

Undergraduate Research Opportunity Program

## Conceptual Design, Mission Planning, and Aeroelastic Analysis of a High Aspect Ratio Solar-Powered UAV

**Author:** Suhail Halaby (02383703)  
**Supervisor:** Dr. Rafael Palacios

### Abstract

This report presents the conceptual design, aeroelastic analysis, and mission planning of a high aspect ratio solar-powered UAV developed under Imperial College's Solar Plane Initiative. The project aimed to establish a multidisciplinary framework for performance prediction and optimization, focusing on aeroelastic modeling using SHARPy, control law tuning via ArduPilot SITL, solar income estimation through shadow-casting simulations, and trajectory planning for endurance and point-to-point missions. The analysis confirmed structural stability across expected flight regimes and demonstrated effective integration of control and planning systems. The airframe construction nears completion, with flight testing anticipated in Autumn 2025.

Aug 2025

# Contents

|   |    |
|---|----|
| List of Figures . . . . .                                       | 2  |
| List of Tables . . . . .  | 3  |
| 1 Introduction . . . . .  | 4  |
| 1.1 Motivation . . . . .  | 4  |
| 1.2 Objectives . . . . .  | 4  |
| 1.3 Airframe Overview . . . . .                                 | 4  |
| 2 Aeroelastic Analysis . . . . .                                | 5  |
| 2.1 Modeling Parameters, Discretization, and Numerics . . . . . | 5  |
| 2.2 Static Analysis . . . . .                                   | 6  |
| 2.3 Linearized Dynamics . . . . .                               | 6  |
| 2.4 Reduced Order Models and Velocity Analysis . . . . .        | 8  |
| 2.5 Dynamic Simulations and Gust Responses . . . . .            | 8  |
| 3 Applied Control Law and SITL Testing . . . . .                | 10 |
| 4 Solar Income Modeling . . . . .                               | 10 |
| 5 Optimal Waypoint Mission Generation . . . . .                 | 11 |
| 5.1 Trajectory Planning for Loitering Missions . . . . .        | 12 |
| 5.2 Trajectory Planning for Point-to-Point Missions . . . . .   | 12 |
| 6 Conclusion and State of Completion . . . . .                  | 13 |

# List of Figures

|    |  |    |
|----|--|----|
| 1  | CAD Rendering of the Aircraft, with Spar Geometry Exposed . . . . .                    | 5  |
| 2  | SHARPy Discretization . . . . .  | 6  |
| 3  | Static Deflection of the Wing Loaded to S.L.F. Conditions . . . . .                    | 6  |
| 4  | A Comparison of Linearized Models . . . . .  | 7  |
| 5  | Root Locus Diagrams, Krylov Subspace ROMs at 15 m/s . . . . .                          | 8  |
| 6  | Velocity Analysis, Krylov Subspace ROMs . . . . .                                      | 8  |
| 7  | BFOR Flight Path Trajectories, 10% 1-cos Gust Response . . . . .                       | 9  |
| 8  | Rigid Body Velocities, 10% 1-cos Gust Response . . . . .                               | 9  |
| 9  | Rigid Body Angular Rates, 10% 1-cos Gust Response . . . . .                            | 9  |
| 10 | Wing Root Loading, 10% 1-cos Gust Response . . . . .                                   | 10 |
| 11 | Wingtip Deflection, 10% 1-cos Gust Response . . . . .                                  | 10 |
| 12 | Day-Long Motion of the Sun at a Latitude of 52, Mid-Aug. 2025 . . . . .                | 11 |
| 13 | Unobstructed Area Fraction Using Sun Location in the Body Frame of Reference . . . . . | 11 |
| 14 | Hub-and-spokes Path Constraints . . . . .  | 12 |
| 15 | Evaluating the Capabilities of A* Based Path Planning over 20 km Missions . . . . .    | 13 |
| 16 | Sate of Completion as of Mid. Aug. 2025 . . . . .                                      | 14 |
| 17 | Modal Solver Settings . . . . .  | 17 |
| 18 | LinearAssembler Solver Settings . . . . .  | 17 |
| 19 | Krylov ROM settings . . . . .  | 18 |
| 20 | Modal Solver Settings . . . . .  | 18 |
| 21 | LinearAssembler Solver Settings . . . . .  | 18 |
| 22 | Velocity Analysis Settings . . . . .   | 18 |

# List of Tables

|   |  |    |
|---|--|----|
| 1 | Key Airframe Statistics . . . . .                                  | 4  |
| 2 | Estimated Structural Properties of Aircraft Sections . . . . .     | 5  |
| 3 | SHARPy Structural Discretization Parameters . . . . .              | 5  |
| 4 | Comparing the Rigid Body Modes Between VLM-Based Methods . . . . . | 7  |
| 5 | Component mass and CG locations from nose datum BFoR . . . . .     | 16 |

# 1 Introduction

## 1.1 Motivation

This project forms part of the Imperial College Aerial Vehicles Solar Plane Initiative (SPI), an endeavor established with the objective of breaking the endurance record for sub-50 kg aircraft.

UROP was directed towards the development of a comprehensive, multidisciplinary framework for performance analysis, which will form critical element in the progression of the initiative. Ultimately, this design framework is intended to establish the feasibility of the current design, maximize its performance potential, and identify avenues for refinement in future iterations.

This report serves as a brief summary of the efforts undertaken during the 8-week period.

## 1.2 Objectives

The development of the aircraft is contingent on the following objectives, each of which has been addressed to some extent during the duration of the research opportunity. Ultimately, this project seeks to develop a suitable design methodology for future endeavors. As such, the results of this investigation are to be interpreted in the context of being representative of the first iteration of many. The objectives associated with the design process can be allocated into four categories.

**Aeroelastic Analysis:** This will involve using VLM based methods to characterize the open-loop dynamics of the aircraft. This process will ultimately yield the information needed to understand the aircraft's flight envelope.

**SITL Controller Tuning:** Utilizing a suitable model of the aircraft in conjunction with Ardupilot-Simulink SITL tools to tune the autopilot.

**Solar Income Modeling:** Developing high fidelity tools for solar income modeling, sufficient for providing reasonable endurance estimates and useful metrics for optimal mission planning.

**Mission Planning:** Using the solar income model, the control law, and the aircraft's flight envelope to inform the planning of optimal waypoint missions.

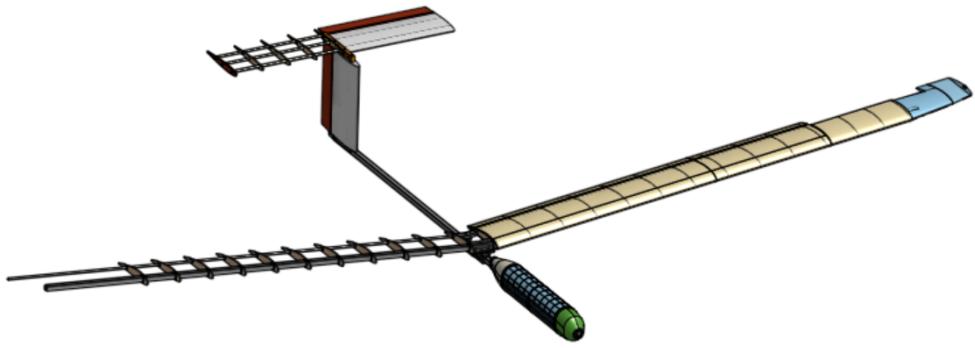
## 1.3 Airframe Overview

The current ICAV-SPI airframe exploits a great deal of conventional model-making wisdom; thin sheets of balsa wood are used to create a solid skin, attached by thicker balsa wood ribs to an aluminum main spar of constant cross-sectional area. There is also a tubular CFRP secondary spar, which also serves as the hinge for the wing-mounted control surfaces. The tail is constructed similarly, but the aluminum spar is substituted for a set of smaller CFRP rods. Once construction is complete, the aircraft's aerodynamic surfaces will be draped in a thin, iron-on plastic film. Table 1 and figure 1 illustrate the proportions of the aircraft.

