Clouding Seeding UAV Final Report

MAE 154a: Final Report on Clouding Seeding UAV

By: Carter Hallum, Mohammed Elshal, Walyon Tang

Instructor: Damien Toohey

March 21, 2025

Table of Contents

Introduction

In California lack of rain has been a recurring and devastating problem across california. Recent droughts, forest fires, and other water failures have been felt throughout the state. To address this issue, a new method known as cloud seeding is being studied. The main method of chemical cloud seeding involves flying a plane through clouds that meet the criteria and releasing chemicals that will help induce precipitation. A recent trend of cloud seeding has been adopted in many countries and studies are being conducted on its effectiveness. Many countries and places have started to adopt cloud seeding, which is a method of inducing clouds to rain through chemicals as a way to combat this. Cloud seeding has been around since the 1940s and studies show that clouding seeding increases precipitation and clouds after cloud seeding. Our Goal is to deliver more autonomy to farmers and other communities in the central valley.

Mission Statement

The Cloud Seeding UAV will be able to launch from short runways and fly on a designated path through predicted cloud formations moving over the area. During the flight sensors on UAV will measure particle sizes, moisture and other conditions. If the cloud meets ideal cloud seeding conditions the UAV will release chemical flares that contain the necessary clouding seeding chemicals. After completing its mission the UAV will land back on the runway it took off from.

Mission Requirements

Following the mission statement above we designed some mission requirements out UAV must meet. These Requirements were set based on other manned cloud seeding planes that had similar size, weight, and payload capacity. The requirements are the following:

- Able to operate at altitude ceiling to 20,000 ft
- Range of 400 nmi
- Carry a 200lbs of chemical payload
- Able to take off and land on short runways/roads (3000ft.)
- Ability to fly through gusts up to 30 knots
- Onboard Navigational hardware and software

- Onboard Weather hardware and software
- Must be able to fly at low temperatures

Payload

At the heart of this mission is the chemical payload, which plays a critical role in inducing precipitation. Typically, chemicals such as silver iodide or potassium iodide are used because they act as effective nucleating agents, promoting the formation of ice crystals within clouds. There are two main types of flares used, each tailored to specific cloud conditions:

- Glaciogenic Flares: These are used in cold clouds. They stimulate ice formation, enabling the growth of ice crystals into precipitation-sized particles.
- Hygroscopic Flares: These are used in warm clouds. They attract moisture, encouraging the coalescence of water droplets into larger droplets that eventually lead to precipitation.

Together, these chemical strategies are key to modifying cloud microphysics and enhancing precipitation formation during the mission.

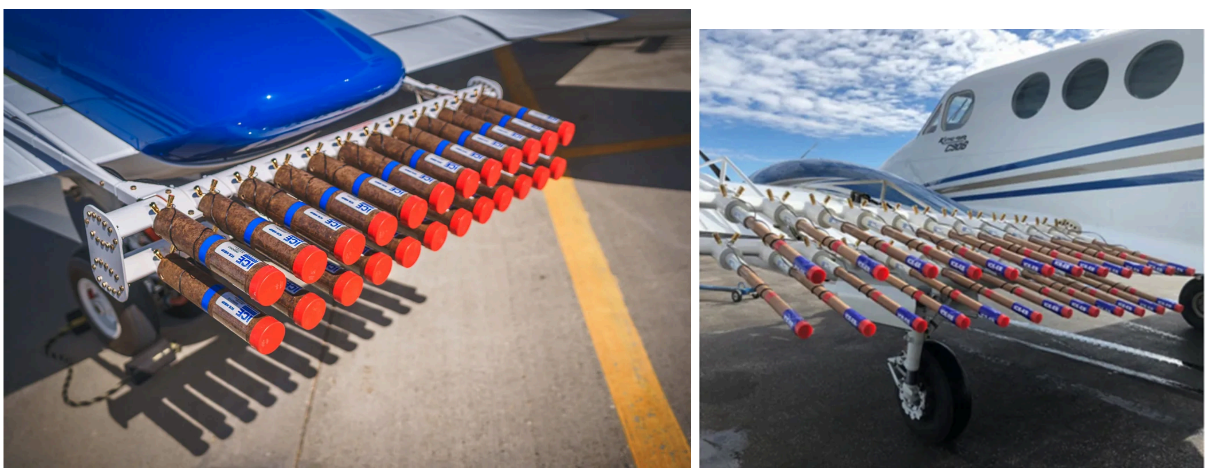


Figure 2.1 Shows types of flares used in cloud seeding.

Additional Components

- HGuide n580 Inertial-GNSS Navigator: Provides onboard navigation and flight control by integrating inertial sensors with GNSS data for accurate positioning and attitude determination.

Figure 2.2 Shows HGuide n580 Inertial-GNSS Navigator.

- Cloud Particle Sensor: A spectrometer that measures cloud droplets between 2 µm and 50 µm, handling concentrations up to 2,000 particles/cm³ to assess cloud microstructure.

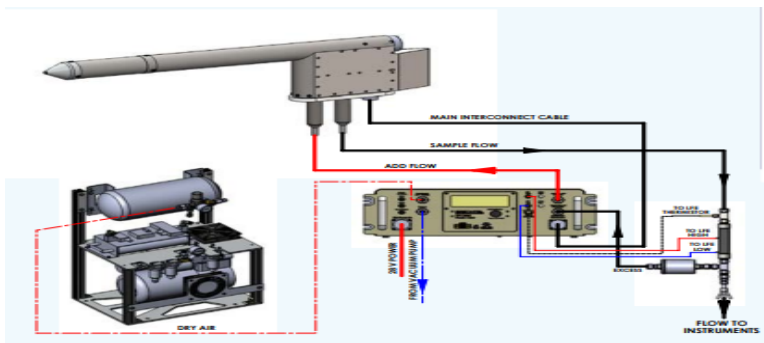


Figure 2.3 Shows cloud flare system.

- Cloud Condensation Nuclei Counter: Quantifies the number of particles that can act as condensation nuclei, essential for understanding the potential for cloud droplet formation.

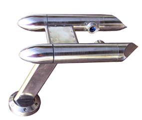


Figure 2.4 Shows CDP-2 Cloud Droplet Probe.

- Ground Based Weather Radar: Tracks pre-flight weather and cloud locations to ensure the mission is launched under favorable conditions and to guide flight planning.
- Electrically Heated Surface: Provides de-icing and anti-icing for critical surfaces to maintain proper aerodynamic performance in cold weather conditions.
- Chemical Flare Payload system: Dispenses nucleating agents (e.g., silver iodide or potassium iodide) to trigger ice crystal formation in clouds, enhancing precipitation.

Final Plane Design

Layout: Our airplane features a conventional fixed-wing design with a nose-mounted engine for streamlined thrust and efficiency. Its fuselage serves as a central body that houses the payload, and fuel, while a robust tri-fixed landing gear ensures reliable ground handling. Chemical flares are mounted beneath the wing for effective deployment during cloud seeding operations.

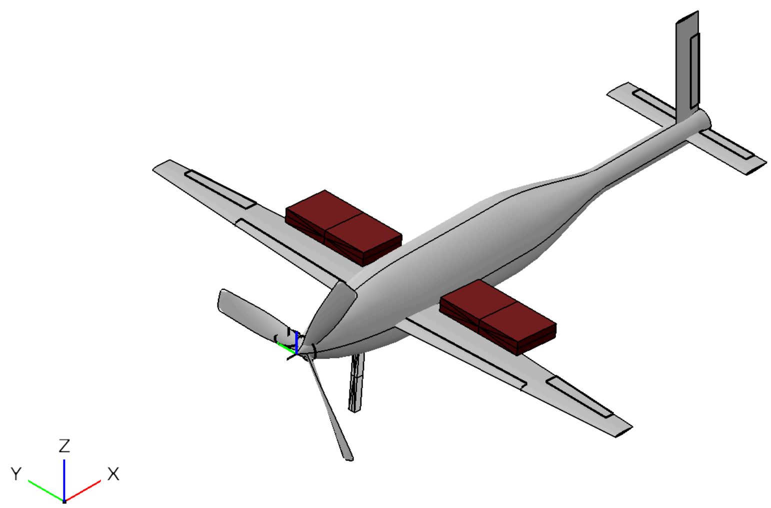


Figure 3.1 Final CAD of UAV

Longitudinal Component Locations: The X-locations of key components (wing, engine, fuel, payload, sensors, landing gear, tail) are plotted along with the center of gravity (xcg) and neutral point (xnp) for aircraft balance analysis.

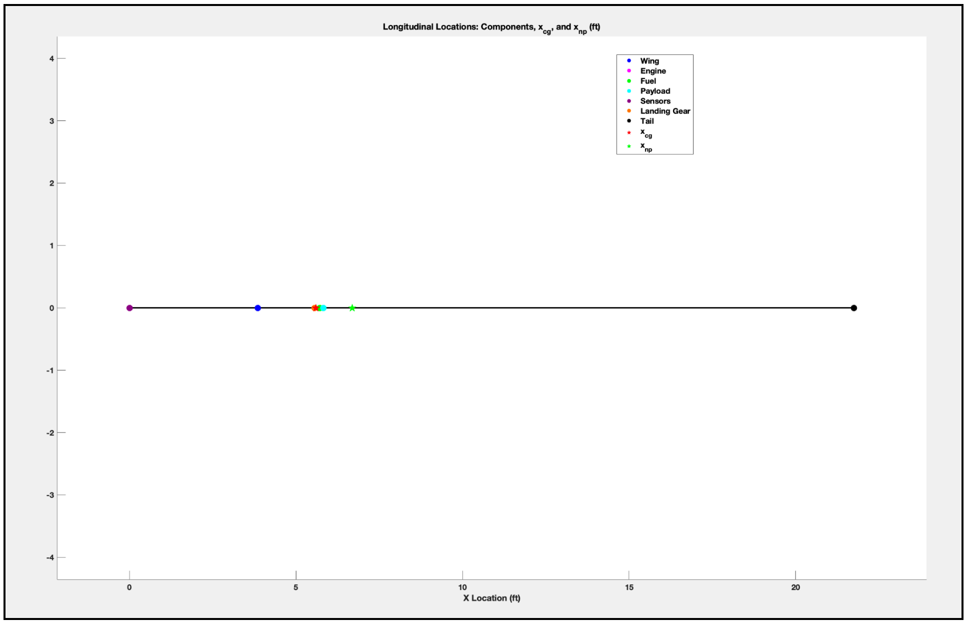


Figure 3.2 Shows location of each component along with CG and NP location.

