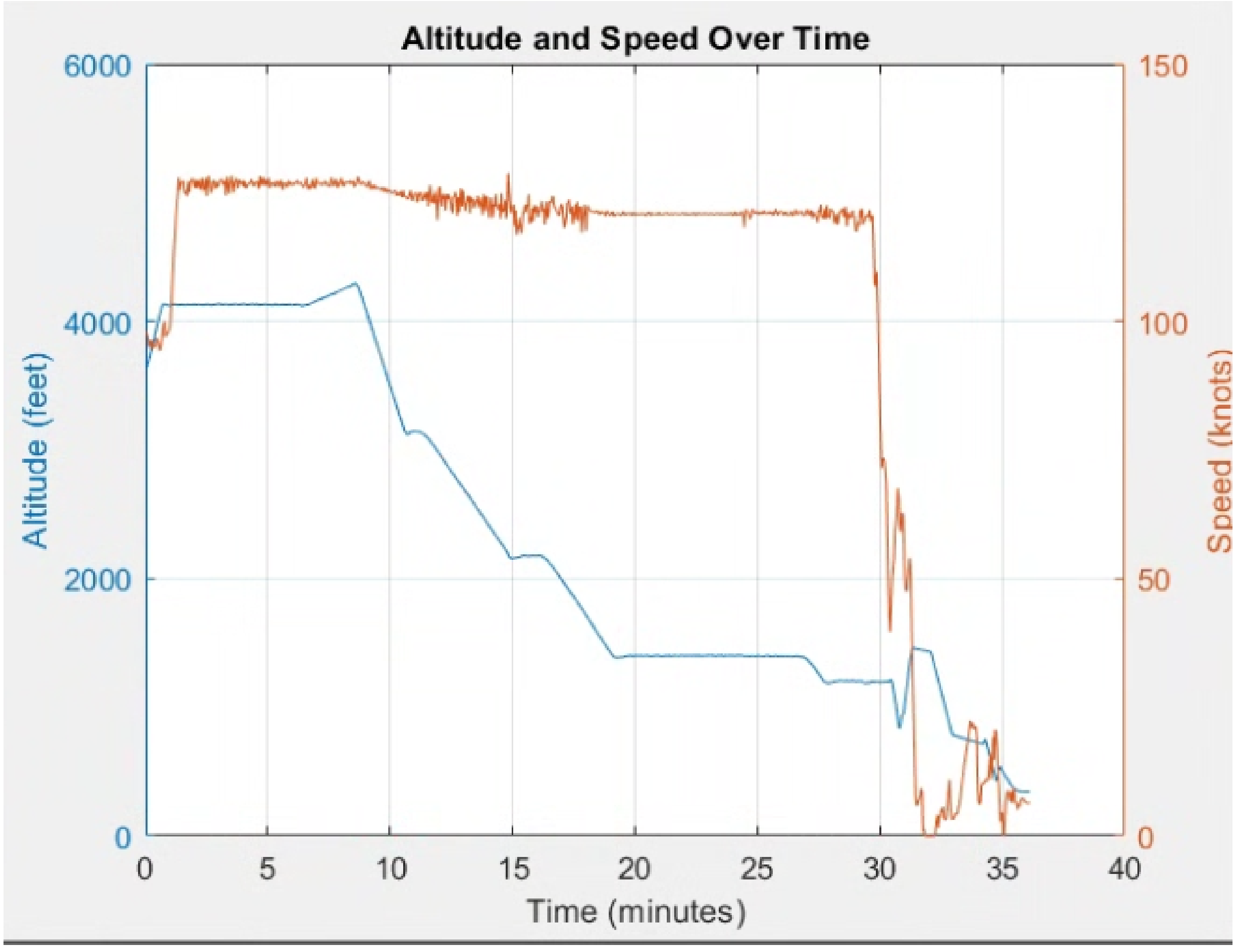


Figure 7: Altitude and Speed Plot

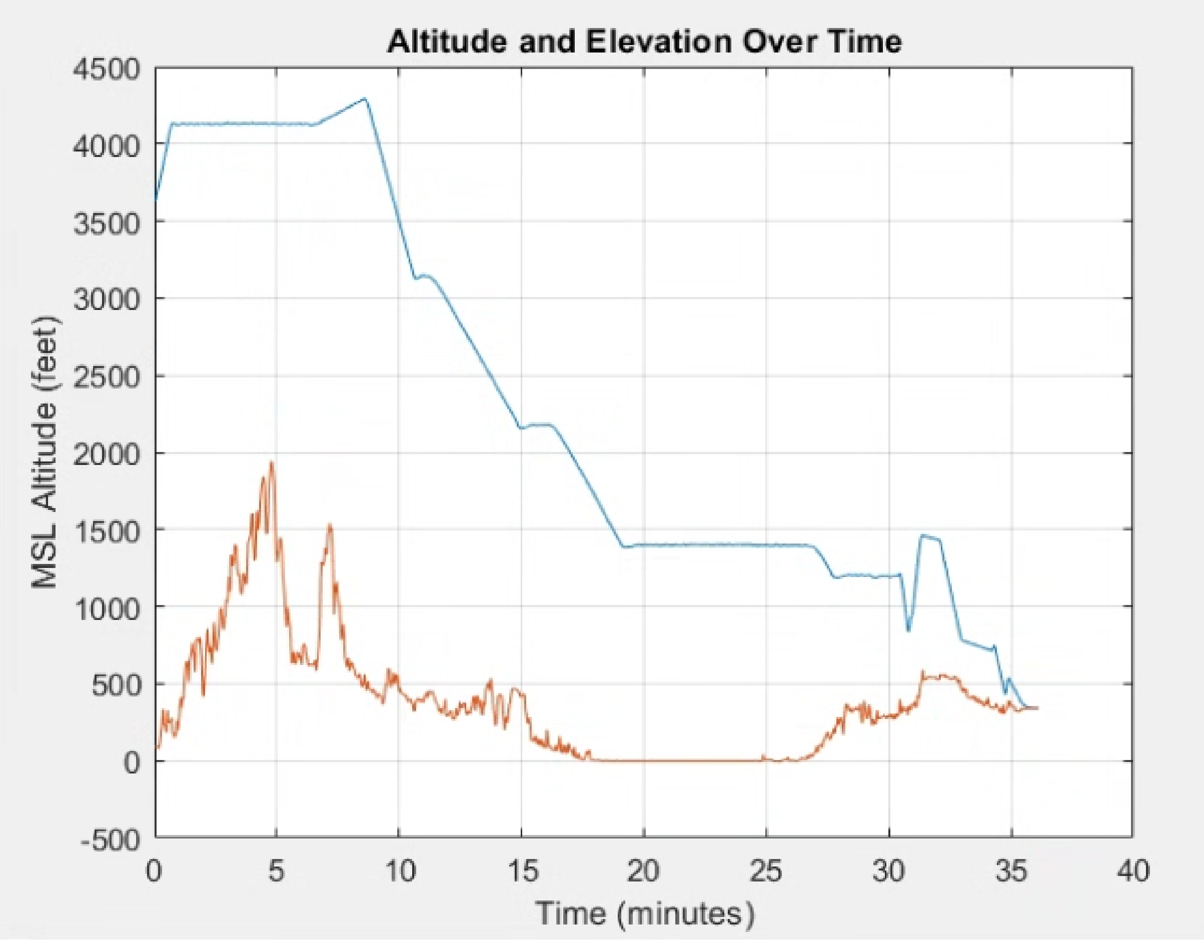


Figure 8: Altitude and Terrain Plot

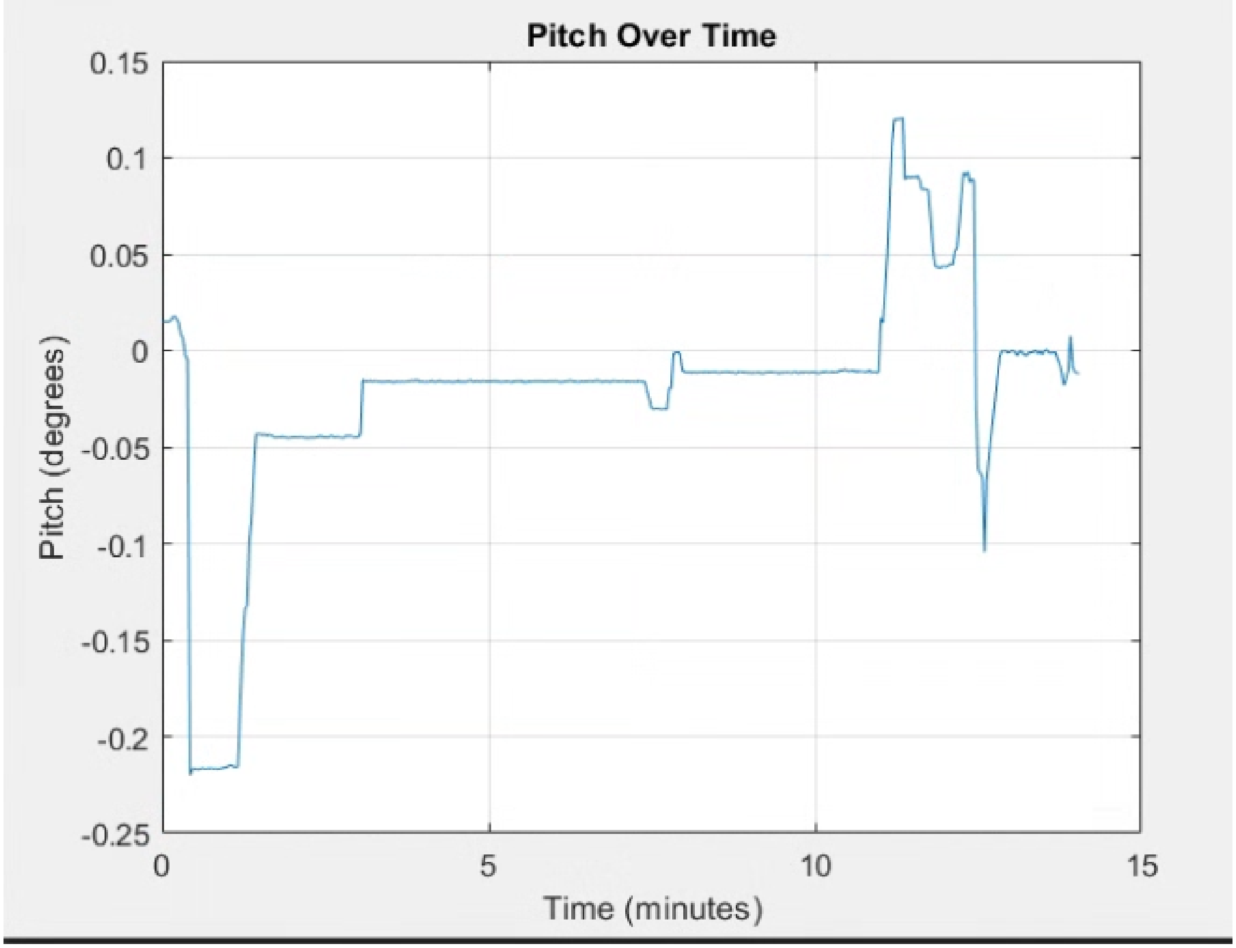


Figure 9: Pitch Plot

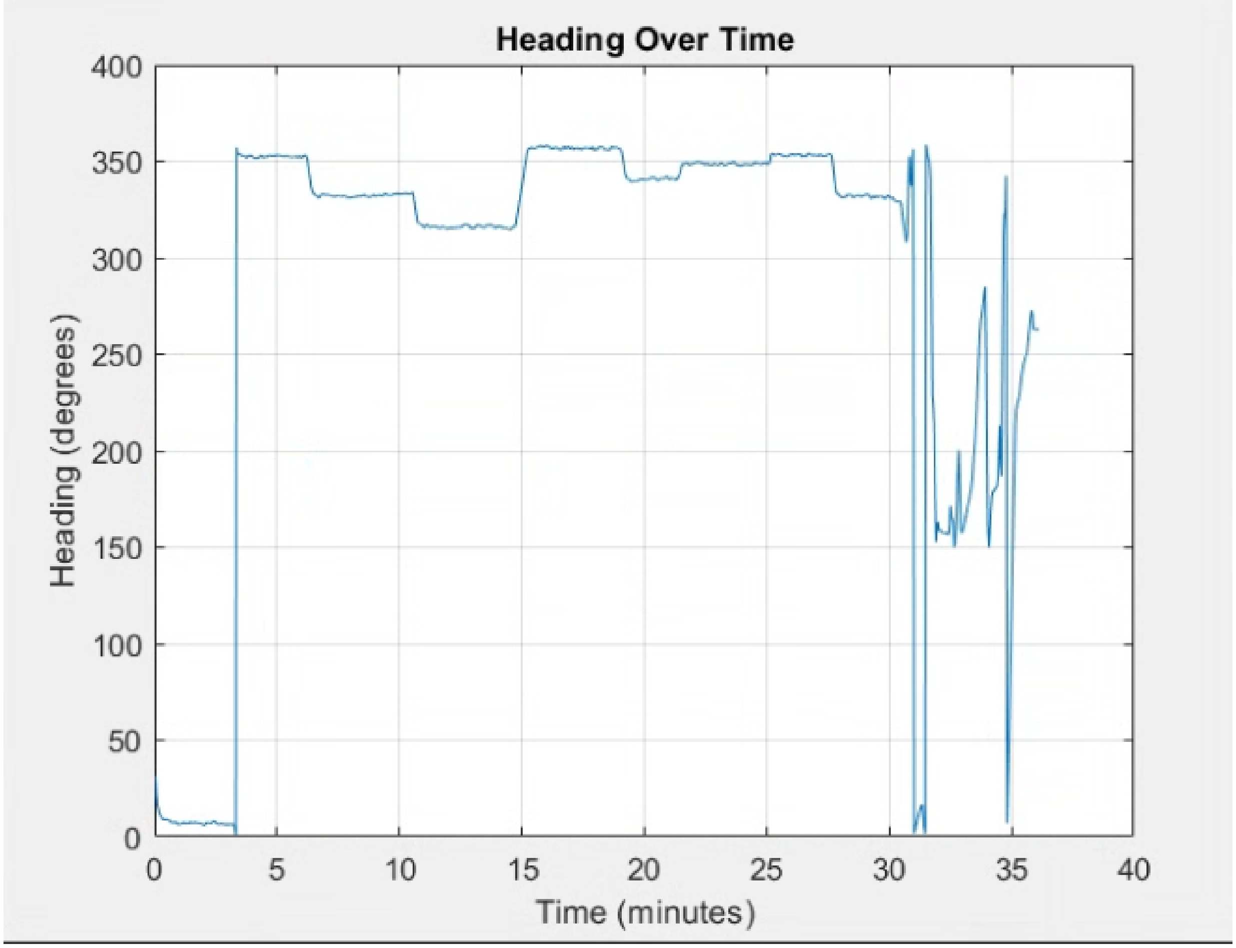


Figure 10: Heading Plot

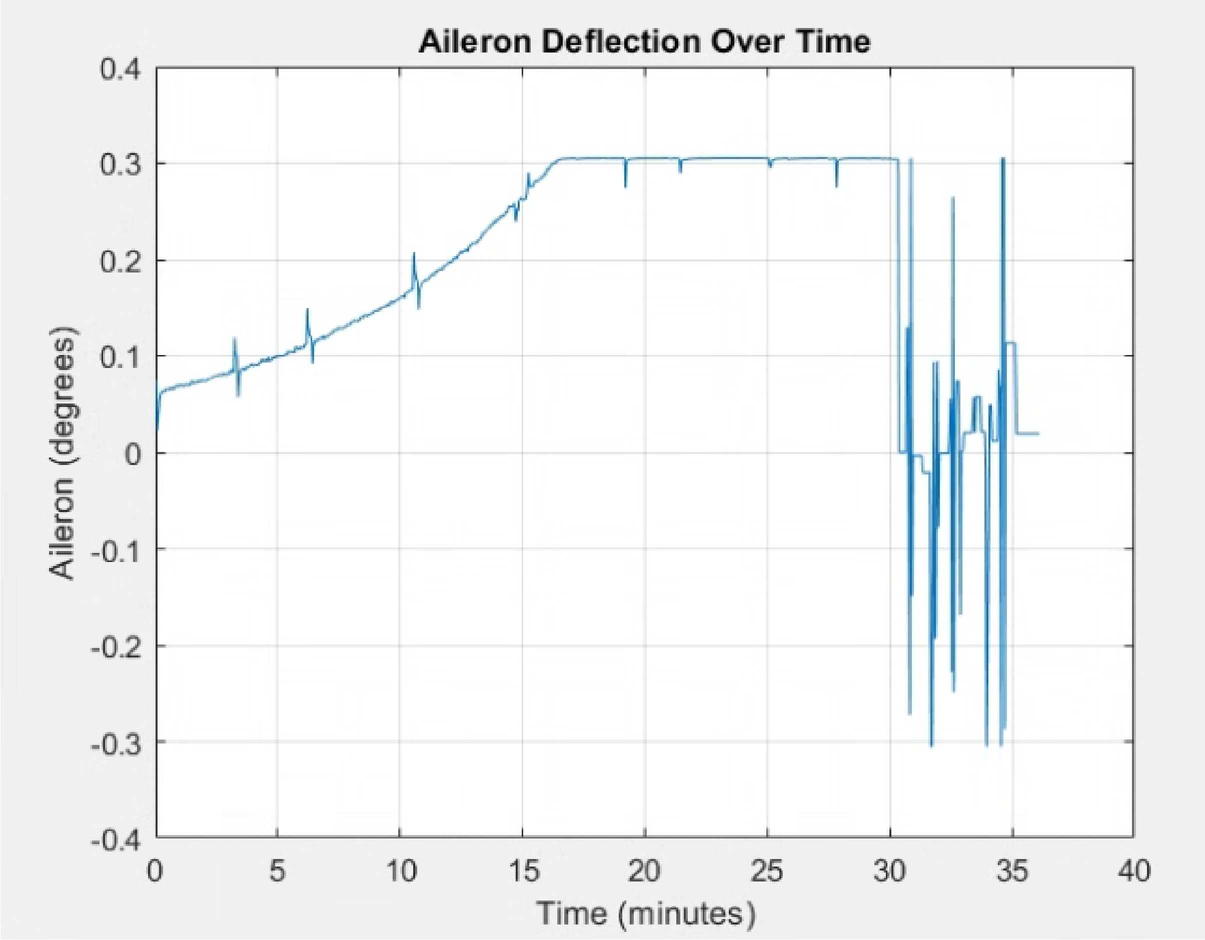


Figure 11: Aileron Deflection Plot

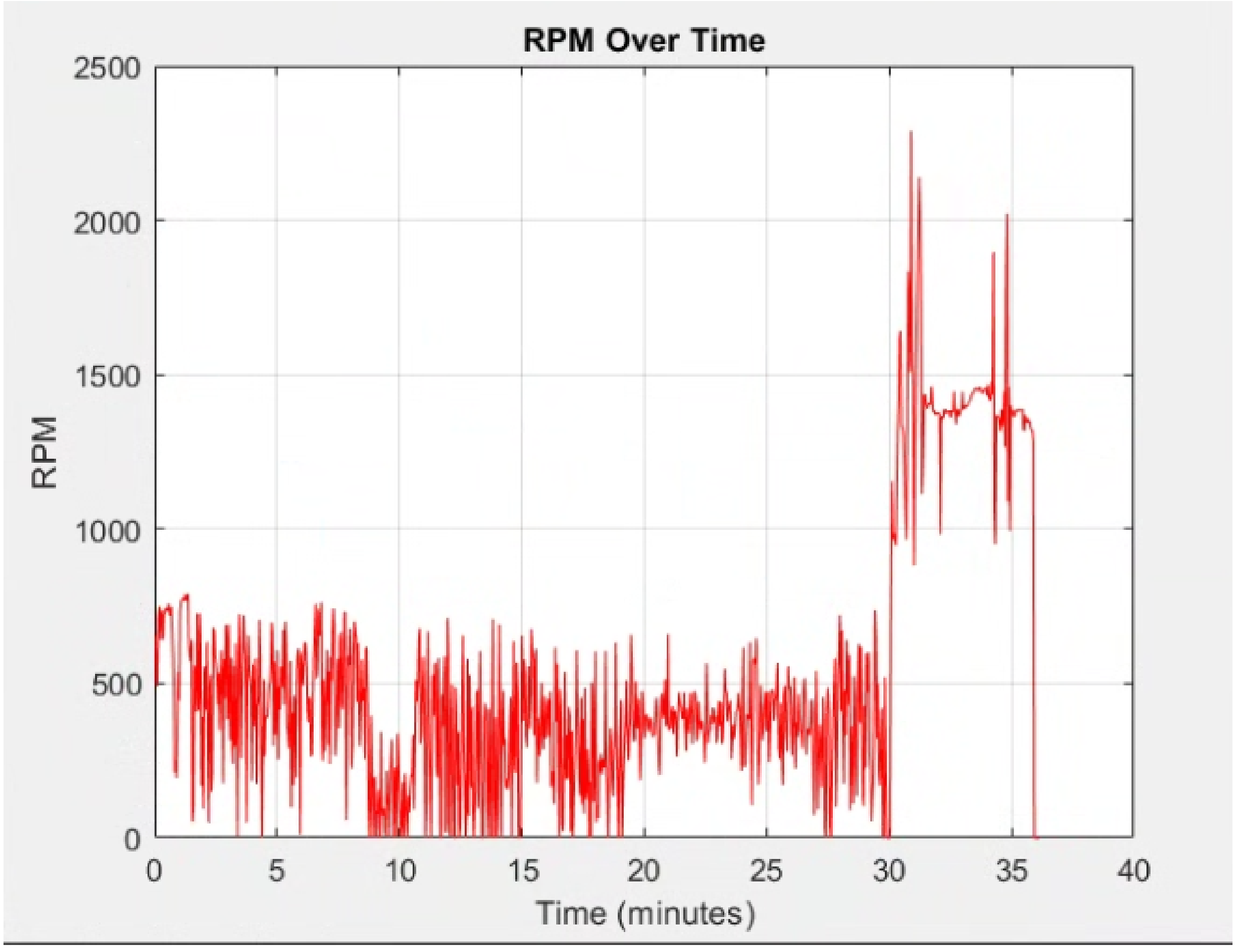


Figure 12: RPM Plot

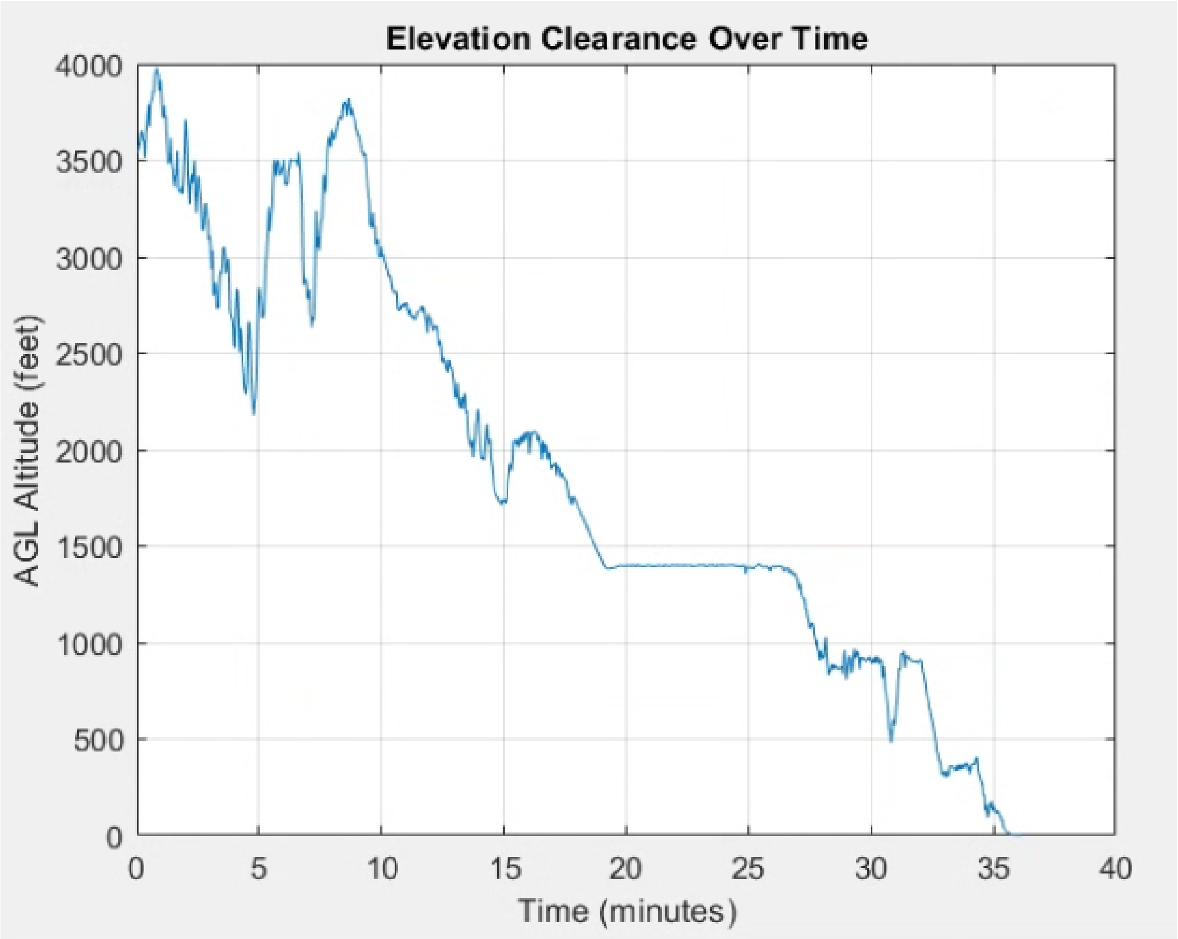


Figure 13: Terrain Clearance Plot

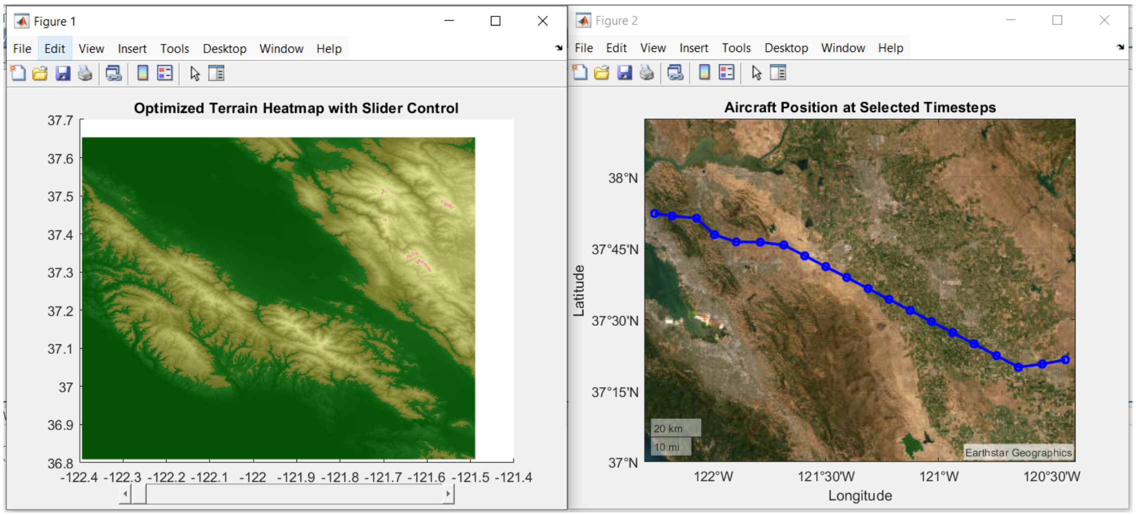


Figure 14: Terrain overlay after initial vertical takeoff and transition to horizontal flight

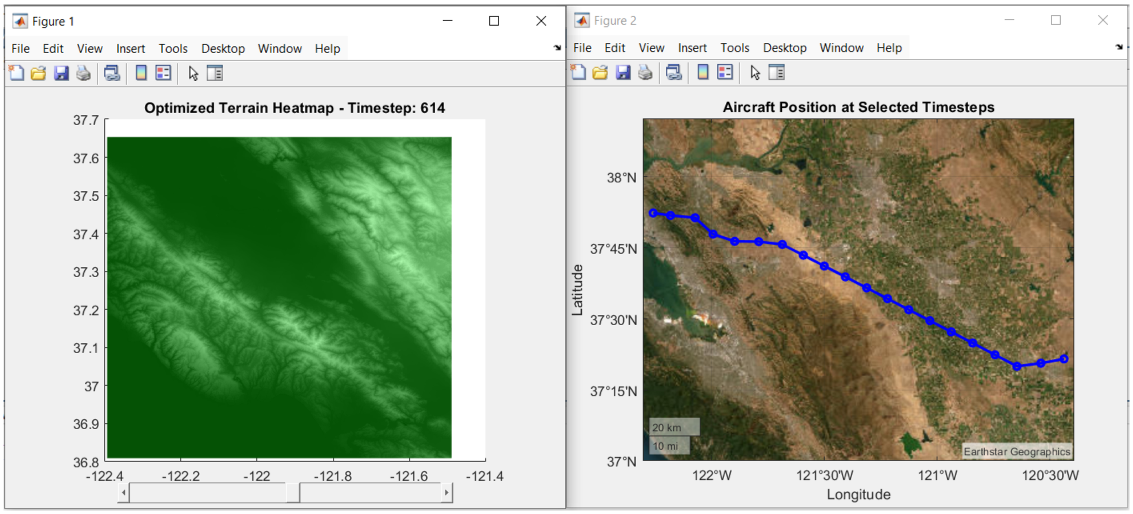


Figure 15: Terrain overlay midway through flight

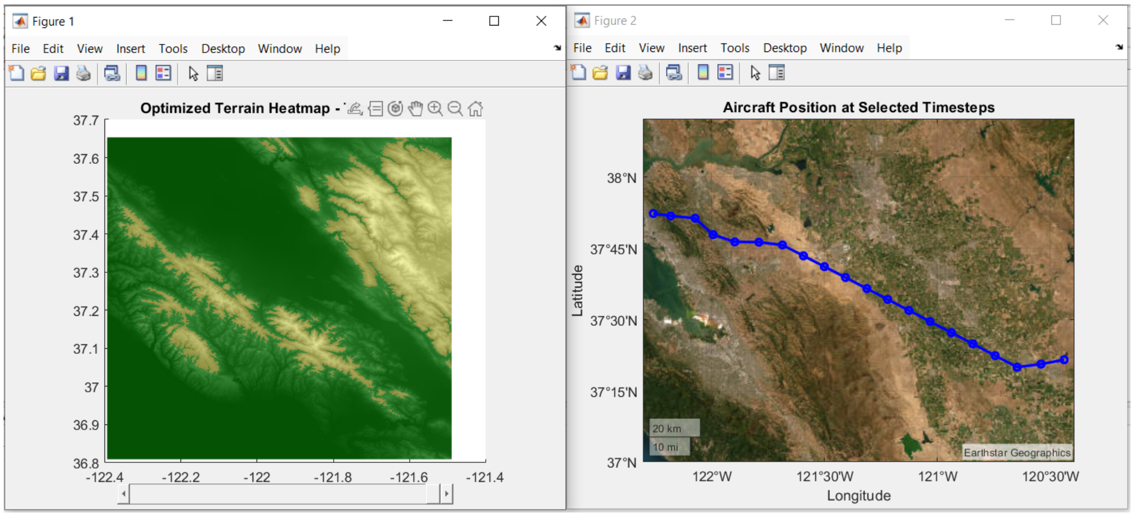
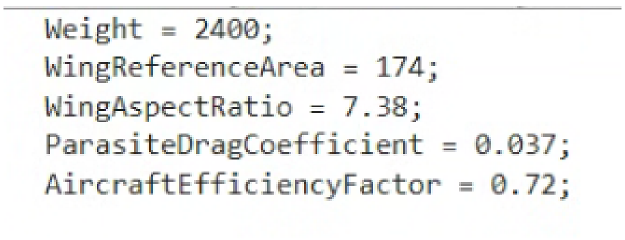


Figure 16: Terrain overlay on descent



```
Weight = 2400;
WingReferenceArea = 174;
WingAspectRatio = 7.38;
ParasiteDragCoefficient = 0.037;
AircraftEfficiencyFactor = 0.72;
```