This airframe is intended to be a cost effective (yet modular) testbed for the powertrain and for future designs. It features many hot-appeasable components (such as the wing root attachment point, the avionics/-payload bay, the tail, etc.). While not entirely optimized in it's current state, taking the hot-swappable parts further will be a key focus in the near future. Prior to the UROP, significant analysis was conducted on this airframe using XFLR5 [1].

**Table 1:** Key Airframe Statistics

|                  |      |
|------------------|------|
| Length (m)       | 2.05 |
| Wingspan (m)     | 4.20 |
| Height (m)       | 0.52 |
| Wing Area (m)    | 1.05 |
| Aspect Ratio     | 16.8 |
| MTOW (kg)        | 7.2  |
| Capacity (KWh)   | 0.3  |
| Thrust-to-Weight | 0.21 |



**Figure 1:** CAD Rendering of the Aircraft, with Spar Geometry Exposed

## 2 Aeroelastic Analysis

### 2.1 Modeling Parameters, Discretization, and Numerics

The inertial and structural properties of each of the surfaces is estimated. The process of doing so draws upon the CAD renderings of the aircraft, which yield good estimates of the geometric sectional parameters, along with estimates on component masses and moments of inertia.

**Table 2:** Estimated Structural Properties of Aircraft Sections

| Surface  | Mass (kg) | Span (m) | GA ( $\text{Nm}^{-1}$ ) | EA ( $\text{Nm}^{-1}$ ) |
|----------|-----------|----------|-------------------------|-------------------------|
| Wing     | 3.007     | 4.20     | $6.50 \times 10^6$      | $9.10 \times 10^6$      |
| Fuselage | 0.652     | 2.05     | $5.89 \times 10^6$      | $8.24 \times 10^6$      |
| Tail     | 0.325     | 1.00     | $1.92 \times 10^6$      | $2.69 \times 10^6$      |
| Fin      | 0.310     | 0.40     | $1.92 \times 10^6$      | $2.69 \times 10^6$      |

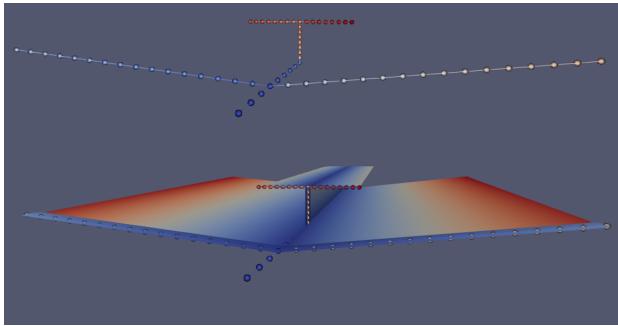
| Surface  | GJ ( $\text{Nm}^2$ ) | EI <sub>y</sub> ( $\text{Nm}^2$ ) | EI <sub>z</sub> ( $\text{Nm}^2$ ) | M ( $\text{kgm}^{-1}$ ) | J ( $\text{kgm}$ )    | Elastic Axis |
|----------|----------------------|-----------------------------------|-----------------------------------|-------------------------|-----------------------|--------------|
| Wing     | $9.86 \times 10^4$   | $7.57 \times 10^2$                | $1.37 \times 10^5$                | 0.716                   | $6.94 \times 10^{-3}$ | 0.229        |
| Fuselage | $6.70 \times 10^2$   | $4.69 \times 10^2$                | $4.69 \times 10^2$                | 0.318                   | $3.62 \times 10^{-5}$ | –            |
| Tail     | $3.64 \times 10^4$   | $2.30 \times 10^2$                | $5.09 \times 10^4$                | 0.325                   | $9.82 \times 10^{-4}$ | 0.420        |
| Fin      | $4.53 \times 10^4$   | $2.30 \times 10^2$                | $6.34 \times 10^4$                | 0.775                   | $1.29 \times 10^{-3}$ | 0.420        |

In order to accurately capture the mass distribution of the fuselage, many of the nose components (including the avionics, powertrain, and fuselage fairings) were modeled as point masses. Considering that these components are near-symmetric, lightweight, and seldom extend a significant distance from the vehicle’s centerline, their moments of inertia about their centers can be neglected. A list of the applied point masses is available in appendix A.

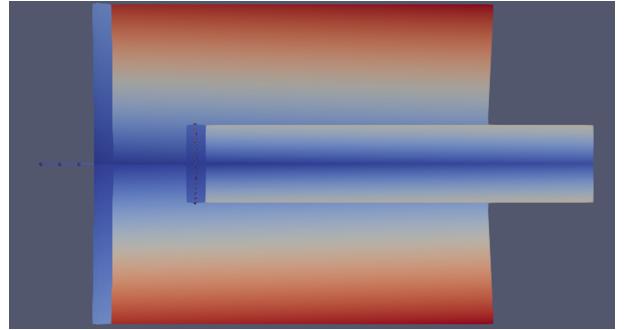
The discretization of the model’s underlying beam structures draws heavily from SHARPY’s Simple-HALE tutorial case; the element type and element quantities remain largely unchanged. Notably, the ICAV-SPI airframe has significant portion of it’s fuselage length located ahead of the wing’s leading edge. As such, the wings are affixed to the fuselage node corresponding most closely to the true offset. This entails specifying a fuselage node spacing that best captures the true offset of the wings. Moreover, this process involved altering SHARPY’s connectivity matrix to represent the new structural and aerodynamic boundary conditions at the wing root. As for the aerodynamic surfaces, the wings were given a camber line corresponding to that of NACA 4415 and the tail surfaces remained flat. Control surfaces were defined, matching the dimensions of those outlined during the CAD process.

**Table 3:** SHARPY Structural Discretization Parameters

|                           |   |
|---------------------------|---|
| Nodes per Beam Element    | 3 |
| Half Span Wing Elements   | 8 |
| Fuselage Elements         | 4 |
| Horizontal Stab. Elements | 8 |
| Vertical Sta. Elements    | 4 |

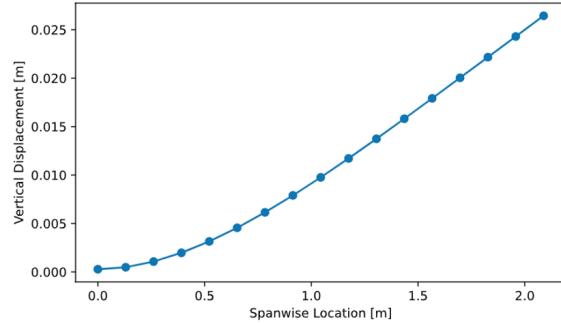


(a) Isometric Views, Surfaces and Beam Geometry



(b) Top View

**Figure 2:** SHARPy Discretization



**Figure 3:** Static Deflection of the Wing Loaded to S.L.F. Conditions

## 2.2 Static Analysis

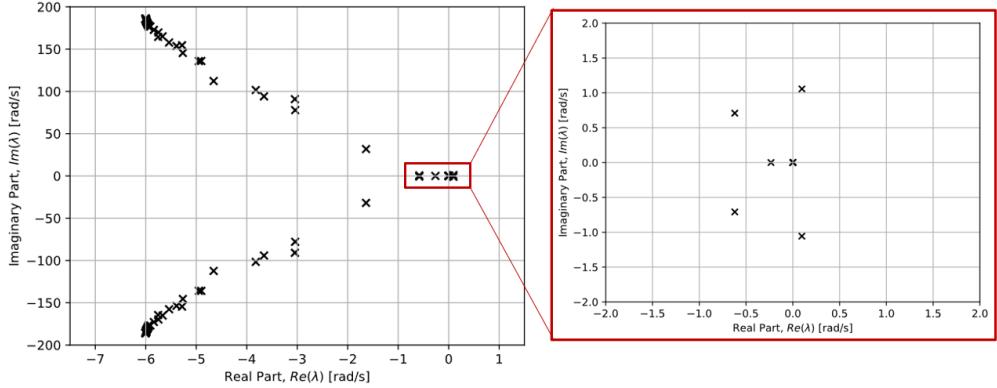
In order to validate the model, it remains to evaluate the wingtip deflection associated with steady level flight at the intended cruise speed of 15 m/s. In order to obtain the loading conditions corresponding to steady level flight, the aircraft was trimmed. The obtained wingtip deflection was 26.1 mm, or roughly 10.57 % of the reference chord. This figure was then compared to the analytical deflection of a cantilever beam under uniform loading, which stood at 34.2 mm, serving as an upper bound to the realm of reasonable estimates.