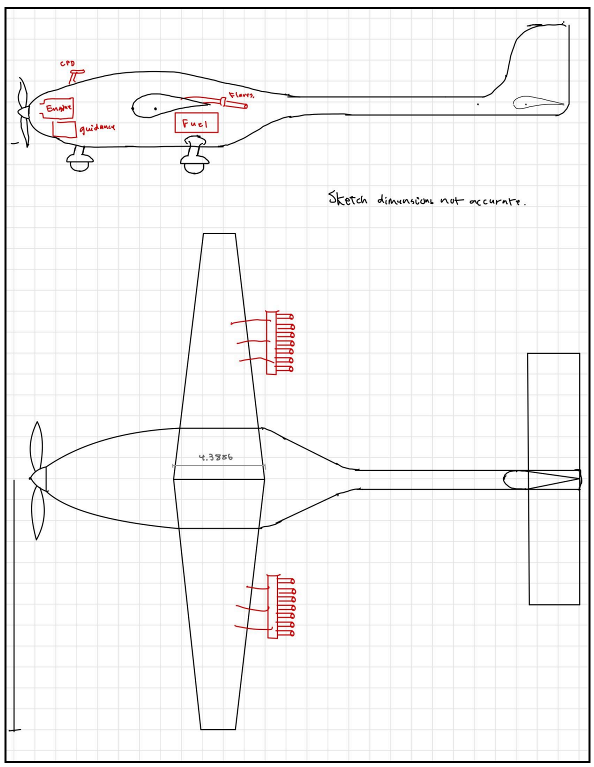


Figure 3.3 Sketch of components locations

Specs

Spec Name	**Value**	Surface Area of Horizontal Tail	7.84 ft^2
Total Length	21.76 ft	Horizontal Tail Aspect Ratio	6.59
Body Length	15.6 ft	Horizontal Tail Span	7.18 ft
Body Width	3.16 ft	Horizontal Tail Chord	2.16 ft
Body Height	2.86	Horizontal Tail Incidence Angle	1.82 deg
Length of Extension	6.14 ft	Horizontal Tail Airfoil	NACA0012
Diameter of Extension	1 ft	Surface Area of Vertical Tail	6.04 ft^2
Wing Surface Area	86.14 ft^2	Vertical Tail Tail Aspect Ratio	4.61
Wing Aspect Ratio	6.867	Vertical Tail Span	5.28 ft
Wing Taper Ratio	0.367	Vertical Tail Chord	2.13 ft
Wing Span	24.32 ft	Vertical Tail Airfoil	NACA0012
Root Chord	6.29 ft	MTOW	674.8 lb
Airfoil	NACA0012		

Table 3.1 showing all specs of UAV.

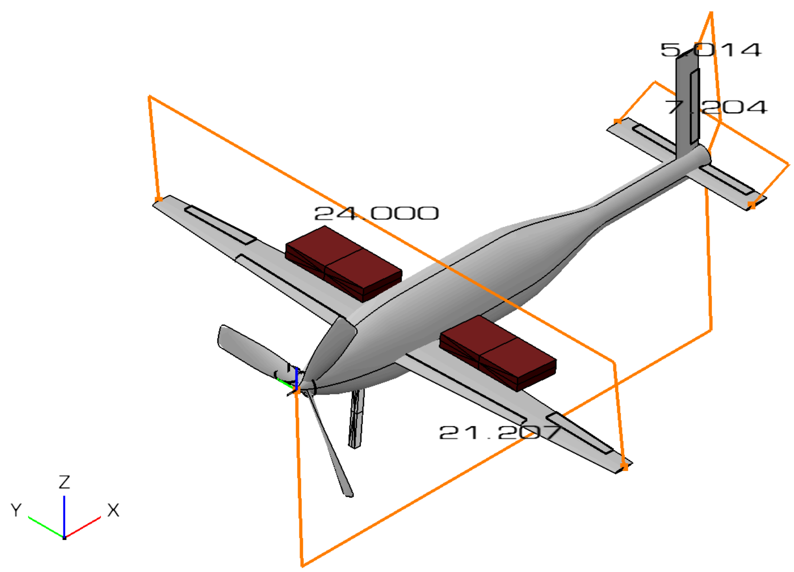


Figure 3.4 Shows CAD model with measurements for the UAV.

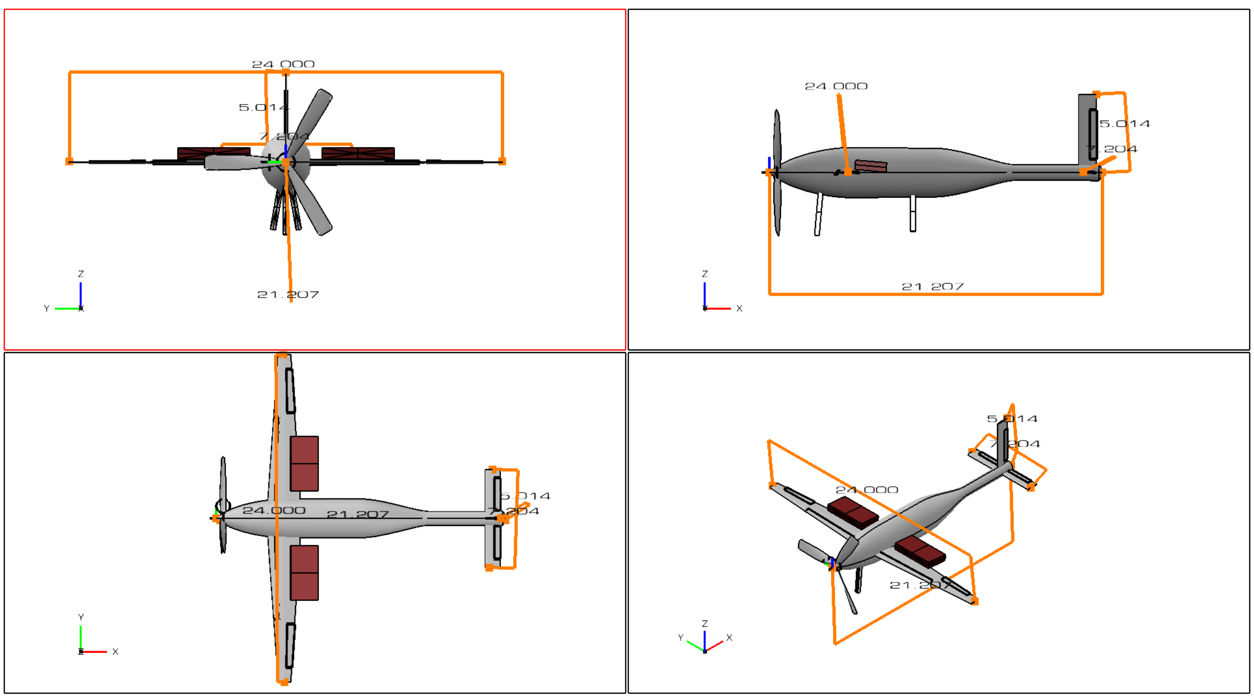


Figure 3.5 Four different view of CAD Model With Measurements

Performance

Propulsion:

To finalize engine choice, a list of 20 different engines were analyzed based on their weights and horsepower and the results are shown in the figure below. A scatter plot with a linear fit is generated to analyze how engine weight scales with horsepower.

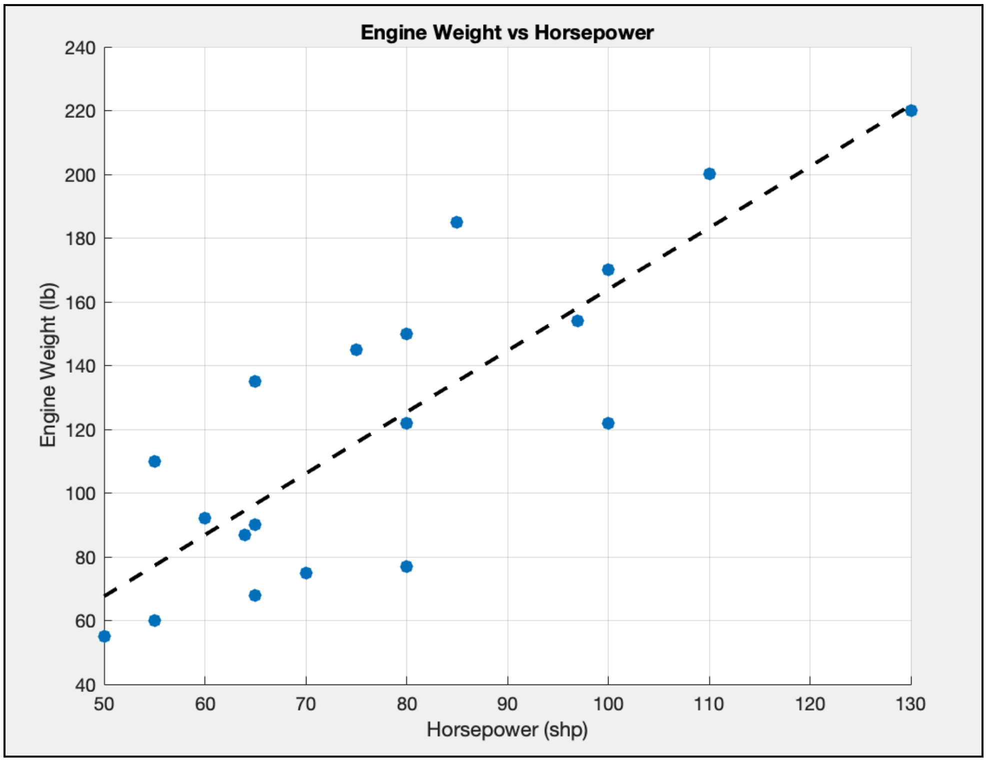


Figure 3.6 Shows 20 different engines' weights plotted vs. horsepower.

Chosen Engine Information: is Aixro XR-50 Engine

- 4-stroke, liquid-/air-cooled.
- Power: 50 HP at 8750 rpm.
- Weight: ~50 lbs (dry weight).

Compact and lightweight with excellent drivability and low vibration which make it an ideal choice for high-performance sports and challenging gust conditions.

Fuel: We are using Gasoline 91 as our fuel, which has a density of 6.07 lb/gal, ensuring that each gallon contributes a known weight to the overall mass budget. The engine's specific fuel consumption (SFC) is 0.7 slug/hp/hr, a key metric that indicates the amount of fuel burned per unit of engine power over time and helps us estimate fuel efficiency under various flight conditions. In our design, the fuel weight is 97.6 lb, and the fuel tank capacity is 16.07 gallons, setting practical limits on the aircraft's range and endurance. These fuel parameters are integrated into our weight and balance calculations, influencing the overall performance, center-of-gravity, and flight envelope of the aircraft.

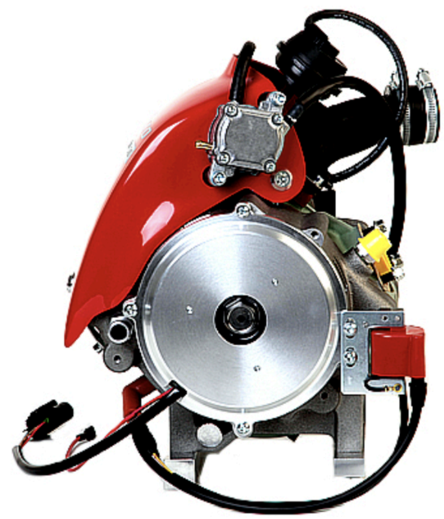


Figure 3.7 Aixro XR-50 Engine.

Propeller Efficiency (η):

- **Data Source:** The propeller class loads experimental data from CSV files corresponding to different pitch angles (α). Each CSV contains columns for the advance ratio J, thrust coefficient CT, and power coefficient CP (These values were obtained visually from the NACA640 charts provided in class).
- **Double Interpolation** method were performed for a given J and desired pitch angle (α), the code first interpolates within the dataset for two bracketing pitch angles, then linearly interpolates between those values.
- **Efficiency Formula:** Then efficiency is computed as:

$$\eta = \frac{CT}{CP} \times J$$

- The method **plotEfficiency** (or plotEfficiencies for multiple angles) generates a curve of efficiency versus *J*. This curve shows how the propeller's performance changes with the advance ratio for a fixed pitch as shown in the figure below.