Figure 17: Best Glide Parameters Defined for An Aircraft

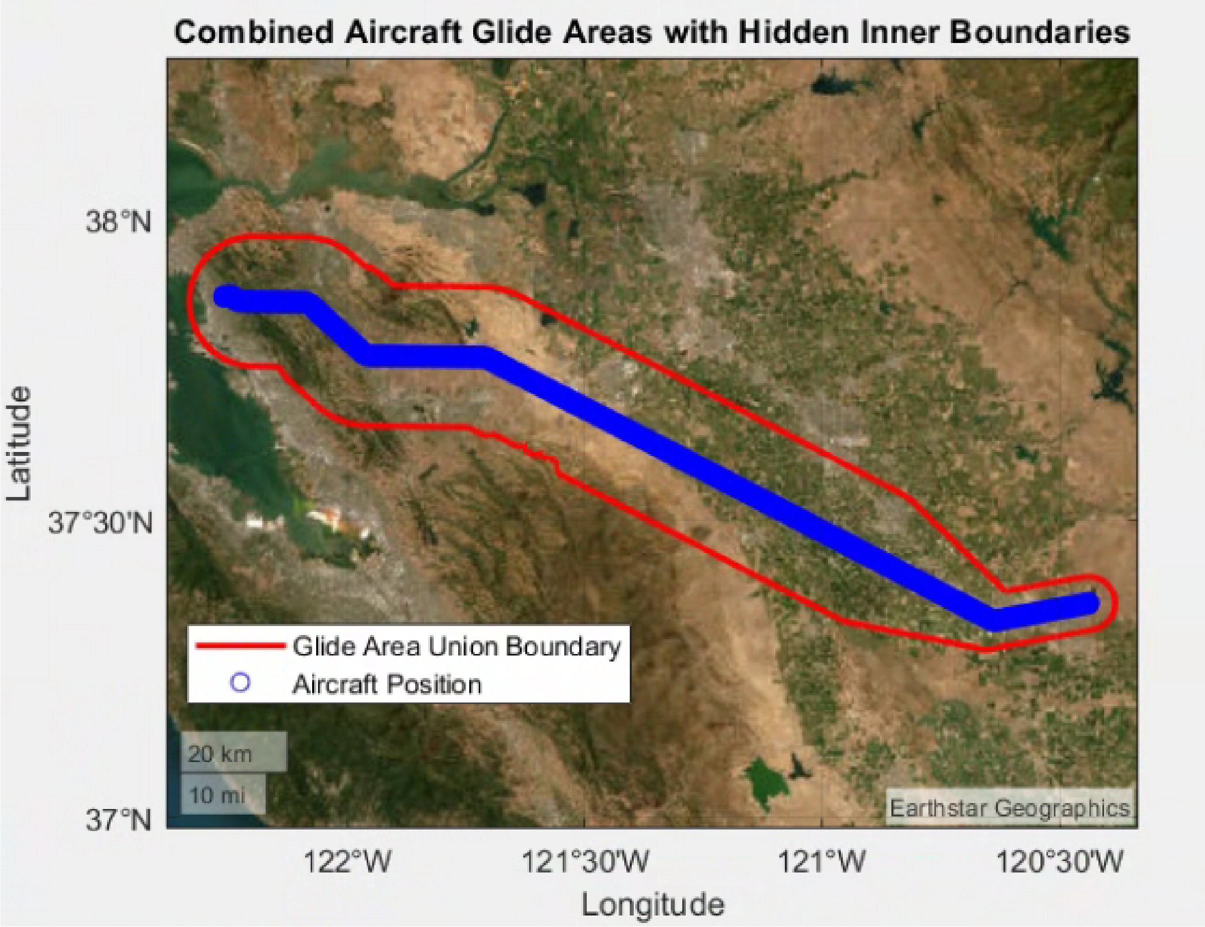


Figure 18: Union Glide Boundary Example

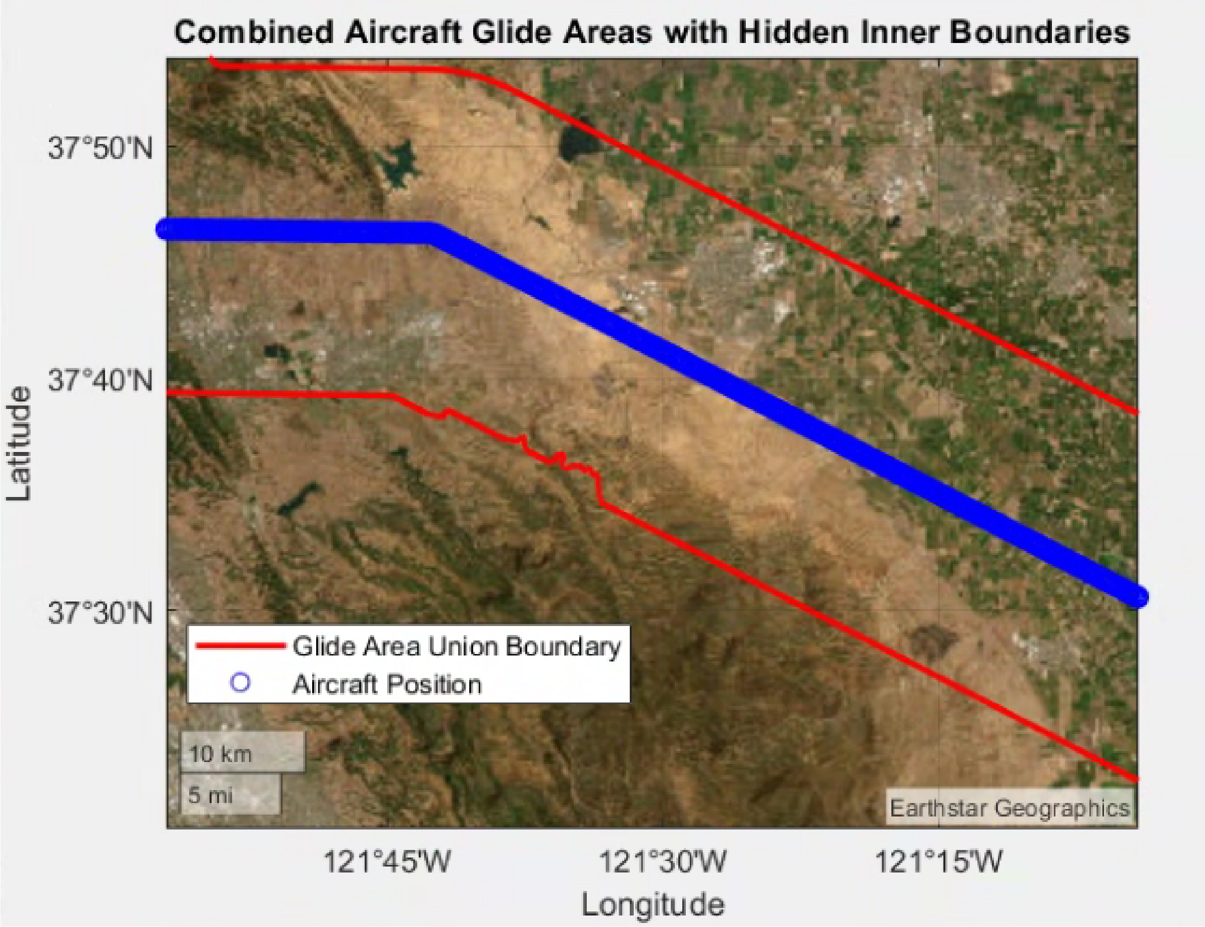


Figure 19: Union Glide Boundary Example - Close up to Display Terrain Restriction

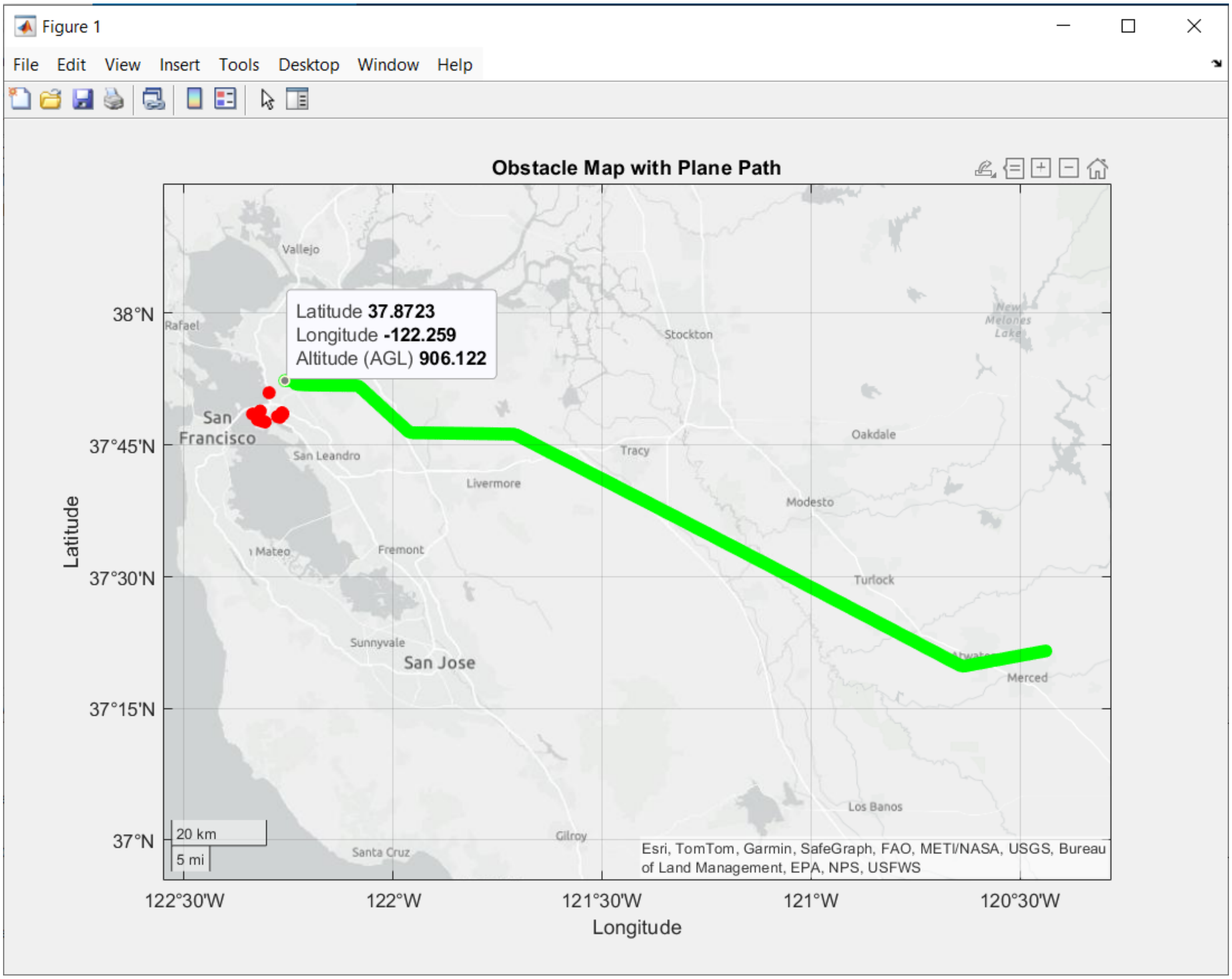


Figure 20: Red dot cluster indicates hazards within threshold values: 500 ft./6 nmi

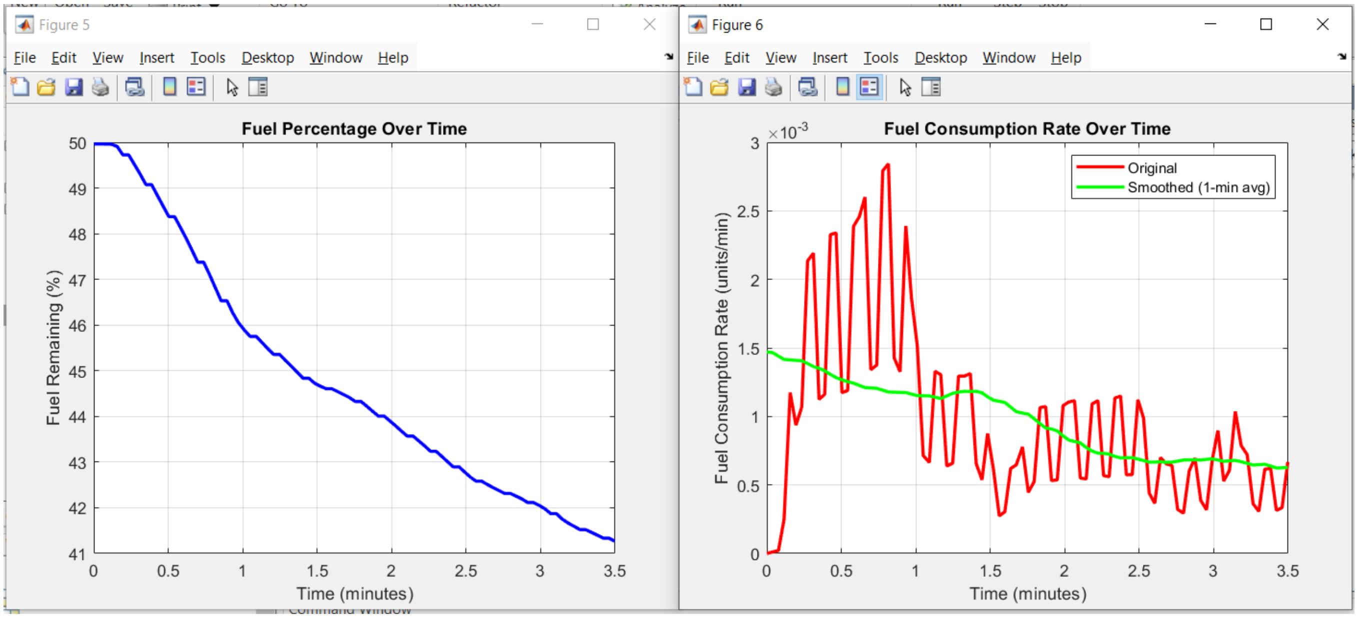


Figure 21: Battery Consumption and level (Fuel) over time

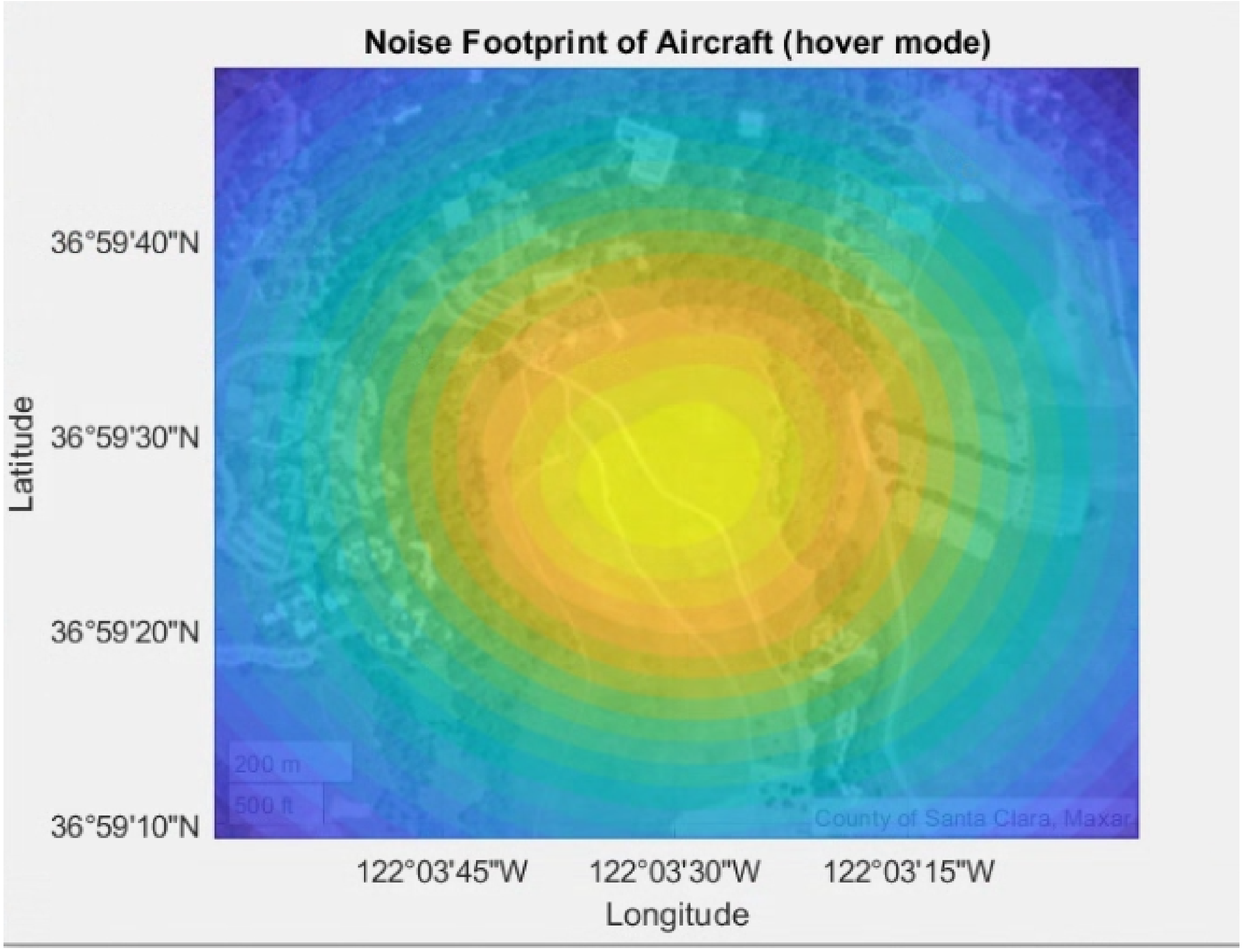


Figure 22: Noise Plot

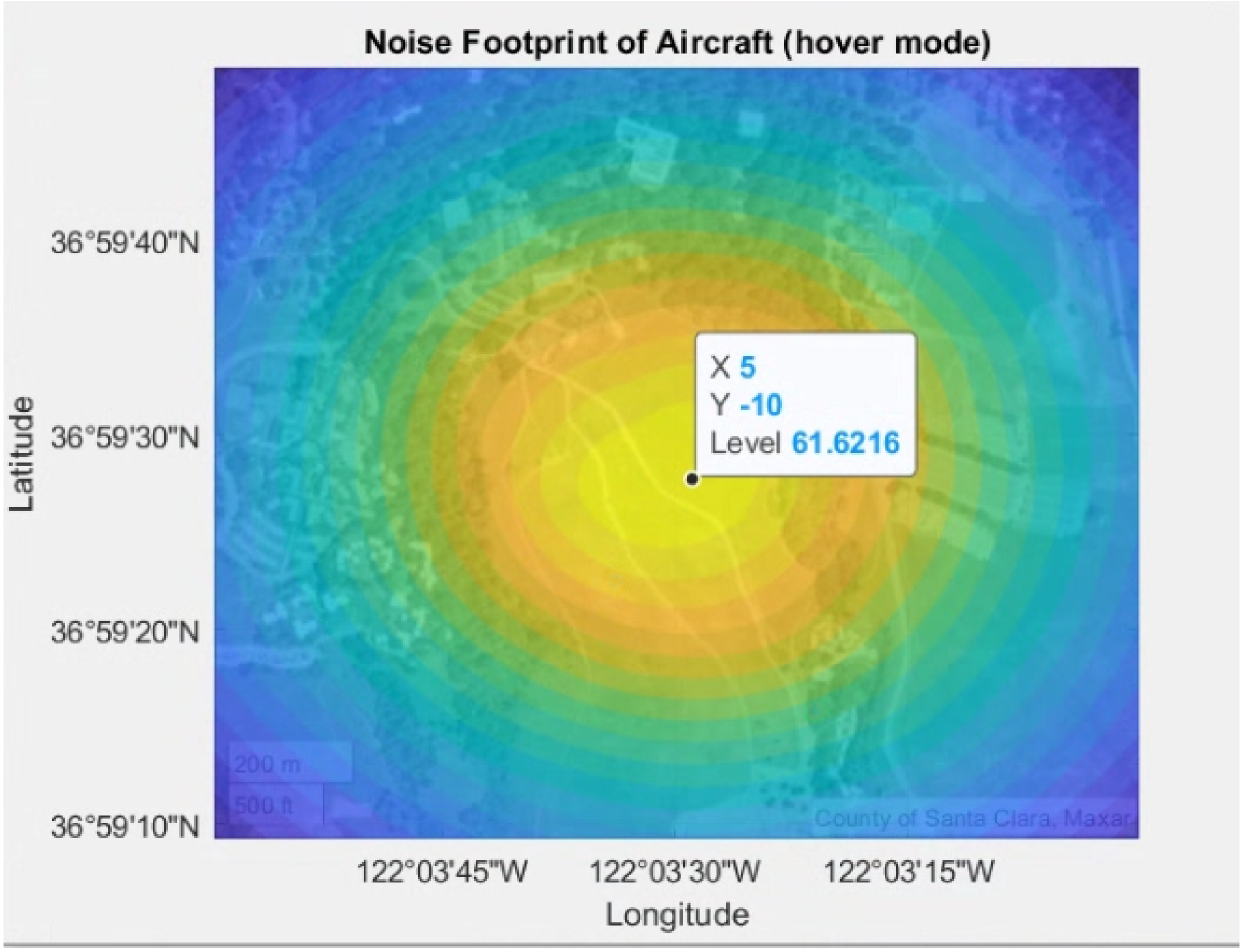


Figure 23: Noise Plot with decibels displayed

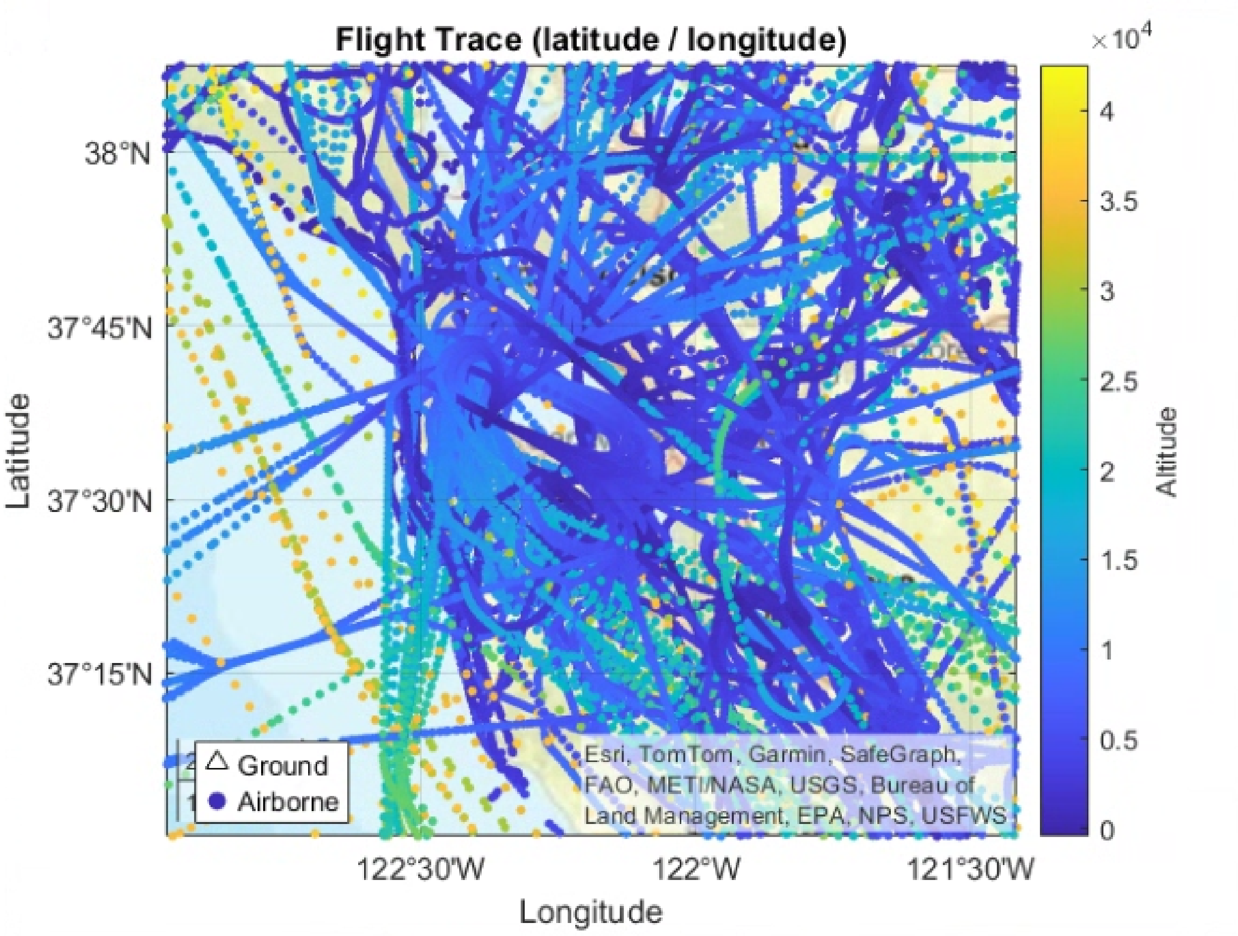

Flight Trace (latitude / longitude)
38°N
37°45'N
37°30'N
37°15'N
Latitude
122°30'W
122°W
121°30'W
Longitude
△ Ground
● Airborne
Esri, TomTom, Garmin, SafeGraph, FAO, METI/NASA, USGS, Bureau of Land Management, EPA, NPS, USFWS
×10⁴
4
3.5
3
2.5
2
1.5
1
0.5
0
Altitude