## 2.3 Linearized Dynamics

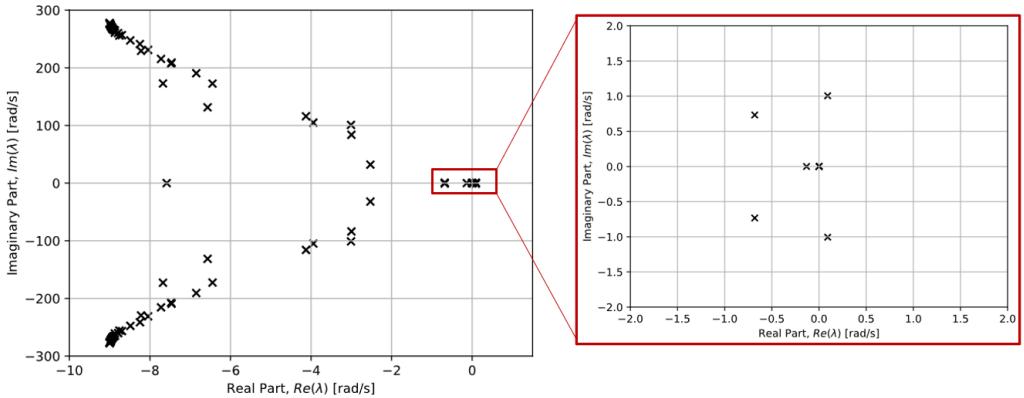
The aircraft was linearized about a trimmed equilibrium at 15 m/s. Recovering a linear state-space model of the aircraft involved configuring SHARPy's *Modal* and *LinearAssembler* solvers. An exact account of the settings is available in appendix B.

Using these settings, the system's eigenvalues can be analyzed; specifically, two key comparisons can be drawn. First, the aircraft's rigid body modes can be compared to the results obtained using XFLR5. Second, the distribution of the eigenvalues on the root locus diagram can be examined in the context of how the model was discretized. The number of chord wise vortex ring elements used evidently has a profound impact on the specifics of the assembled linear models.

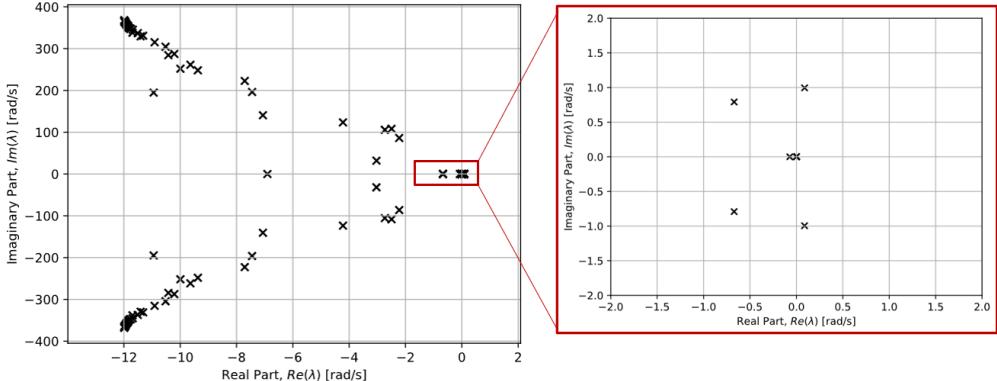
The rigid body modes remain largely unchanged, however, the eigenvalue corresponding to the spiral mode progresses towards the positive real direction. The eigenvalues associated with the vehicle's structural dynamics retain the same structure, presenting as heavily damped, often at high natural frequencies. The eigenvalues associated with the UVLM's aerodynamic modes differ quite significantly as the number of chord-wise panels increases, and understandably so. The aerodynamics-dominated eigenvalues with the largest real components (representative of the most significant behaviors) are present at roughly the same location for all three trials. These are clustered near -2 on the real axis.



(a) Root locus Diagram, 4 Chordwise Panels



(b) Root locus Diagram, 6 Chordwise Panels



(c) Root locus Diagram, 8 Chordwise Panels

Figure 4: A Comparison of Linearized Models

The eigenvalues corresponding from the rigid body modes produced with 8 chord-wise panels can be compared to the results obtained from XFLR5. By matching the mode shapes to the relevant flight dynamics mode, the open loop dynamics of the aircraft are observed to be largely consistent (in the frequency domain) between either method, but with occasional discrepancies. While this isn't a like-for-like comparison, it serves as a sensibility check for the vehicle's rigid body modes.

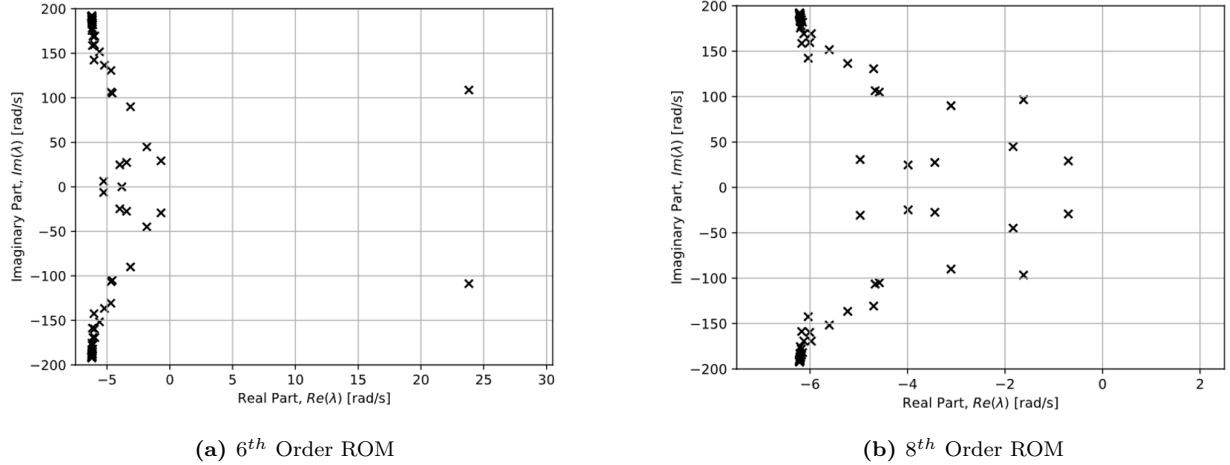
**Table 4:** Comparing the Rigid Body Modes Between VLM-Based Methods

| Dynamic Mode    | XFLR                | SHARPy Linearization |
|-----------------|---------------------|----------------------|
| Phugoid         | $-0.006 \pm 0.821i$ | $0.983 \pm 1.002i$   |
| SPPO            | $-7.118 \pm 6.303i$ | $-3.068 \pm 16.604i$ |
| Dutch Roll      | $-0.470 \pm 2.402i$ | $-0.718 \pm 0.822i$  |
| Spiral          | 0.579               | -0.083               |
| Roll Subsidence | -11.190             | -7.654               |