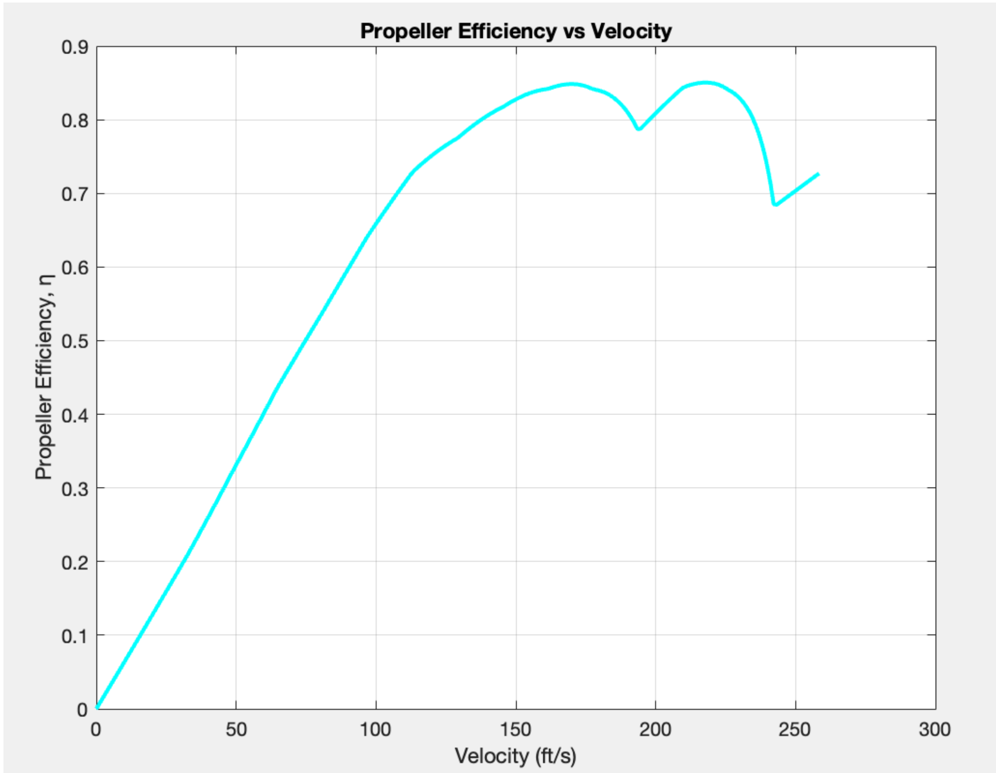


Figure 3.8 Shows NACA0012 (3 blades) propeller efficiency vs velocity.

Drag Calculations and Curves

- The matlab code computes and plots all types of drag as follows (calculated Cd0, CL for the use airfoil specifically):
    1. **Parasite Drag:** Using the calculated CD0 value (from the wing's airfoil data) and the formula.
    2. **Induced Drag:** Using the induced drag factor k and the lift coefficient CL
    3. **Total Drag:** The sum of the above two components is then plotted versus velocity. These curves illustrate the drag contributions across different flight speeds.

$$D = \frac{1}{2}\rho V^2 S C_{D_p} + \frac{1}{2}\rho V^2 S C_{D_i}$$

Parasite Drag **Induced Drag**

The following curve was then obtained. As seen the minimum drag of 37.28 lbf occurs at cruise speed of 101 ft/s.

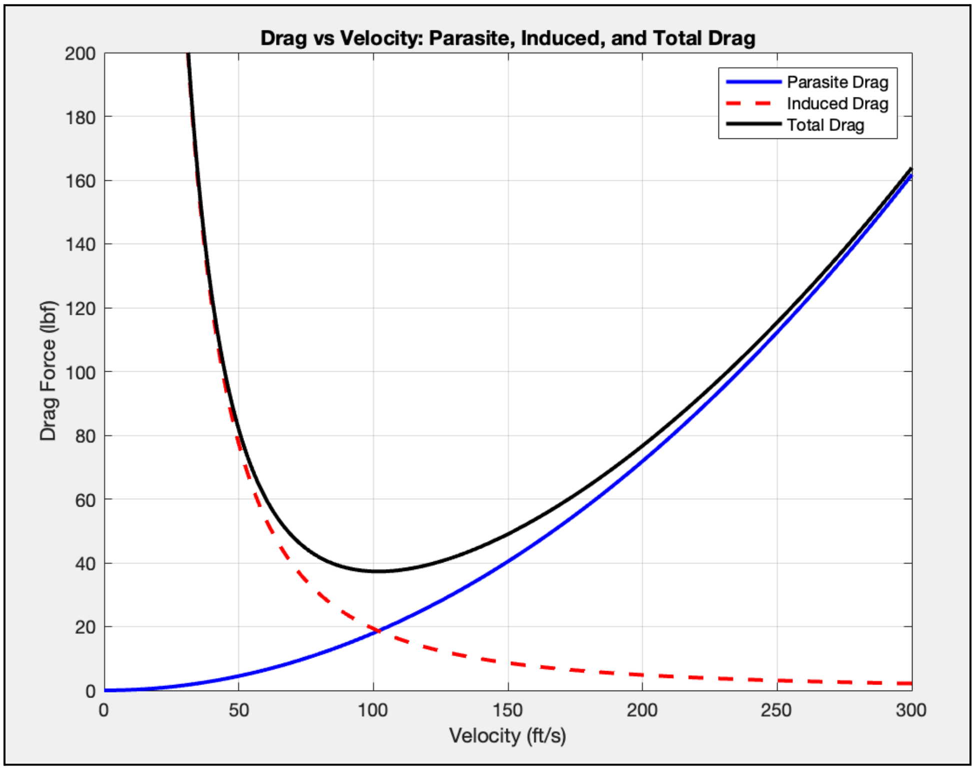


Figure 3.9 Shows Parasite, Induced and Total drag plotted vs velocity.

Additional Curves

- Lift vs. Angle of Attack (CL vs. α): The code retrieves the lift coefficients for a range of angles from the wing and tail data and plots these to form the lift curve.

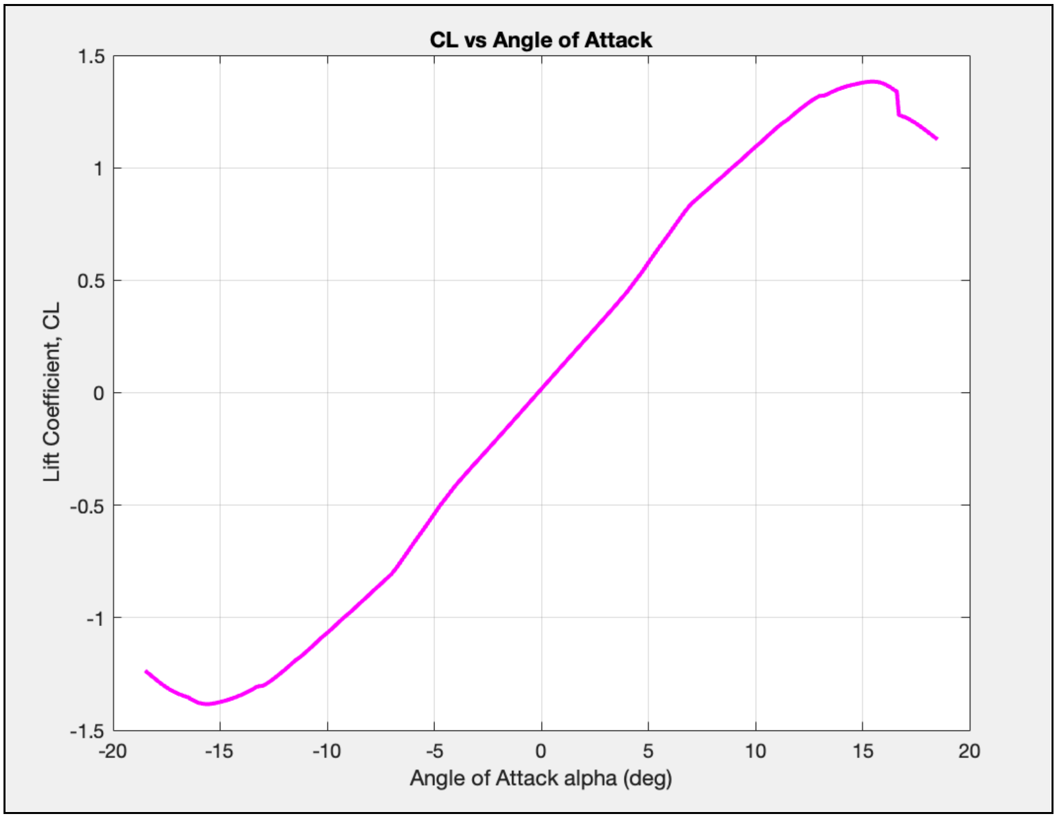


Figure 3.10 Shows a plot of coefficient of lift vs. angle of attack.

- Drag vs. Angle of Attack (CD vs. α): Similarly, drag coefficients are computed (combining parasite and induced components) over a range of angles and plotted.

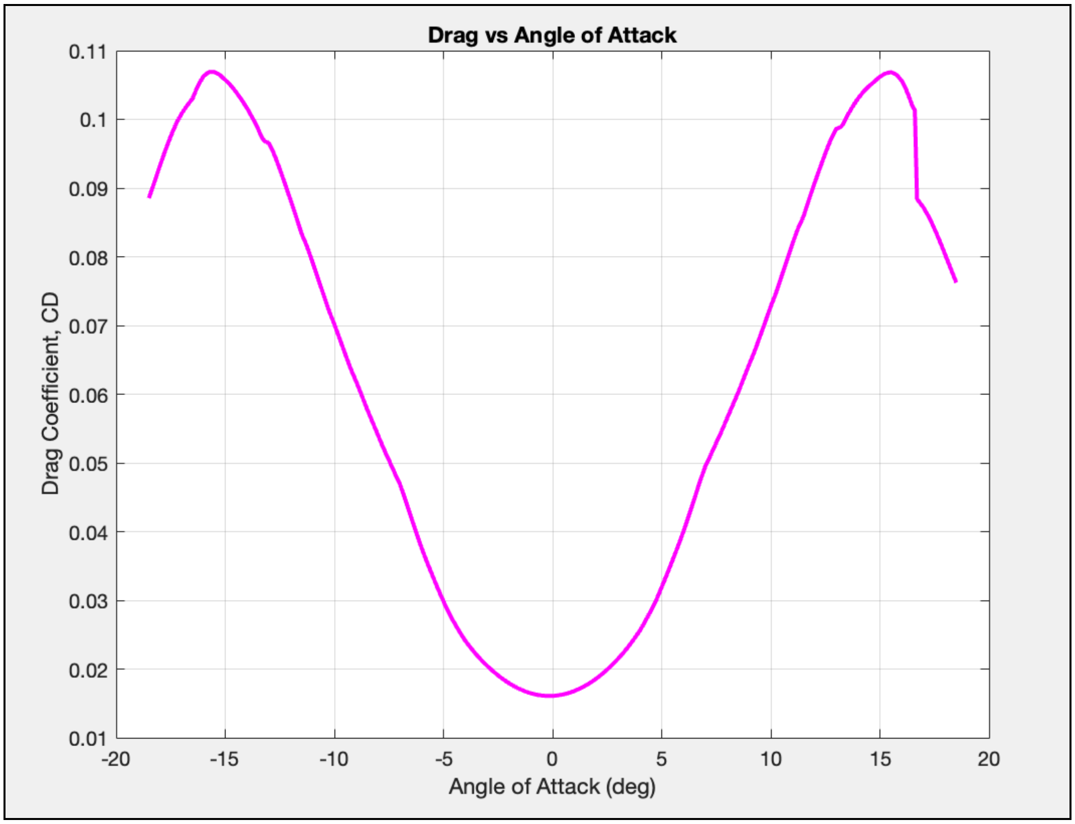


Figure 3.11 Shows a plot of coefficient of drag vs. angle of attack.