Figure 24: All ADS-B records from 12/30/2024

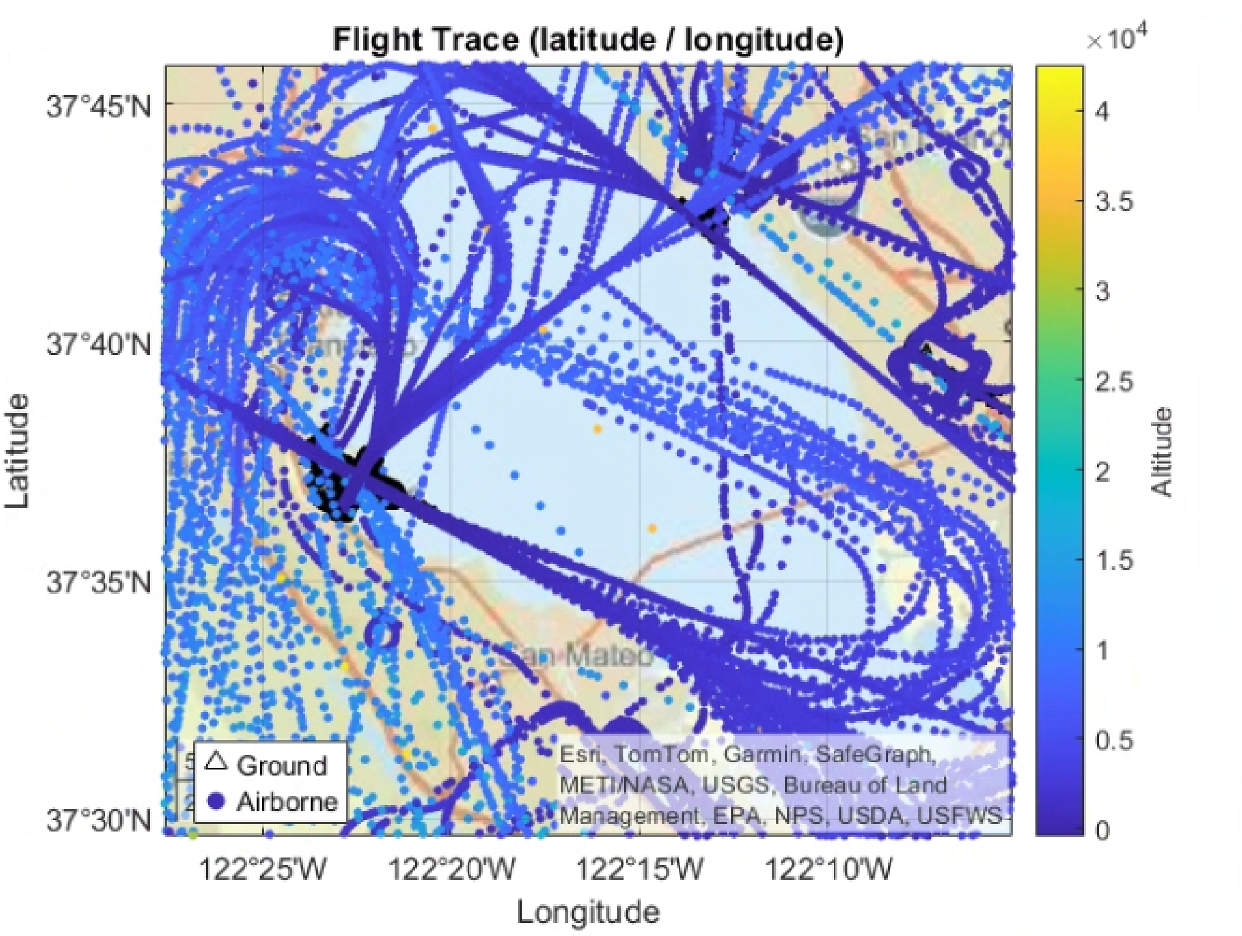


Figure 25: All ADS-B records from 12/30/2024 - Close up of SF Bay

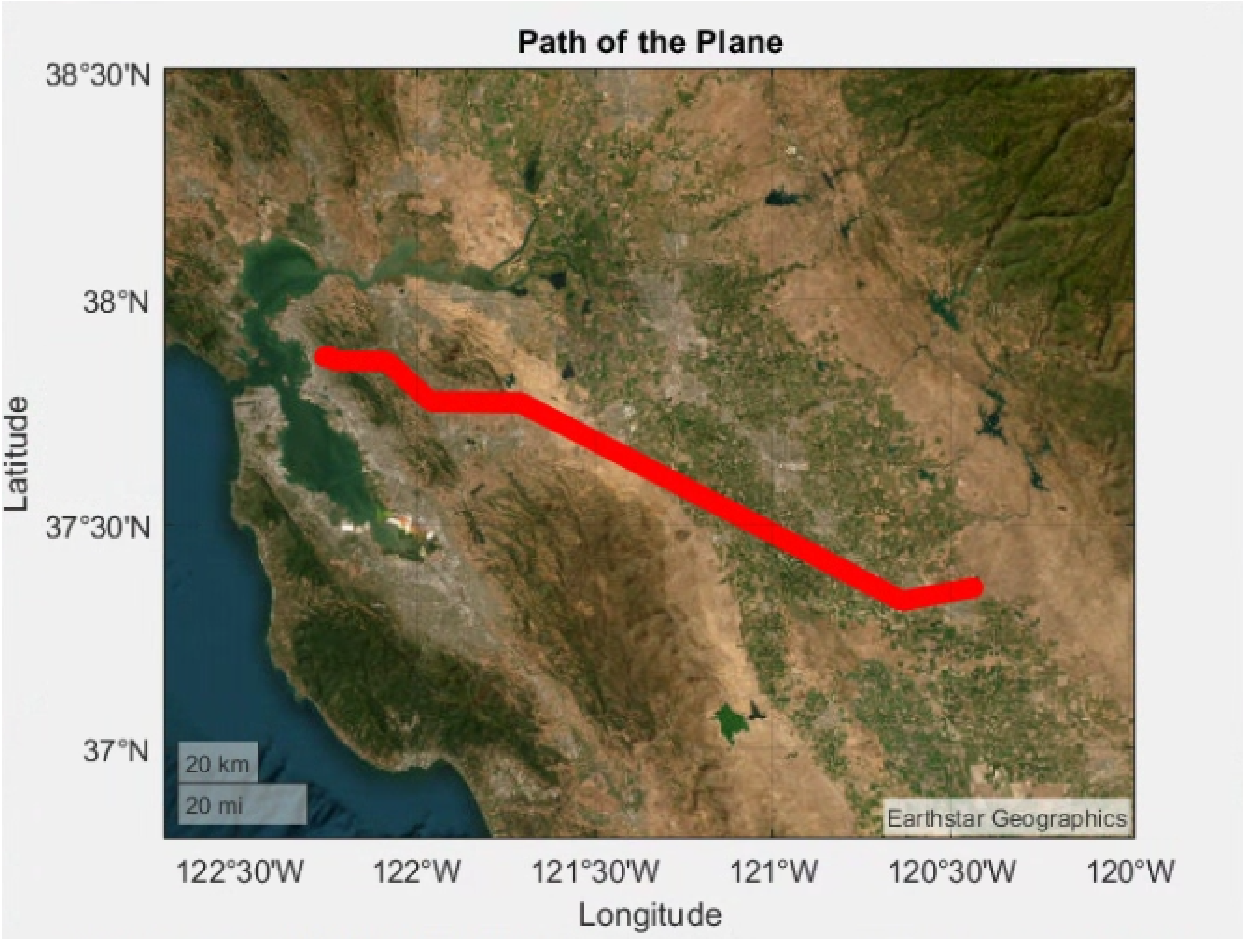


Figure 26: Berkeley to Merced Route Flown in MSFS 24

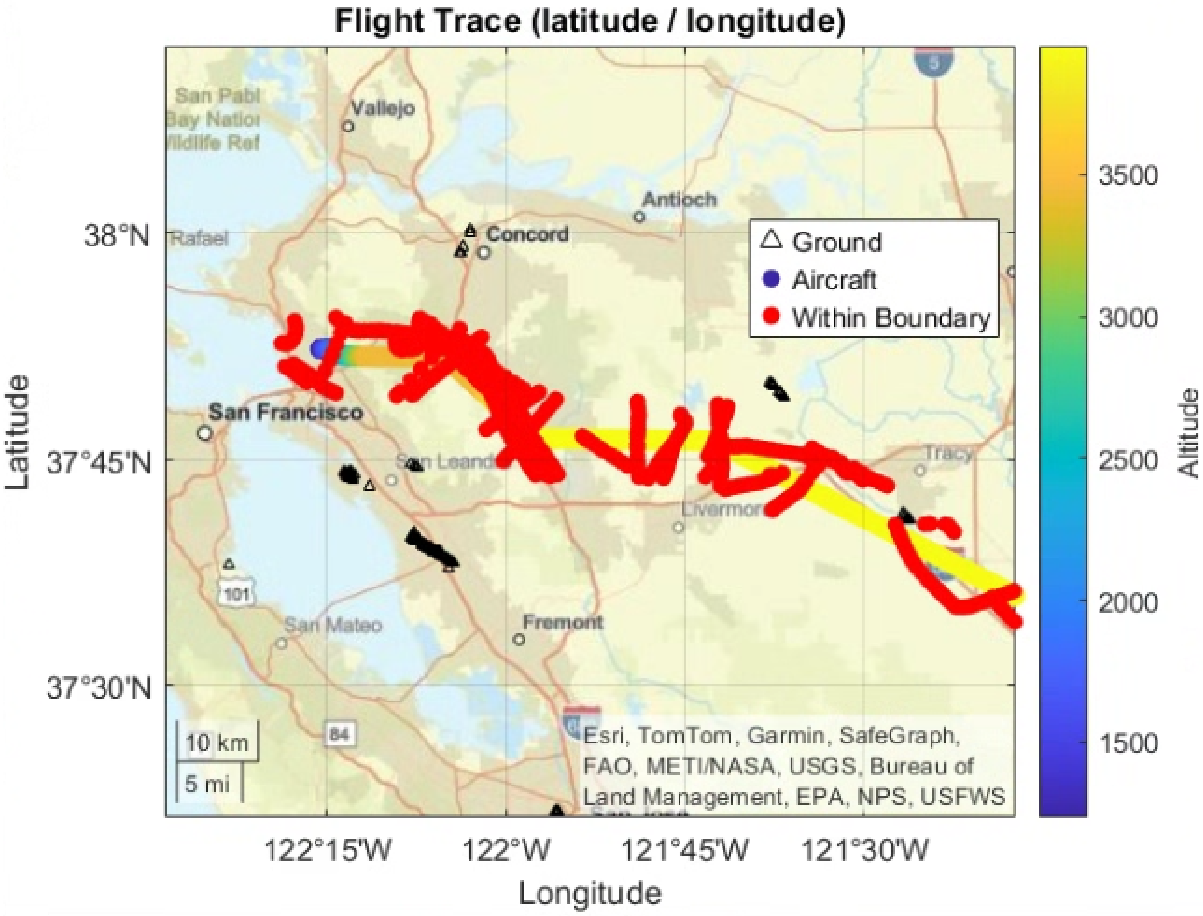


Figure 27: Traffic Conflicts Highlighted in Red

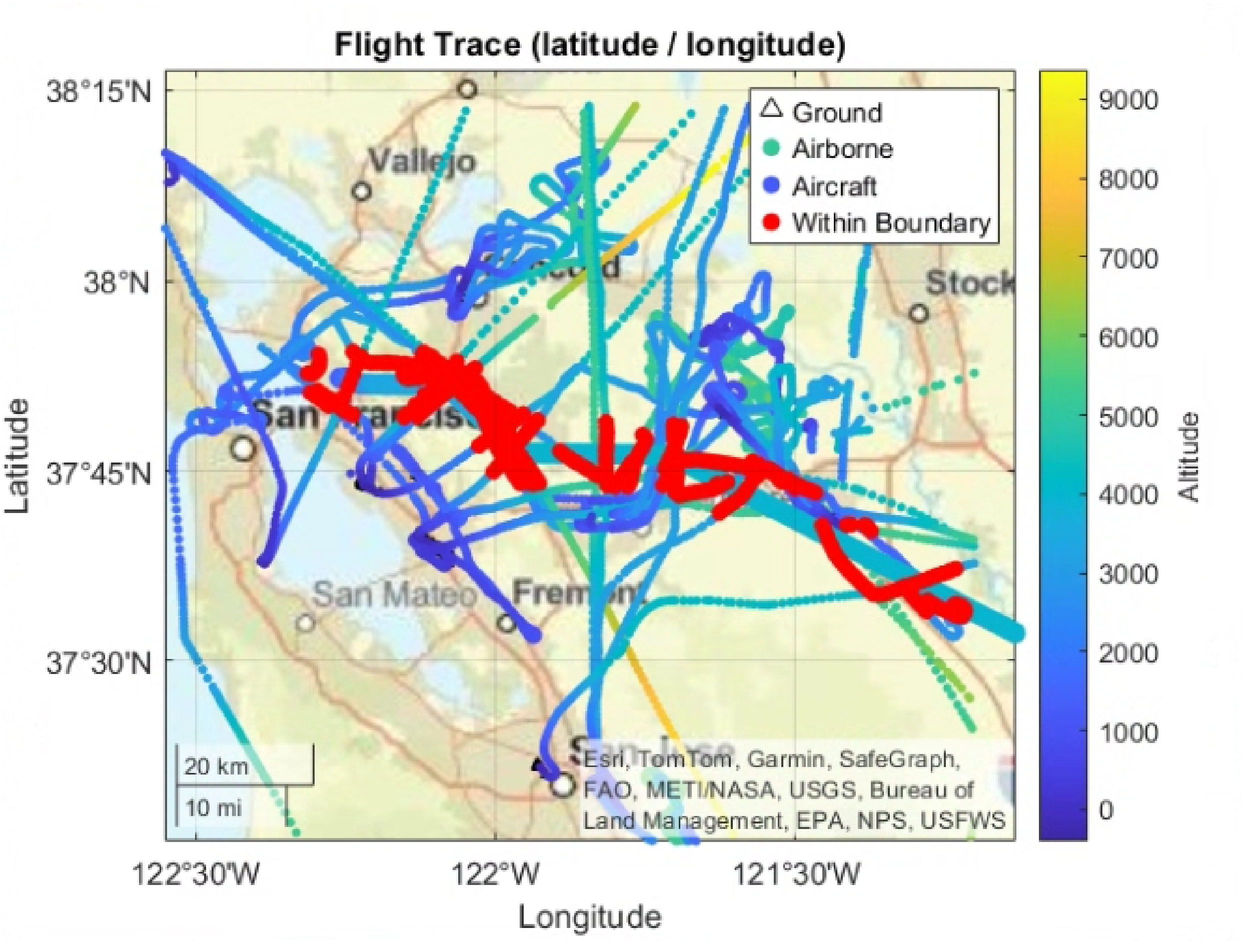


Figure 28: Traffic Conflicts with Full Trace of Aircraft who Conflict

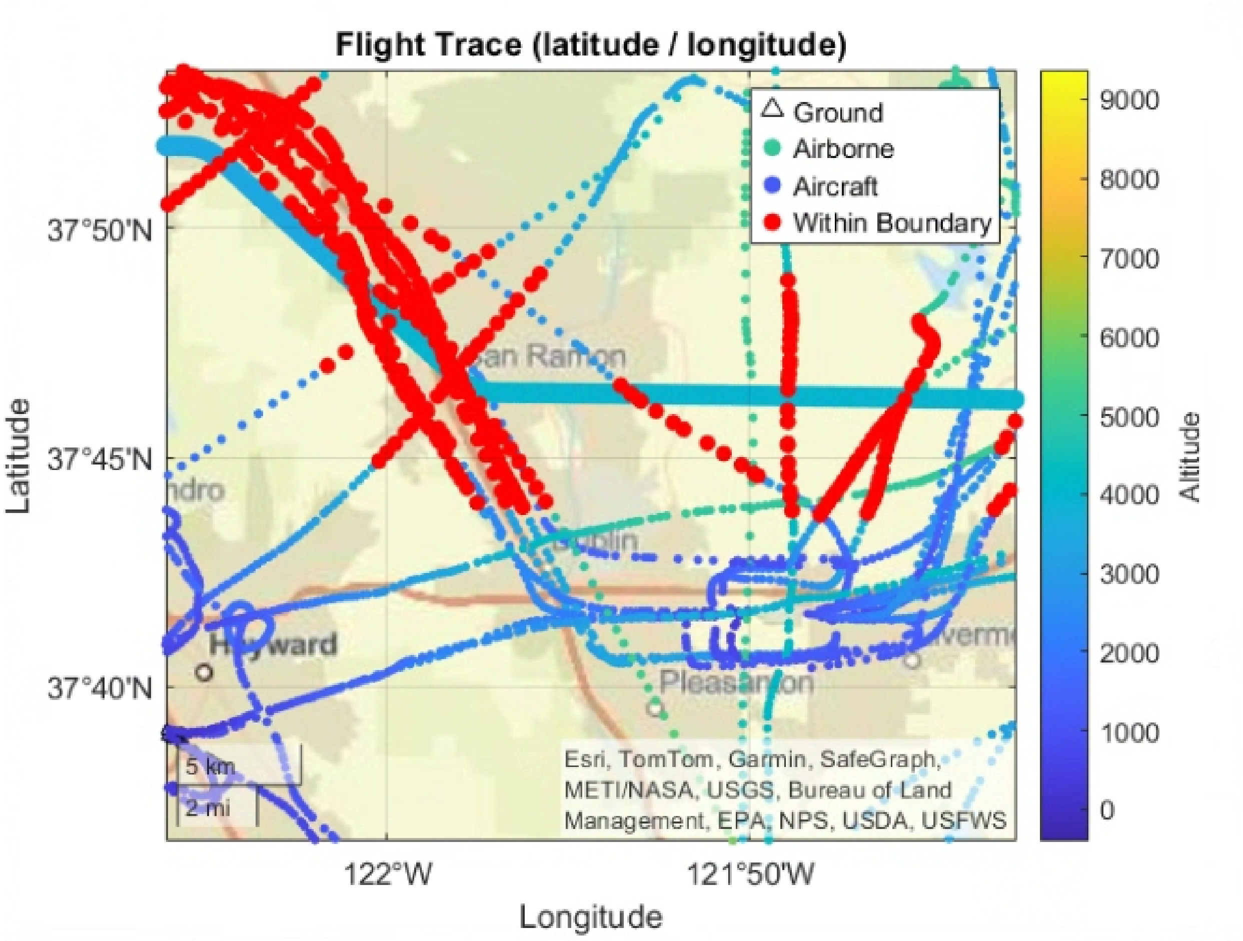


Figure 29: Traffic Conflicts close to Livermore

7.2 Section 3: Routing

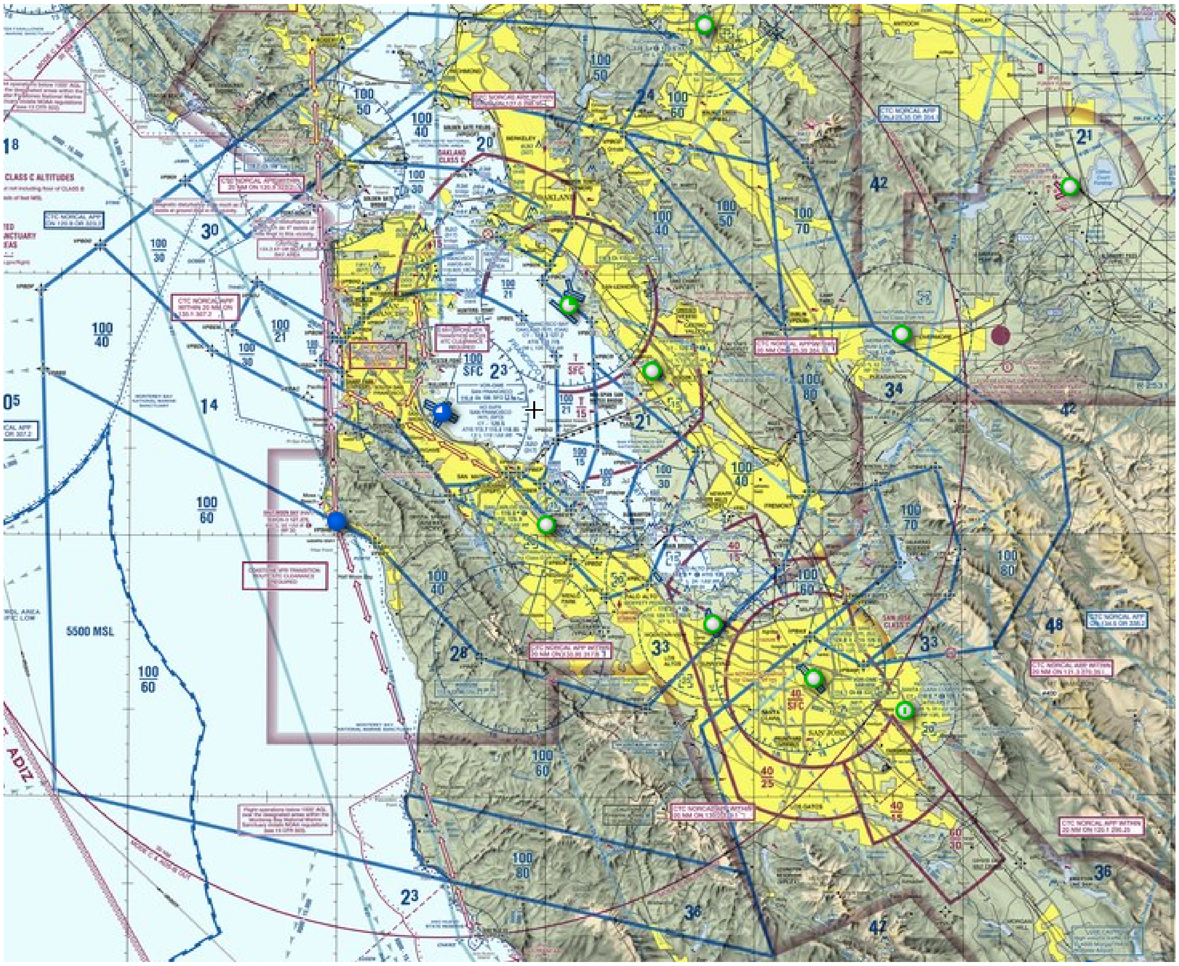


Figure 30: San Francisco Terminal Area Chart

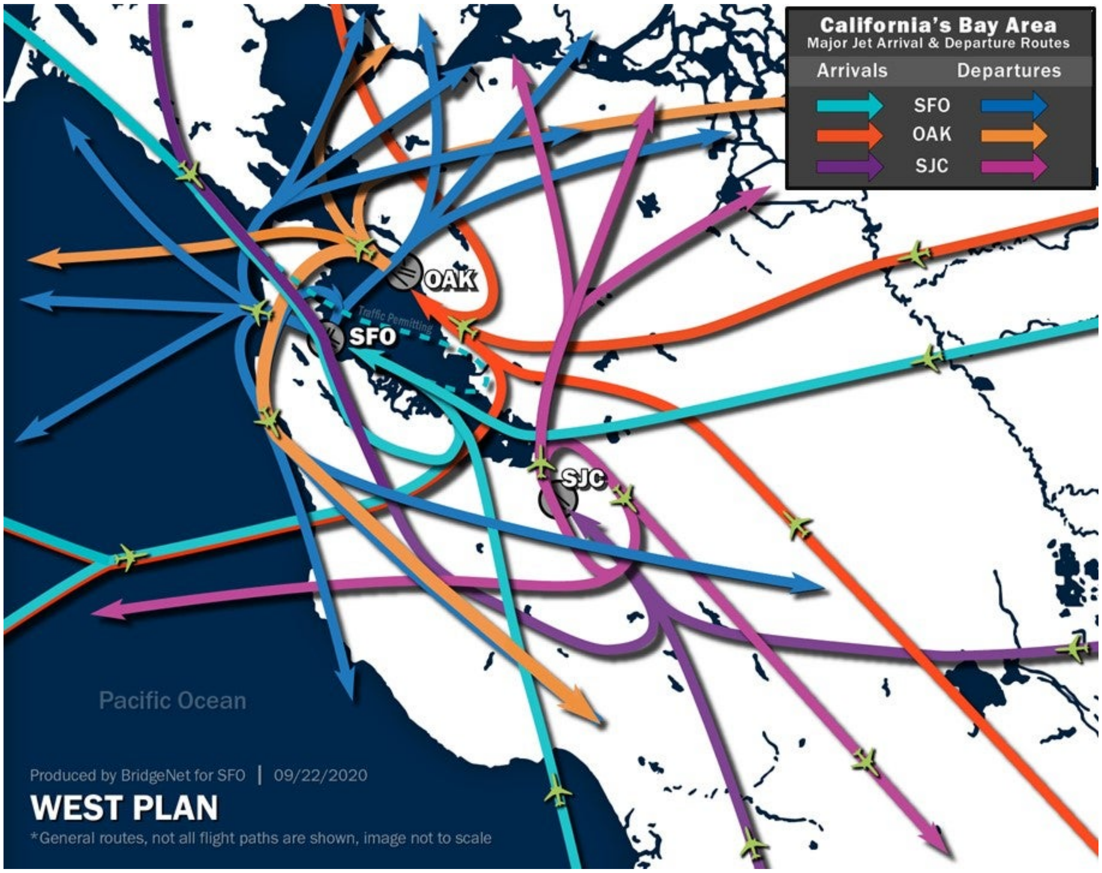

California's Bay Area
Major Jet Arrival & Departure Routes
Arrivals
Departures
SFO
OAK
SJC
OAK
SFO
SJC
Traffic Permitting
Pacific Ocean
Produced by BridgeNet for SFO | 09/22/2020
WEST PLAN
*General routes, not all flight paths are shown, image not to scale

Figure 31: Norcal Commercial Traffic Routing

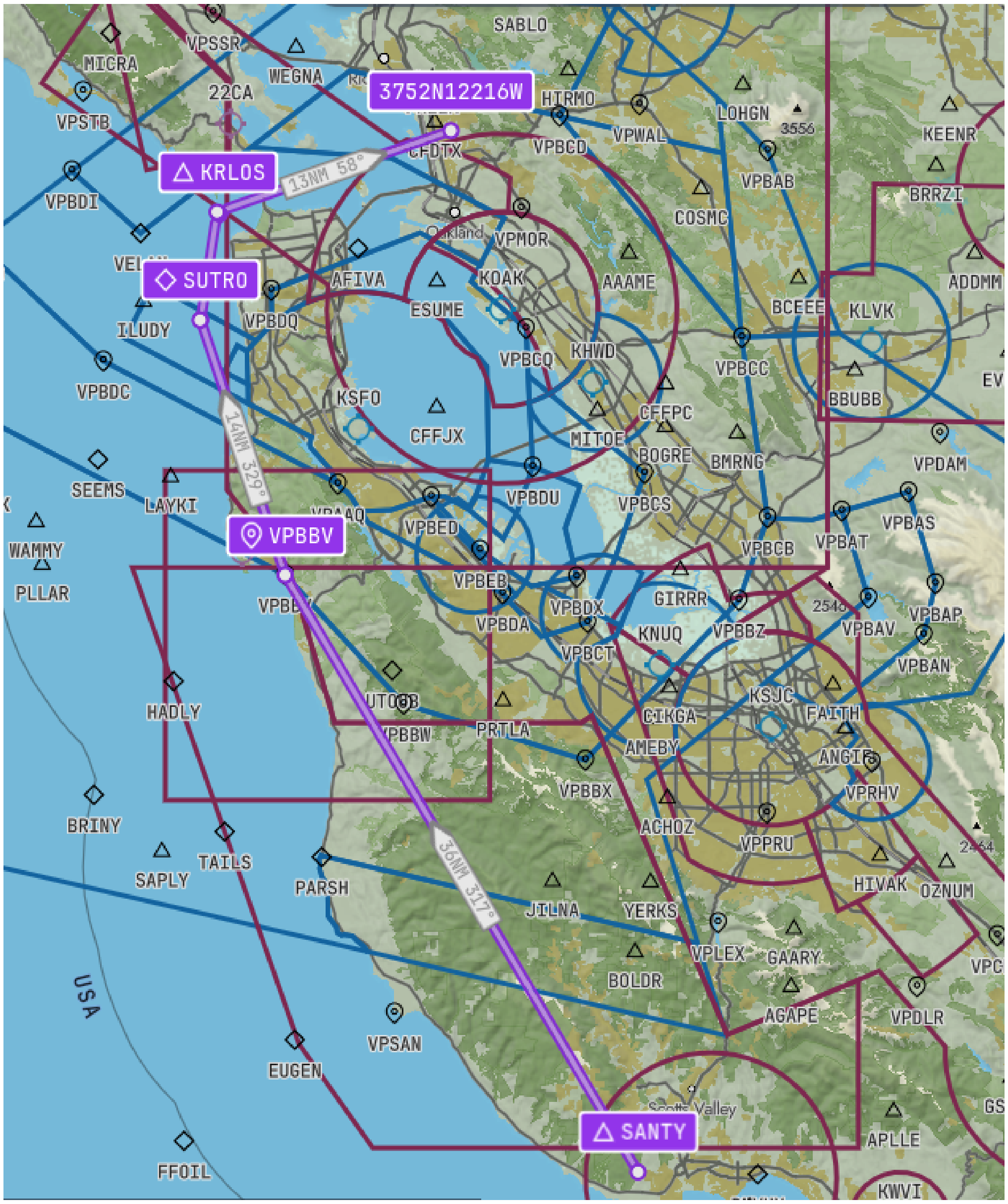


Figure 32: Creating a Flight Plan Manually in the MSFS Flight Planner Tool

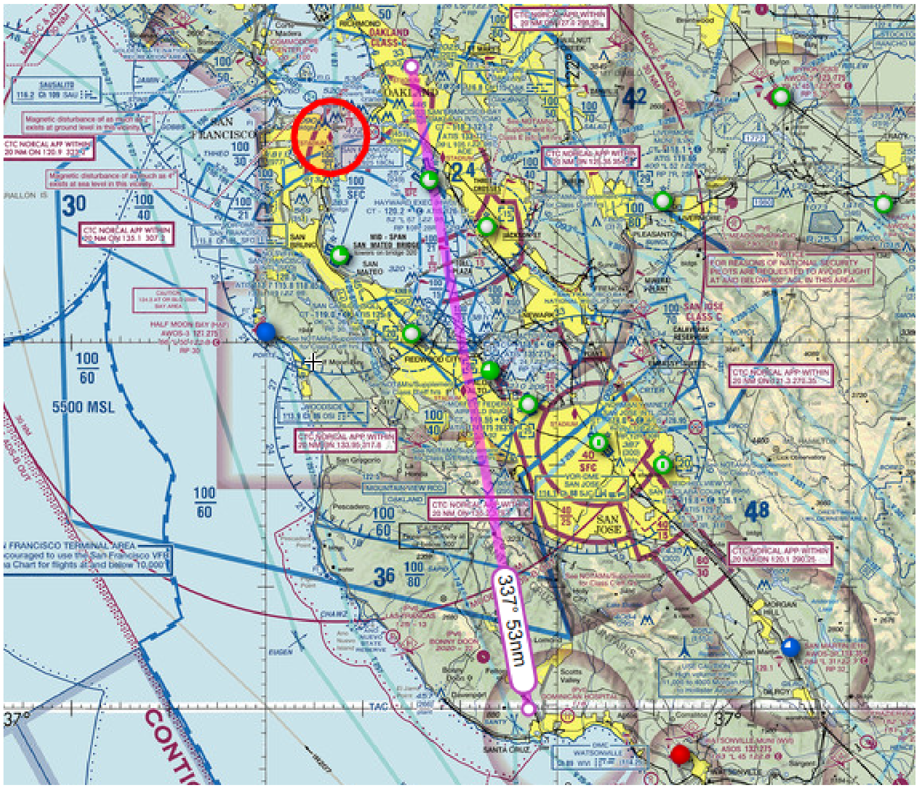


Figure 33: Direct Route between Santa Cruz and Berkeley

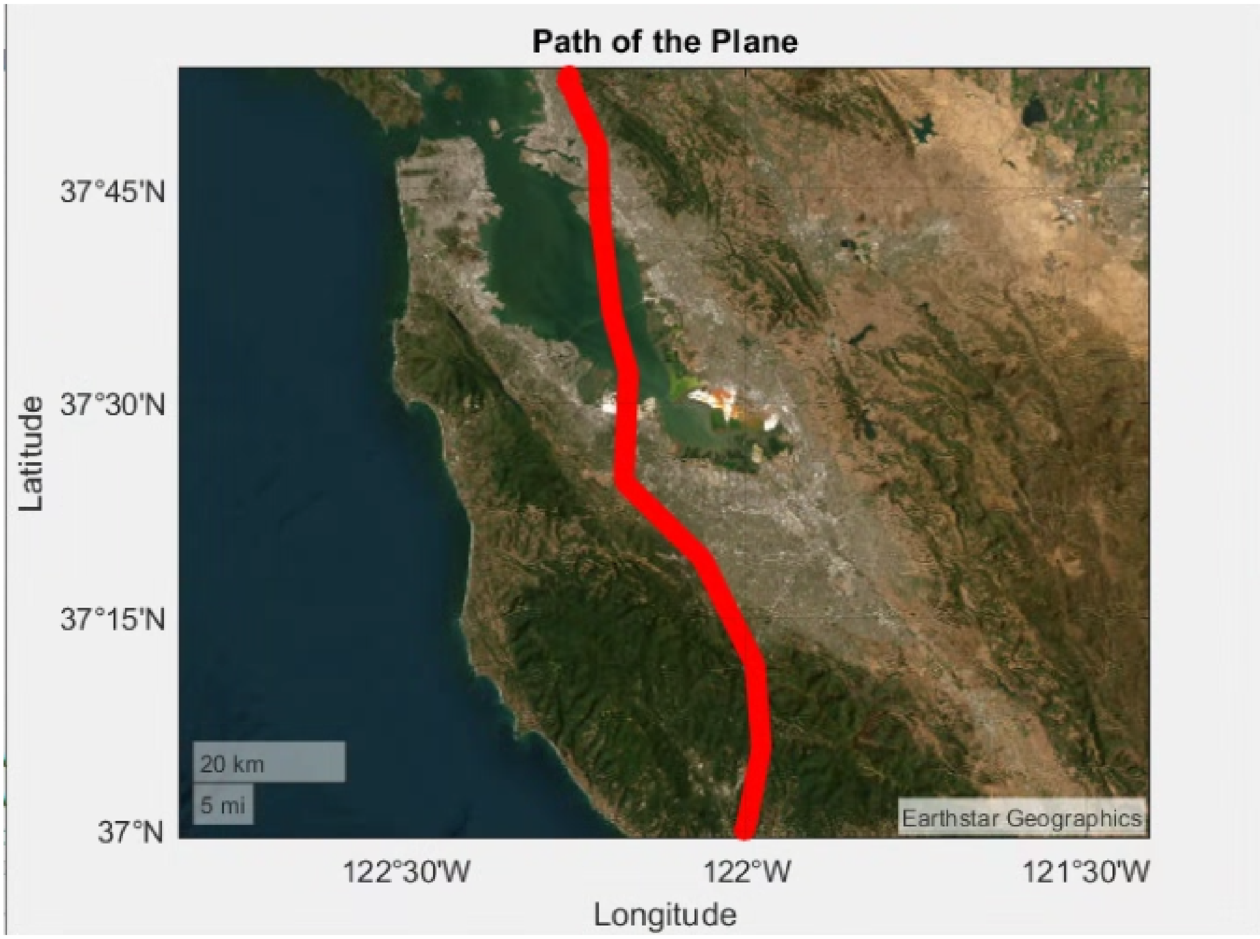


Figure 34: Santa Cruz to Berkeley Proposed Route to Minimize Travel Time and Remain in VFR Flight Routes (Over the Midspan of the Santa Mateo Bridge)

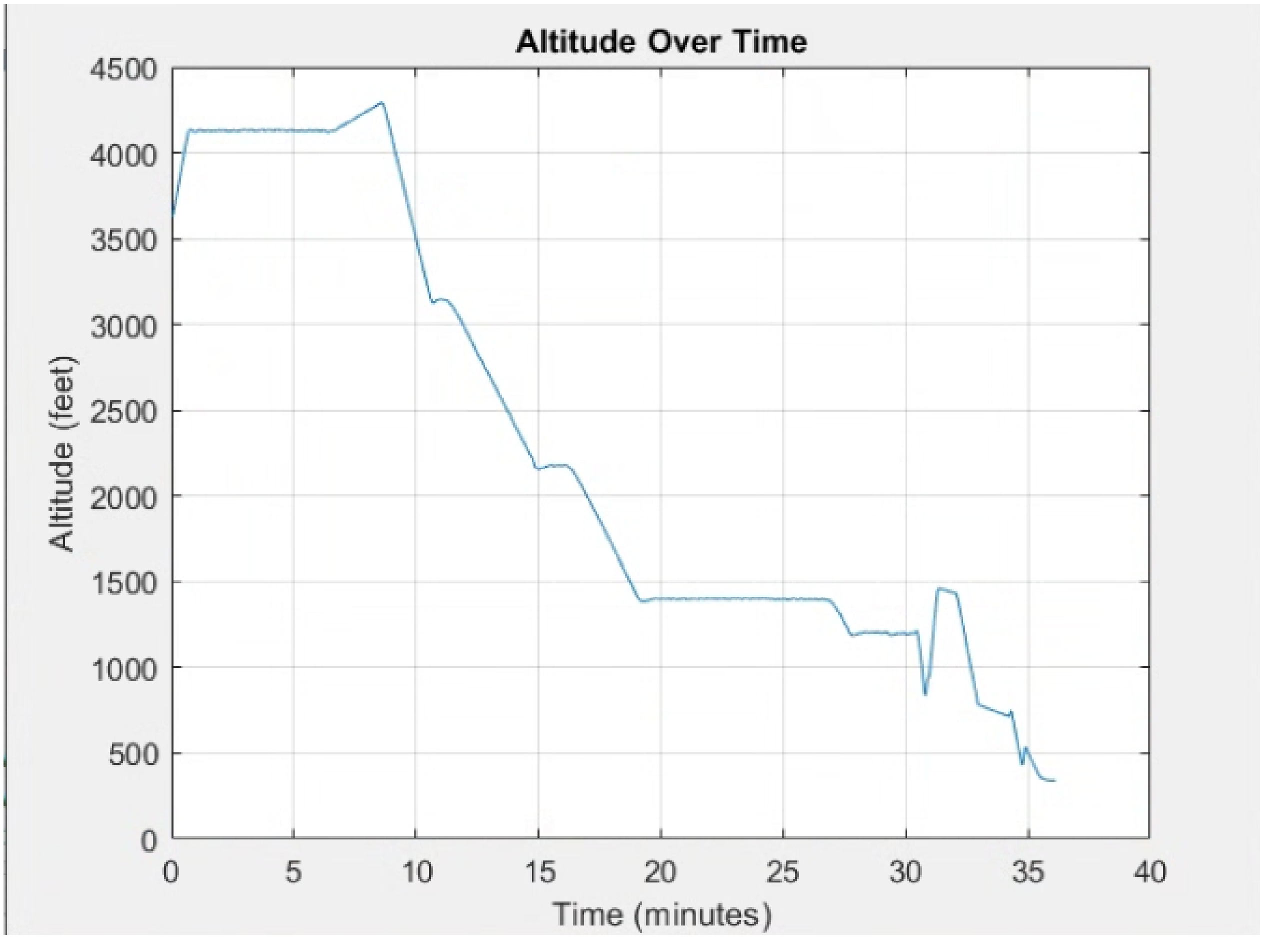


Figure 35: Santa Cruz to Berkeley Proposed Route Altitude over Time

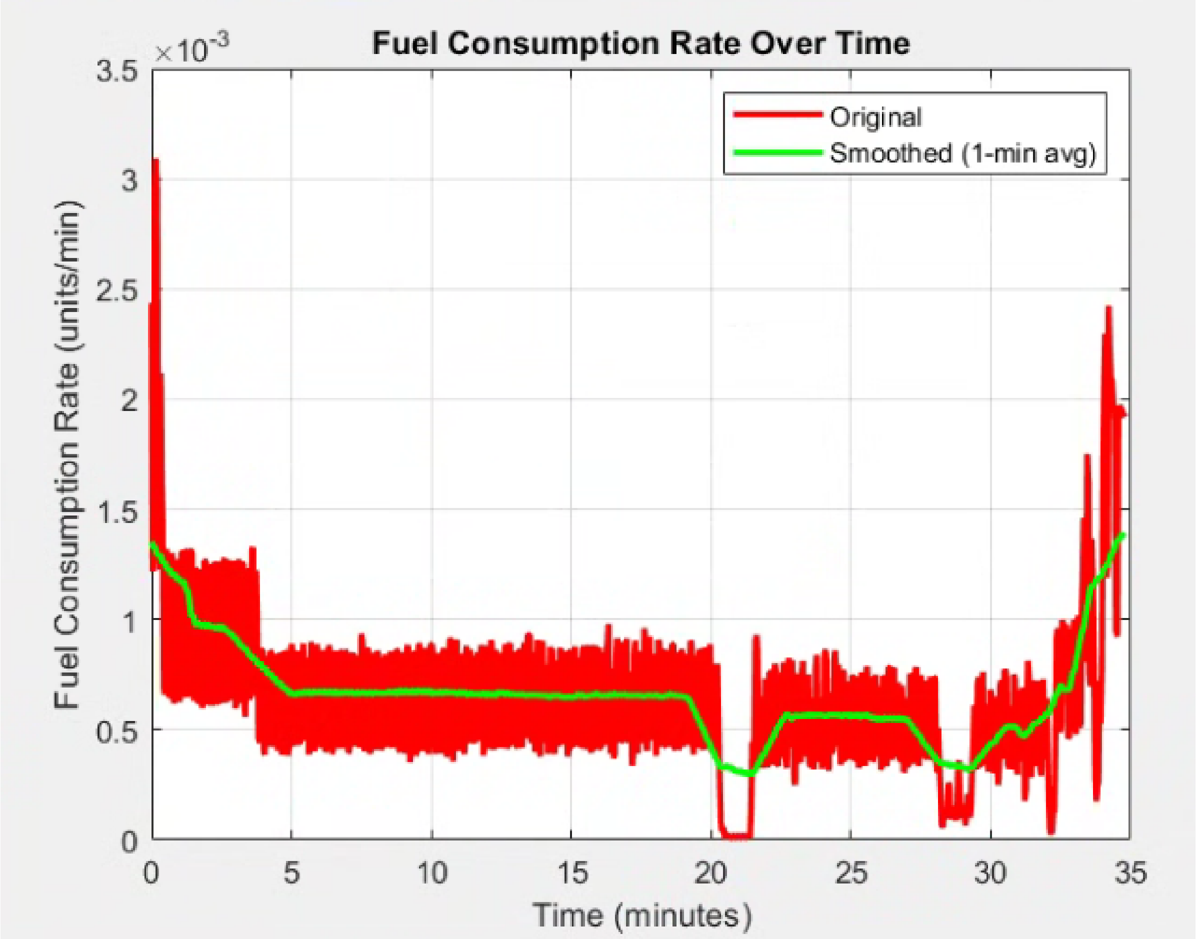


Figure 36: Energy (Fuel) Consumption Rate of Proposed
Santa Cruz to Berkeley VFR Route

