CAL STATE LA
CALIFORNIA STATE UNIVERSITY, LOS ANGELES

College of Engineering, Computer Science, and Technology
Capstone Senior Design

July 9, 2024

Re: Certificate of Award for the IIT-Bombay UAS Team:

- ROHAN SURESH MEKALA
- JASKARAN SINGH
- ROHAN BADGUJAR
- RAHUL AGARWAL

On behalf of the California Unmanned Aerial Systems Competition (C-UASC) held June 22, 2024:

It is my pleasure to certify that the team from IIT-Bombay earned the highest score for our Design Innovation Competition and for their accompanying simulation. They are awarded:

FIRST PLACE: Design Innovation

for their well-designed VTOL (Vertical Takeoff and Landing) UAS! CONGRATULATIONS!

Michael Thorburn, Ph.D.

Michael Thorburn

Organizer, C-UASC at Mojave Air & Space Port, Mojave California, USA
AircraftCompetitions@calstatela.edu
https://www.calstatela.edu/ecst/uav-competitions

(m) 1-562-506-3021

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY, INDIA

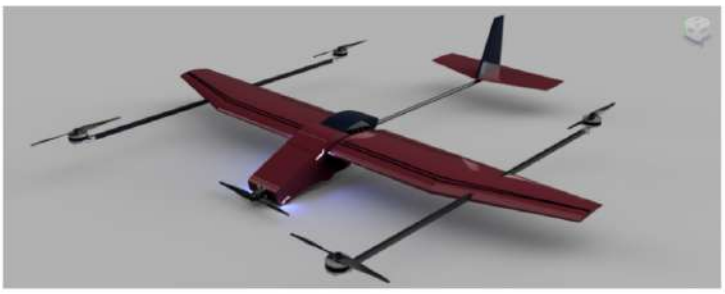
UMIC-AEROVE

Unmesh Mashruwala Innovation Cell

California Unmanned Aerial Systems Competition (CUASC)

Team members

- Jaskaran Singh (2nd year Mechanical)
- Rohan Mekala (2nd year Mechanical)
- Rohan Badgujar (2nd year Mechanical)
- Arnav Dutt Sharma (2nd year Chemical)
- Rahul Agarwal (2nd year Electrical)
- Rohan Joshi (1st year Mechanical)
- Aagam Kasaliwal (1st year Mechanical)
- Jeet Gurbani (1st year Aerospace)
- Prashant Vashisht (1st year Mechanical)
- Ashish Ranjan (1st year Mechanical)



PegasusX432

Contents

1 DESIGN ANALYSIS

1.1 ANALYSIS

The 4+1 configuration was chosen as opposed to the tilt-rotor configuration for our VTOL aircraft due to the following reasons:

- Simplicity and Reliability: A 4+1 configuration typically involves fewer moving parts than tilt rotors. Simplicity often translates to reliability, reducing the chances of mechanical failures. Fewer moving parts also mean lower maintenance requirements and costs over the aircraft's lifespan.
- Weight Efficiency: Tilt rotors require complex mechanisms to transition between vertical and horizontal flight modes, including swiveling nacelles and rotor systems. These mechanisms add weight to the aircraft, reducing its overall efficiency. In contrast, a 4+1 configuration can be lighter and more weight-efficient, enhancing performance and payload capacity.
- Aerodynamic Efficiency: Tilt rotors often compromise aerodynamic efficiency due to the need for rotor systems optimized for vertical and horizontal flight. In contrast, the fixed engines used in a 4+1 configuration can be optimized specifically for their intended roles, maximizing efficiency in both flight modes. This optimization can lead to improved battery efficiency and range.
- Flight Control Complexity: Tilt rotor aircraft require sophisticated flight control systems to manage vertical and horizontal flight transitions and maintain stability and control throughout the flight envelope. In comparison, a 4+1 configuration simplifies flight control, as each set of motors can be controlled independently for vertical or horizontal flight, reducing complexity and potential points of failure.

It was decided to proceed with the conventional tail setup not to increase the aircraft system's complexities further. To decide upon the most optimum airfoils for the main wing and the horizontal and vertical tails, the performance and stability analysis was done in XFLR5 for various flight parameters. Using historical data, it was possible to make the initial guesses for some geometrical parameters such as wingspan and Aspect Ratio. With the help of these initial estimates of the flight parameters, various prototypes could be developed with varying wingspans, control surface sizing, etc. By performing preliminary flight tests (such as glide and dive tests) using these various prototypes, a fixed set of parameter values was decided for our first design iteration, VTOL version 1.0. With the first iteration fully defined, recursive weight iterations were started in order to converge to a desirable total weight for the aircraft.

$$W_0 = \frac{W_{\text{payload}} + W_{\text{battery}} + W_{\text{motor}}}{1 - \frac{W_{\text{st}}}{W_0} - \frac{W_{\text{pr}}}{W_0}} \tag{1}$$

The battery weight would then be computed using the following formula,

$$W_{\text{battery}} = \frac{W_0 \times V \times t}{\left(\frac{L}{D}\right) \times \text{SED}} \tag{2}$$