- Coefficient of Lift vs. Velocity curve: the coefficient of lift CL is then plotted against velocity which is an important plot as it indicates the stall speed which determines in turn the length of the runway.

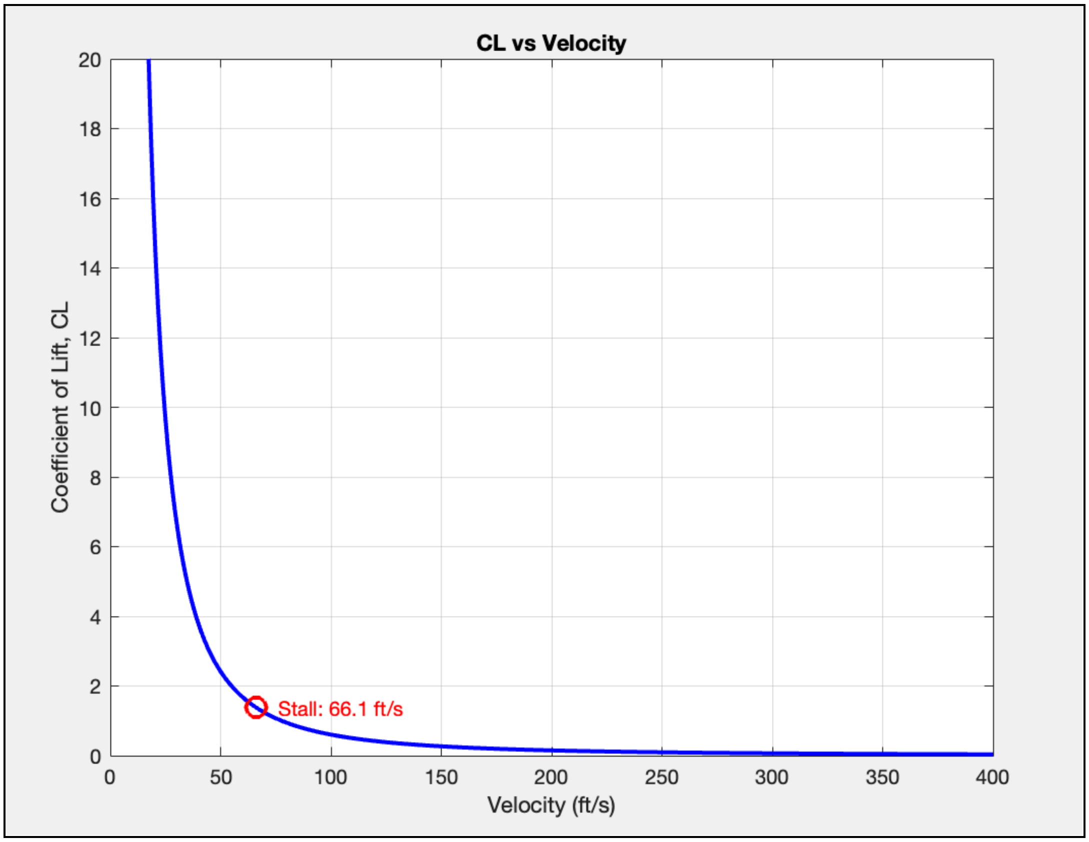


Figure 3.12 Shows a plot of coefficient of lift vs. velocity and marking a stall speed of 66 ft/s.

Power Available vs. Power Required

- **Power Available (PA):** The available power is computed from the propulsion system. The method calc_PA(v) multiplies the engine's rated power (in shaft horsepower) by the propeller efficiency at velocity v (obtained from the propeller's interpolation functions) and converts the result from ft·lb/s to horsepower (by multiplying by 550).
- **Plotting:** For each velocity, PA is computed and then plotted as a solid curve (after conversion to horsepower). This shows how much power is available from the engine/propeller system at different flight speeds.
- **Power Required (PR):** The power required is determined by the drag the aircraft must overcome. Then, the required power is given by:

$$PR = D_{total} \times v$$

- **Plotting:** The required power is computed over the same range of velocities and plotted (often as a dashed curve). Comparing the PA and PR curves helps identify the flight regime where the available power exceeds the power required.

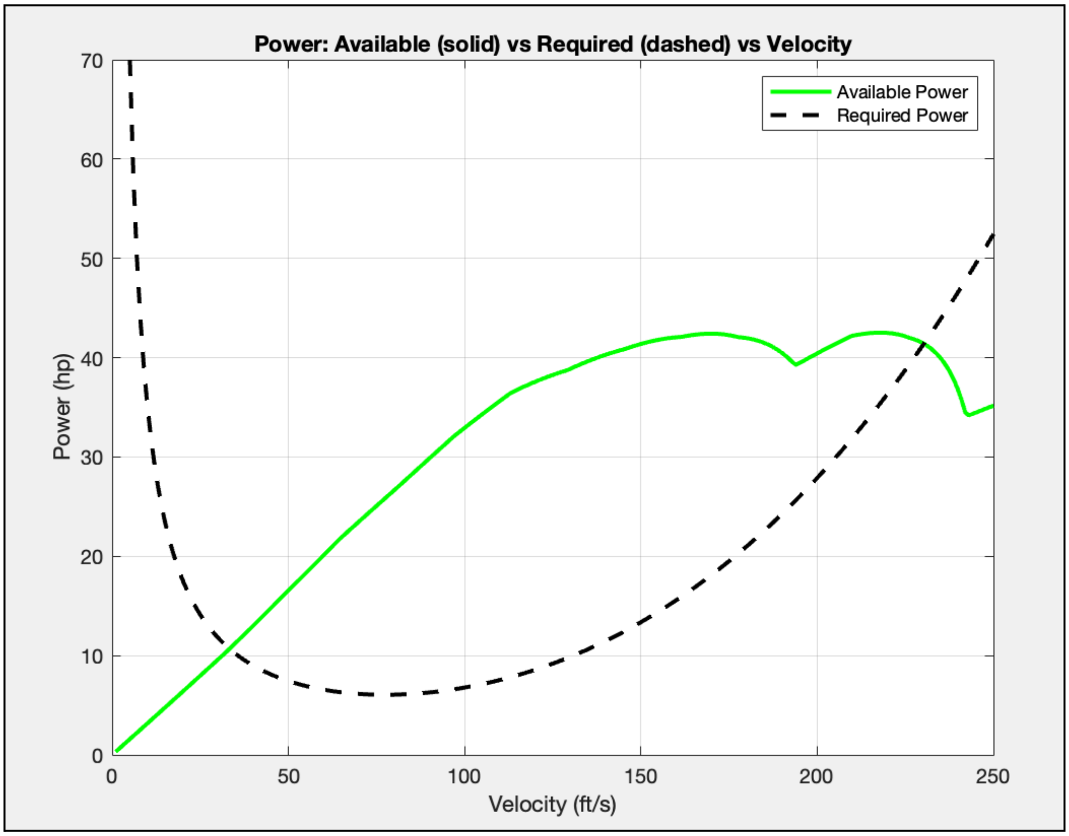


Figure 3.13 Shows a plot of power vs. velocity (intersect at Vmin = 66 and Vmax = 240 ft/s).

Weight

- Weight calculations were performed using the nicolai equations and optimized through various iterations to achieve the lightest viable aircraft.
- The weight pie chart visually breaks down the aircraft's total weight into key components—wing, tail, engine, fuselage, fuel, sensors, landing gear, and payload. Each slice represents the percentage contribution of that component to the overall weight. This helps designers quickly assess the weight distribution and identify any components that are disproportionately heavy, which is critical for ensuring balanced performance and stability.

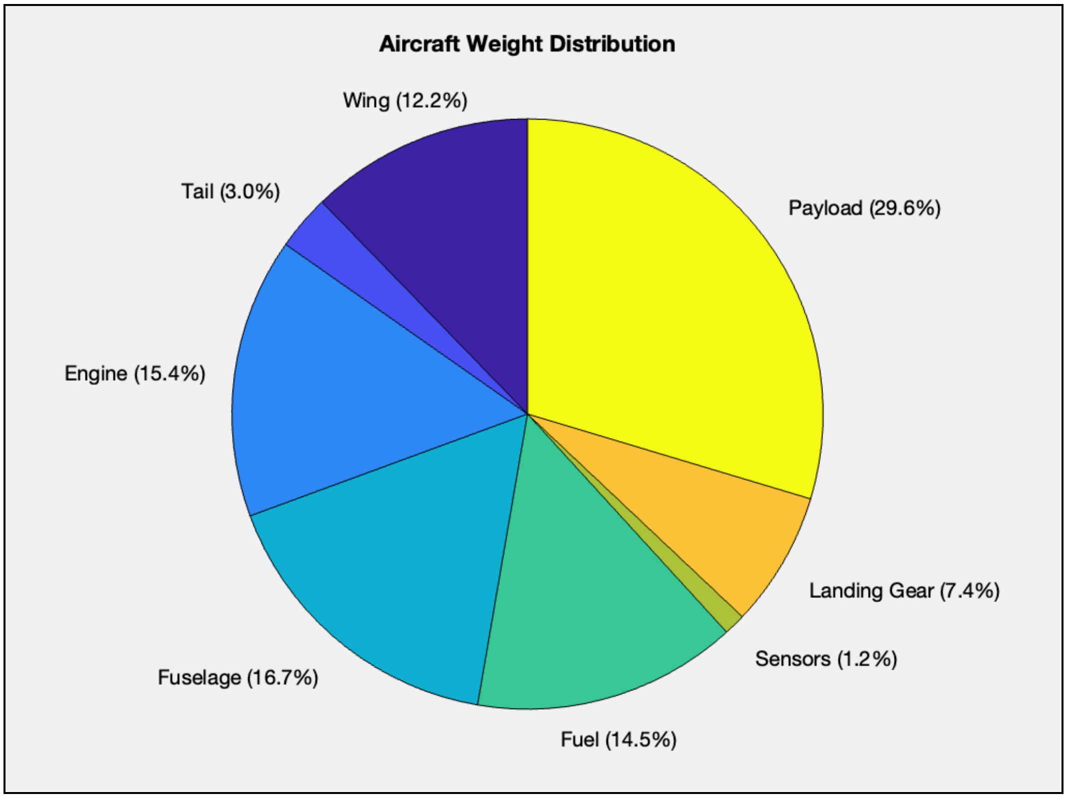


Figure 3.14 Shows a pie chart with all weights of components.

Stability Analysis

This section evaluates the aircraft's stability and trim conditions by computing key aerodynamic and performance parameters, and then it compares the aircraft's center of gravity (CG) to its neutral point (NP) for various extreme loading cases.