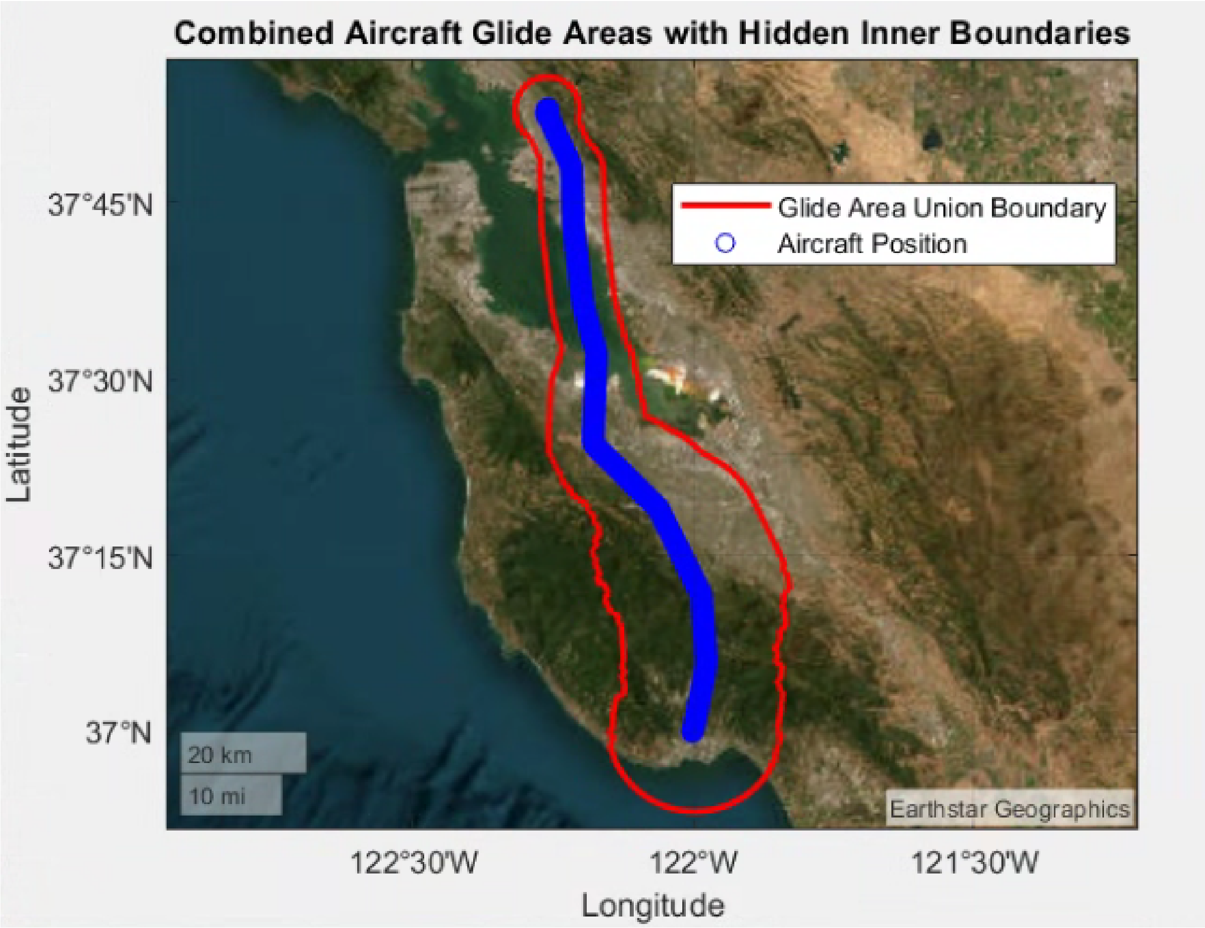


Figure 37: Glide range union boundary for Proposed Santa Cruz to Berkeley Route

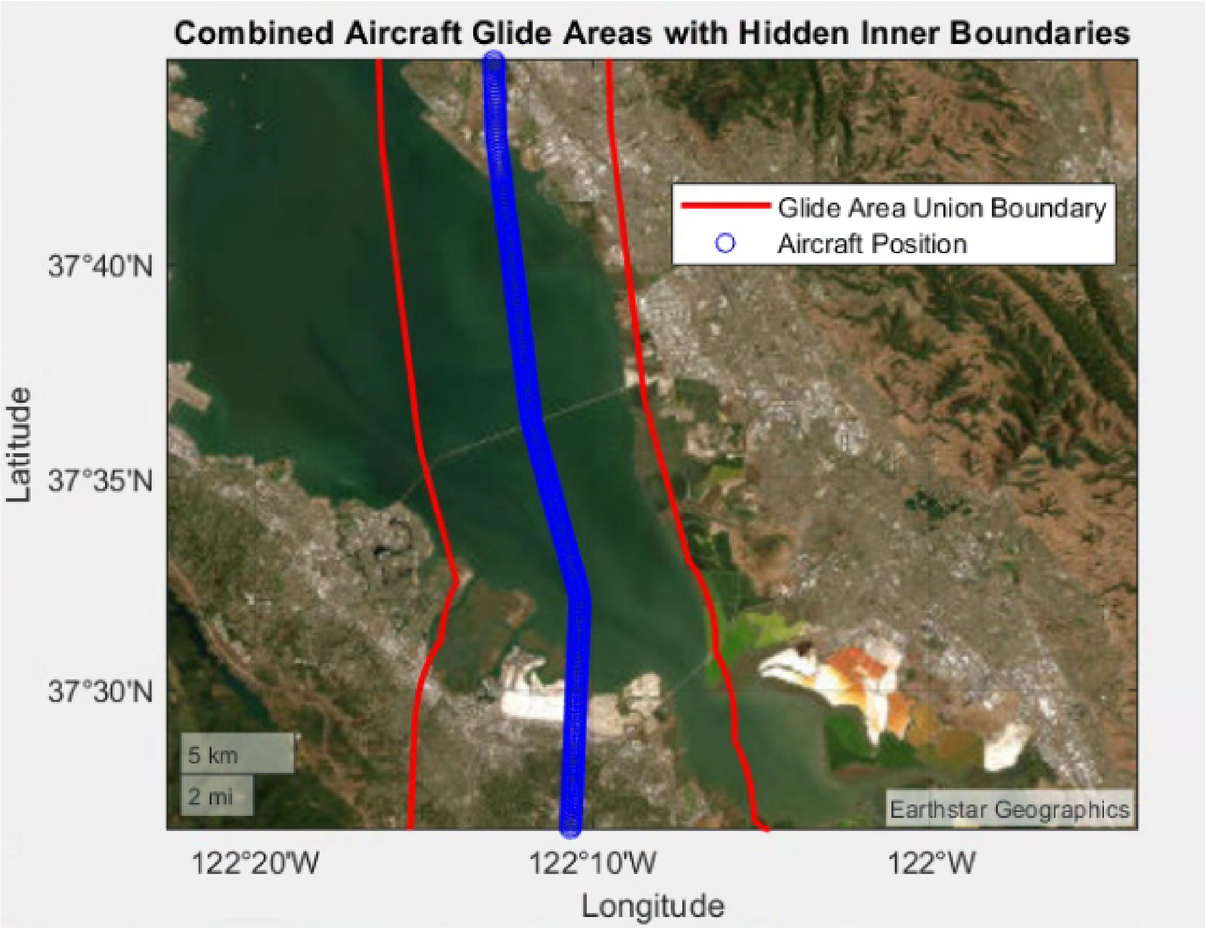


Figure 38: Glide range union boundary (Close up of SF Bay) for Proposed Santa Cruz to Berkeley Route

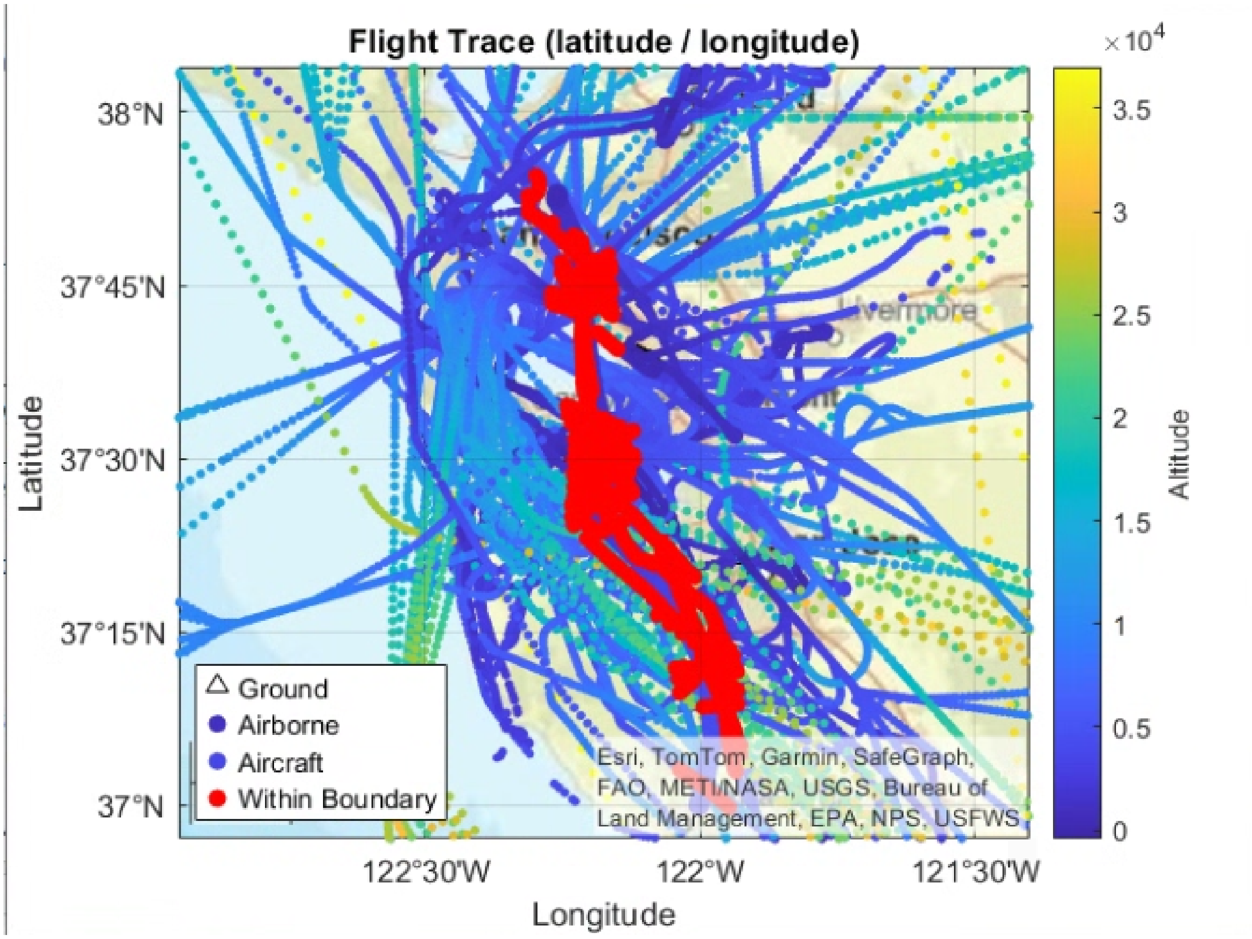


Figure 39: All Conflicts (IFR Spacing) on Proposed Santa Cruz to Berkeley Route

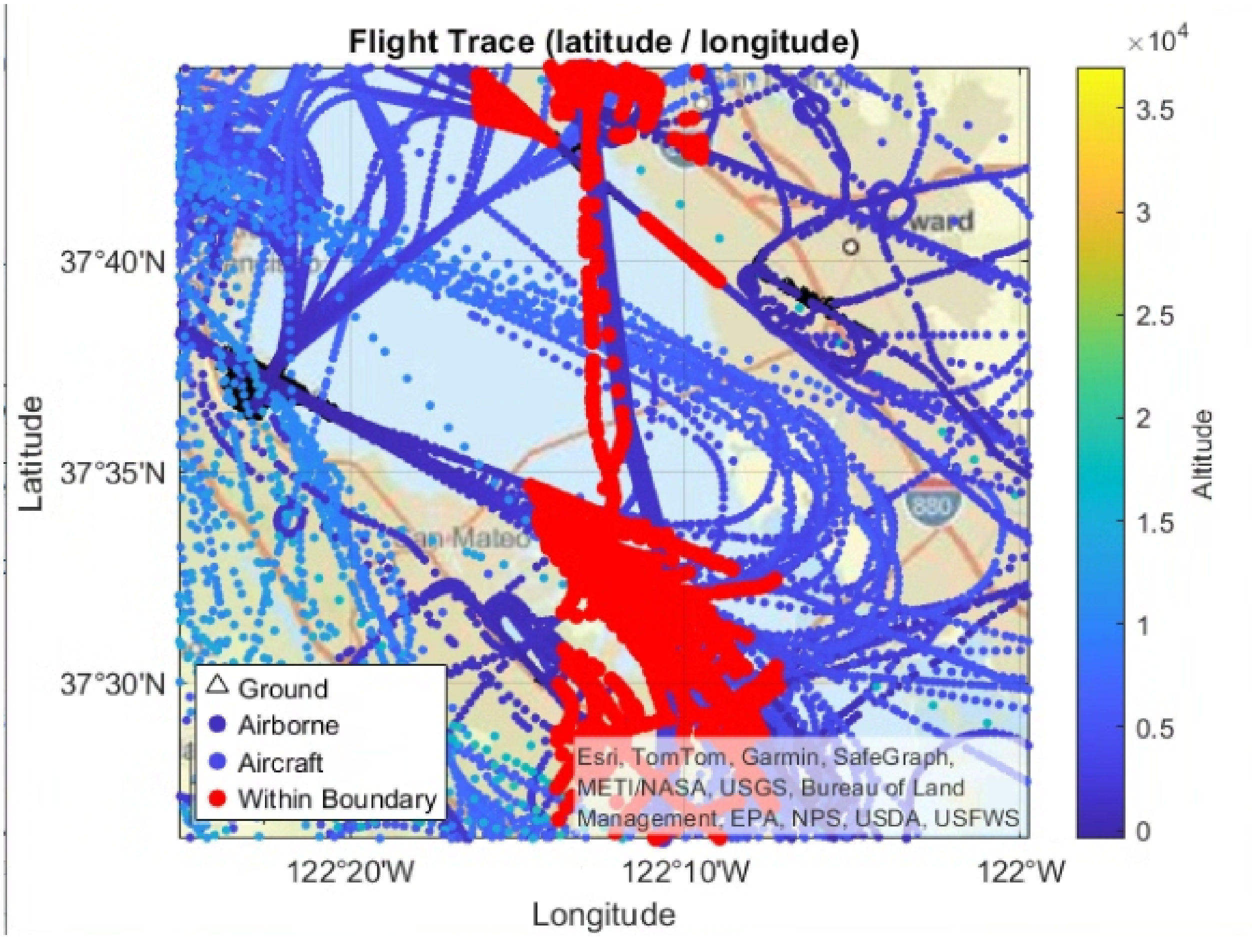

Flight Trace (latitude / longitude)
Latitude
37°40'N
37°35'N
37°30'N
Longitude
122°20'W
122°10'W
122°W
△ Ground
● Airborne
● Aircraft
● Within Boundary
Altitude
×10^4
3.5
3
2.5
2
1.5
1
0.5
0
Esri, TomTom, Garmin, SafeGraph, METI/NASA, USGS, Bureau of Land Management, EPA, NPS, USDA, USFWS

Figure 40: Close up of Conflicts (IFR Spacing) for SFO and OAK Arrivals and Departures on Proposed Santa Cruz to Berkeley Route

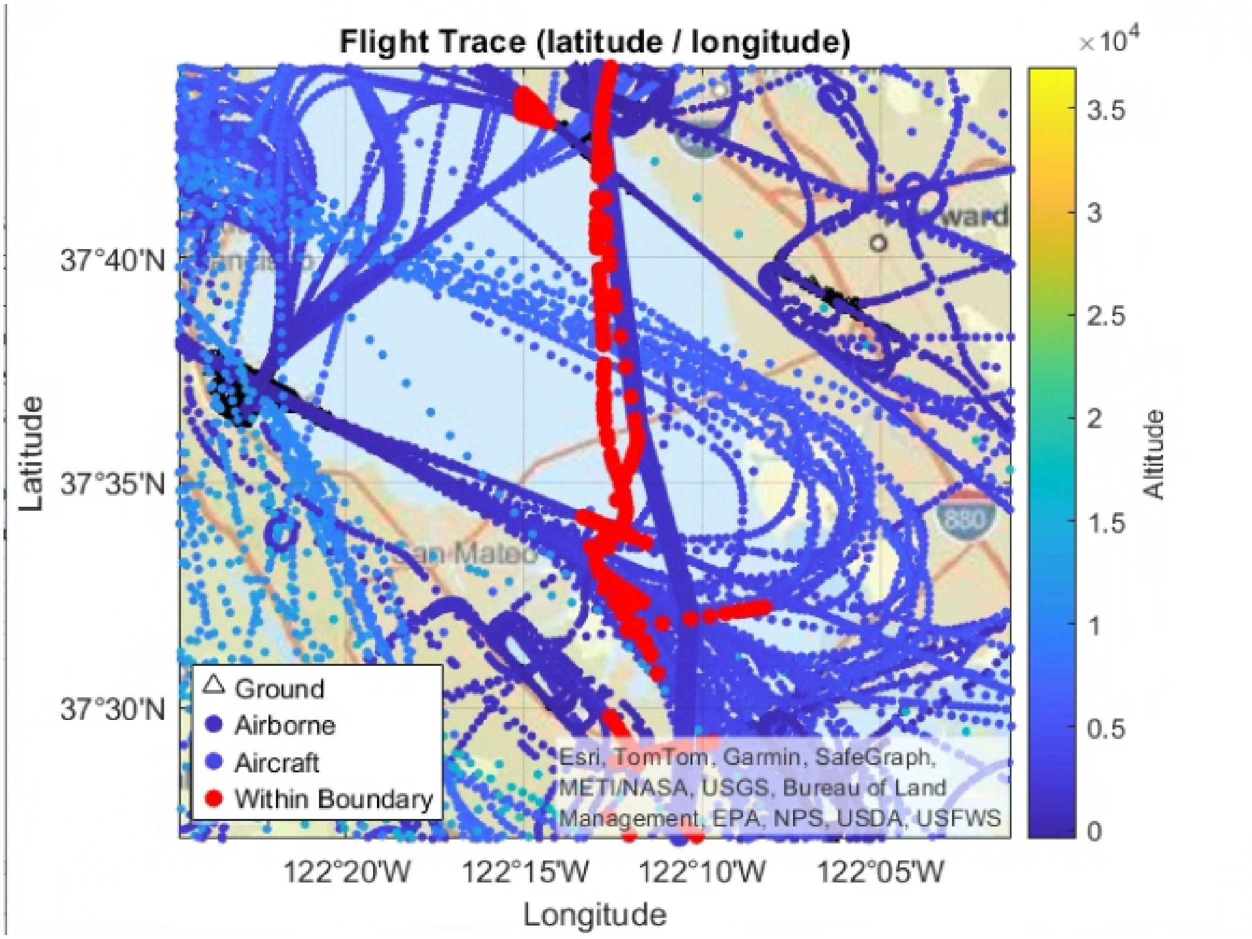


Figure 41: Conflicts (Reduced Spacing) on Proposed Santa Cruz to Berkeley Route

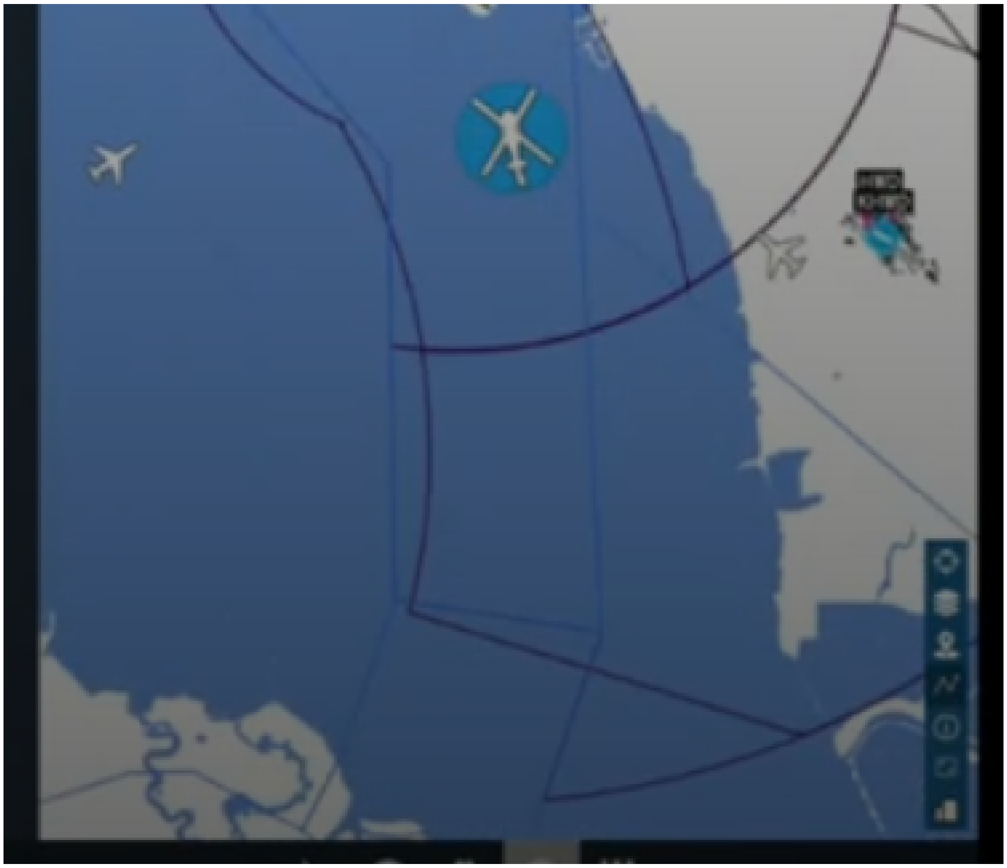
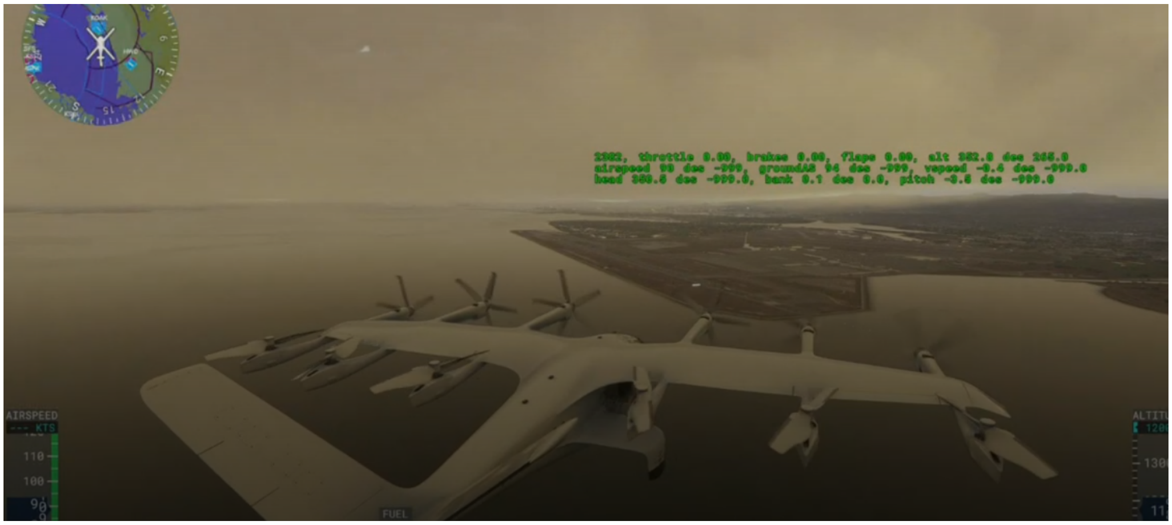
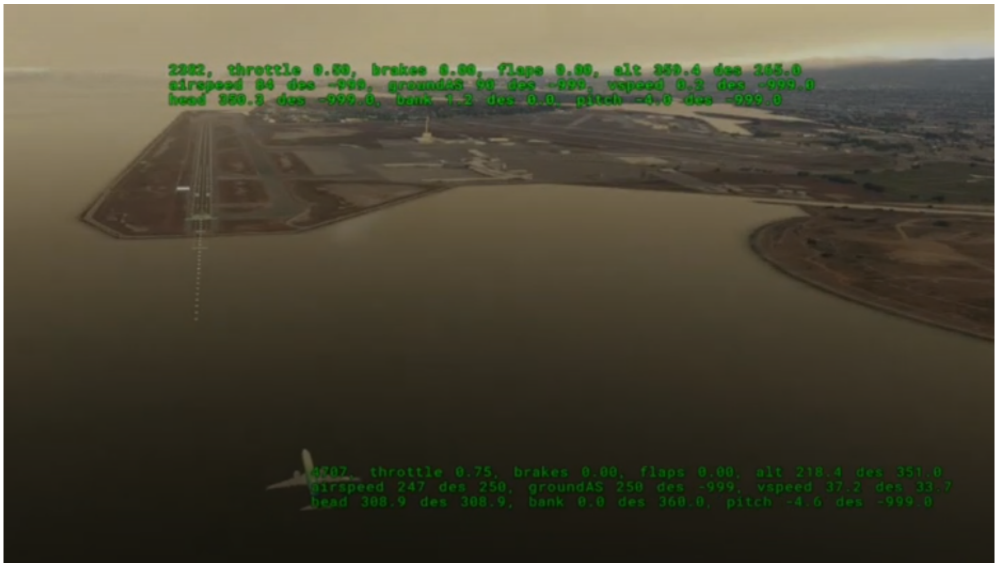


Figure 42: Scenic Scenario Setup



```
2382, throttle 0.00, brakes 0.00, flaps 0.00, alt 382.0 des 265.0
airspeed 90 des -999, groundAS 94 des -999, vspeed -0.4 des -999.0
head 350.6 des -999.0, bank 0.1 des 0.0, pitch -3.6 des -999.0
```

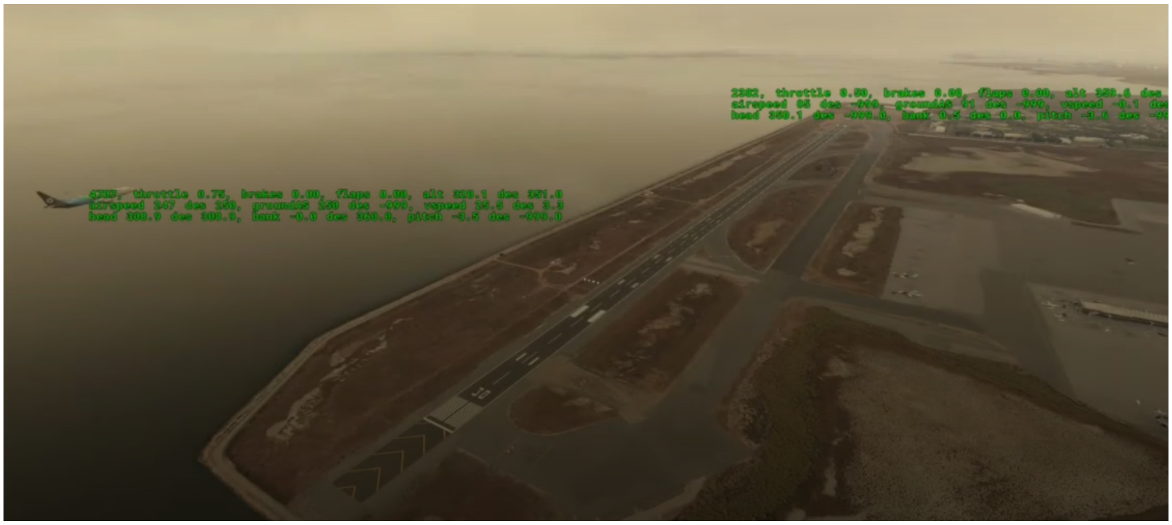
Figure 43: Archer Midnight Flying ahead on VFR route



```
2882, throttle 0.50, brakes 0.00, flaps 0.00, alt 350.4 des 365.0
airspeed 84 des -999, groundAS 90 des -999, vspeed 0.2 des -999.0
head 350.3 des -999.0, bank 1.2 des 0.0, pitch -4.0 des -999.0
```

```
4707, throttle 0.75, brakes 0.00, flaps 0.00, alt 218.4 des 351.0
airspeed 247 des 250, groundAS 250 des -999, vspeed 37.2 des 33.7
head 308.9 des 308.9, bank 0.0 des 360.0, pitch -4.6 des -999.0
```

Figure 44: 737 on Final Approach into 30, about to perform a go-around with an eVTOL flying ahead.



```
4707, throttle 0.75, brakes 0.00, flaps 0.00, alt 320.1 des 351.0
airspeed 247 des 250, groundAS 250 des -999, vspeed 25.5 des 3.3
head 308.9 des 308.9, bank -0.0 des 360.0, pitch -3.5 des -999.0
```

```
2382, throttle 0.50, brakes 0.00, flaps 0.00, alt 350.6 des
airspeed 85 des -999, groundAS 91 des -999, vspeed -0.1 des
head 350.1 des -999.0, bank 0.5 des 0.0, pitch -3.6 des -99
```