## 2.4 Reduced Order Models and Velocity Analysis

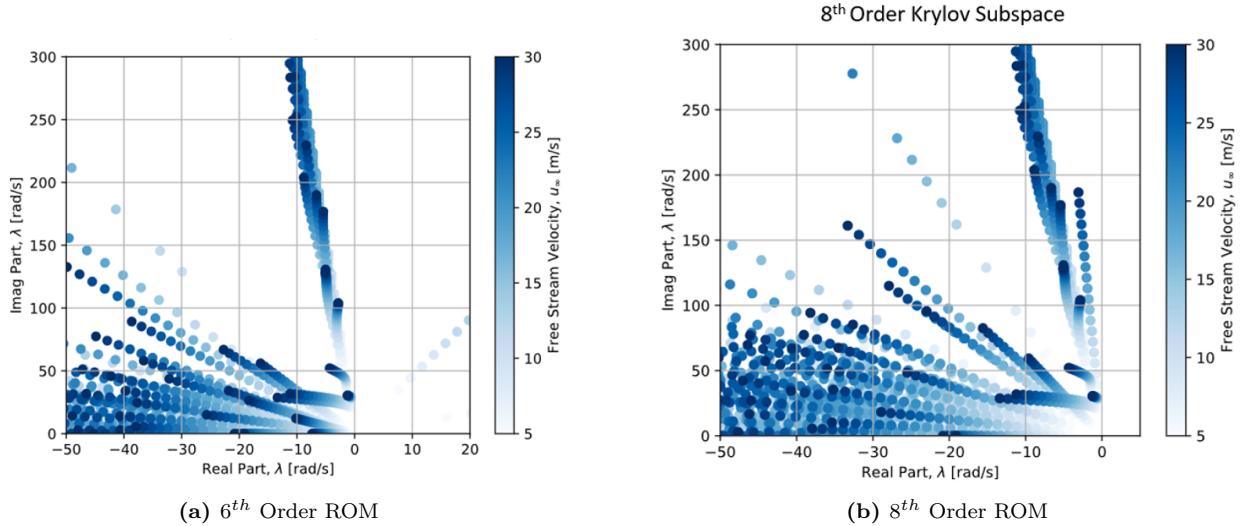
Reduced order models were obtained in the form of usitlizing modal projection alongside 6<sup>th</sup> and 8<sup>th</sup> order Krylov subspace methods. The specific settings used by SHARPy's solvers are evidenced in appendix C.

The root locus plots displayed in figure 5 display the eigenvalues of the two ROMs, plotted with the rigid body modes neglected. Figure 5a illustrates a key feature of Arnoldi iteration; the reduced order model isn't guaranteed to inherit the stability of the system. A pair of eigenvalues that possess a large positive real value are present. These eigenvalues aren't representative of any physical behavior, but are purely numerical artifacts of the Arnoldi iteration. Transitioning to an 8th order ROM addresses this issue, and the result closely resembles the previously displayed linearized examples.



**Figure 5:** Root Locus Diagrams, Krylov Subspace ROMs at 15 m/s

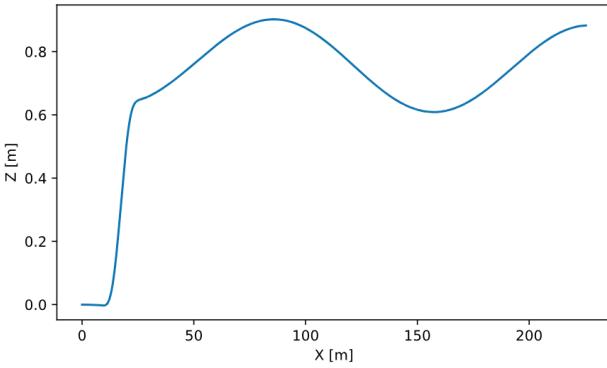
Following these results, a velocity analysis was conducted. The goal was to develop an improved understanding of the aircraft's flight envelope; any unstable aeroelastic behavior was to be identified and characterized. As discussed previously, the 6<sup>th</sup> order model produces some (otherwise worrying) non-physical behavior. Fortunately, conducting the velocity analysis with the 8<sup>th</sup> order ROM reveals very little concerning behavior. By these estimates, flutter (or any other aeroelastic phenomena) does not pose a significant threat to the airframe until well beyond the expected regime of flight velocities, if at all.



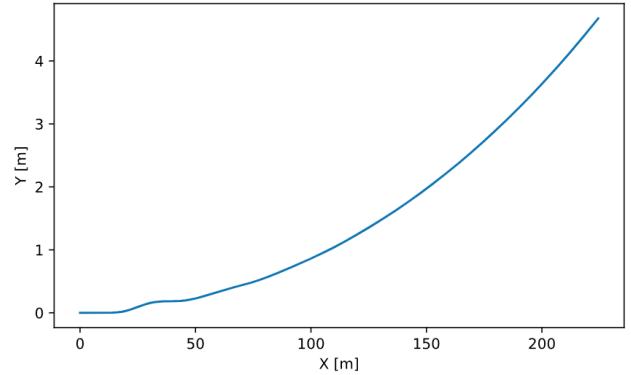
**Figure 6:** Velocity Analysis, Krylov Subspace ROMs

## 2.5 Dynamic Simulations and Gust Responses

It remains to evaluate the gust response of the aircraft using 10% intensity 1-cos gusts applied in the longitudinal and lateral directions. Ultimately, the structural loading at the wing root can be quantified. The trajectory of the aircraft in response to either gust gives key insights into the aircraft's dynamics; they capture nonlinearities not accounted for in the previous stability analysis.



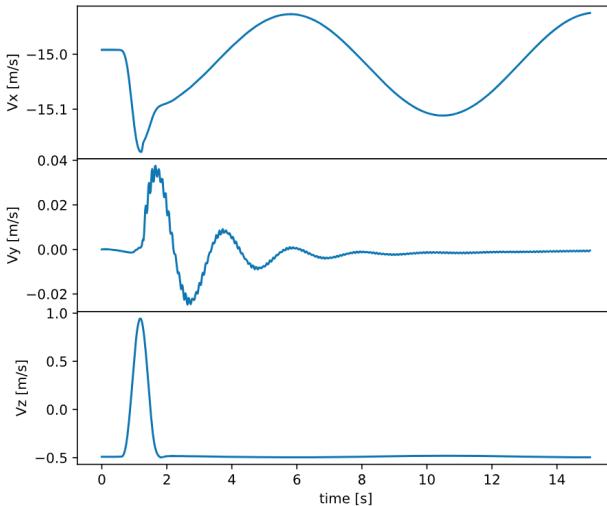
(a) Long. Trajectory, Long. Gust



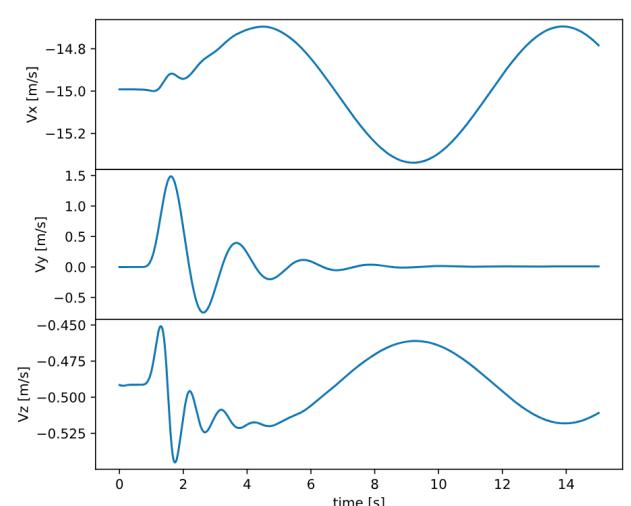
(b) Lat. Trajectory, Lat. Gust

**Figure 7:** BFOR Flight Path Trajectories, 10% 1-cos Gust Response

The vehicle's velocities are largely consistent with the expected response to the perturbation. Notably, the lateral gust also excites a slight phugoid motion, evidencing the truly coupled nature of the solvers at play.