- Extracting CG and NP: For four configurations (by varying whether fuel and payload are at maximum or minimum) the code extracts the normalized CG (h_cg) and the neutral point (h_n) values from the aircraft configurations. These represent the positions along the chord (x/c) where the mass (CG) and the aerodynamic balance point (NP) are located.
- Static Margin Determination: The static margin is the difference between the NP and the CG (h_n − h_cg). A positive static margin indicates inherent stability, as it means the NP is aft of the CG. The code calculates these margins for all extreme cases.
- The values are then plotted on a graph where the x-axis represents the normalized distance along the chord and the y-axis serves as a visual grouping for the different cases. Different markers and colors distinguish between the CG and NP for each configuration.

This visual comparison helps designers quickly assess whether all configurations meet the desired stability criteria (typically a static margin between 5% and 30%).

Results:

Static Margin for CG cases	Max Fuel	Min Fuel
Max Payload	0.1741	0.1775
Min Payload	0.1906	0.2001

Table 3.2 showing static margin values for extreme cases.

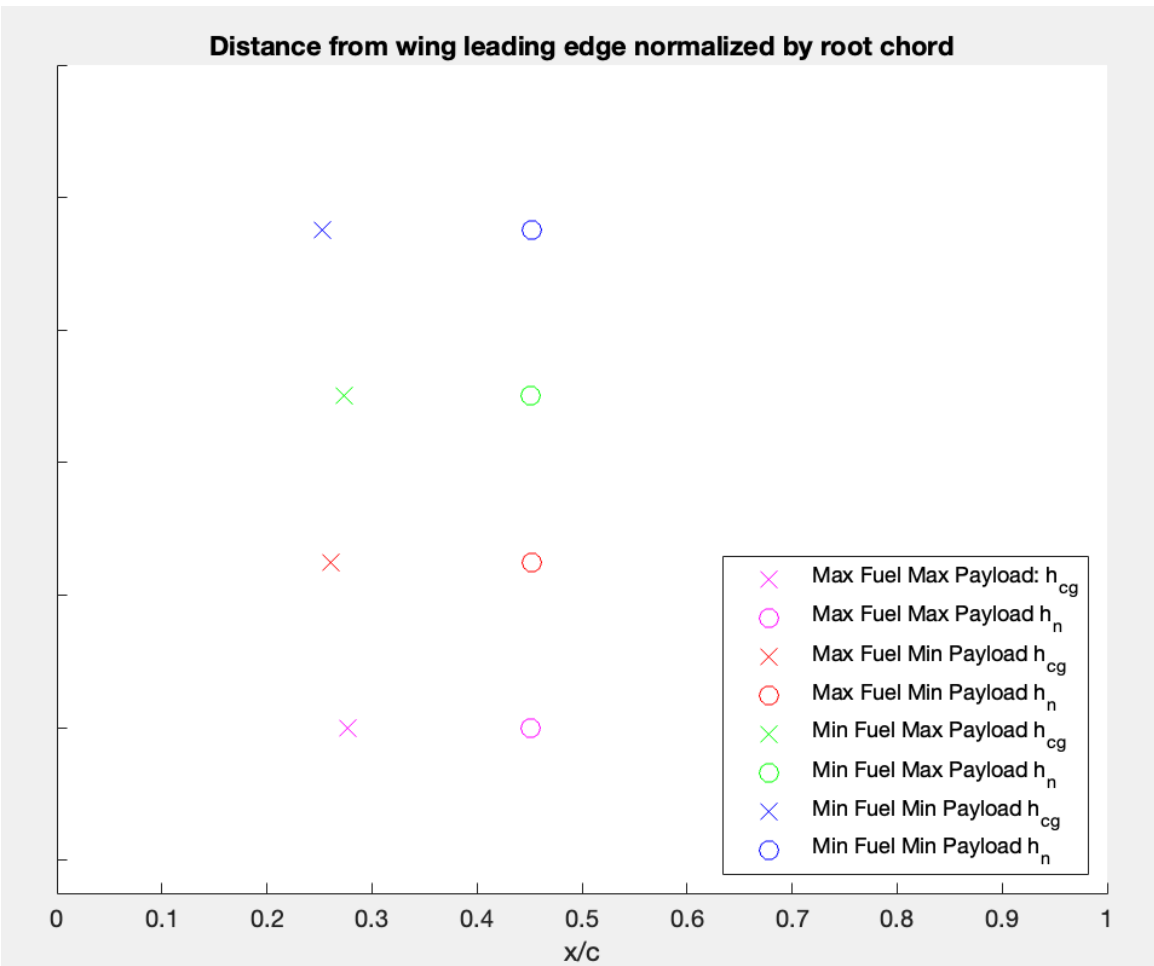


Figure 3.15 Shows distance from wing leading edge normalized by RC.

Stability Derivatives: These stability derivatives approximate how the aircraft's pitch moment responds to changes in angle of attack, pitch rate, roll rate, sideslip, and yaw rate. They rely on simplified formulas that incorporate the wing and tail's lift-curve slopes, the aircraft's tail volume coefficient, and basic geometric parameters (tail moment arm). Negative values (for CM_alpha and CM_q indicate stabilizing effects), an increase in angle of attack or pitch rate produces a restoring pitch moment. Meanwhile, terms like CM_B and CM_R show how lateral or yaw motions can couple into pitch, helping engineers assess whether the design adequately damps these cross-axis effects.

In the code, we created a dedicated method (calc_stability_derivs) that returns a structure of stability derivatives (CM_alpha, CM_q, CM_p, CM_B, CM_R). Inside this method, we pull the necessary geometric and aerodynamic data such as the tail's moment arm (l_t), tail and wing areas, lift-curve slopes, and chord lengths from other parts of the aircraft model. We then apply simplified empirical formulas that relate each derivative to these parameters. For instance, CM_alpha uses a tail volume coefficient and a downwash factor, while CM_q scales by both the tail volume coefficient and the distance from the tail to the wing's aerodynamic reference chord. By assembling these terms step by step, the code systematically computes each derivative, storing the results in a single structure. This approach keeps the stability calculations modular and easy to update if the aircraft geometry or aerodynamic assumptions change.

Below is a table showing computed derivatives:

Derivative	Value
CM_alpha	-1.1811
CM_q	-7.9301
CM_p	-0.9657
CM_B	6.2664
CM_R	-16.0100

Table 3.3 showing computed stability derivatives of the final UAV.

Optimization

(1) Requirement based Optimization

- Initially the process started by defining key aircraft dimensions such as (length, width, depth), payload and component weight targets, and initial guesses for gross weight, drag coefficients, and wing loading. Flight conditions such as cruise speed and air density were also set.
- **Sizing and Layout:** Using the initial gross weight guess, the wing area is estimated from the basic lift equation. The wing geometry (aspect ratio, taper ratio, chord lengths) and tail surfaces (horizontal and vertical tail areas) are sized based on proportions of the wing area. Common moment arms (the assumed station for non-wing masses) are defined for later CG calculations.
- **Iterative Weight and Stability Estimation:** We added an implementation of an iterative loop where it:
    - Calculated component weights using empirical formulas (such as Niccolai equations) for the wing, fuselage, tail, and propulsion.
    - Updated the gross weight guess using an exponential relation tied to range and propeller efficiency.
    - Recomputed wing area and geometry based on the new weight.
    - Determined the aircraft's center of gravity (CG) and neutral point, then calculates static margins (stability margins).
    - Adjusted the tail's aerodynamic center location iteratively to meet a target static margin.
- **Aerodynamic and Performance Calculations:** With the updated geometry, we computed drag (parasite and induced) over a range of velocities, resulting in drag curves. and also calculated power required (from drag and velocity) and compared this with available engine power. Additional performance plots included a Lift and Drag curves used to gauge overall aerodynamic efficiency.
- This method resulted in several graphs are generated to monitor the design:
    - A convergence plot showing how gross weight estimates evolve.

- Drag versus velocity curves (total, parasite, and induced).
- Power curves comparing required power and available power.
- A weight distribution pie chart that breaks down the aircraft's total weight into components (wing, tail, propulsion, fuselage, fuel, etc.), helping designers verify that the weight is appropriately distributed.
- Extreme CG cases and stability margins are also computed and printed to ensure the design meets stability requirements.

Requirement Check (final step):
Finally, the design outputs (such as fuel capacity, wing area, stability margins, and runway distance) are checked against the specified requirements (range, stall speed, maximum speed, static margin, etc.) to verify that the aircraft meets the design goals.

(2) Input Based Optimization

For our Input based optimization system we opted for the Monte Carlo system. For our implementation of this method we had 25 different input variables that would be used to generate the plane. These 25 different input variables would be randomized within +/- 10% of their value. These randomized variables would then generate a single plane using those inputs. Each iteration would generate 2000 planes. This allows us to test more planes with randomness and identify trends in the passing planes to aid in the optimization. A table of all the randomized Inputs can be found below.

Name	Var Name	Units
Wing Area	wing_Sw	ft^2
Wing Taper Ratio	wing_taper	None
Wing Aspect Ratio	wing_AR	None
Wing Root Chord	wing_cr	ft^2
Airfoil Thickness	airfoil_t	ft

Horizontal Tail Area	tail_Sh	ft^2
Vertical Tail Area	tail_Sv	ft^2
Horizontal Tail Chord	tail_ch	ft
Vertical Tail Chord	tail_cv	ft
Horizontal Tail Aspect Ratio	tail_ARh	None
Vertical Tail Aspect Ratio	tail_ARv	None
Engine	prop_choice	None
Propeller Diameter	prop_diam	ft
Propeller Pitch	prop_pitch	Degrees
Weight of Fuel	prop_Wfuel	Lbs
Location of Wing	shape_xwing	Percentage
Location of Engine	shape_xeng	Percentage
Location of Fuel	shape_xfuel	Percentage
Location of Payload	shape_xpayload	Percentage
Location of Sensors	shape_xsensors	Percentage
Location of Landing Gear	shape_xlg	Percentage
Location of Tail	shape_xtail	Percentage
Total Fuselage Length	PSize_fueslen	ft
Body Fuselage Length	PSize_fuesblen	ft
Body Fuselage Width	PSize_fuesbwid	ft

Body Fuselage Height	PSize_fuesbhei	ft

Table 4.1 Table of all randomized inputs

After generating 2000 planes the code would check if they pass or fail according to 6 criteria.