Figure 45: 737 Max in Go Around within 1 mile and 500 feet of eVTOL Aircraft

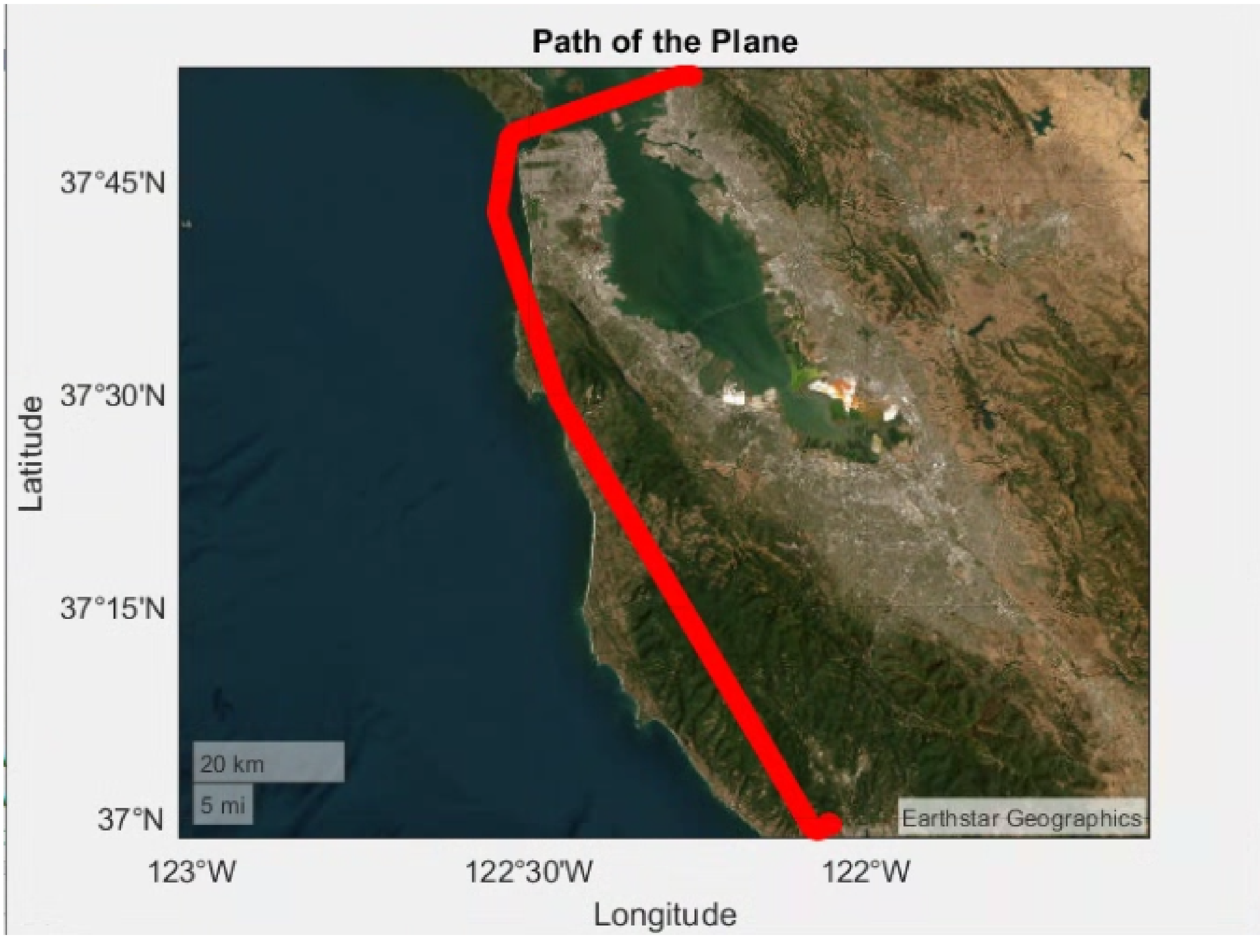


Figure 46: Santa Cruz to Berkeley Proposed Route to Avoid IFR Traffic

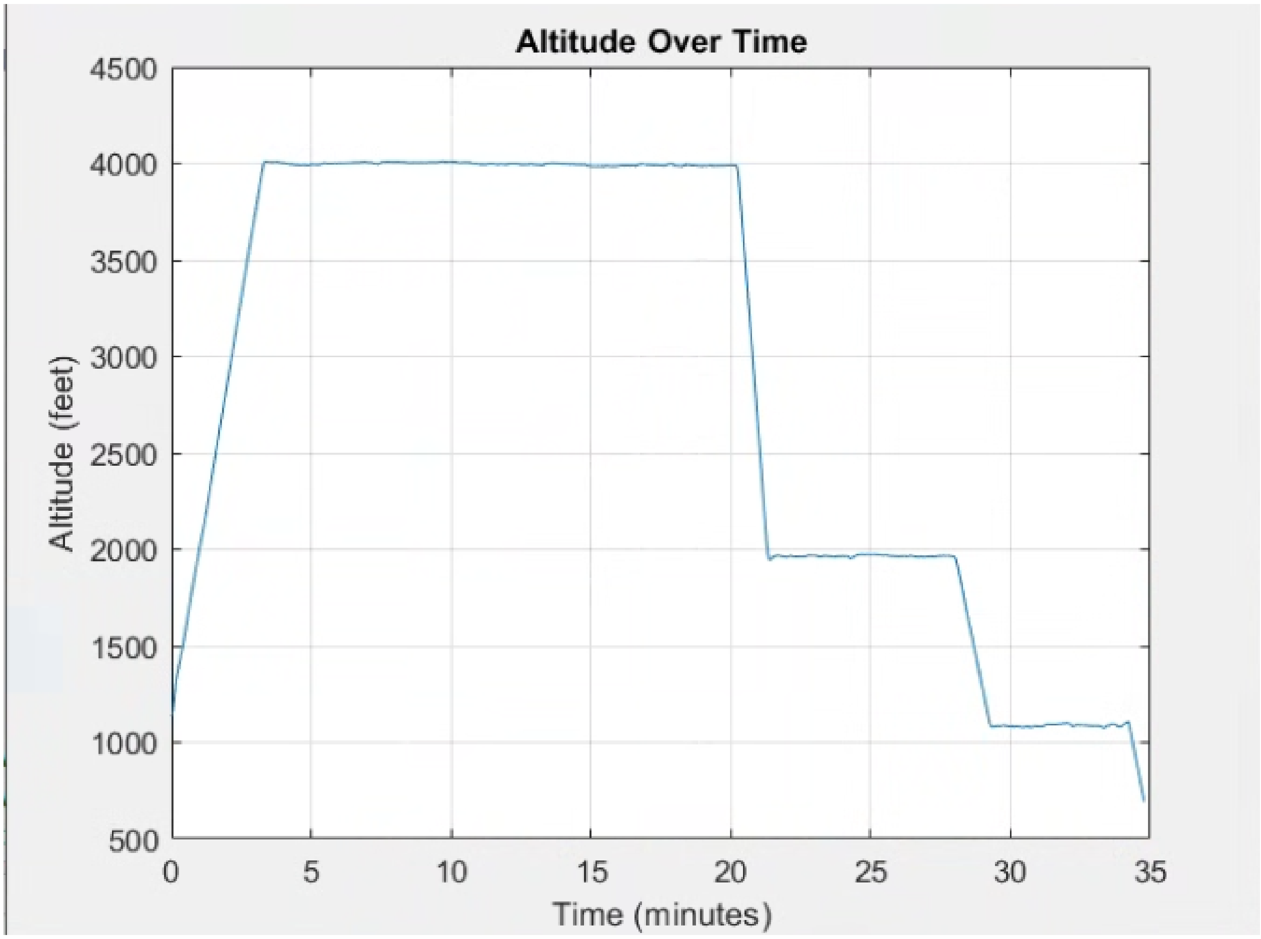


Figure 47: Santa Cruz to Berkeley Proposed IFR Route
Altitude over Time

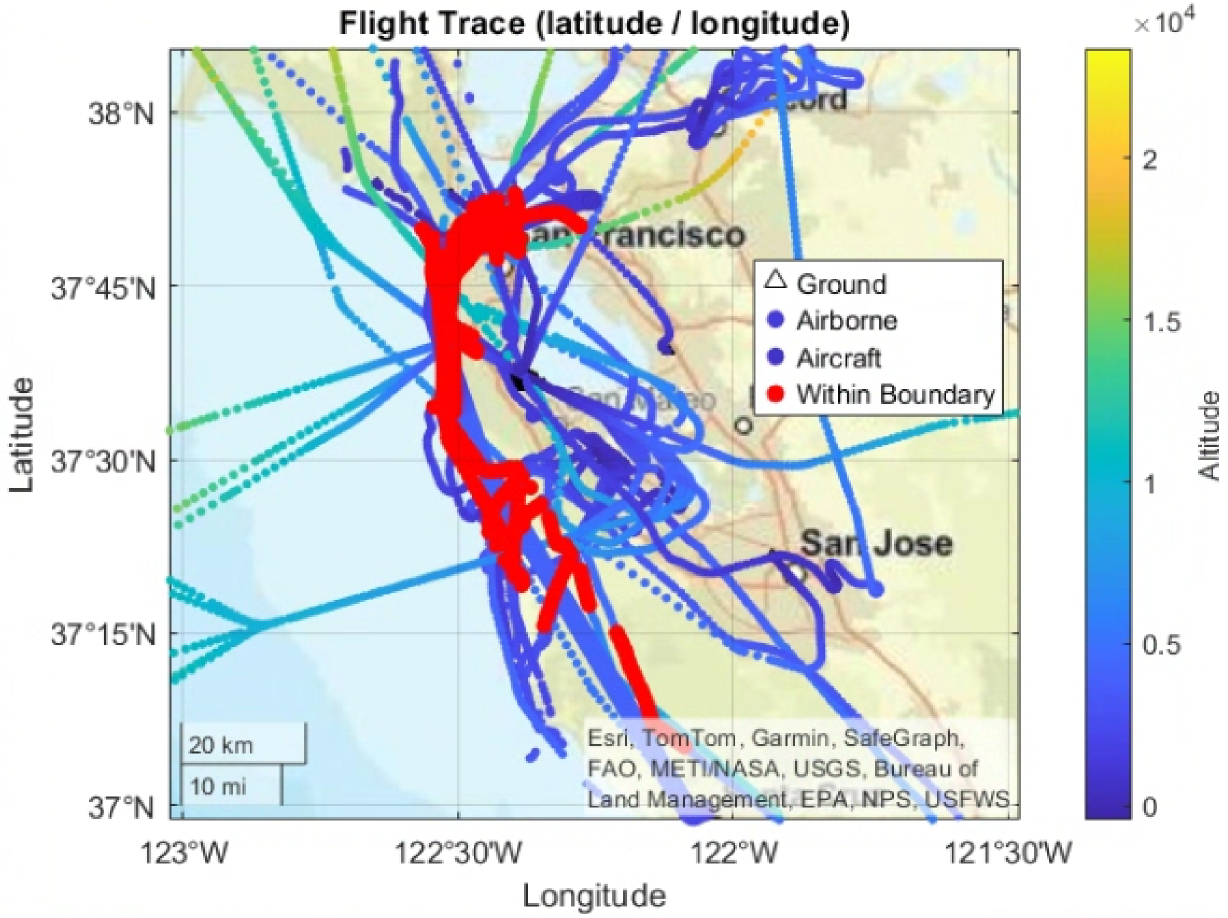


Figure 48: All Conflicts (IFR Spacing) on Proposed Santa Cruz to Berkeley IFR Route

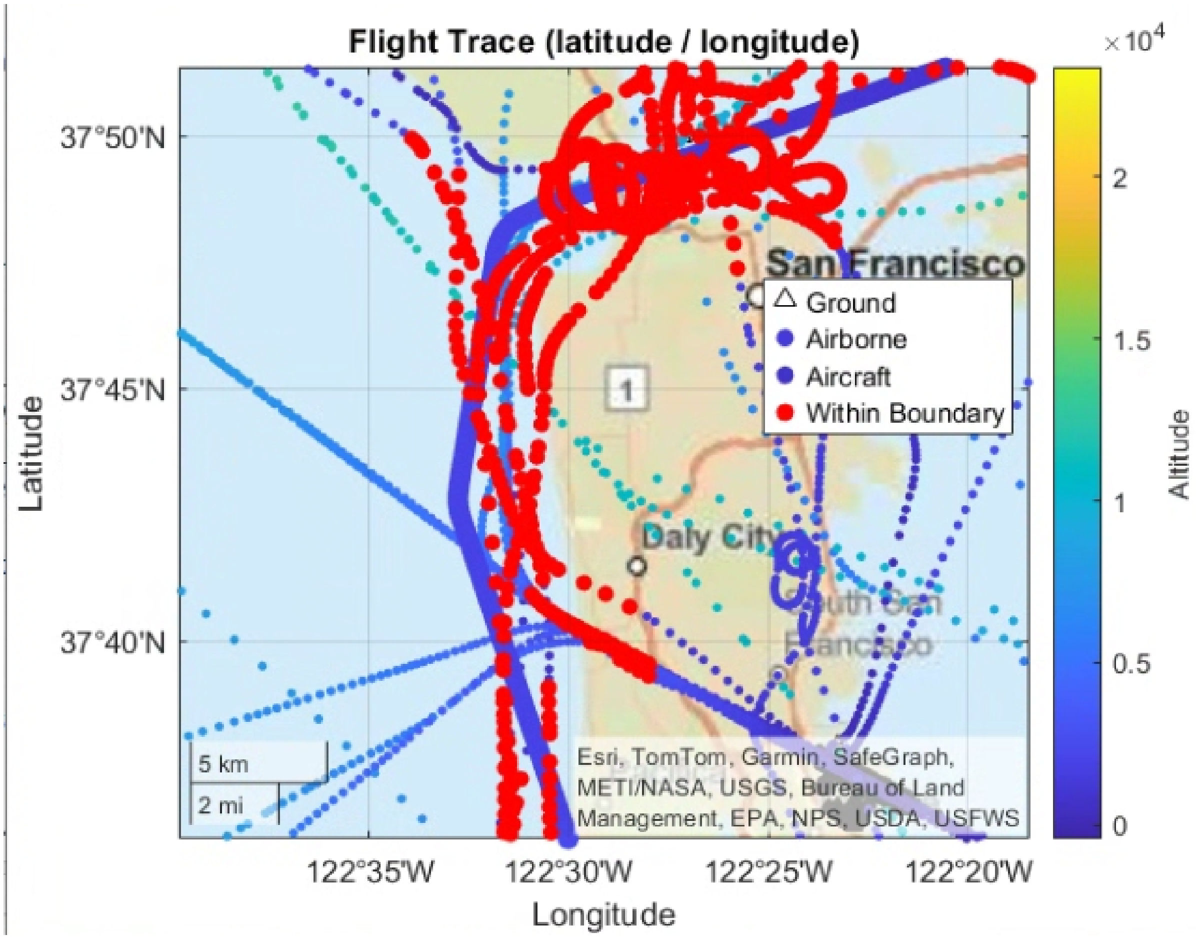


Figure 49: Close up of Conflicts (IFR Spacing) for SFO and OAK Arrivals and Departures on Proposed Santa Cruz to Berkeley IFR Route

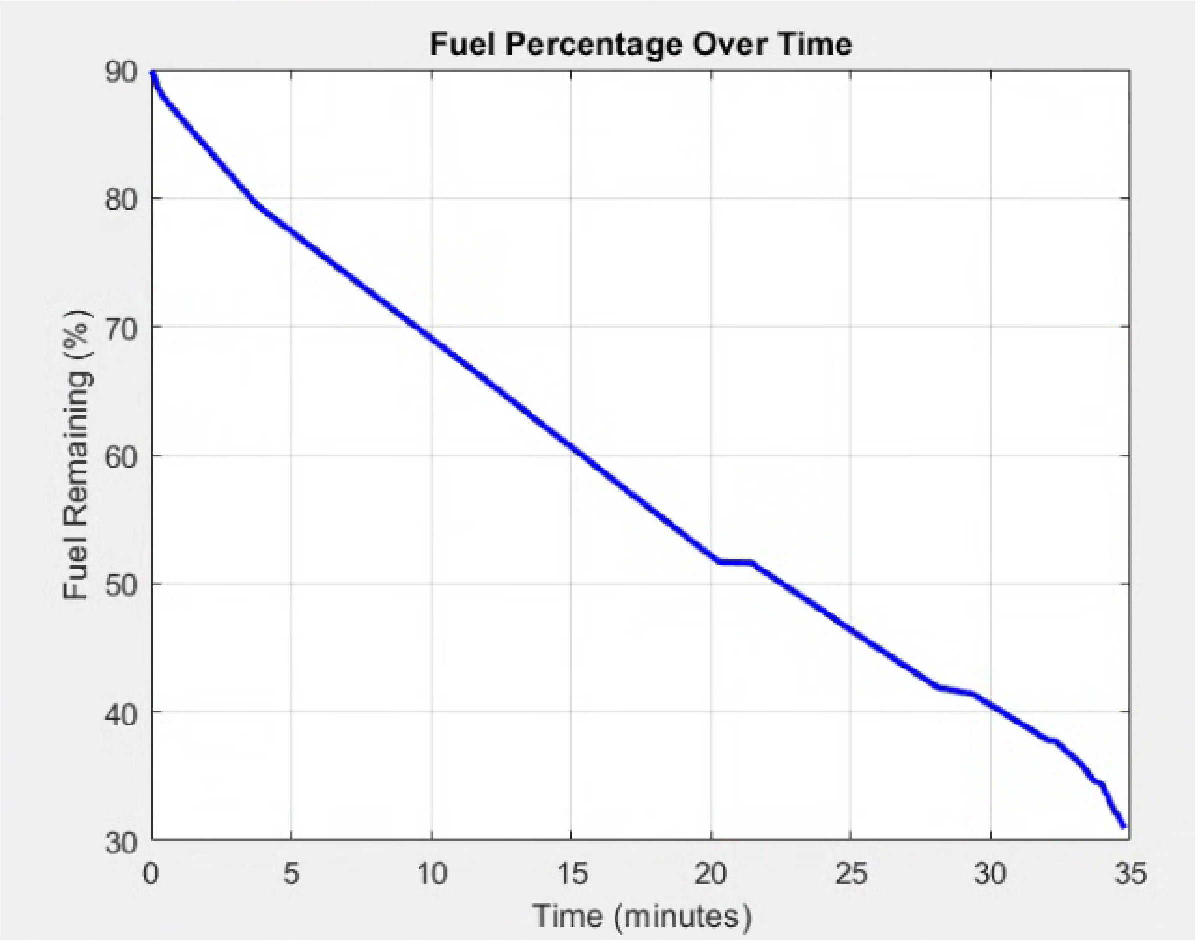


Figure 50: Energy (Fuel) Consumption of Proposed Santa Cruz to Berkeley IFR Route

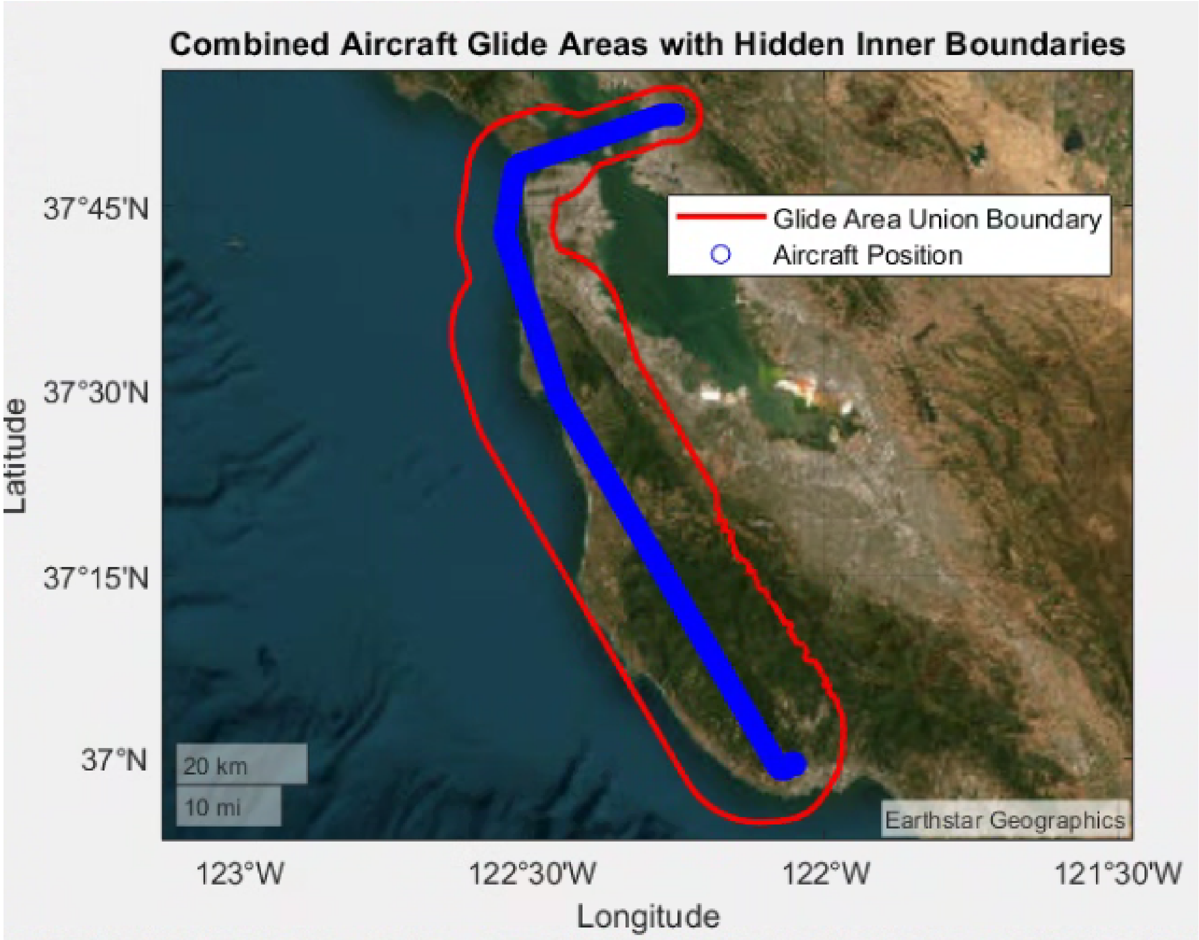


Figure 51: Glide range union boundary for Santa Cruz to Berkeley Proposed Route to Avoid IFR Traffic

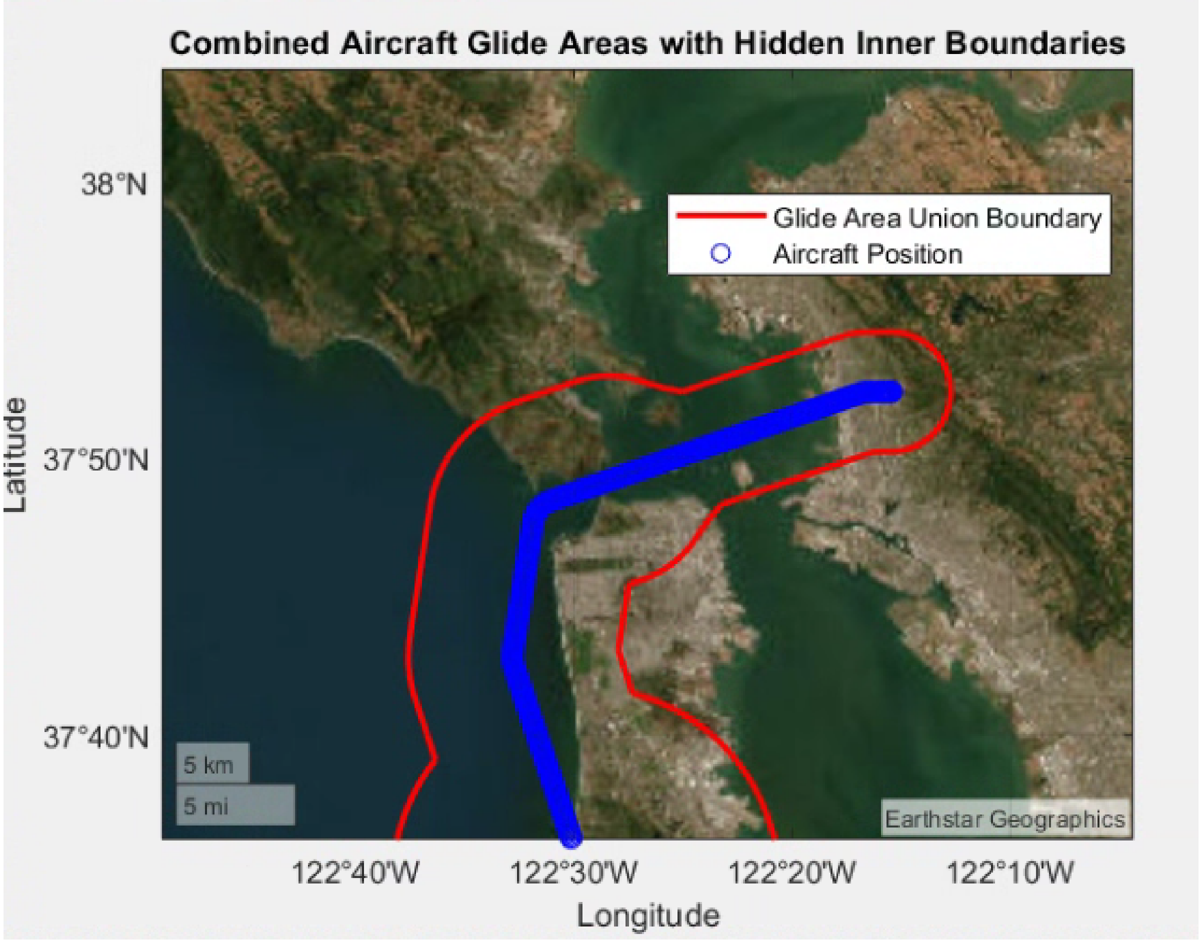


Figure 52: Glide range union boundary (Close up of San Francisco) for Santa Cruz to Berkeley Proposed Route to Avoid IFR Traffic

7.3 Section 4: Terminal Procedures

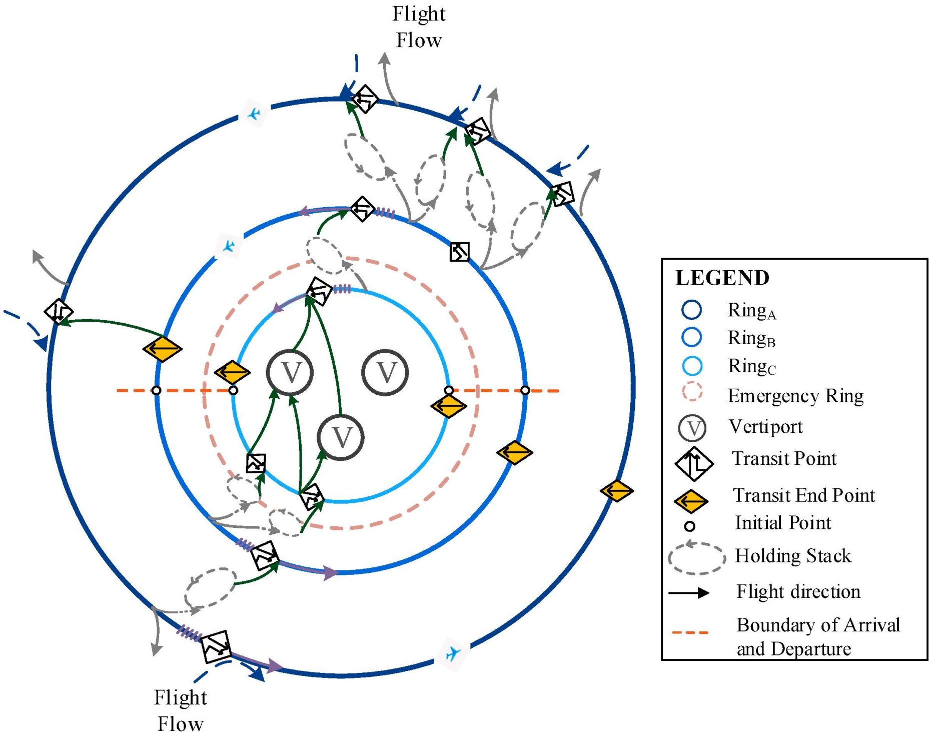


Figure 53: Operational concept design for terminal airspace

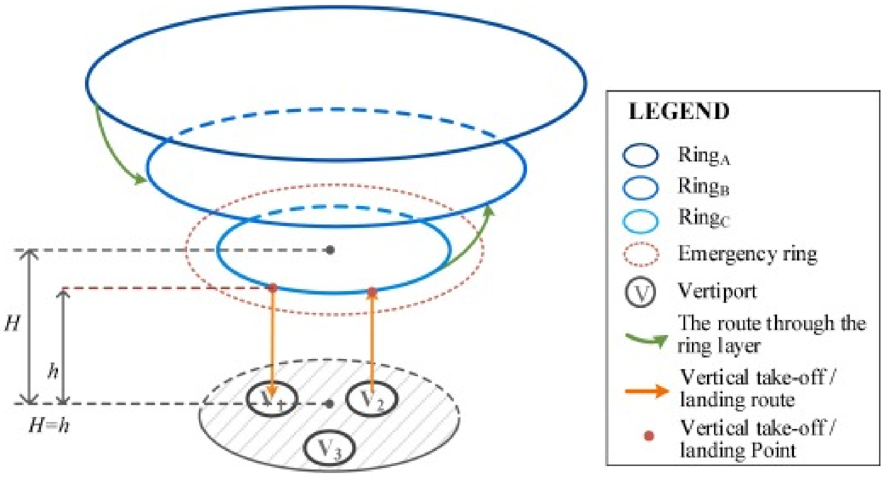
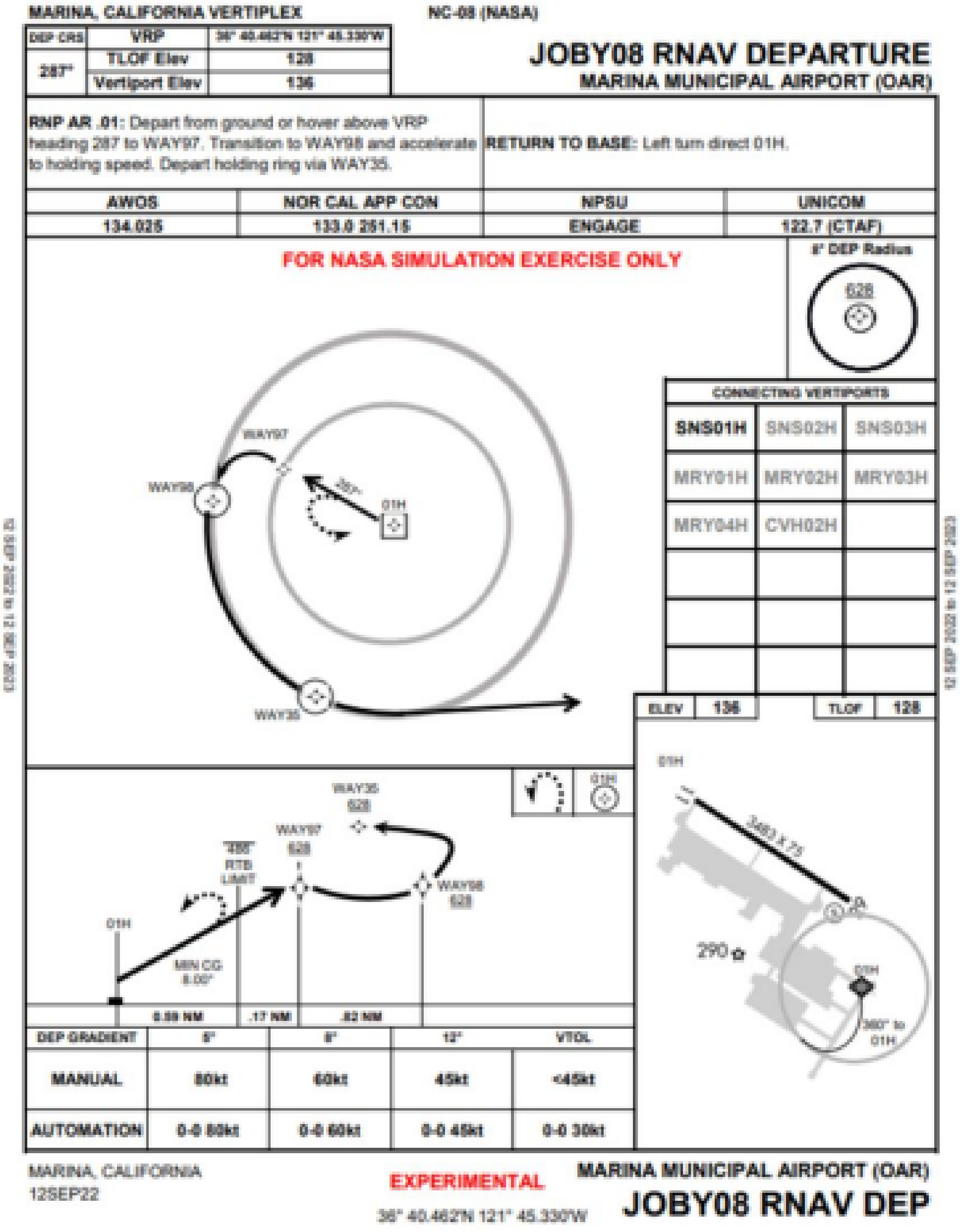

H
h
H=h
V_1
V_2
V_3
LEGEND
Ring$_A$
Ring$_B$
Ring$_C$
Emergency ring
V Vertiport
The route through the ring layer
Vertical take-off / landing route
Vertical take-off / landing Point

Figure 54: 3D Operational concept design for terminal airspace



MARINA, CALIFORNIA VERTIPLEX

NC-08 (NASA)

DEP CRS	VRP	36° 40.462'N 121° 45.330'W
287°	TLOF Elev	128
	Vertiport Elev	136

JOBY08 RNAV DEPARTURE

MARINA MUNICIPAL AIRPORT (OAR)

RNP AR .01: Depart from ground or hover above VRP heading 287 to WAY97. Transition to WAY98 and accelerate to holding speed. Depart holding ring via WAY35.

RETURN TO BASE: Left turn direct 01H.

AWOS	NOR CAL APP CON	NPSU	UNICOM
134.025	133.0 251.15	ENGAGE	122.7 (CTAF)

FOR NASA SIMULATION EXERCISE ONLY

8' DEP Radius

CONNECTING VERTIPORTS		
SNS01H	SNS02H	SNS03H
MRY01H	MRY02H	MRY03H
MRY04H	CVH02H	

ELEV	136	TLOF	128

DEP GRADIENT	5°	8°	12°	VTOL
MANUAL	80kt	60kt	45kt	<45kt
AUTOMATION	0-0 80kt	0-0 60kt	0-0 45kt	0-0 30kt

MARINA, CALIFORNIA
12SEP22

EXPERIMENTAL

36° 40.462'N 121° 45.330'W

MARINA MUNICIPAL AIRPORT (OAR)