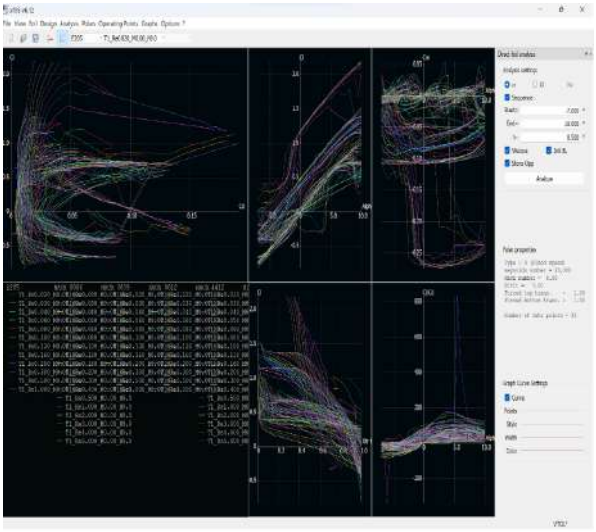
However, a few modifications were made to this formula. In place of the propulsive weight ratio, the weight due to the motors was added, as well as other electrical components. On using the weight of the motors in terms of their ratio with the total weight, a trend of absurd values was observed that didn't correspond to any feasible values of total weight. To solve this problem, the weight of the motors was shifted to the numerator to map it as a function of the total weight. After a few iterations, once the desirable value for the total weight is converged upon a, it is used to compute the lift required. The underlying assumption is that the attempt is made to compute the flight parameters for a steady, level flight, meaning that the flight path angle would be approximately zero. The lift would be equal to the total weight. To add a factor of safety, the lift required was taken to be approximately 1.8 times the weight of the quadplane. Using this value of lift, type 2 analysis was performed in XFLR5, which corresponds to fixed lift, to ascertain the drag and other parameters of importance, such as the pitching moment coefficient cruise velocity, as well as to verify the stability prerequisites using the pitching moment vs. angle of attack plot so that we can proceed with the stability analysis of the various longitudinal and lateral flight modes assuming that these two modes are decoupled using the short perturbation model. Using the value of drag obtained from XFLR5 analysis, the thrust required can be computed using the steady-level approximation. It has been tried to maintain a margin of safety such that the thrust would be enough to sustain the aircraft's weight and ensure that the aircraft still possesses sufficient maneuverability. The computed thrust can then be used in the motor selection process by probing the thrust tables of various motors to narrow down to a motor with the required thrust at around 80 percentage throttle. Further, simulations were performed in XPLANE to estimate

the control surface sizing. It was attempted to map the controls of the VTOL aircraft to a model created in GAZEBO to accomplish the required mission. The CFD simulations validated the results procured from XFLR5, such as the geometrical flight parameters (wingspan, wing area, etc.) and the tail positioning (from the stability analysis). To take the analysis a step further, the CFD simulations were also able to model the effects of the fuselage and the quad frame, which integrates the quad motors into the VTOL system.

2 XFLR5

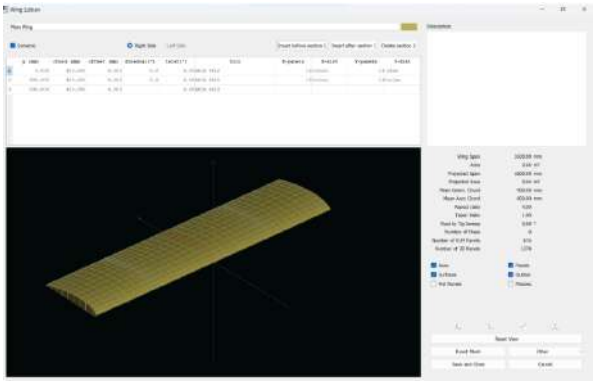
2.1 AEROFOIL SELECTION

Various prospective airfoils were analyzed for our VTOL design in XFLR5 software, which implements the Vortex Lattice Method and panels to interpolate the 2D results it calculates using a mathematical model into 3D analysis for airfoils. Multi-threaded batch analysis from -7 to 10 degrees with a change of 0.5 degrees was performed.