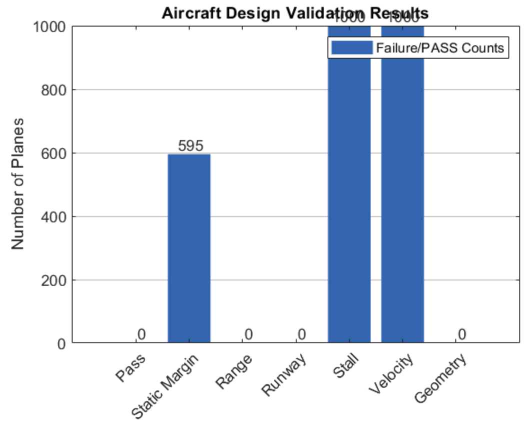
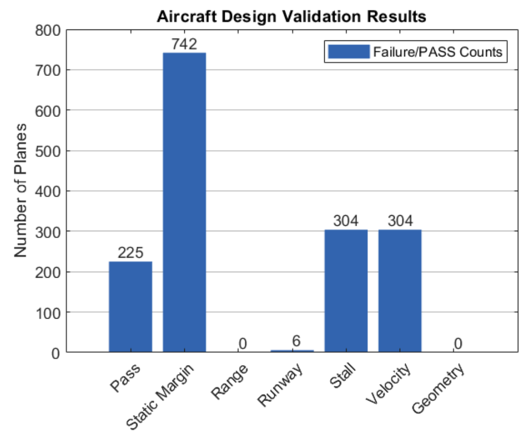
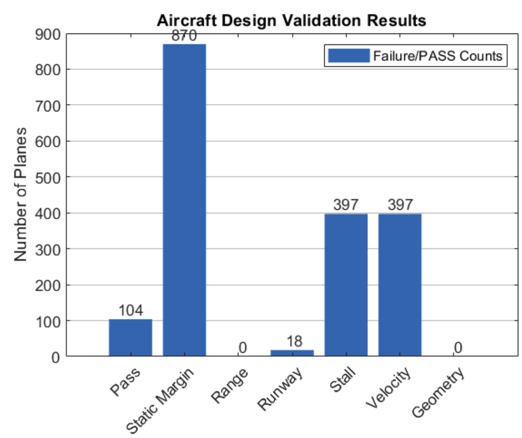
- All stability margins within 0.1 and 0.3
- Range of 400 nmi
- Runway distance
- Stall conditions
- Velocity check to ensure cruise speed make sense
- Geometry check to ensure plane makes sense

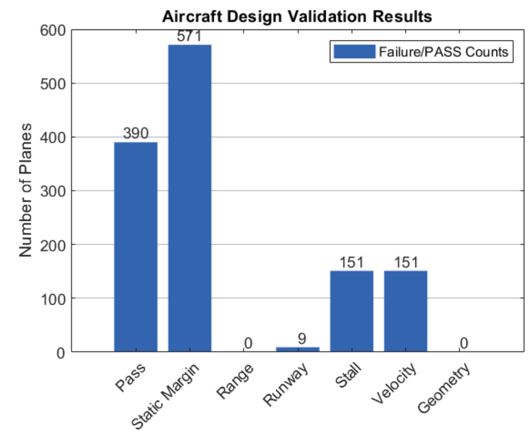
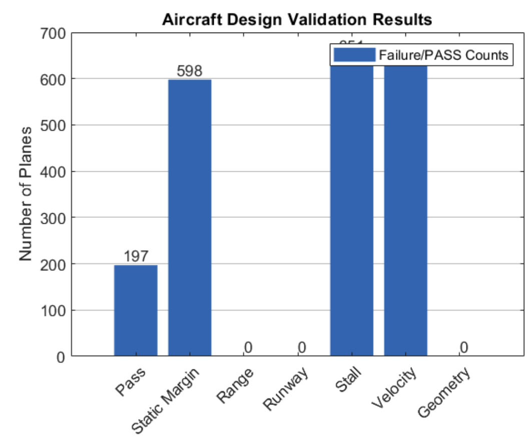
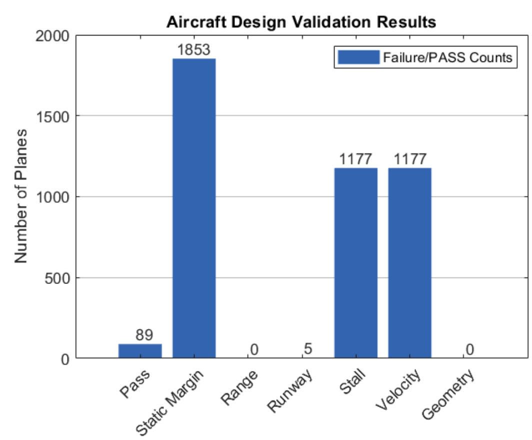
If a plane failed any or multiple of these criteria they would be assigned a value which allowed us to track what failures each plane had. If a plane passed it wouldn't be assigned a value. We also output the best plane from each generation by finding a plane with the smallest weight total and passing all the requirements. From that data we created 21 graphs comparing the total weight to the different inputs. Additionally we generate a histogram showing the passing amount of plane. These graphs in combination with the best plane output allow us to determine the success of the generation and provide insight on the trends. We adjust our inputs for the next generation based on the graphs, trends we noticed, and the best plane output.

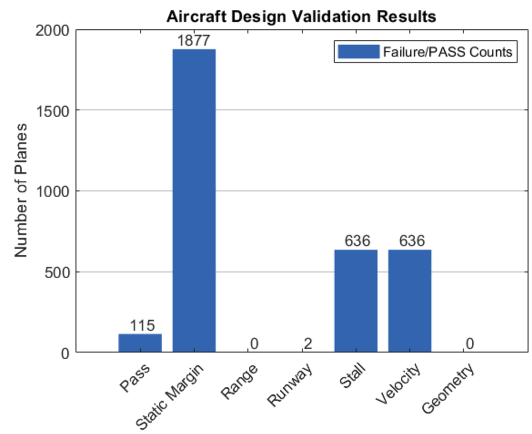
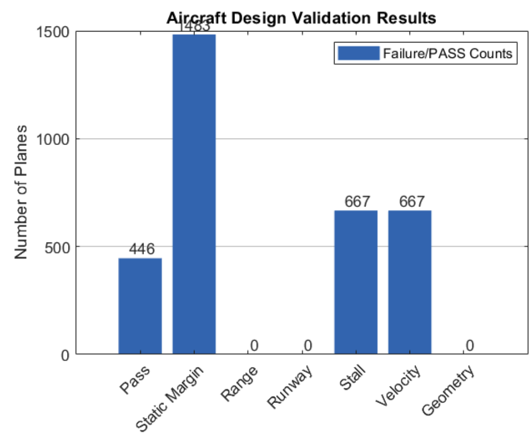
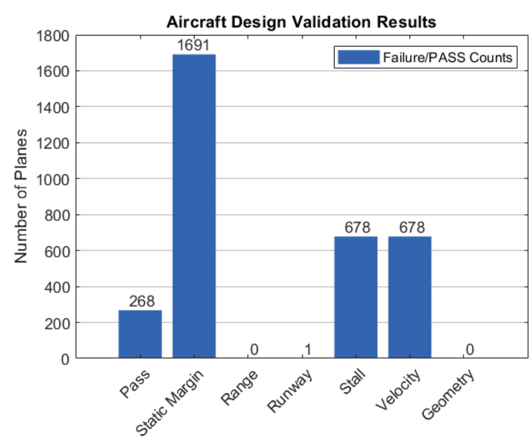
Generations

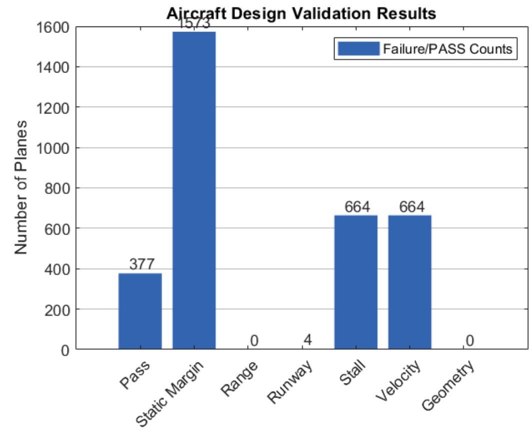
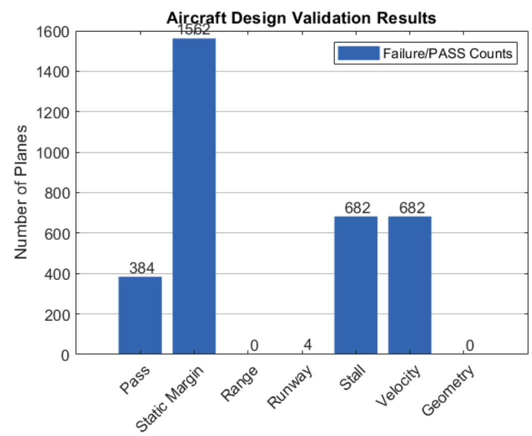
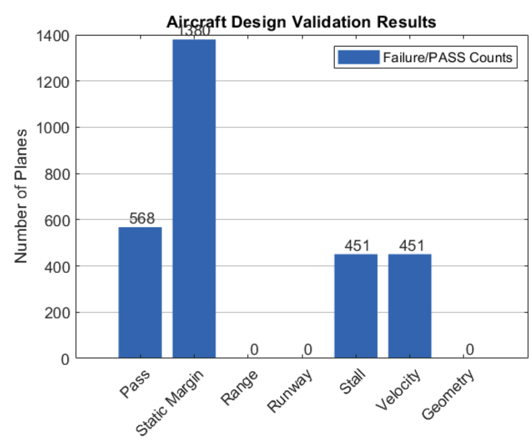
In total there were 16 generations resulting in 27000 planes created total. Each generation and what changed in each generation is recorded in the table below.

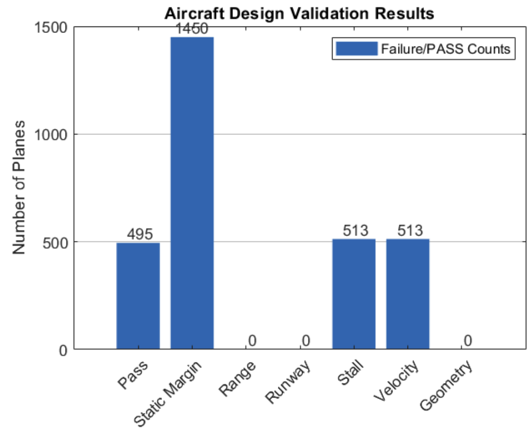
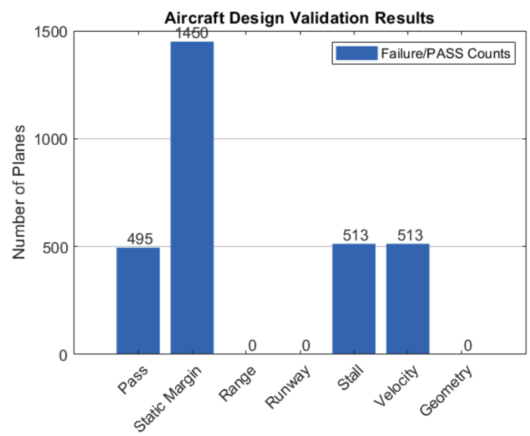
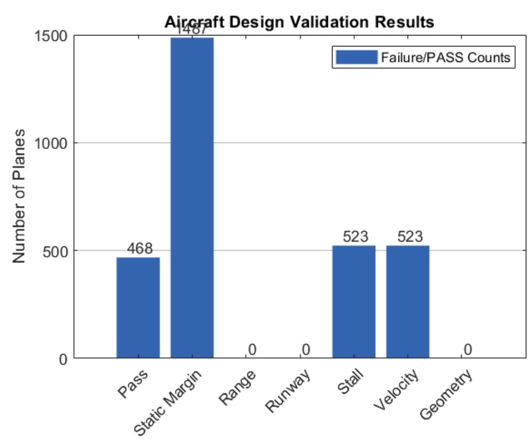
#	What Changed	% Pass	Best plane Weight Total	Histogram of Run

1	Inputs of a reference plane	0	N/A	
2	Used Preliminary Design Inputs	22.5%	903	
3	Adjusted Wing position forward, reduce fuel amount	10.4%	864	

4	Increased Wing area,	39%	852	
5	Adjusted wing areas and length	19%	830	
6	Reduce Wing Area to 70 Adjust Fuselage width and height to 3	4.45%	789	

7	Adjusted Pitch to 27	5.75%	680	
8	Move Wing position forward	22.3%	684	
9	Decreased Body fuselage length	13.4%	678	

10	Increased Total fuselage length and body fuselage length	18.8%	677	
11	Same inputs as before	19.2%	689	
12	Increase Wingspan to 100, reduce fuselage and body fuselage lengths	28.4%	694	