JOBY08 RNAV DEP

Figure 55: Joby 8 Departure Digram

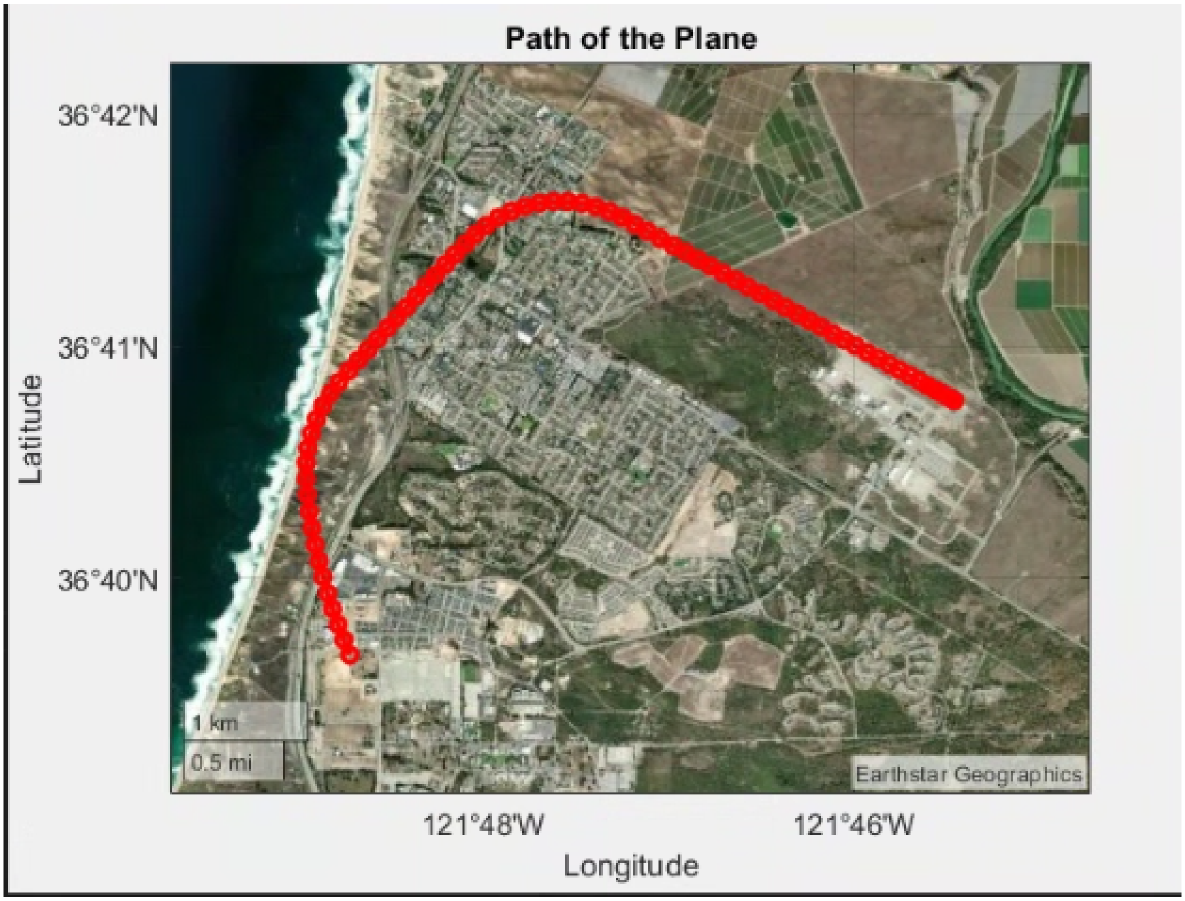


Figure 56: Joby 8 Departure Path Flown

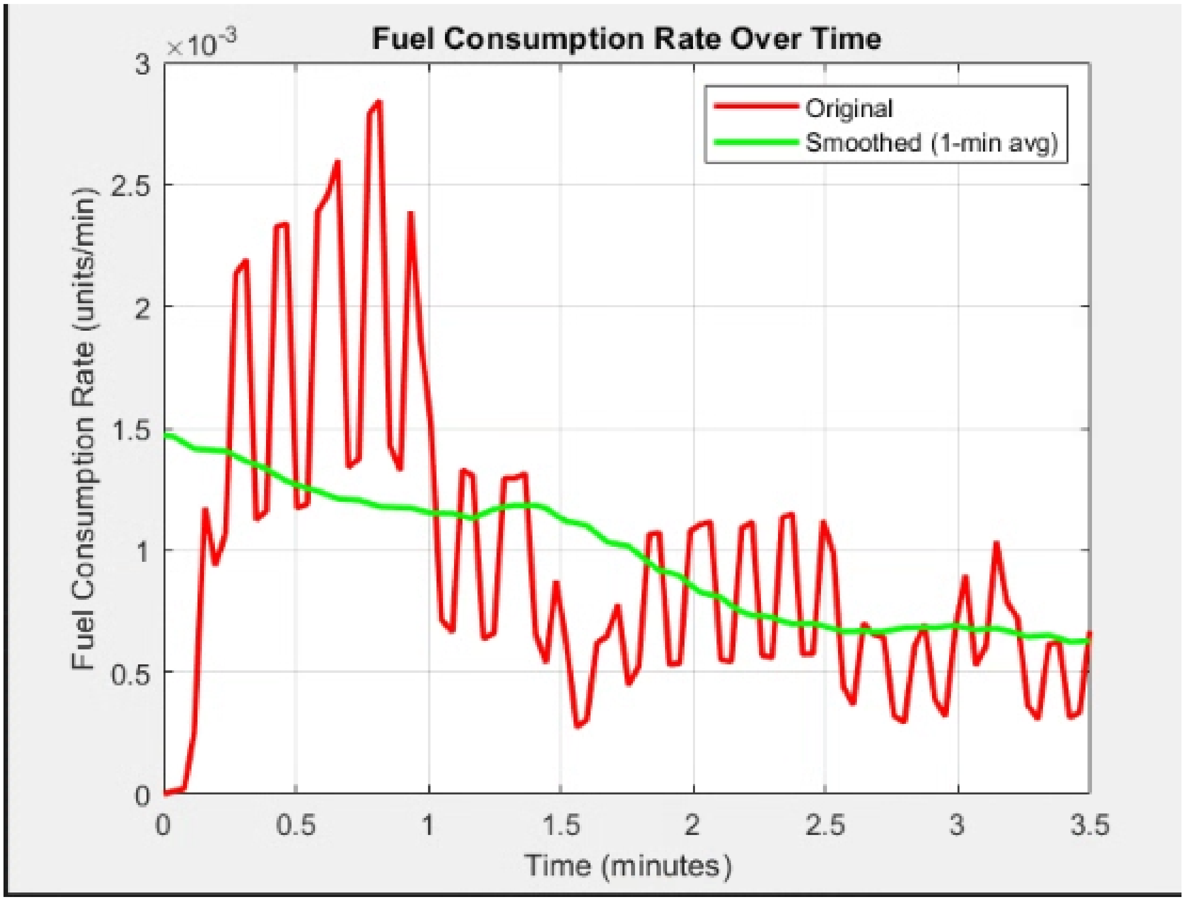


Figure 57: Joby 8 Departure Energy Consumption

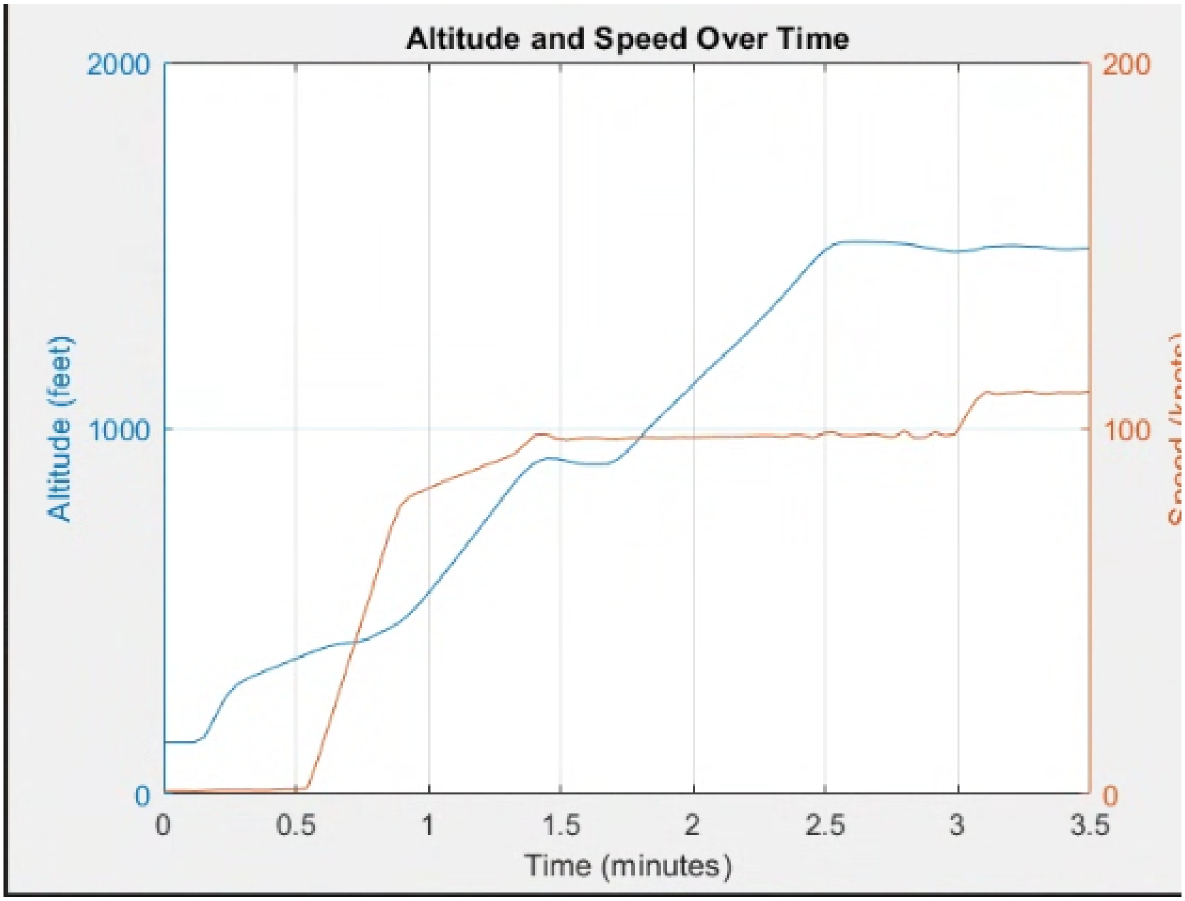


Figure 58: Joby 8 Departure Profile

7.4 Section 5: Scheduling

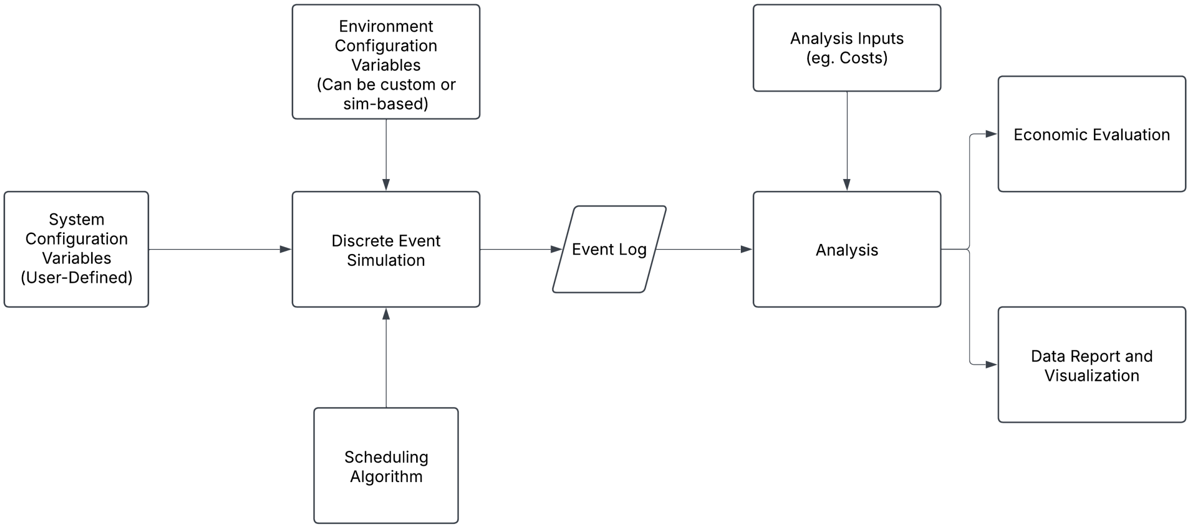
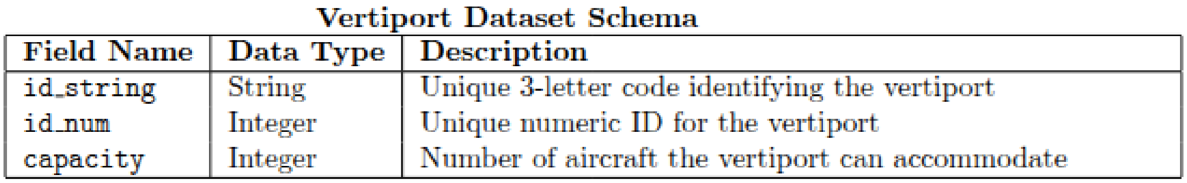

Environment Configuration Variables (Can be custom or sim-based)
Analysis Inputs (eg. Costs)
Economic Evaluation
System Configuration Variables (User-Defined)
Discrete Event Simulation
Event Log
Analysis
Data Report and Visualization
Scheduling Algorithm

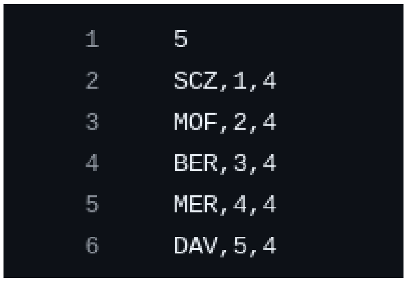
Figure 59: Discrete Event Simulator Process Diagram



Vertiport Dataset Schema

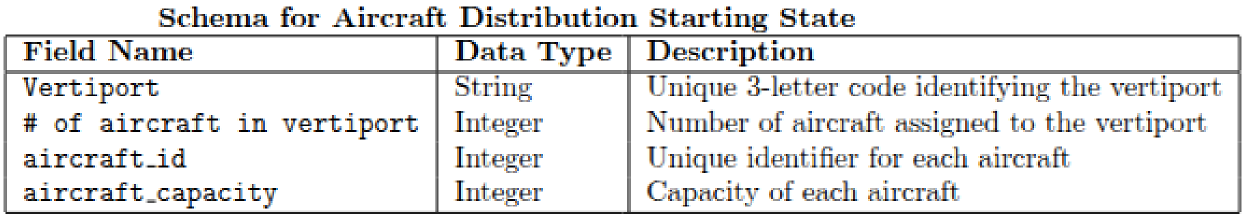
Field Name	Data Type	Description
`id_string`	String	Unique 3-letter code identifying the vertiport
`id_num`	Integer	Unique numeric ID for the vertiport
`capacity`	Integer	Number of aircraft the vertiport can accommodate

Figure 60: Vertiport Data Schema



```
5
SCZ,1,4
MOF,2,4
BER,3,4
MER,4,4
DAV,5,4
```

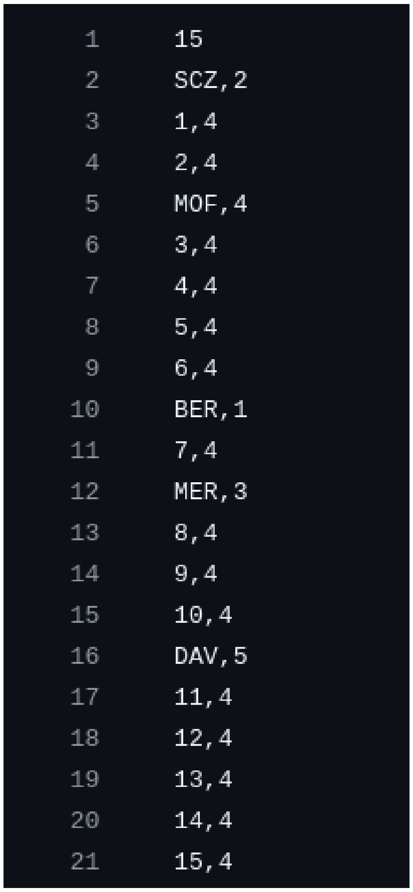
Figure 61: Example Vertiport Definition



Schema for Aircraft Distribution Starting State

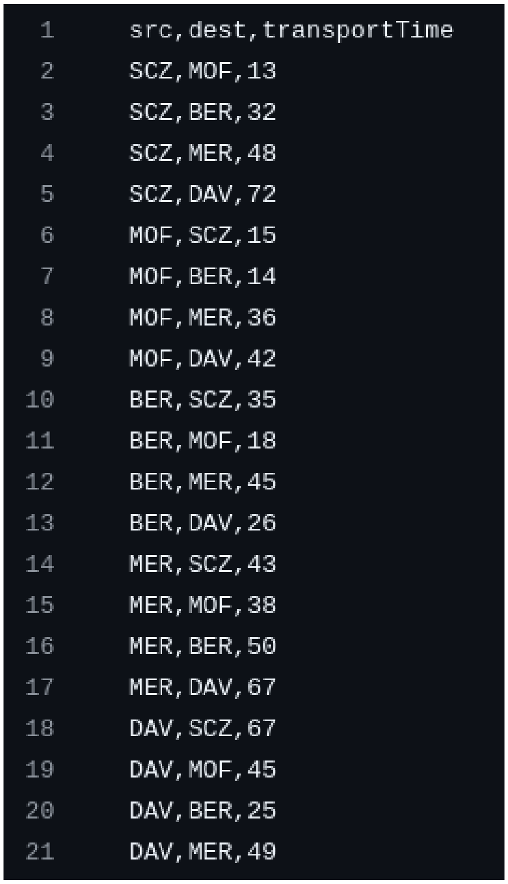
Field Name	Data Type	Description
Vertiport	String	Unique 3-letter code identifying the vertiport
# of aircraft in vertiport	Integer	Number of aircraft assigned to the vertiport
aircraft_id	Integer	Unique identifier for each aircraft
aircraft_capacity	Integer	Capacity of each aircraft

Figure 62: Starting State Data Schema



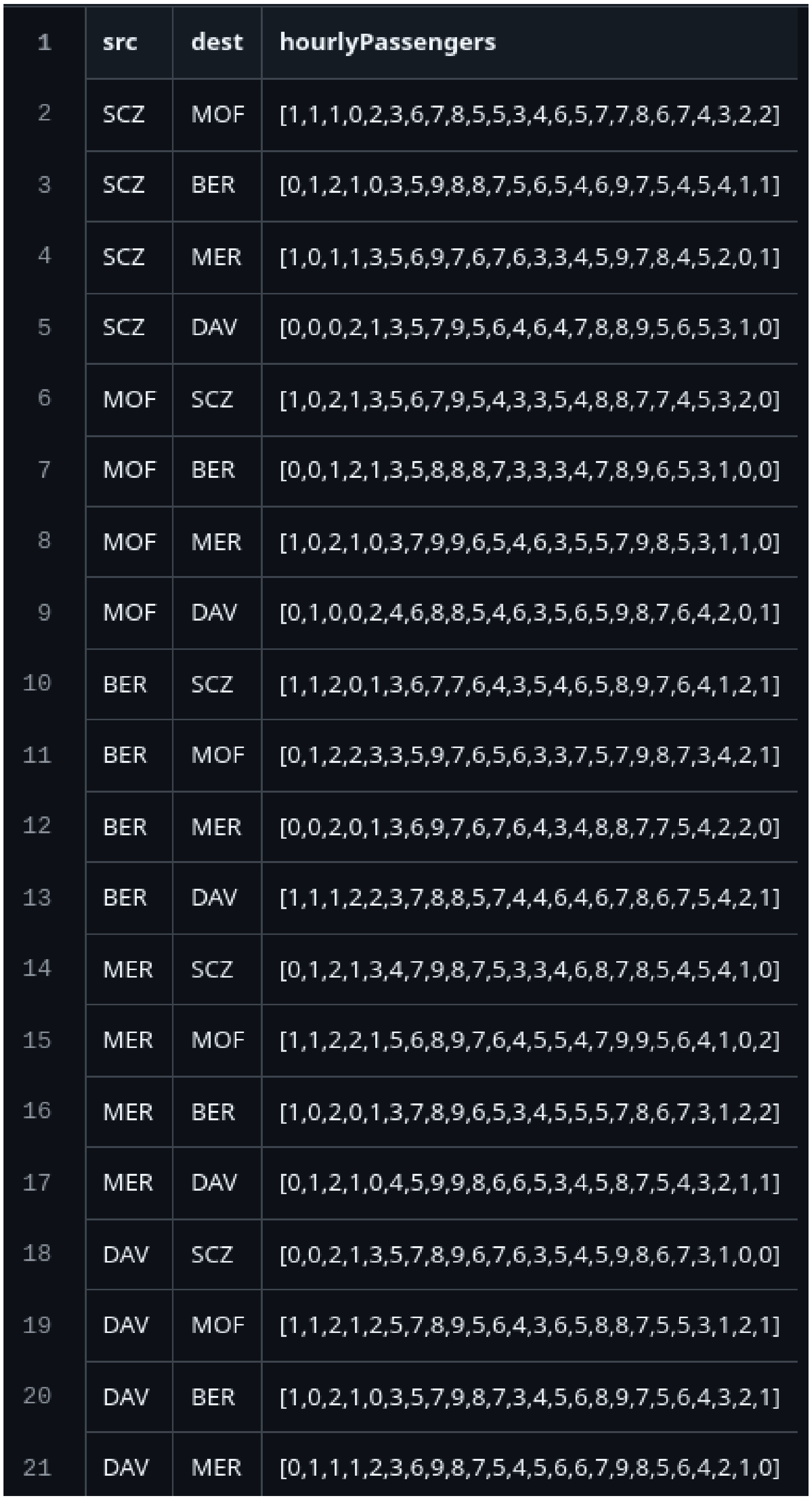
```
15
SCZ,2
1,4
2,4
MOF,4
3,4
4,4
5,4
6,4
BER,1
7,4
MER,3
8,4
9,4
10,4
DAV,5
11,4
12,4
13,4
14,4
15,4
```

Figure 63: Example Starting State Definition



```
src,dest,transportTime
SCZ,MOF,13
SCZ,BER,32
SCZ,MER,48
SCZ,DAV,72
MOF,SCZ,15
MOF,BER,14
MOF,MER,36
MOF,DAV,42
BER,SCZ,35
BER,MOF,18
BER,MER,45
BER,DAV,26
MER,SCZ,43
MER,MOF,38
MER,BER,50
MER,DAV,67
DAV,SCZ,67
DAV,MOF,45
DAV,BER,25
DAV,MER,49
```

Figure 64: Example Transport Times Definition (Derived from Simulator Tests)



src	dest	hourlyPassengers
SCZ	MOF	[1,1,1,0,2,3,6,7,8,5,5,3,4,6,5,7,7,8,6,7,4,3,2,2]
SCZ	BER	[0,1,2,1,0,3,5,9,8,8,7,5,6,5,4,6,9,7,5,4,5,4,1,1]
SCZ	MER	[1,0,1,1,3,5,6,9,7,6,7,6,3,3,4,5,9,7,8,4,5,2,0,1]
SCZ	DAV	[0,0,0,2,1,3,5,7,9,5,6,4,6,4,7,8,8,9,5,6,5,3,1,0]
MOF	SCZ	[1,0,2,1,3,5,6,7,9,5,4,3,3,5,4,8,8,7,7,4,5,3,2,0]
MOF	BER	[0,0,1,2,1,3,5,8,8,8,7,3,3,3,4,7,8,9,6,5,3,1,0,0]
MOF	MER	[1,0,2,1,0,3,7,9,9,6,5,4,6,3,5,5,7,9,8,5,3,1,1,0]
MOF	DAV	[0,1,0,0,2,4,6,8,8,5,4,6,3,5,6,5,9,8,7,6,4,2,0,1]
BER	SCZ	[1,1,2,0,1,3,6,7,7,6,4,3,5,4,6,5,8,9,7,6,4,1,2,1]
BER	MOF	[0,1,2,2,3,3,5,9,7,6,5,6,3,3,7,5,7,9,8,7,3,4,2,1]
BER	MER	[0,0,2,0,1,3,6,9,7,6,7,6,4,3,4,8,8,7,7,5,4,2,2,0]
BER	DAV	[1,1,1,2,2,3,7,8,8,5,7,4,4,6,4,6,7,8,6,7,5,4,2,1]
MER	SCZ	[0,1,2,1,3,4,7,9,8,7,5,3,3,4,6,8,7,8,5,4,5,4,1,0]
MER	MOF	[1,1,2,2,1,5,6,8,9,7,6,4,5,5,4,7,9,9,5,6,4,1,0,2]
MER	BER	[1,0,2,0,1,3,7,8,9,6,5,3,4,5,5,5,7,8,6,7,3,1,2,2]
MER	DAV	[0,1,2,1,0,4,5,9,9,8,6,6,5,3,4,5,8,7,5,4,3,2,1,1]
DAV	SCZ	[0,0,2,1,3,5,7,8,9,6,7,6,3,5,4,5,9,8,6,7,3,1,0,0]
DAV	MOF	[1,1,2,1,2,5,7,8,9,5,6,4,3,6,5,8,8,7,5,5,3,1,2,1]
DAV	BER	[1,0,2,1,0,3,5,7,9,8,7,3,4,5,6,8,9,7,5,6,4,3,2,1]
DAV	MER	[0,1,1,1,2,3,6,9,8,7,5,4,5,6,6,7,9,8,5,6,4,2,1,0]

Figure 65: Example Passenger Demand Definition

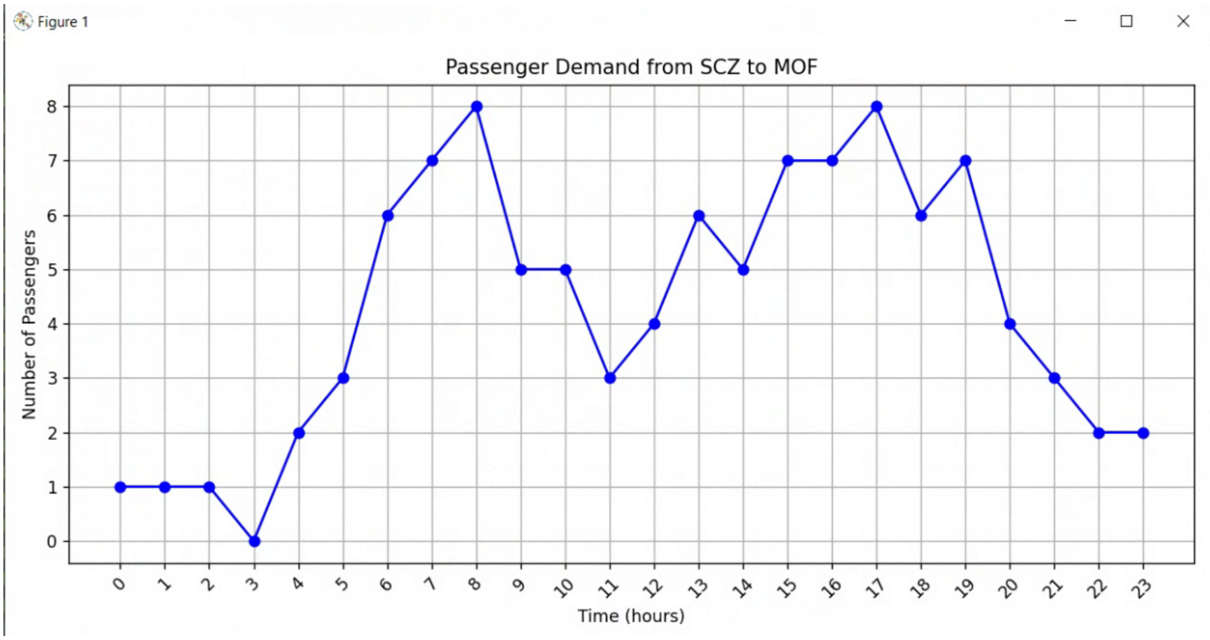
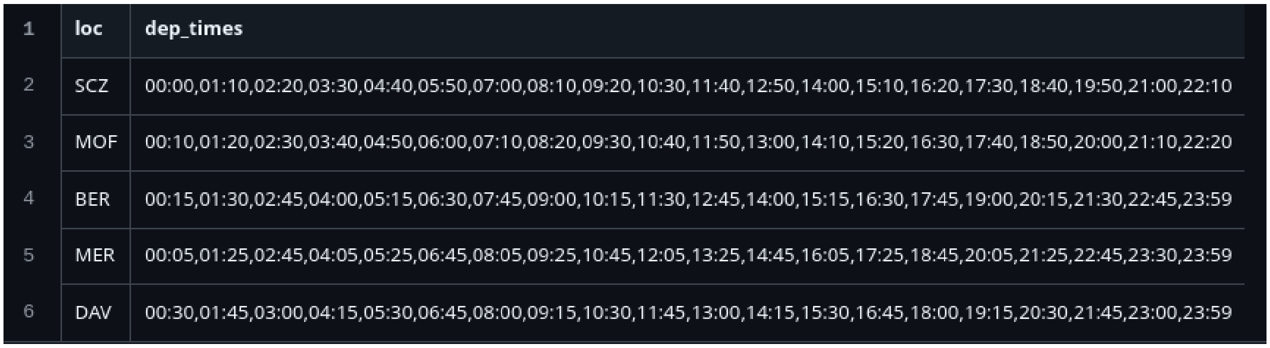


Figure 66: Example Passenger Demand Graph from Santa Cruz to Moffett Field



loc	dep_times
SCZ	00:00,01:10,02:20,03:30,04:40,05:50,07:00,08:10,09:20,10:30,11:40,12:50,14:00,15:10,16:20,17:30,18:40,19:50,21:00,22:10
MOF	00:10,01:20,02:30,03:40,04:50,06:00,07:10,08:20,09:30,10:40,11:50,13:00,14:10,15:20,16:30,17:40,18:50,20:00,21:10,22:20
BER	00:15,01:30,02:45,04:00,05:15,06:30,07:45,09:00,10:15,11:30,12:45,14:00,15:15,16:30,17:45,19:00,20:15,21:30,22:45,23:59
MER	00:05,01:25,02:45,04:05,05:25,06:45,08:05,09:25,10:45,12:05,13:25,14:45,16:05,17:25,18:45,20:05,21:25,22:45,23:30,23:59
DAV	00:30,01:45,03:00,04:15,05:30,06:45,08:00,09:15,10:30,11:45,13:00,14:15,15:30,16:45,18:00,19:15,20:30,21:45,23:00,23:59

Figure 67: Example Ground Transport Times Definition

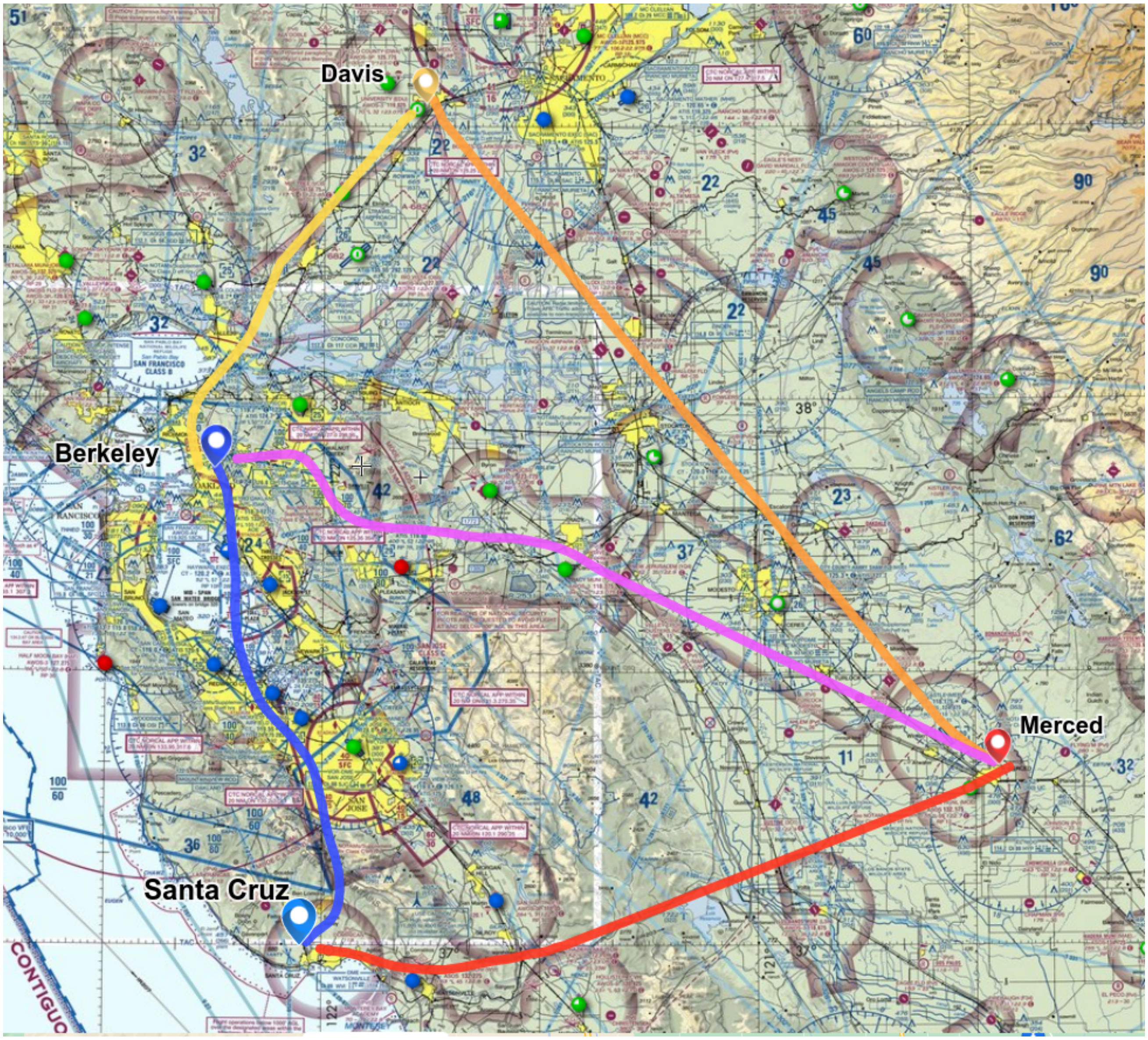


Figure 68: Previously Proposed Routes

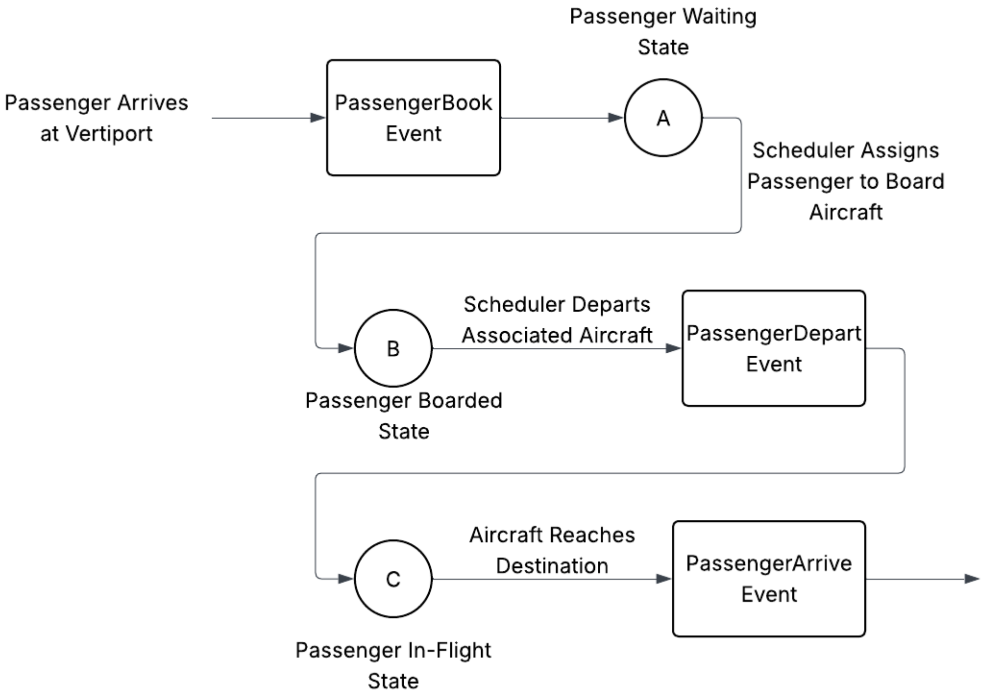

Passenger Waiting State
Passenger Arrives at Vertiport
PassengerBook Event
A
Scheduler Assigns Passenger to Board Aircraft
B
Scheduler Departs Associated Aircraft
PassengerDepart Event
Passenger Boarded State
C
Aircraft Reaches Destination
PassengerArrive Event
Passenger In-Flight State