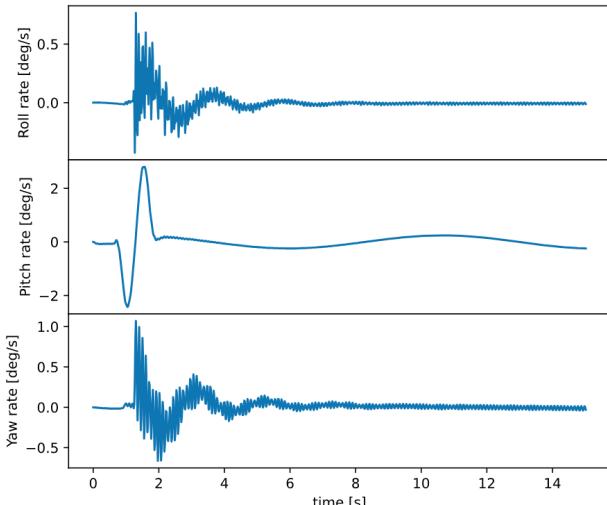
(a) Longitudinal Gust Response



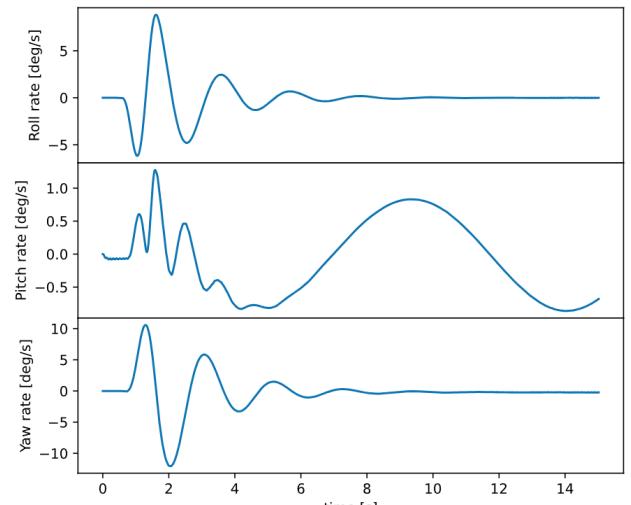
(b) Lateral Gust Response

**Figure 8:** Rigid Body Velocities, 10% 1-cos Gust Response

The angular velocities testify to much of the same information, matching the conventional expectations for an aircraft's rigid body modes.

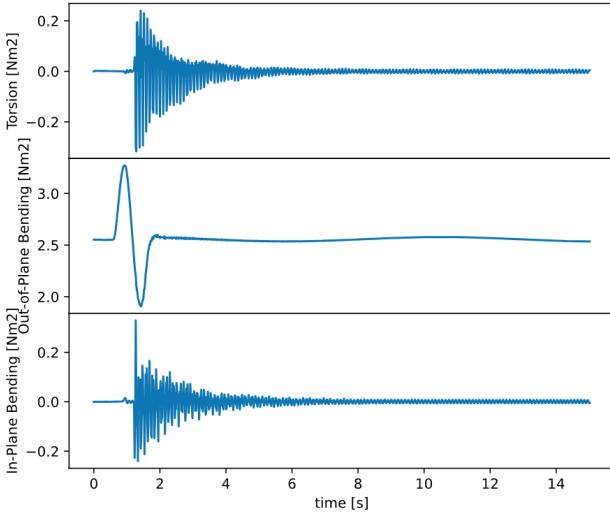


(a) Angular Rates, Long. Gust

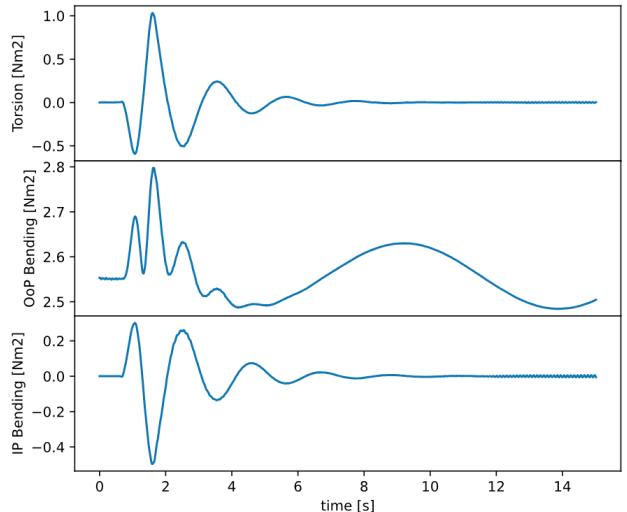


(b) Angular Rates, Lat. Gust

**Figure 9:** Rigid Body Angular Rates, 10% 1-cos Gust Response

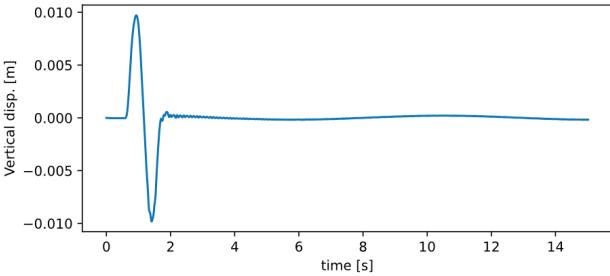


(a) Loading, Long. Gust

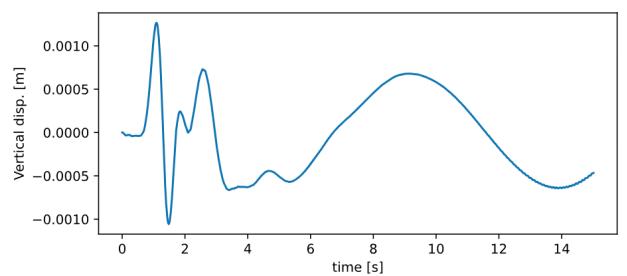


(b) Loading, Lat. Gust

**Figure 10:** Wing Root Loading, 10% 1-cos Gust Response



(a) Wingtip Deflection, Long. Gust



(b) Wingtip Deflection, Lat. Gust

**Figure 11:** Wingtip Deflection, 10% 1-cos Gust Response

### 3 Applied Control Law and SITL Testing

With the open-loop dynamics of the aircraft characterized, it remains to tune the autopilot; in this case ArduPilot [2] is the specific system intended to be used. With the intention of being largely model independent, ArduPilot is highly robust. It makes use of a total energy control system (TECS) [3], an intuitive nonlinear controller to manage the longitudinal behavior of the aircraft. TECS generates a reference climb gradient, along with a throttle command.