13	Decrease Wingspan to 90 and increase horizontal tail to 9	24.7%	694	
14	Decrease Horizontal tail to 7	24.7%	690	
15	Used Best Plane input from 12	23.4%	674	

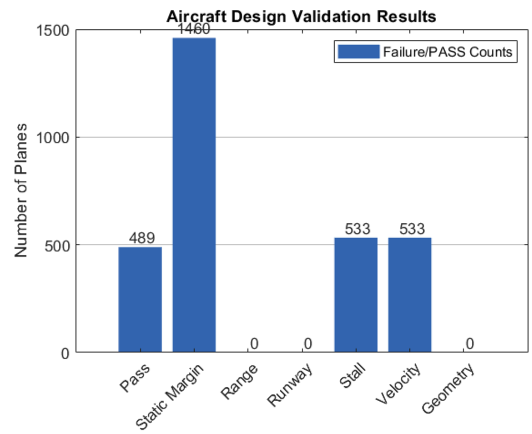
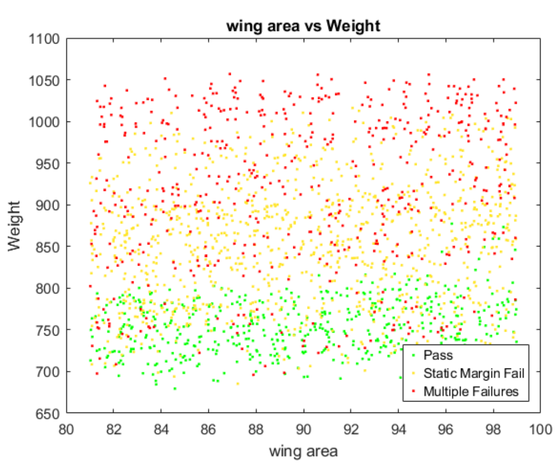
16	Used Best Plane input from 15	24.4%	679	

Table 4.2 shows changes each generation success %, best plane total weight, and a histogram of the iteration

After these generations the final generation's best plane is used as our final plane design. Through our generations we adjusted the first order effects that would be the Wing area and total length. As the generations went on we were able to adjust the fuselage height and width as well as horizontal tail area. In addition to the histogram each generation also generated 21 graphs displaying the total weight vs input variable and whether it passed or not. These graphs allowed us to analyze each input variable and identify trends to help focus our inputs. In the end our final generation best plane is the plane we used as our final plane design.

Final Generation

Below are some of the graphs generated by our final generation of planes. The graphs display the weight vs input variable for the final generation. The green circles represent the panes that pass, yellow is for static margin failure, and red is for multiple failures.



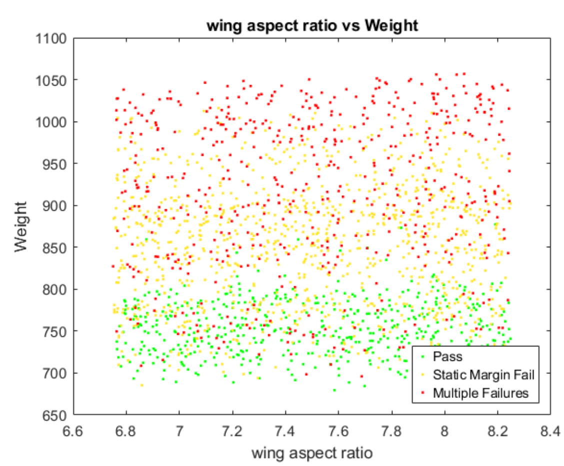


Fig 4.2.1: Weight vs Wing area and Weight vs Wing Aspect ratio

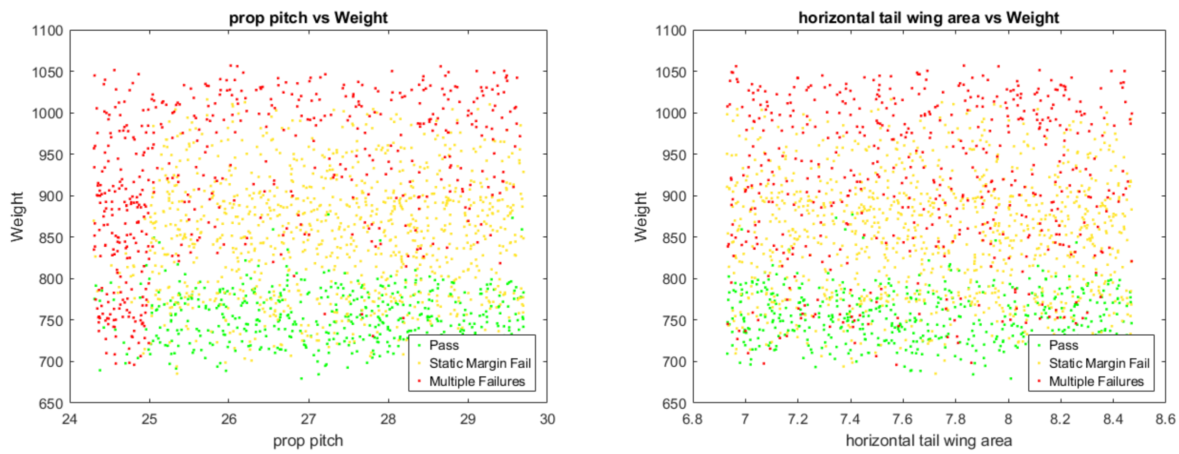


Fig 4.2.2: Weight vs Prop Pitch and Horizontal tail wing area vs Weight

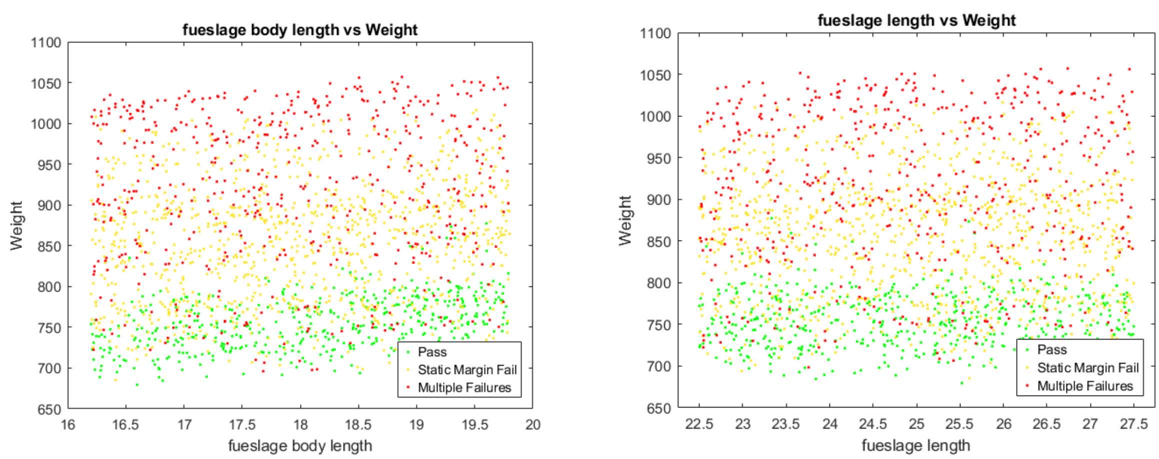


Fig 4.2.3: Fuselage Body length and total Fuselage length vs Weight

Guidance and Navigation Controls

To execute our UAV mission there are two parts of the process: preflight ground control and inflight operation. Before the flight using weather radars and other forms of ground control to map out weather patterns and determine the clouds we want the UAV to fly through. After mapping out the clouds, their heading, and our flight plane we can move onto the flight stage.

The following section covers the various flight strategies we attempted and implemented for our UAV to carry out the missions.

Strategy

Pursuit (Waypoint Navigation):

- Strategy: The pursuer aims to directly follow the target's path, trying to maintain a direct line of sight to the target.
- Outcome: While seemingly intuitive, this strategy can lead to overshooting or unnecessary energy expenditure, especially if the target changes course.
- Implementation: By using waypoint navigation, a waypoint would be placed at the cloud's location, and as a result the aircraft's heading will constantly updated to face be pointed the cloud until interception

Constant Bearing, Decreasing Range (Proportional Navigation):

- Strategy: The pursuer maintains a constant angle (bearing) to the target, regardless of the target's movement.
- Outcome: If a constant bearing to the cloud is held the aircraft and cloud are on a collision course.
- Implementation: By using proportional navigation, a method used to plot a perfect interception course by holding the target at a constant angle relative to the aircraft's heading the aircraft

Benefits of Constant Bearing, Decreasing Range Strategy

- Interception: Constant bearing is often more efficient for interception than pure pursuit, as it can lead to a shorter interception distance.
- Energy Efficiency: Pursuit can be less energy-efficient, especially if the target changes course frequently.

Overview

The GNC computer implements a three-phase mission for a cloud-seeding UAV in a compressed simulation. The mission is designed based on two main components: the

UAV as the pursuer, and the cloud as the target. The Cloud (or Target) is moving in a straight-line trajectory at a constant speed and is assumed to always be flying at a fixed altitude of 20,000 feet. The UAV (or Pursuer) starts at sea level (0 ft) and begins gradually flying toward the cloud and keeps climbing until it reaches 20,000 ft during the (Approach & Climb) phase. Once the UAV is close enough horizontally and at the correct altitude, it enters the (Follow phase) to track the cloud. After following for a preset enough duration to complete the mission, it then enters the (Return & Descend) phase, returning to its initial home position and descending back to sea level.

The overall system consists of two main functions one is a guidance law function called porp_guid which replaces the waypoint_guid function in the pioneer simulator. The porp_guid calculates the commanded bank angle and waypoint based on the current UAV and cloud positions. A separate plotting function (called show_map) is used to accumulate and visualize the full 3D trajectories of both the UAV and the cloud, as well as a fixed reference (initial positions) at sea level.

Explanation of The porp_guid Function

- **Purpose:** The function is used to compute the horizontal guidance command for the UAV and to manage its altitude. It uses an augmented proportional navigation (PN) logic for horizontal steering and a simple altitude controller for gradual ascending motion. The function operates using a state machine that divides the mission into three phases:
- **Phase 1 (Approach & Climb):** The UAV flies toward the cloud while climbing from 0 ft to 20,000 ft.
- **Phase 2 (Follow):** Once the UAV is near the cloud (within a horizontal threshold and almost at the target altitude), it maintains a 20,000-ft altitude and follows the cloud.
- **Phase 3 (Return & Descend):** After following for a fixed duration, the UAV returns to its initial (home) position and descends back to sea level.
- **Code Breakdown (variables explanation)**: The function uses persistent variables to maintain state between calls:
  phase: The current mission phase (1, 2, or 3).
  start_time: Timer to measure elapsed time in the current phase.
  init_pos: The initial horizontal position of the UAV (used for returning home).

UAV_alt: The current altitude of the UAV.

last_call_time: Used to compute the time step (dt) for altitude updates.

A Global Variable of *CURRENT_UAV_ALT*: This global variable mirrors the persistent altitude UAV_alt and is used by the plotting function for visualization.

target_alt: which is set to 20,000 ft.

climb_rate is set as climb rate to ascend and descend within the simulation.

approach_thresh (1,000 ft): defines the horizontal distance threshold for transitioning from the Approach phase.

altitude_tol (100 ft): defines the tolerance for altitude error before switching to the Follow phase.

follow_duration (3 s for simulation purposes but actual duration will be much higher) is the period during which the UAV follows the cloud.