Figure 69: Passenger State Diagram

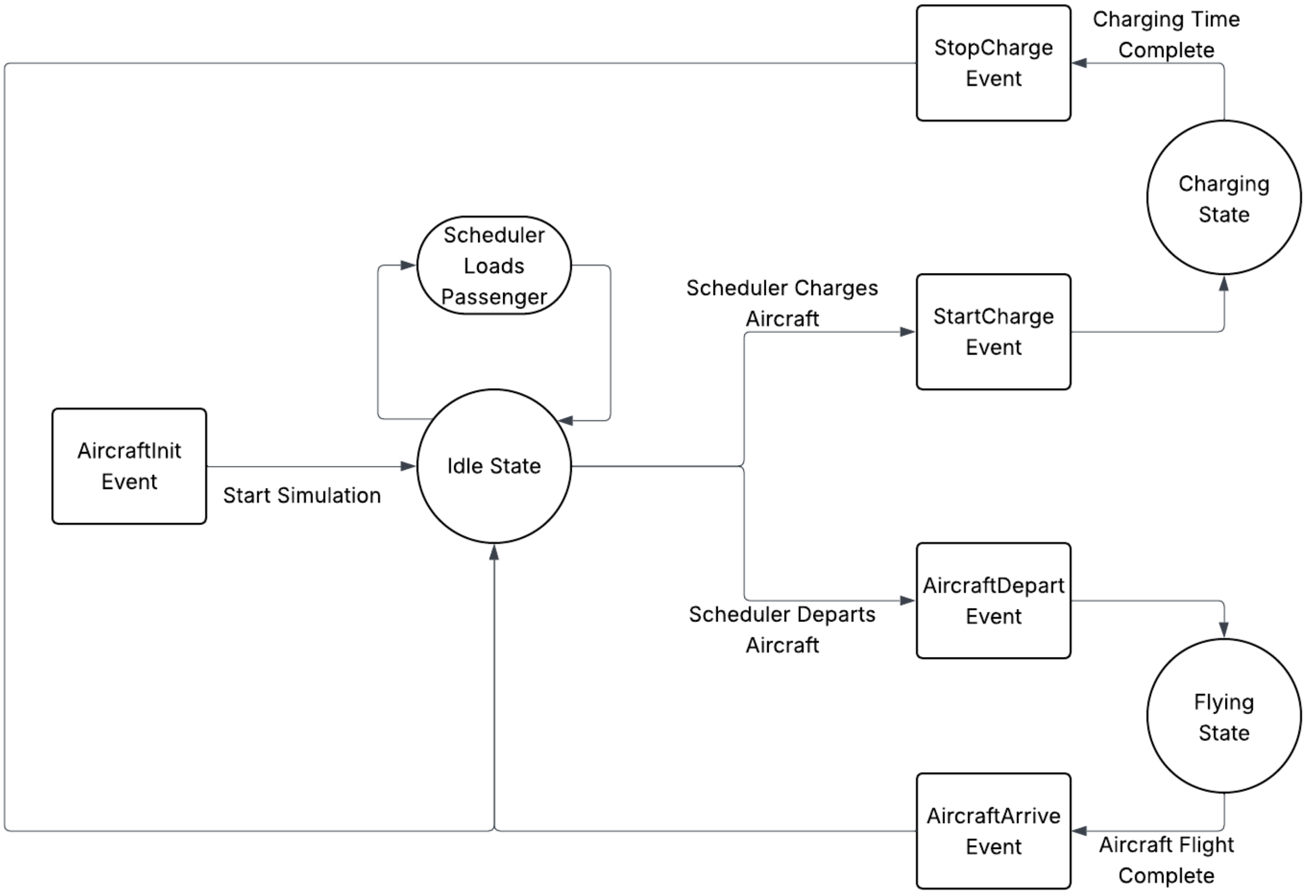


Figure 70: Aircraft State Diagram

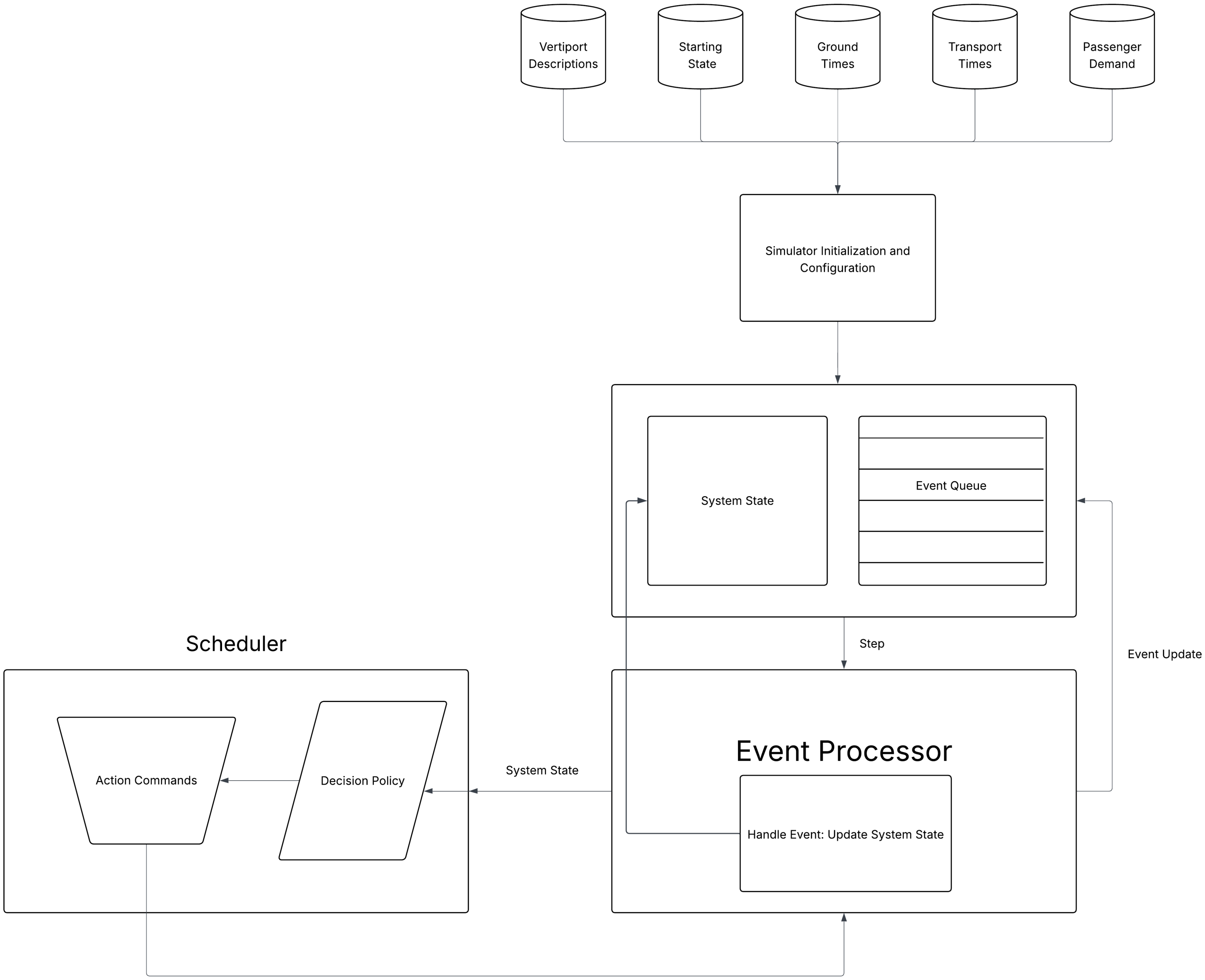
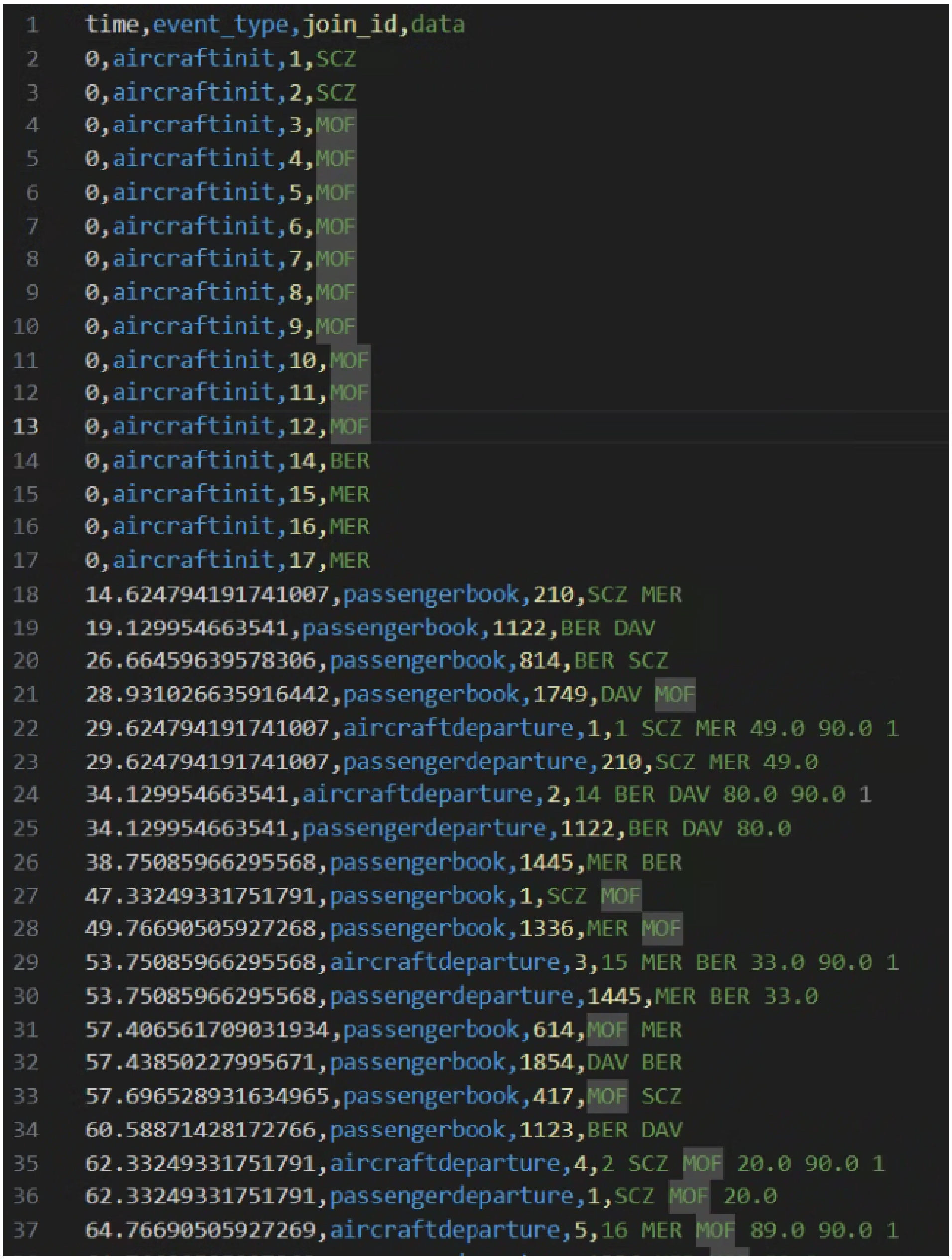


Figure 71: Scheduling Simulation Architecture



```
time,event_type,join_id,data
0,aircraftinit,1,SCZ
0,aircraftinit,2,SCZ
0,aircraftinit,3,MOF
0,aircraftinit,4,MOF
0,aircraftinit,5,MOF
0,aircraftinit,6,MOF
0,aircraftinit,7,MOF
0,aircraftinit,8,MOF
0,aircraftinit,9,MOF
0,aircraftinit,10,MOF
0,aircraftinit,11,MOF
0,aircraftinit,12,MOF
0,aircraftinit,14,BER
0,aircraftinit,15,MER
0,aircraftinit,16,MER
0,aircraftinit,17,MER
14.624794191741007,passengerbook,210,SCZ MER
19.129954663541,passengerbook,1122,BER DAV
26.66459639578306,passengerbook,814,BER SCZ
28.931026635916442,passengerbook,1749,DAV MOF
29.624794191741007,aircraftdeparture,1,1 SCZ MER 49.0 90.0 1
29.624794191741007,passengerdeparture,210,SCZ MER 49.0
34.129954663541,aircraftdeparture,2,14 BER DAV 80.0 90.0 1
34.129954663541,passengerdeparture,1122,BER DAV 80.0
38.75085966295568,passengerbook,1445,MER BER
47.33249331751791,passengerbook,1,SCZ MOF
49.76690505927268,passengerbook,1336,MER MOF
53.75085966295568,aircraftdeparture,3,15 MER BER 33.0 90.0 1
53.75085966295568,passengerdeparture,1445,MER BER 33.0
57.406561709031934,passengerbook,614,MOF MER
57.43850227995671,passengerbook,1854,DAV BER
57.696528931634965,passengerbook,417,MOF SCZ
60.58871428172766,passengerbook,1123,BER DAV
62.33249331751791,aircraftdeparture,4,2 SCZ MOF 20.0 90.0 1
62.33249331751791,passengerdeparture,1,SCZ MOF 20.0
64.76690505927269,aircraftdeparture,5,16 MER MOF 89.0 90.0 1
```

Figure 72: Event Log Example

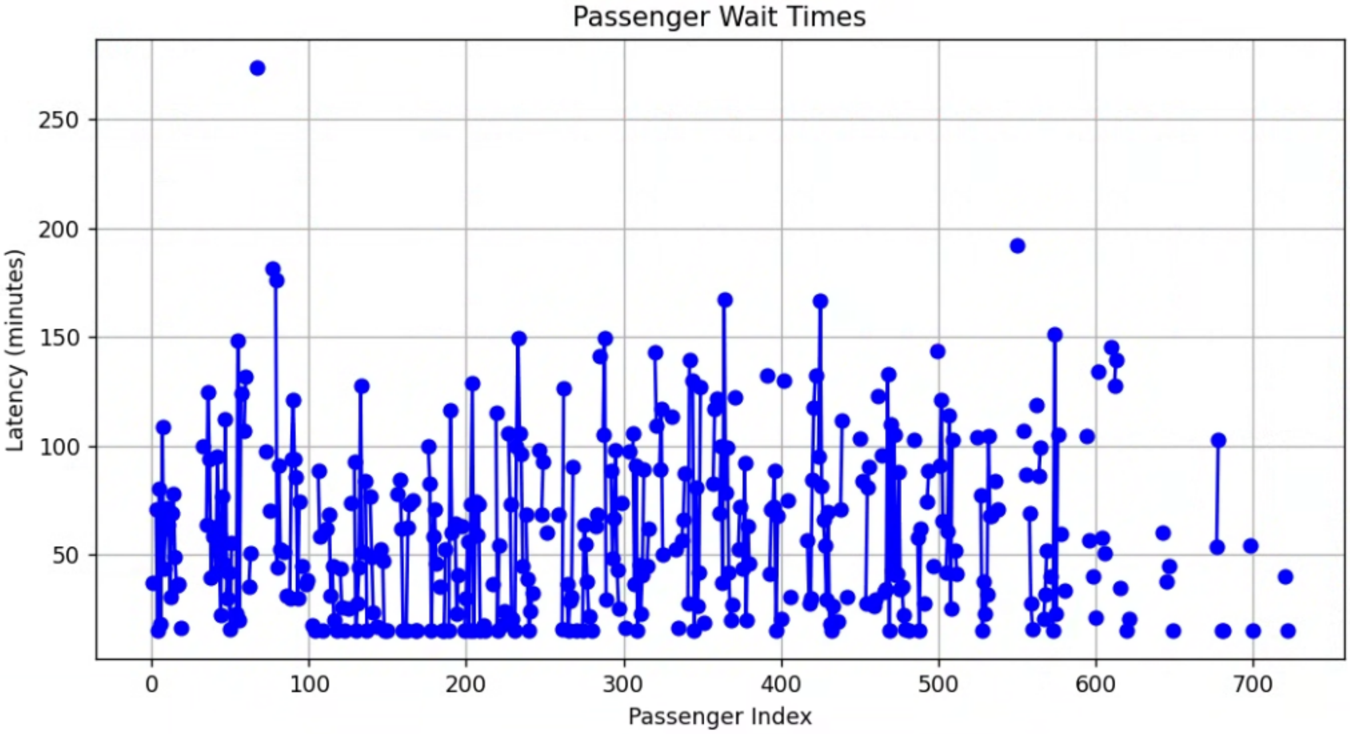


Figure 73: Example Wait Time Index

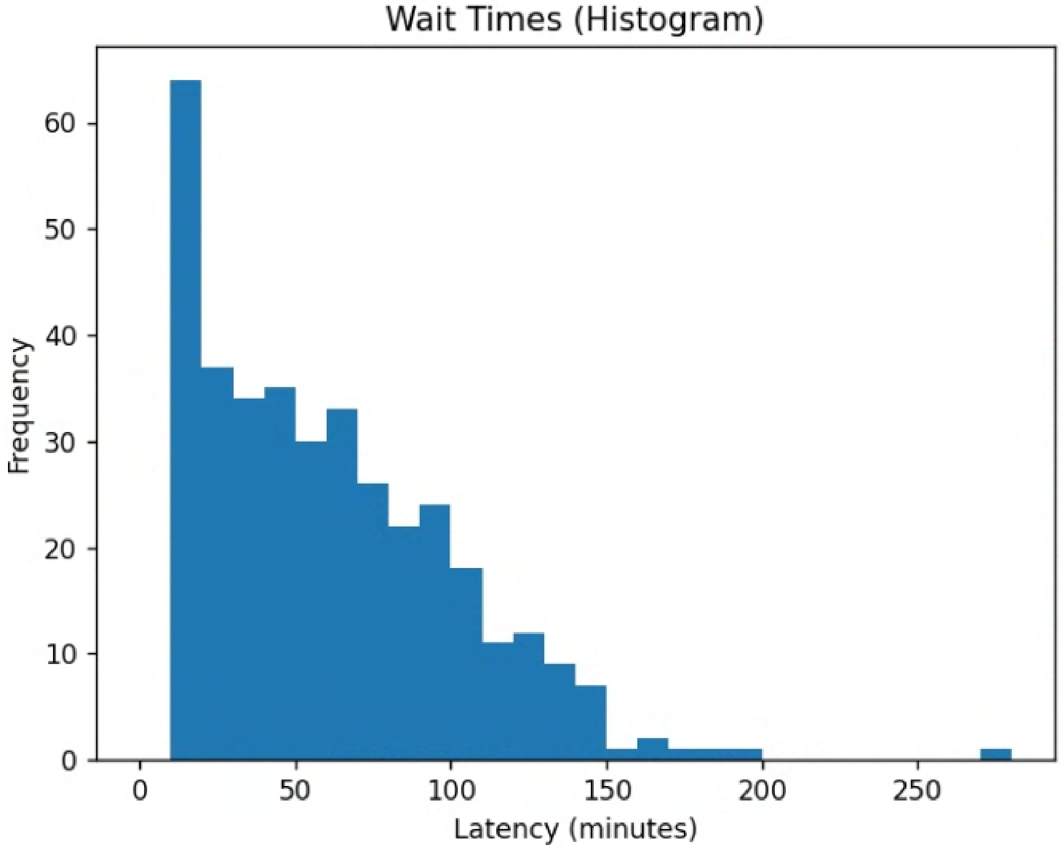


Figure 74: Example Wait Time Histogram

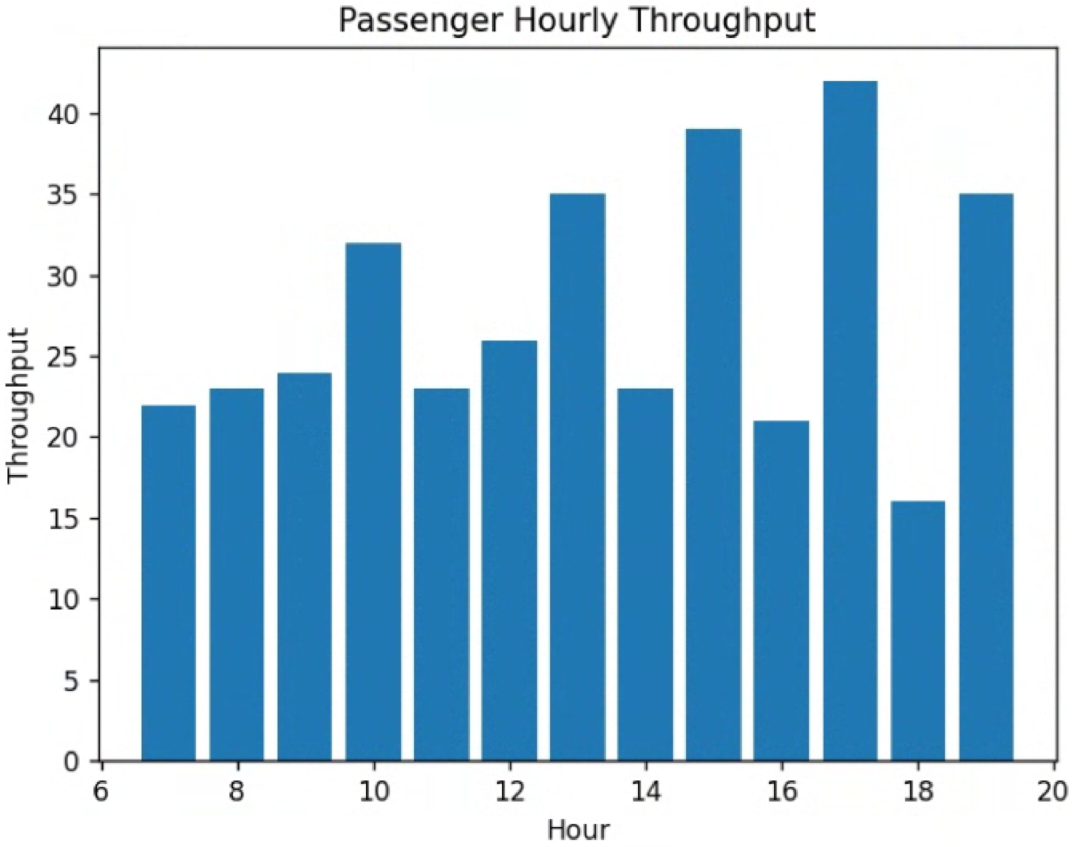


Figure 75: Example Throughput Histogram

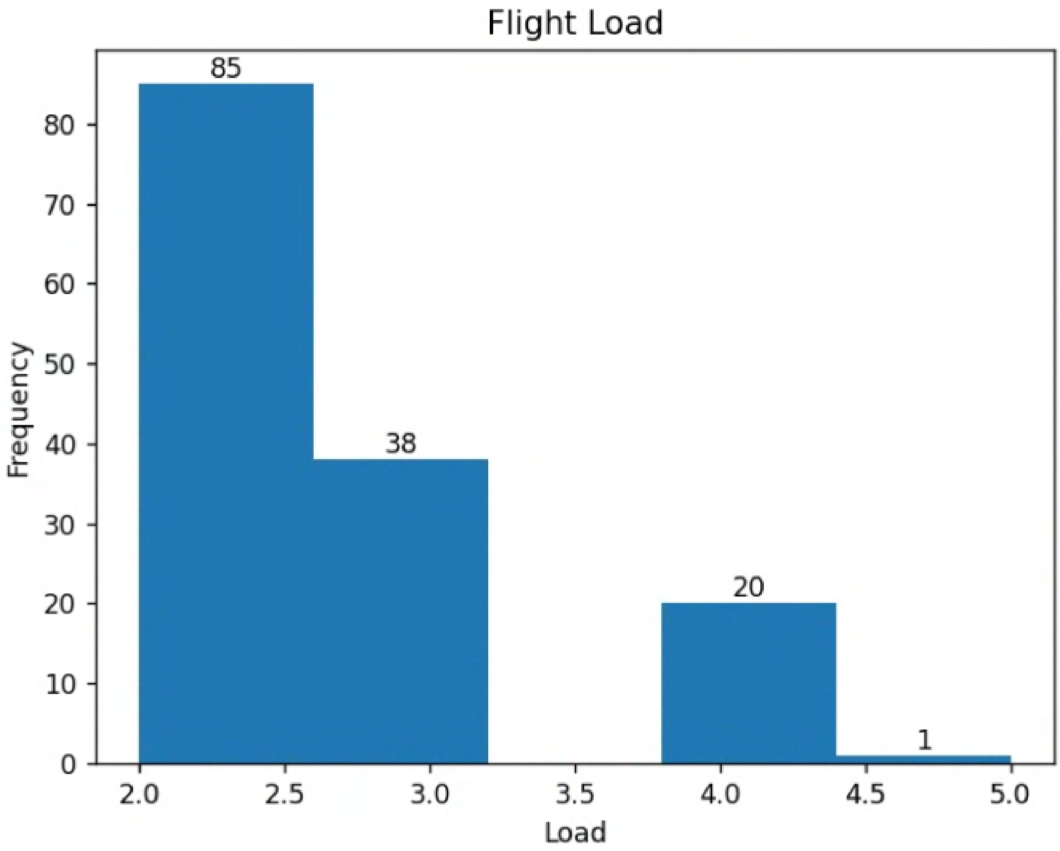
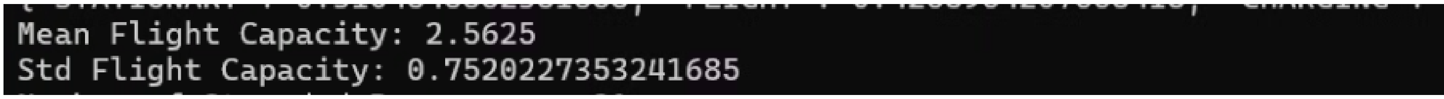
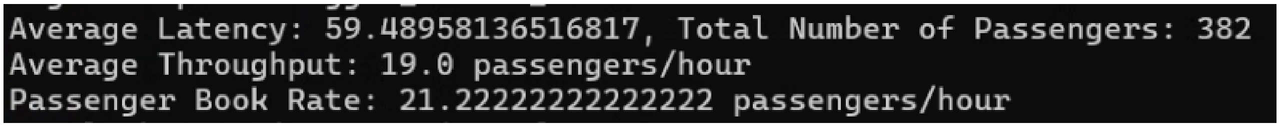


Figure 76: Example Flight Load Histogram



```
Mean Flight Capacity: 2.5625
Std Flight Capacity: 0.7520227353241685
```

Figure 77: Example Flight Load Data



```
Average Latency: 59.48958136516817, Total Number of Passengers: 382
Average Throughput: 19.0 passengers/hour
Passenger Book Rate: 21.22222222222222 passengers/hour
```

Figure 78: Average Wait Time, Throughput, and Book Rate statistics

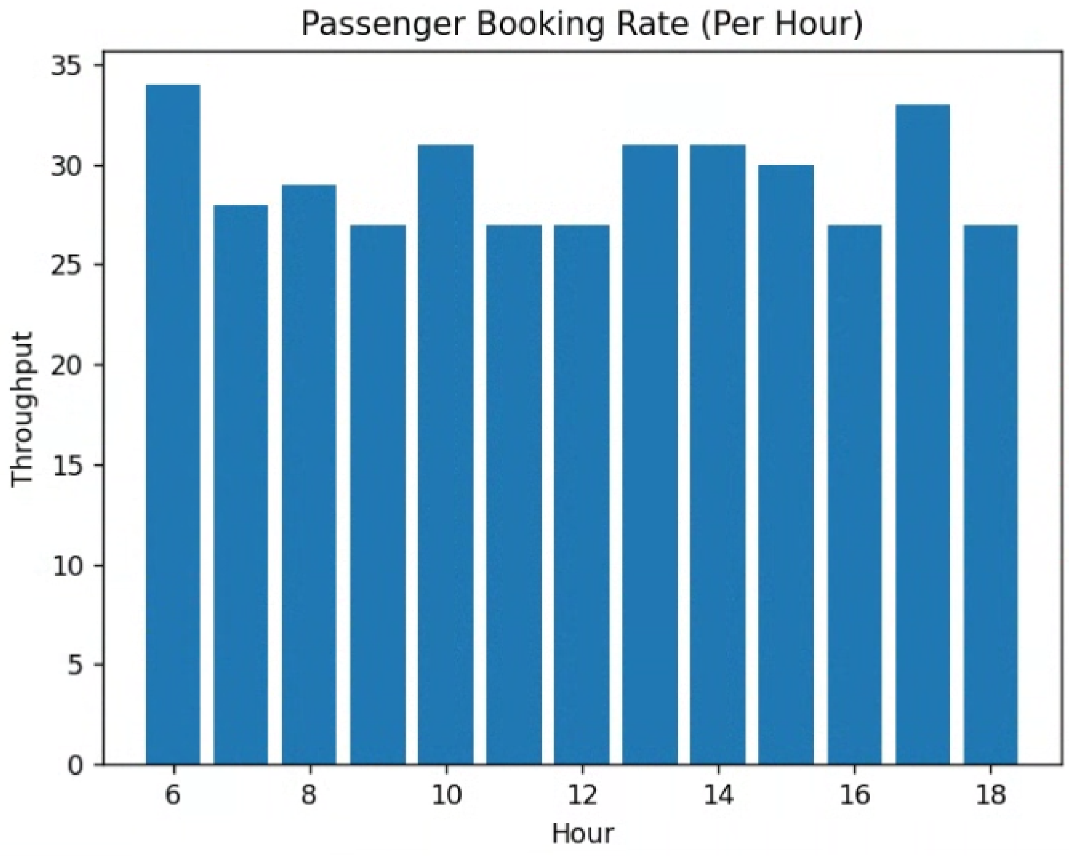
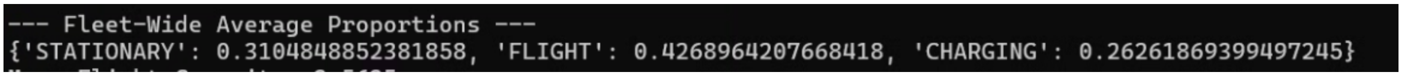


Figure 79: Example Passenger Book Rate Histogram (Derived from Passenger Demand)



```
--- Fleet-Wide Average Proportions ---
{'STATIONARY': 0.3104848852381858, 'FLIGHT': 0.4268964207668418, 'CHARGING': 0.26261869399497245}
```

Figure 80: Example Cycle Data (In Average Percentage per Cycle)

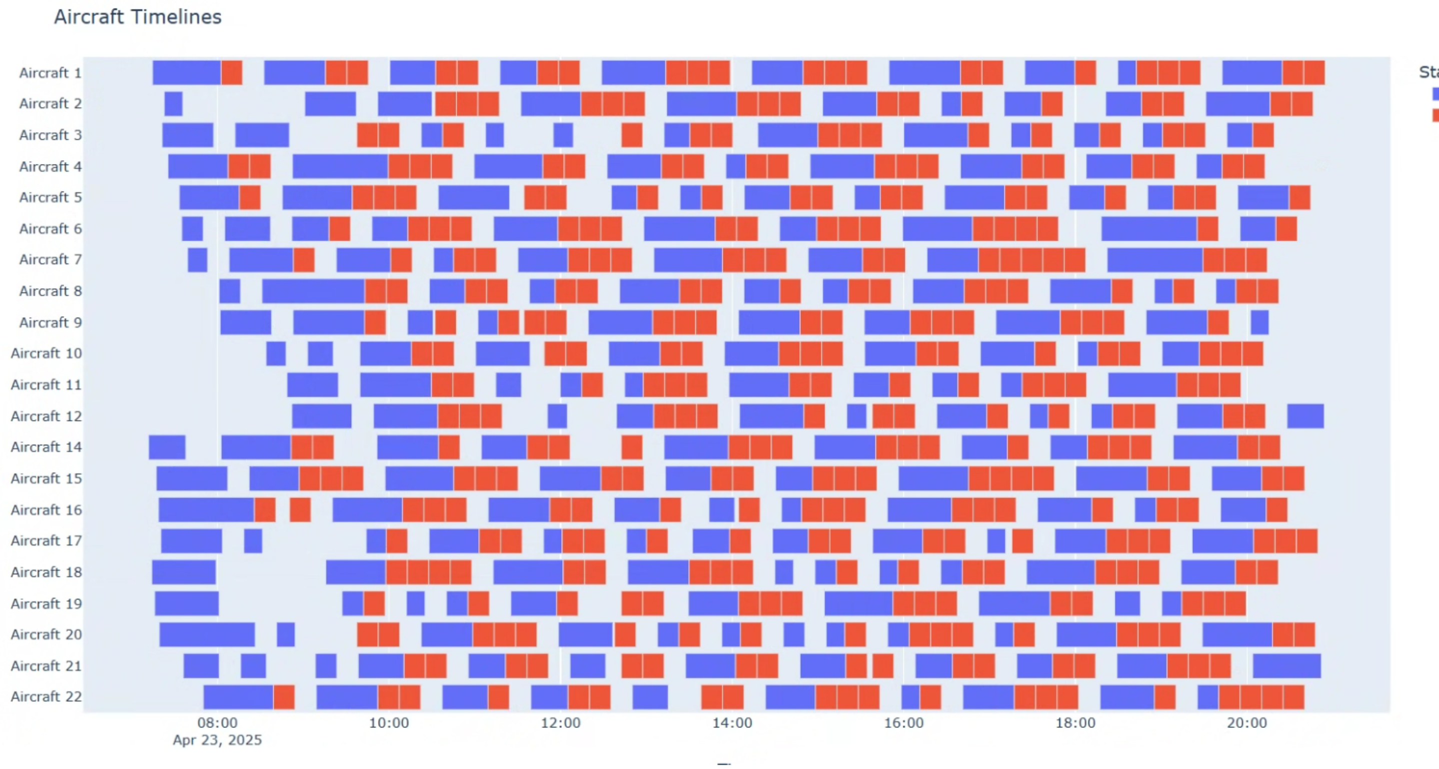
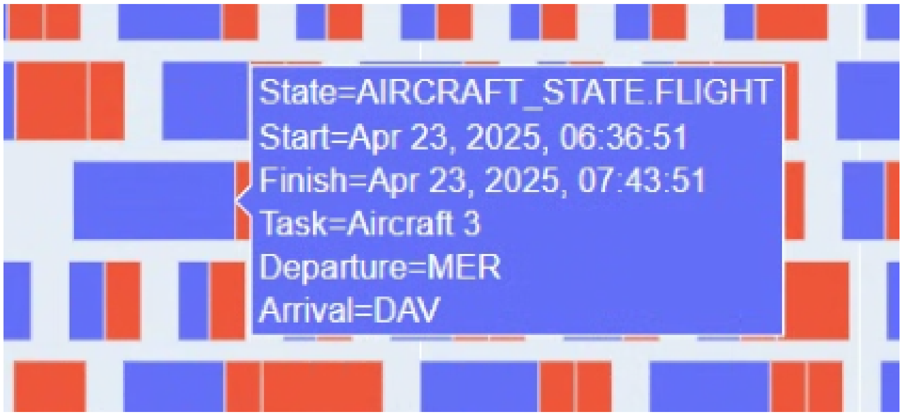
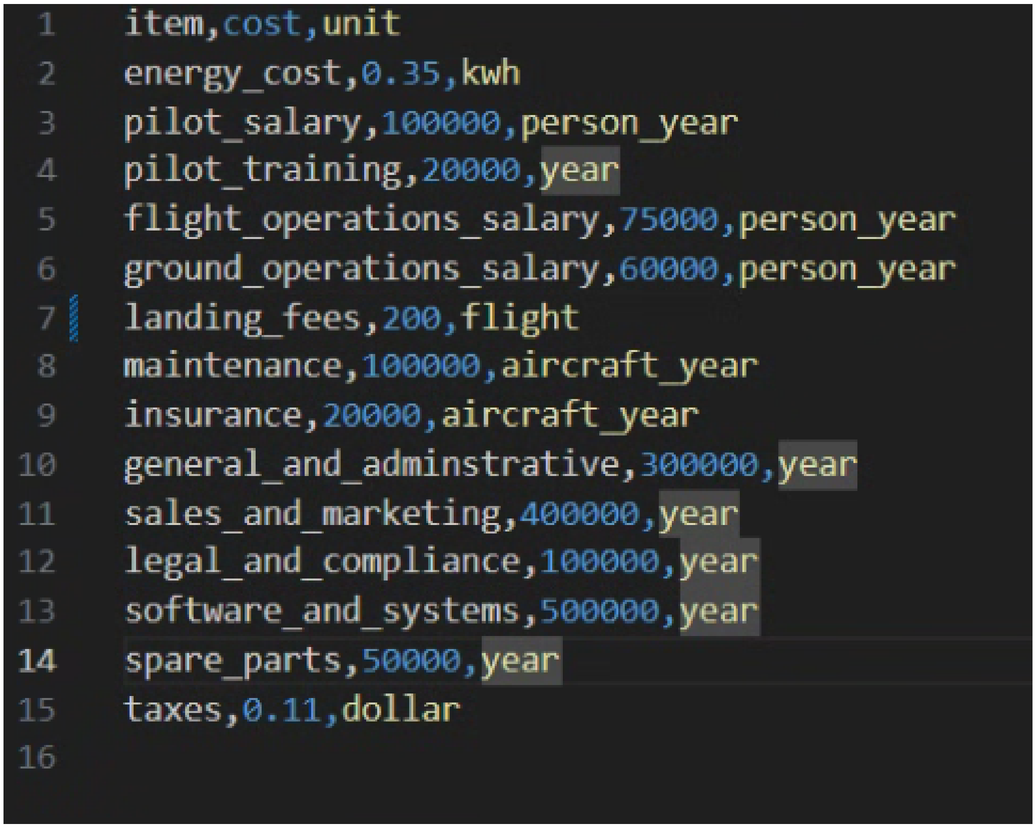