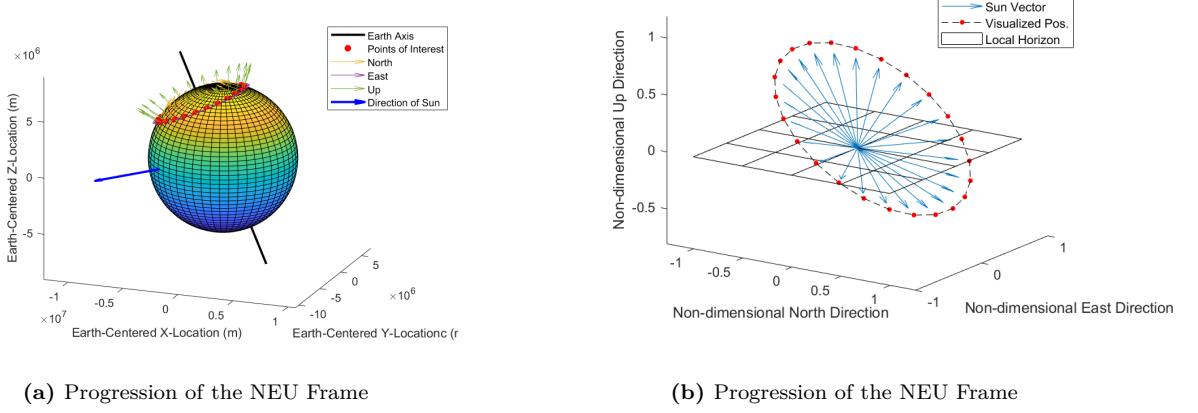
Moreover, Autopilot makes use of L1 Navigation [4] to navigate waypoint missions, this system commands a lateral acceleration based on the aircraft's cross-track error and a specified 'look-forward' distance. Using several basic approximations, the acceleration command generates a reference bank angle. The reference bank angle and climb gradients are achieved by a set of inner-loop roll and pitch PID controllers. Rudder inputs are given by a PID controller based turn coordinator.

Utilizing the reduced order linear state-space model of aircraft in conjunction with a dedicated ArduPilot software-in-the-loop (SITL) connector for Simulink [5] allows for the controller's gains to be tuned. While immensely useful for flight testing preparation process, the available SITL connectors come with several flaws. Importantly, they require significant user involvement during the initialization phase and do not lend themselves to being initiated from a MATLAB script. Owing to the simplicity of the control law and the abundance of documentation, ArduPilot's functionality can be completely reconstructed as a Simulink block diagram. This complete Simulink file can be used for the sake of trajectory planning; it is conducive to various optimization methods which rely upon utilizing this simulation to evaluate the performance of the aircraft with regards to a specific set of input data.

### 4 Solar Income Modeling

While meteorological data on solar intensity is widely available, the quantity of energy harnessed by the aircraft is highly dependent on several other factors. Most notably, at certain attitudes the geometry of the aircraft has a significant role to play. With T-tail aircraft in particular, the horizontal and vertical stabilizers often cast shadows over the main wing, potentially obstructing a significant portion of the available solar cells.

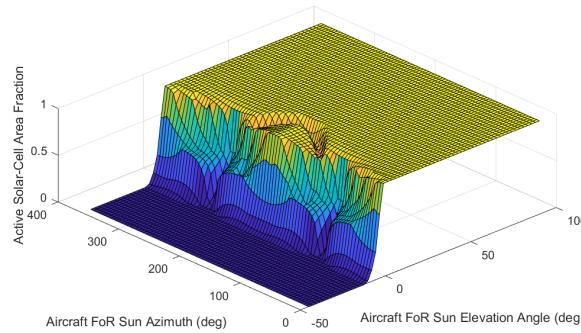
Modeling this phenomenon relies on two things: an accurate estimate of the sun's relative location in the sky, and a means of casting shadows between the vehicle's surfaces. The former is simple to address, and was tackled using a quaternion-based approach to track the sun's motion in the local NEU frame (shown below in figure 12), relying on well established astronomical data [6]. The direction of the Sun vector can then be resolved into the body frame of reference, based on the Euler angles of the aircraft (which are reflected directly in the assembled state space model).



**Figure 12:** Day-Long Motion of the Sun at a Latitude of 52, Mid-Aug. 2025

The development of the shadow casting system was tackled by discretizing the aircraft's surfaces into quadrilateral elements, with the solar cells being represented similarly. The program iterates through the individual solar cells, identifying any objects residing upstream of the cell's tangent plane. The program then projects the vertices of said surfaces onto this plane. The information regarding these projected polygons is stored in the form of a MATLAB Poly-shape object, allowing for the area of intersection to be identified. The end result is an array of values representing the fraction of obstructed area for each solar cell; these results may be processes with knowledge of diode placement [7] to represent the vehicle's true solar income.

The shadow casting system is used to populate a set of lookup tables that represent the fraction of solar energy extracted at various azimuth and elevation angles, with the act of interpolation occurring while running the primary Simulink file. This is suitable for rigid aircraft, as it avoids calling the shadow casting system unnecessarily. In the case where significant deflections are expected, the cast shadows can be evaluated at every timestep. Figure 13 represents the output data produced by this system; a series of energy reductions occur when either the H-stab, V-stab or opposing wing cast shadows over especially large segments of the solar cell arrays. The most prominent feature of this graph is the large dip in the center, this corresponds to the sun being directly behind the aircraft, with the H-stab casting its largest possible shadow over the wing.



**Figure 13:** Unobstructed Area Fraction Using Sun Location in the Body Frame of Reference

## 5 Optimal Waypoint Mission Generation

In an effort to prolong the aircraft's endurance, optimal waypoint placement must be considered. However, the act of trajectory planning is heavily intertwined with mission specific objectives. As such, two path planning strategies were implemented, one oriented towards point-to-point navigator and the other oriented toward prolonged loitering. These two path planners will be applied in conjunction with each other to fit a wide variety of mission profiles.

Meteorological information provided from the OpenMeteo API [8] proved vital to successfully implement the trajectory planning techniques. The core set of data provided consists of wind speeds and directions at altitudes of 10, 80, and 120 meters above the surface. The availability of wind forecasts at multiple altitudes within the model-permitted airspace yield useful insights when it comes to optimizing the location of the waypoints in 3D space. This information can be supplemented with figures on cloud cover and precipitation, such that more risk-conscious methods can be tested.

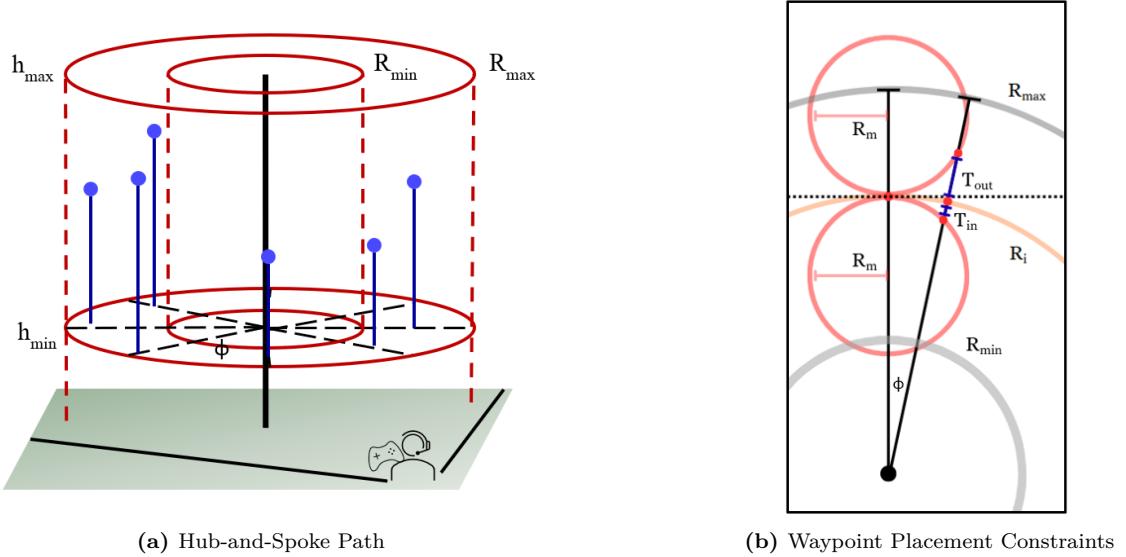
## 5.1 Trajectory Planning for Loitering Missions

The optimization problem takes the following form; the set of paths capable of being flown by the vehicle are determined by a hub-and-spoke track. The path can be fine-tuned by adjusting the radii and altitudes corresponding to the individual waypoints; thus, a path with 8 waypoints possesses 16 degrees of freedom. Because running the high fidelity Simulink model is a lengthy/computationally heavy process, Bayesian optimization is ideal for this application; tight control can be maintained over the number of function evaluations.