- **Altitude Update Mechanism:** The altitude update uses the elapsed time dt (computed from last_call_time) to increment or decrement the current altitude (UAV_alt) toward the desired altitude (alt_des). A default value is provided if dt is too small, ensuring that altitude changes are perceptible within the simulation time frame.
- **Error Calculation**: The horizontal error is computed between the UAV's current position and the desired waypoint (which changes based on the mission phase).
- **Heading Command:** The desired heading is calculated using atan2(delta_E, delta_N).
- **Bank Angle Command:** An augmented PN law calculates the required lateral (normal) acceleration:

$$a_{normal} = N \times V_c \times \lambda_{dot}$$

- Where λ_{dot} is approximated by dividing the heading error by a time constant. This acceleration is then converted into a commanded bank angle (φ_comm) using a small-angle approximation. The bank angle is limited to ±30°.

Explanation of the show_map Function

- **Purpose:** The function is responsible for the visualization of the UAV and cloud trajectories in a 3D plot. It accumulates the positions over time using persistent arrays and then displays:
- *UAV Trajectory (Blue)*: A blue line shows the path and a blue dot represents the current position.

- *Cloud Trajectory (green):* A green dot represents the cloud and is always plotted moving with a constant speed at a fixed altitude of 20,000 ft.
- **Trajectory Data Variables:**
  uavTrajectory: Stores every recorded position of the UAV as a row vector [pE, pN, UAV_alt].
  cloudTrajectory: Stores the cloud's positions (with altitude fixed at 20,000 ft).
  Plotting: Plots the accumulated trajectories using plot3 for a 3D visualization. It also sets appropriate axis limits and labels to allow clear visualization of the horizontal (pE, pN) and vertical (altitude) dimensions.

Simulator Blocks:

The following figures display the changes we made to the pioneer simulator block in order to implement our proportional navigation system.

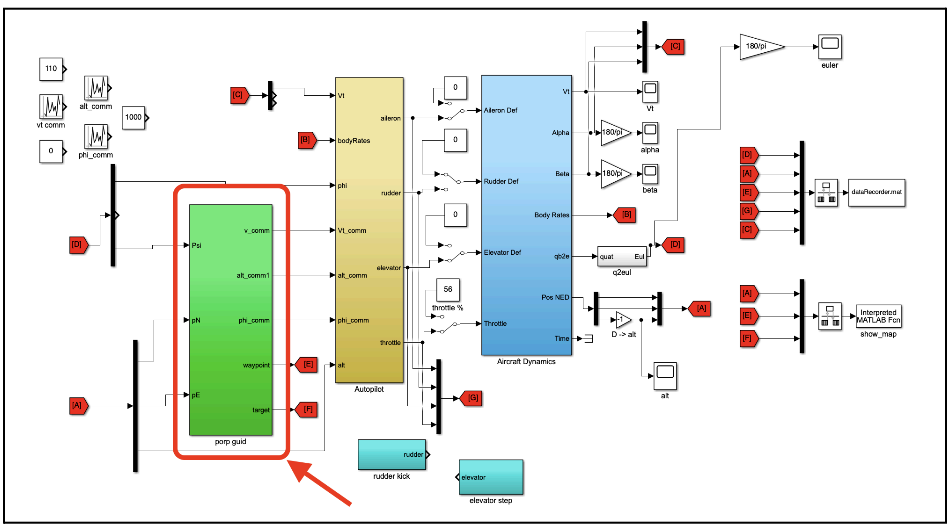


Figure 5.1 Shows the pioneer simulator blocks with the replaced porp_guid function.

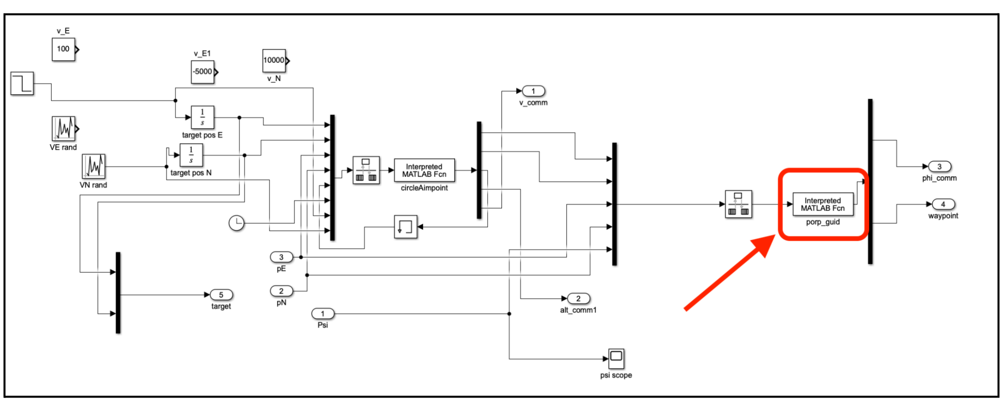


Figure 5.2 Shows the pioneer simulator blocks with the replaced porp_guid function.

Results

The results of our simulation can be seen in the following figures. The first figure shows the trajectory of the UAV in 3 dimensions through the entire mission. The second figure shows the cloud in green and the UAV intercepting the cloud to allow for the payload to be deployed.

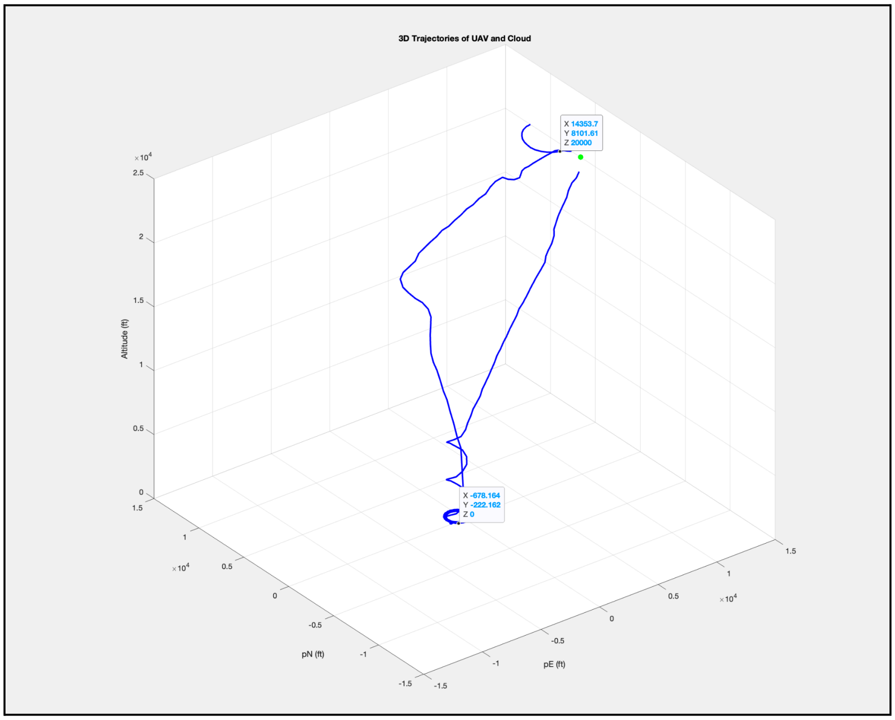


Figure 5.3 Shows the trajectory of the UAV in 3D performing the mission and returning to initial position.

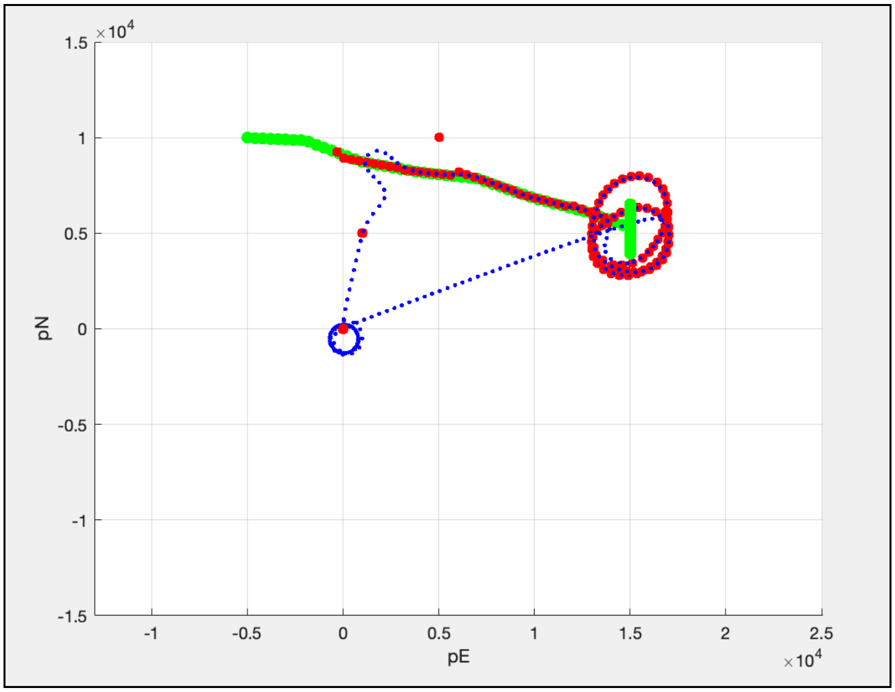


Figure 5.4 Shows the trajectory of the UAV in 2D performing the mission and returning to initial position.

Conclusion

This report presents a preliminary design and analysis of a cloud seeding aircraft that meets essential performance and stability requirements. The design features a fixed-wing configuration with a nose-mounted engine, a fuselage that integrates payload and fuel storage, a robust tri-fixed landing gear, and chemical flares mounted under the wing for effective cloud seeding. Stability derivatives and weight estimations have been rigorously computed to ensure balanced flight characteristics, while aerodynamic and performance analyses validate the aircraft's operational envelope.

Future improvements: Future improvements could involve enhancing the weight and fuel efficiency through advanced optimization algorithms and high-fidelity simulations. Incorporating adaptive wing and control surface technologies may further improve stability and maneuverability under variable atmospheric conditions. Additionally, exploring variable-pitch propellers and refined propulsion models could yield better power management, ultimately increasing range and endurance. These advancements would contribute to a more robust and versatile aircraft design for cloud seeding missions.

References

Gray, Jennifer, and Rachel Ramirez. "Scientists in the US Are Flying Planes into Clouds to Make It Snow More." *CNN*, Cable News Network, 14 Mar. 2022, www.cnn.com/2022/03/14/weather/cloud-seeding-weather-modification-wxn/index.html.

https://www.dropletmeasurement.com/product/cloud-droplet-probe/

https://www.cloud-seeding-technologies.com/products/flare-racks-and-mounts/

http://airfoiltools.com/

https://aeroenginesaz.com/en/brand_aixro

https://www.sciencedirect.com/science/article/pii/S0169809523001850

https://onlinelibrary.wiley.com/doi/10.1155/2022/3128657#:~:text=This%20study%20applies%20a%20novel,and%20ground%2Dbased%20precipitation%20detection.

https://www.avbuyer.com/articles/special-missions-aircraft/aerial-cloud-seeding-the-rainmakers-of-aerial-work-aviation-113771#:~:text=Cloud%20seeding%20is%20primarily%20carried,to%20form%20within%20the%20cloud.

https://openvsp.org/