Figure 81: Aircraft Scheduling Timeline: Red - Charging, Blue - Flight, Blank - Idle



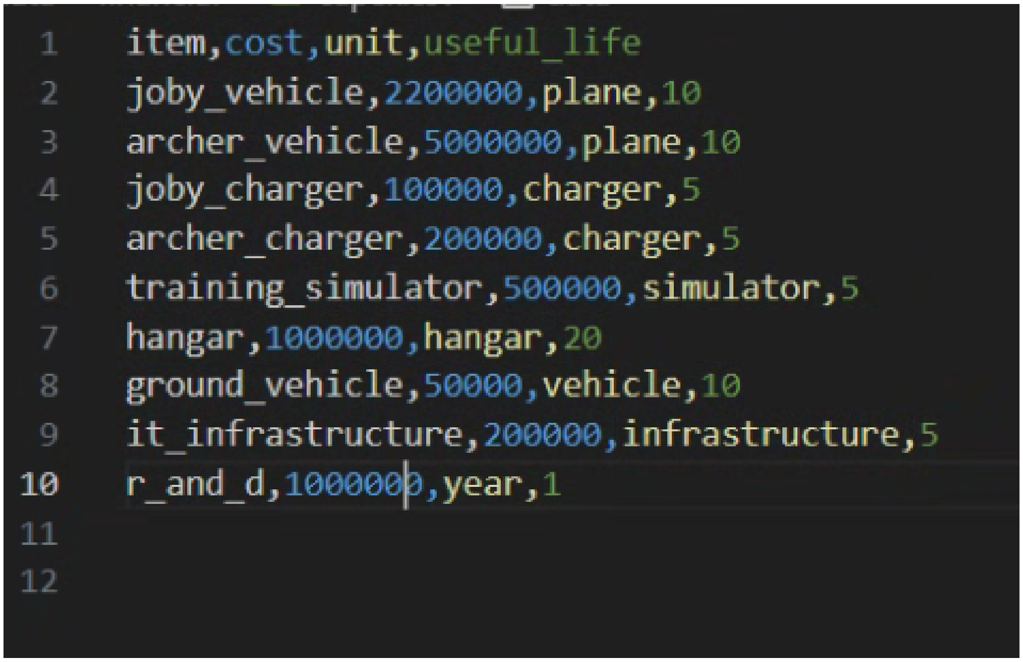
State=AIRCRAFT_STATE.FLIGHT
Start=Apr 23, 2025, 06:36:51
Finish=Apr 23, 2025, 07:43:51
Task=Aircraft 3
Departure=MER
Arrival=DAV

Figure 82: Hovering over Aircraft Flight on Timeline Chart



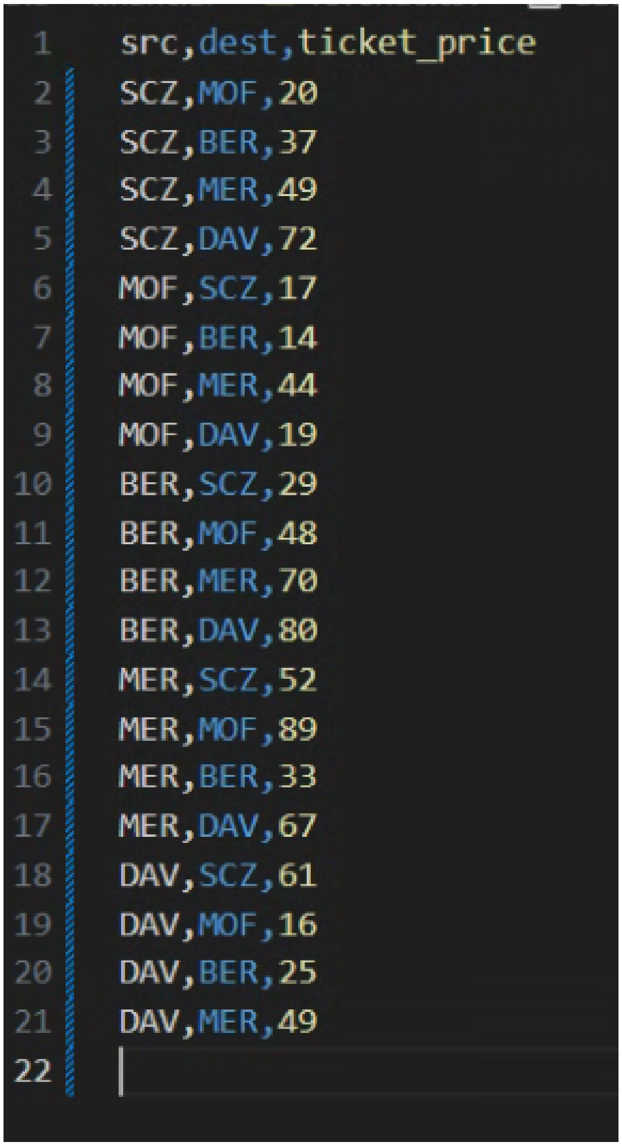
```
item,cost,unit
energy_cost,0.35,kwh
pilot_salary,100000,person_year
pilot_training,20000,year
flight_operations_salary,75000,person_year
ground_operations_salary,60000,person_year
landing_fees,200,flight
maintenance,100000,aircraft_year
insurance,20000,aircraft_year
general_and_adminstrative,300000,year
sales_and_marketing,400000,year
legal_and_compliance,100000,year
software_and_systems,500000,year
spare_parts,50000,year
taxes,0.11,dollar

```

Figure 83: Example OPEX information



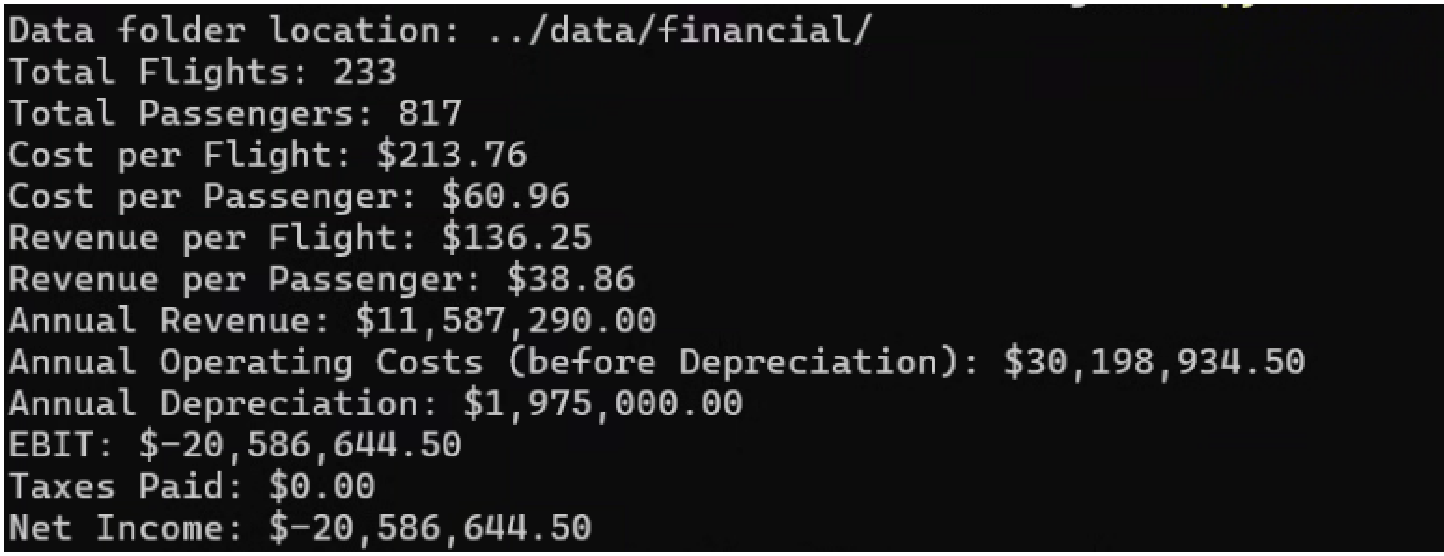
```
item,cost,unit,useful_life
joby_vehicle,2200000,plane,10
archer_vehicle,5000000,plane,10
joby_charger,100000,charger,5
archer_charger,200000,charger,5
training_simulator,500000,simulator,5
hangar,1000000,hangar,20
ground_vehicle,50000,vehicle,10
it_infrastructure,200000,infrastructure,5
r_and_d,1000000,year,1
```

Figure 84: Example CAPEX information



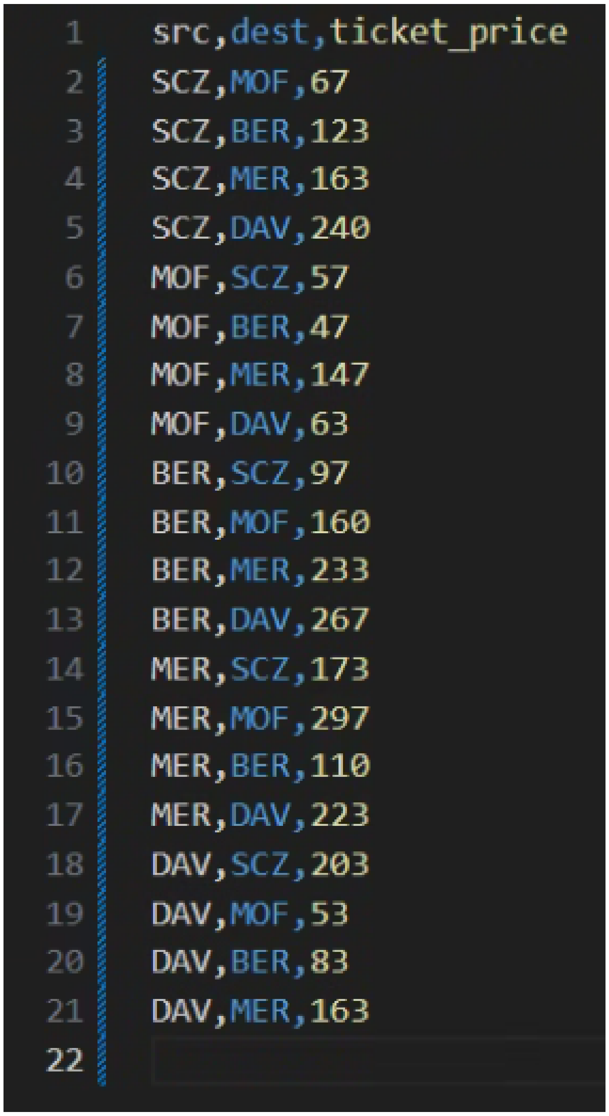
```
src,dest,ticket_price
SCZ,MOF,20
SCZ,BER,37
SCZ,MER,49
SCZ,DAV,72
MOF,SCZ,17
MOF,BER,14
MOF,MER,44
MOF,DAV,19
BER,SCZ,29
BER,MOF,48
BER,MER,70
BER,DAV,80
MER,SCZ,52
MER,MOF,89
MER,BER,33
MER,DAV,67
DAV,SCZ,61
DAV,MOF,16
DAV,BER,25
DAV,MER,49
```

Figure 85: Cheap Ticket Prices



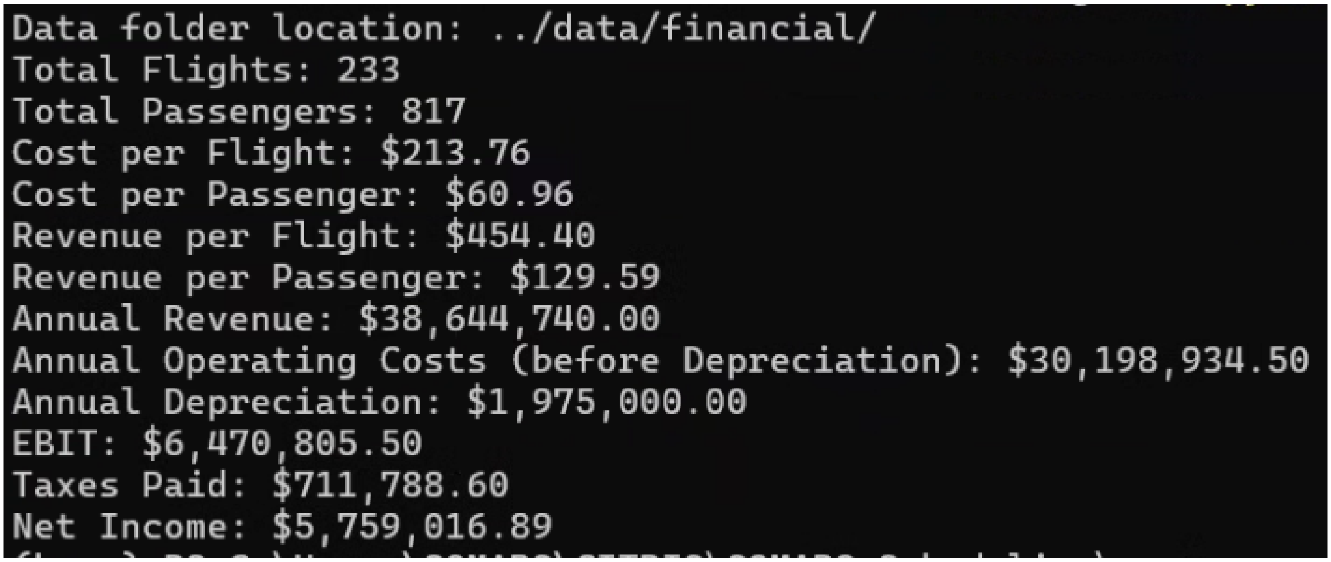
```
Data folder location: ../data/financial/
Total Flights: 233
Total Passengers: 817
Cost per Flight: $213.76
Cost per Passenger: $60.96
Revenue per Flight: $136.25
Revenue per Passenger: $38.86
Annual Revenue: $11,587,290.00
Annual Operating Costs (before Depreciation): $30,198,934.50
Annual Depreciation: $1,975,000.00
EBIT: $-20,586,644.50
Taxes Paid: $0.00
Net Income: $-20,586,644.50
```

Figure 86: Financial Outcome with Cheap Ticket Prices



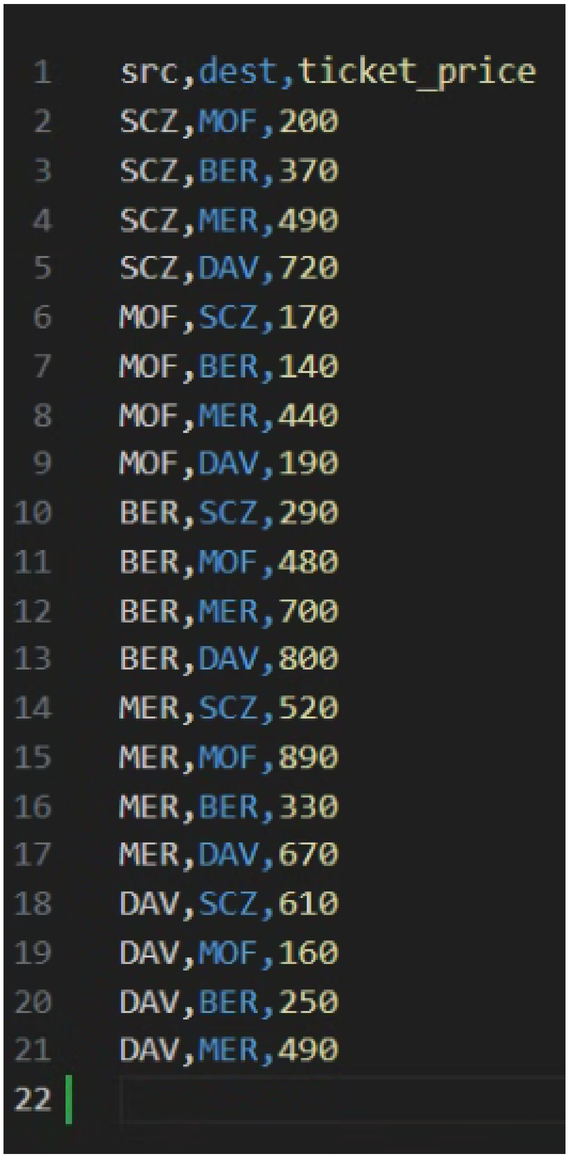
```
src,dest,ticket_price
SCZ,MOF,67
SCZ,BER,123
SCZ,MER,163
SCZ,DAV,240
MOF,SCZ,57
MOF,BER,47
MOF,MER,147
MOF,DAV,63
BER,SCZ,97
BER,MOF,160
BER,MER,233
BER,DAV,267
MER,SCZ,173
MER,MOF,297
MER,BER,110
MER,DAV,223
DAV,SCZ,203
DAV,MOF,53
DAV,BER,83
DAV,MER,163

```

Figure 87: Moderate Ticket Prices



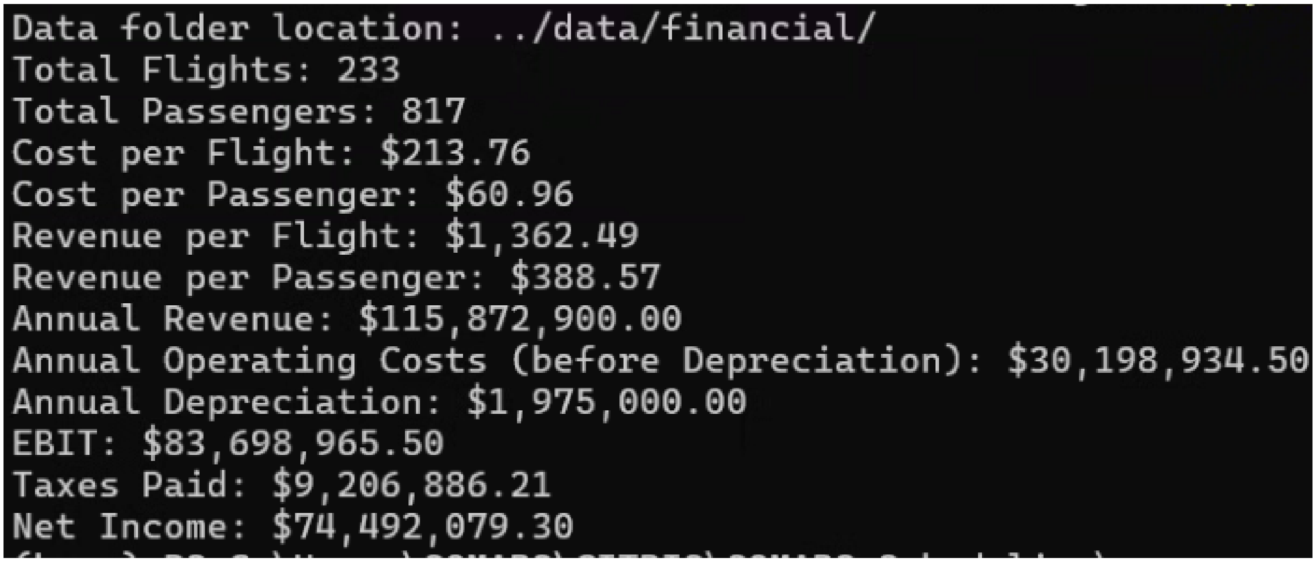
```
Data folder location: ../data/financial/
Total Flights: 233
Total Passengers: 817
Cost per Flight: $213.76
Cost per Passenger: $60.96
Revenue per Flight: $454.40
Revenue per Passenger: $129.59
Annual Revenue: $38,644,740.00
Annual Operating Costs (before Depreciation): $30,198,934.50
Annual Depreciation: $1,975,000.00
EBIT: $6,470,805.50
Taxes Paid: $711,788.60
Net Income: $5,759,016.89
```

Figure 88: Financial Outcome with Moderate Ticket Prices



```
src,dest,ticket_price
SCZ,MOF,200
SCZ,BER,370
SCZ,MER,490
SCZ,DAV,720
MOF,SCZ,170
MOF,BER,140
MOF,MER,440
MOF,DAV,190
BER,SCZ,290
BER,MOF,480
BER,MER,700
BER,DAV,800
MER,SCZ,520
MER,MOF,890
MER,BER,330
MER,DAV,670
DAV,SCZ,610
DAV,MOF,160
DAV,BER,250
DAV,MER,490
```

Figure 89: Expensive Ticket Prices



```
Data folder location: ../data/financial/
Total Flights: 233
Total Passengers: 817
Cost per Flight: $213.76
Cost per Passenger: $60.96
Revenue per Flight: $1,362.49
Revenue per Passenger: $388.57
Annual Revenue: $115,872,900.00
Annual Operating Costs (before Depreciation): $30,198,934.50
Annual Depreciation: $1,975,000.00
EBIT: $83,698,965.50
Taxes Paid: $9,206,886.21
Net Income: $74,492,079.30
```

Figure 90: Financial Outcome with Expensive Ticket Prices

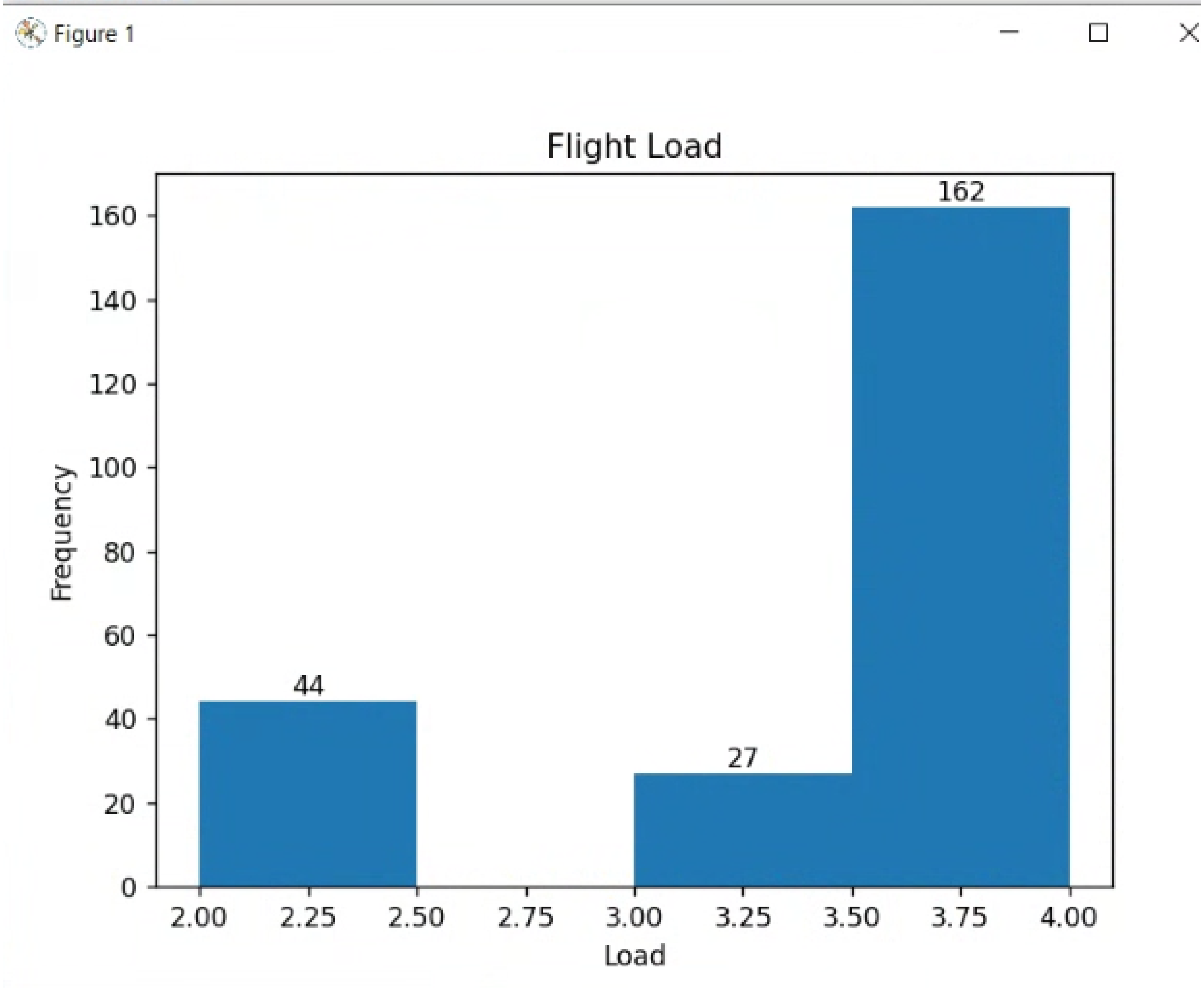


Figure 91: Flight Load of Financial Model Example

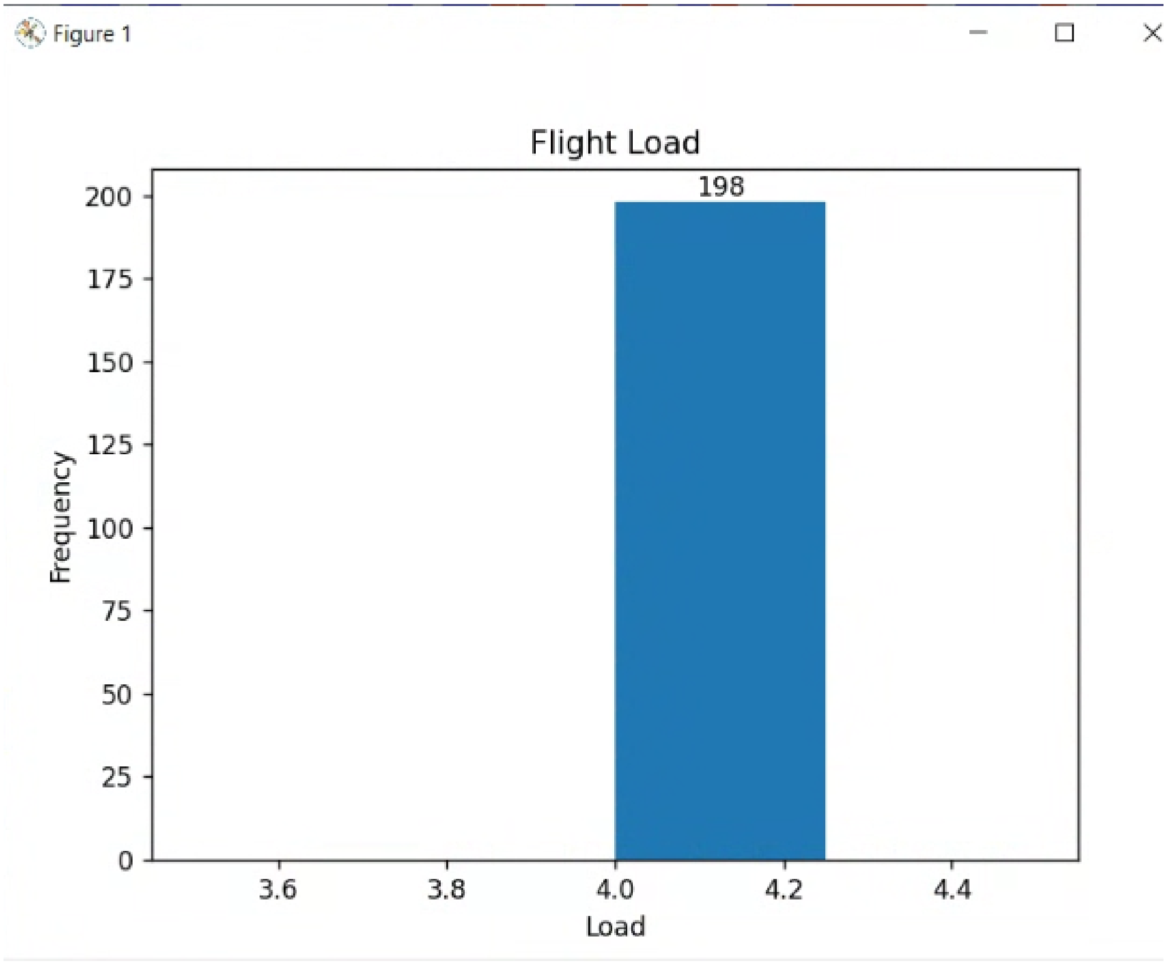
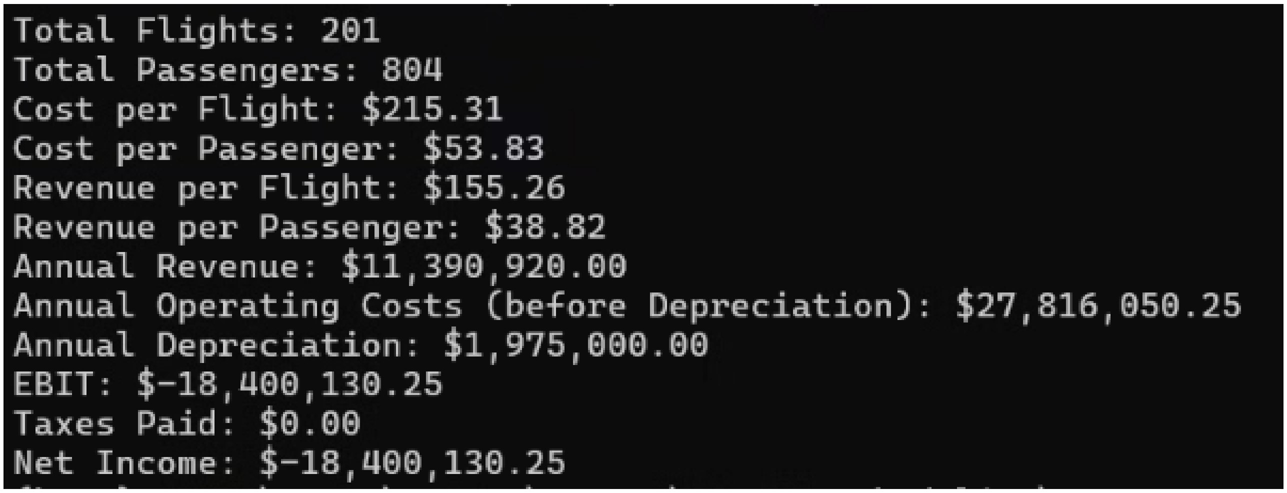


Figure 92: Restricted Flight Load to Strictly 4 Passengers



```
Total Flights: 201
Total Passengers: 804
Cost per Flight: $215.31
Cost per Passenger: $53.83
Revenue per Flight: $155.26
Revenue per Passenger: $38.82
Annual Revenue: $11,390,920.00
Annual Operating Costs (before Depreciation): $27,816,050.25
Annual Depreciation: $1,975,000.00
EBIT: $-18,400,130.25
Taxes Paid: $0.00
Net Income: $-18,400,130.25
```

Figure 93: Financial Outcome with Cheap Ticket Prices
With Load at 4 Passengers

8 Citations

References

[1] Joby Aviation 2024 10K Report

[2] Calculating Best Glide Quantities - Matlab

[3] Advanced Air Mobility Implementation Plan

[4] FINANCIAL FEASIBILITY STUDY OF eVTOL AIRCRAFT

[5] RL-based Scheduling of an AAM Traffic Network

[6] eVTOL Arrival Sequencing and Scheduling for On-Demand Urban Air Mobility

[7] Evaluating eVTOL Network Performance and Fleet Dynamics through Simulation-Based Analysis

[8] Urban Air Mobility Airspace Integration Concepts and Considerations

[9] A Traffic Management Framework for On-Demand Urban Air Mobility Systems

[10] FAA - Time-Based Flow Management

[11] Terminal area control rules and eVTOL adaptive scheduling model for multi-vertiport system in urban air Mobility

[12] UAM Instrument Flight Procedure Design and Evaluation in the Joby Flight Simulator

[13] FAA Order JO 7110.65BB

[14] Aeronautical Information Manual Chapter 3 Section 2

[15] Operational Considerations regarding On-Demand Air Mobility: A Literature Review and Research Challenges

[16] Best Practices for Predicting Acoustics of a Single Rotor Using the NASA RVLT Conceptual Design Toolchain

[17] Microsoft Flight Simulator Flight Plan Definitions

[18] Microsoft Flight Simulator Flight Plan Tool

[19] Microsoft Flight Simulator Scenery Editor

[20] Microsoft Flight Simulator Simconnect SDK

[21] Microsoft Flight Simulator Departure and Approach Definitions and Properties

[22] Scenic Programming Language

[23] Centralized Github Repository for Our Code and Experiments

[24] How MathWorks is Enabling Supernal's Advanced Air Mobility Development with Integrated Simulation Systems

9 Index

AAM - Advanced Air Mobility
eVTOL - Electric Vertical Takeoff and Landing
FAA - Federal Aviation Administration
IFR - Instrument Flight Rules
NASA - National Aeronautics and Space Administration
OAK - Oakland International Airport
RTCA - Radio Technical Commission for Aeronautics
SFO - San Francisco International Airport
VFR - Visual Flight Rules
STAR - Standard Terminal Arrival Route
SID - Standard Instrument Departure