The loiter loop planner is of great utility for endurance record attempts; there are several practical constraints that make this the case. First, it is expected that aircraft will be stretching its limits, often operating near the lowest possible state-of-charge. Hence, the aircraft must remain within LOS at all times; this means that the waypoint loop can be contained by a simple cylindrical bounding box. This simplifies the box constraints being applied to the optimization variables. Second, the weather conditions for any potential record attempts will be well known; a flight window will be selected where near-ideal forecasts are observed.

This allows for the performance of the vehicle to be evaluated in terms of energy expenditures alone, unlike the point-to-point planner. Moreover, the flight will occur within a dedicated airspace (such as a model club). As long as the aircraft is situated within its bounding box, there will likely be very little need to about avoid obstacles.

Moreover, the waypoints can be constrained by imposing soft limits on the waypoint's radial placement. By using an imposed minimum maneuver radius, along with the location of the previous waypoint in the sequence, an upper and lower bound can be placed on the radial location of the next waypoint. Implementing these bounds implicitly makes the assumption that the aircraft always approaches the current waypoint tangentially. While this is often the case, these bounds can be made more relaxed or more conservative to suit the level of attainability desired from the path itself. Since ArduPilot's implementation of L1 relies on the vehicle lying within a specified waypoint radius to consider the waypoint 'reached', imperfect application of the radial waypoint location constraints is compensated for by the robustness of ArduPilot's navigation system.



**Figure 14:** Hub-and-spokes Path Constraints

## 5.2 Trajectory Planning for Point-to-Point Missions

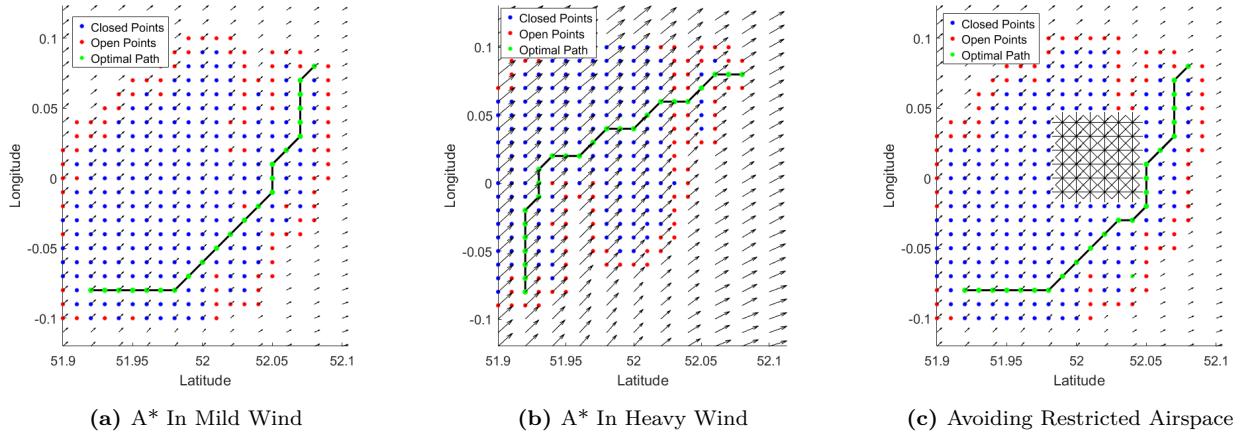
On the other hand, A\* is well suited for point-to-point navigation. Its implementation in this context relies on simplifications to the aircraft's dynamics and solar income model; this involves evaluating the aircraft's performance under the assumption of static equilibrium. The end result is effective at preserving useful information, such as the impact of the shadow cast by the tail over the wing, or the amount of throttle needed to maintain the vehicle's airspeed.

$$\dot{\mathbf{x}} = \mathbf{A}[\mathbf{x} + \mathbf{x}'] + \mathbf{B}\mathbf{u} \Rightarrow \mathbf{u}_{se} = -\mathbf{B}^+(\mathbf{A}[\mathbf{x}_{se} + \mathbf{x}'_{ta}]) \quad (1)$$

This A\* based approach allows for high degrees of flexibility when formulating the relative cost functions. Ultimately, the vehicle can be incentivized to stray clear of cloud cover, precipitation, or exceptionally gusty patches of wind. Moreover, using an obstacle bitmap, high risk or restricted airspace may be avoided entirely. This allows the aircraft's missions to be conducted within the limitations of UK SORA Authorizations (Specific Operations Risk Assessment) [9]. For instance, areas with a population density greater than 50 people per square kilometer must be avoided to remain in compliance.

However, this method comes with one primary drawback; A\* is only optimal to the resolution of the grid. The generated path often consists of sharper-than-necessary corners. While this issue is present at all grid sizes, it is particularly bad at either extreme. It is possible to address this issue through curve fitting techniques, yet this remains to be implemented.

As a whole, this algorithm is most efficient for planning shorter 'stints' between high importance control points. For instance, long distance journeys, eg. London to Edinburgh necessitate navigating through an elaborate maze of operational and airspace restrictions. Instead of running the algorithm from the starting location to the end, the mission can be subdivided further. Below is a procedure that details how this algorithm may be deployed in practice.



**Figure 15:** Evaluating the Capabilities of A\* Based Path Planning over 20 km Missions

1. Select key control points along the mission's path. These should be spaced approximately 20-60 kilometers apart. These may coincide with important geographic features.
2. Utilize the A\* point-to-point planner to identify the best path between the first two control points at the time of launch.
3. Use the full fidelity Simulink model to evaluate the aircraft's performance and to monitor its time of arrival. Record the relevant performance statistics.
4. Re-run the point-to-point planner, this time using the update time of day based on arrival time estimate provided by Simulink model.
5. Repeatedly iterate through the remaining control points until the destination is reached or until the mission is declared infeasible.

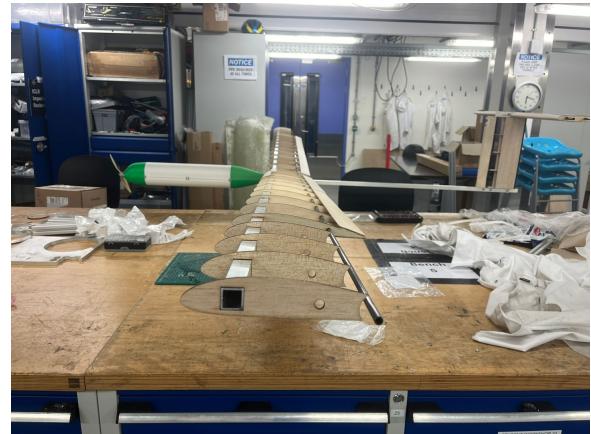
## 6 Conclusion and State of Completion

In summary, the analysis and methods implemented throughout the course of the UROP have created the foundation needed to comprehensively predict and manipulate the performance of the aircraft. Construction of the airframe is approaching completion, the following figures illustrate the extent to which the vehicle was assembled as of late August, 2025. It remains to drape the aircraft's film skin over the balsa wood structure. It is estimated that the maiden flight will occur in late September or early October, when the weather conditions are deemed suitable.

While the project set out to achieve many goals, it fell short when it came to verifying the aircraft's structural dynamics through ground vibration testing. While an experimental setup was devised and the corresponding risk assessments were secured, the procurement of a data acquisition system proved to be exceedingly difficult during the time span of the UROP. However, given that the risk assessments were approved, this option remains high on the list of priorities for next steps, promising to bring important insight toward validating the SHARPy model.



(a) Frontal View



(b) Left Side View



(c) Aft-Left View



(d) Aft-Right View

**Figure 16:** State of Completion as of Mid. Aug. 2025

# Bibliography

- [1] XFLR5 project, Apr 2025. URL <https://sourceforge.net/projects/xflr5/>.
- [2] Jordi Munoz and Chris Anderson. Ardupilot. Dec 2019. URL <https://ardupilot.org/ardupilot/>.
- [3] Kevin R. Bruce. NASA B737 flight test results of the total energy control system (NASA-CR-7782E5), Jan 1987.
- [4] Sanghyuk Park, John Deyst, and Jonathan P. How. A new nonlinear guidance logic for trajectory tracking, Sep 2004.
- [5] Kiril Boychev. Ardupilot software-in-the-loop (SIL) connector for simulink, Mar 2025. URL <https://uk.mathworks.com/matlabcentral/fileexchange/114315-ardupilot-software-in-the-loop-sil-connector-for-simulink>.
- [6] Cerese Albers. NASA planetary data system. May 2025.
- [7] R. Seigwart and P. Oettershagen. High fidelity solar power income modeling for solar-electric aircraft, Sep 2015.
- [8] Openmeteo API tools, Jan 2022. URL <https://open-meteo.com/>.
- [9] Civil Aviation Authority. UK regulation (EU) 2019/947 on rules and procedures for the operation of unmanned aircraft, Jun 2019. URL [https://regulatorylibrary.caa.co.uk/2019-947/Content/map/00140\\_art.\\_11\\_Rules\\_for\\_conducting\\_an\\_operational\\_risk\\_assessment.htm](https://regulatorylibrary.caa.co.uk/2019-947/Content/map/00140_art._11_Rules_for_conducting_an_operational_risk_assessment.htm).

## Appendix A: Applied Point Masses

**Table 5:** Component mass and CG locations from nose datum BFoR

| Part              | $CG_x$ (mm) | $CG_y$ (mm) | $CG_z$ (mm) | Mass (kg) | Type      |
|-------------------|-------------|-------------|-------------|-----------|-----------|
| Cowling           | 60          | 0           | -8          | 0.110     | fuselage  |
| Motor             | 36          | 0           | 0           | 0.343     | fuselage  |
| Motor mount       | 80          | 0           | 1           | 0.051     | fuselage  |
| Brace 1           | 115         | 0           | 0           | 0.046     | fuselage  |
| Avionics panels   | 225         | 0           | 0           | 0.012     | fuselage  |
| Misc. Avionics    | 225         | 0           | 0           | 0.150     | fuselage  |
| Fuselage panels 1 | 225         | 0           | 3           | 0.180     | fuselage  |
| Brace 2           | 335         | 0           | 0           | 0.046     | fuselage  |
| Battery cluster 1 | 390         | 0           | 0           | 0.436     | fuselage  |
| Battery cluster 2 | 485         | 0           | 0           | 0.436     | fuselage  |
| Fuselage panels 2 | 445         | 0           | 3           | 0.180     | fuselage  |
| Brace 3           | 554         | 0           | 0           | 0.045     | fuselage  |
| Aft panels        | 630         | 0           | -10         | 0.120     | fuselage  |
| Wing mounts 1     | 790         | 0           | -17         | 0.200     | wing root |
| Wing mounts 2     | 910         | 0           | -21         | 0.050     | wing root |
| Tail mounts       | 2030        | 0           | 0           | 0.060     | tail      |

## Appendix B: SHARPy Linear-Assembler Settings

```
settings['Modal'] = {'print_info': True,
                     'use_undamped_modes': True,
                     'NumLambda': 30,
                     'rigid_body_modes': True,
                     'write_modes_vtk': 'on',
                     'print_matrices': 'on',
                     'continuous_eigenvalues': 'off',
                     'dt': dt,
                     'plot_eigenvalues': False}
```

Figure 17: Modal Solver Settings

```
settings['LinearAssembler'] = {'linear_system': 'LinearAeroelastic',
                               'linear_system_settings': {
                                   'beam_settings': {'modal_projection': False,
                                                     'inout_coords': 'modes',
                                                     'discrete_time': True,
                                                     'newmark_damp': 0.05,
                                                     'discr_method': 'newmark',
                                                     'dt': dt,
                                                     'proj_modes': 'undamped',
                                                     'use_euler': 'on',
                                                     'num_modes': 40,
                                                     'print_info': 'on',
                                                     'gravity': 'on',
                                                     'remove_dofs': []},
                                   'aero_settings': {'dt': dt,
                                     'integr_order': 2,
                                     'density': rho,
                                     'remove_predictor': False,
                                     'use_sparse': False,
                                     'remove_inputs': ['u_gust']},
                                   'track_body': 'on'
                               }}
```

Figure 18: LinearAssembler Solver Settings

## Appendix C: SHARPy ROM Settings

```
settings['Modal'] = {'print_info': True,
                     'use_undamped_modes': True,
                     'NumLambda': 30,
                     'rigid_body_modes': True,
                     'write_modes_vtk': 'on',
                     'print_matrices': 'on',
                     'continuous_eigenvalues': 'off',
                     'dt': dt,
                     'plot_eigenvalues': False}
```

**Figure 19:** Krylov ROM settings

```
settings['Modal'] = {'print_info': True,
                     'use_undamped_modes': True,
                     'NumLambda': 30,
                     'rigid_body_modes': False,
                     'write_modes_vtk': 'on',
                     'print_matrices': 'on',
                     'continuous_eigenvalues': 'off',
                     'dt': 0,
                     'plot_eigenvalues': False}
```

**Figure 20:** Modal Solver Settings

```
settings['LinearAssembler'] = {'linear_system': 'LinearAeroelastic',
                               'linear_system_settings': {
                                   'beam_settings': {'modal_projection': True,
                                                     'inout_coords': 'modes',
                                                     'discrete_time': True,
                                                     'newmark_damp': 0.05,
                                                     'discr_method': 'newmark',
                                                     'dt': dt,
                                                     'proj_modes': 'undamped',
                                                     'use_euler': 'off',
                                                     'num_modes': 40,
                                                     'print_info': 'on',
                                                     'gravity': 'on',
                                                     'remove_dofs': []},
                                   'aero_settings': {'dt': dt,
                                                     'scalingDict': {'length': 0.5 * chord_ref,
                                                                    'speed': u_inf,
                                                                    'density': rho},
                                                     'integ_order': 2,
                                                     'density': rho,
                                                     'remove_predictor': True,
                                                     'use_sparse': True,
                                                     'remove_inputs': ['u_gust'],
                                                     'rom_method': 'Krylov',
                                                     'rom_method_settings': {'Krylov': rom_settings}},
                                   'track_body': 'on'
                               }}
```

**Figure 21:** LinearAssembler Solver Settings

```
settings['AsymptoticStability'] = {'print_info': 'on',
                                    'modes_to_plot': [],
                                    'display_root_locus': 'off',
                                    'frequency_cutoff': 0,
                                    'export_eigenvalues': 'on',
                                    'velocity_analysis': [5, 30, 20],
                                    }
```

**Figure 22:** Velocity Analysis Settings