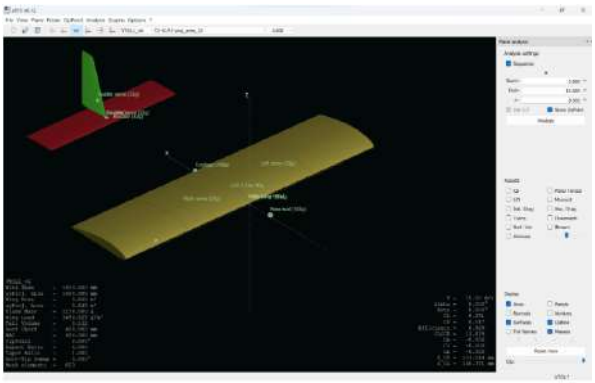
2.1

Transitional behavior in some of the foils was identified, which was investigated using Type 4 Analysis. It showed the variation of Cl and Cd with Re. For improved performance at lower Re, the forced transition points were changed. The analysis of separation bubbles showed that the least drag was observed for those angles of attack where the bubble was at the trailing edge. Flapped foil was performed from -6 to 10 degrees with a change of 0.25 degrees. On observing some irrational points in the polar graphs, those polars were identified and the particular operating point was located. The analysis was run with new AOA values while remembering to initialize the Boundary Layer as a new AOA range was being used. Inverse analysis has also been investigated to get modified airfoils (full/mixed inverse) by specifying the required plots. Direct wing and plane design analysis was performed from -7 to 10 degrees with a change of 0.5 degrees. XFLR5 interpolates this 2D data into 3D. Sometimes, this convergence fails because XFLR5 doesn't have the required 2D data to interpolate ("out of envelope" issue). The solution to this issue is explained later. In wing and plane design, all the respective sections were added and defined, and panels were applied properly at section boundaries (the number and type of panels are 13 and cosine, respectively). A z offset (of 10 mm) was given to the main wing, fin, and elevator to prevent division by 0. A further offset of 5 mm was given to the servos.



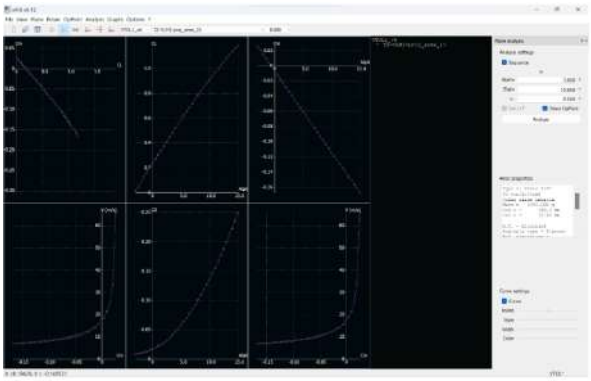
2.2

Additional point masses were added after defining mass distributions in the wing, elevator, and fin.



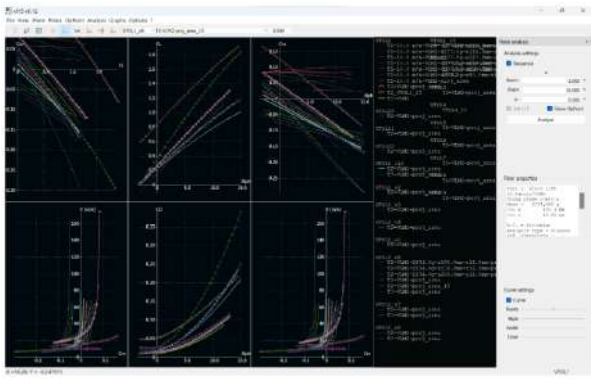
2.3

Performed Type 2 (cruise) analysis with VLM2 and used plane inertia and wing planform projected on the xy plane.



2.4

Performed analysis from -3 to 15 degrees with a change of 0.5 degrees. Verified log file for successful analysis. For flapped wings, two sections were inserted at the same location and applied panels properly (only 1 panel at the first of the two intersecting sections). To select the final airfoils for the wing, elevator, and fin, polar curves of all the possible combinations were plotted, and the airfoil combination was chosen with the required polar characteristics and L/D ratio.



2.5

For importing airfoils, the txt file from the UIUC airfoil database was downloaded, arranged in descending order using Excel, and converted to a dat file. Viscous analysis errors: XFLR5 uses 2D data from direct analysis and interpolates it into 3D in wing and plane design. To fix "out of the envelope" issues, Cl values that were out of the flight envelope were checked. Then, the direct analysis was revisited, and the AOA range was increased to get those Cl values. Whenever Re values were out of the envelope, direct analysis was performed to determine suitable Re. Direct analysis was performed with a sufficiently large AOA range as a good practice. After a certain point, this solution may stop working because the Vortex Lattice Method is linear, and hence, for it, Cl will increase linearly with AOA and will demand Cl values for interpolation, which is not possible to be to produce in the direct analysis even on enlarging the AOA range as stalling will take place before that Cl is reached. Stability prerequisites: Analysed Cm vs. AOA and Cm vs. Cl graphs. Both should have a negative slope, and Cl should be positive when Cm=0. Neutral Point identification: In the Cm vs. AOA graph, Type 1 Analysis was run with different CG positions till the NP was found (when Cm is constant). Static Margin = (Xnp - Xcg)/MAC The plane inertia was redefined using the static margin, and a Type 2 analysis was conducted. Then, the L/D ratio was determined, AOA was balanced, Cl was designed, and cruise velocity and other important parameters were determined from the various polar graphs.

Possible Airfoil Combinations

Symmetrical (for elevator and fin	Asymmetrical (for main wing)
NACA 0006	NACA 4412
NACA 0009	Eppler 205
NACA 0012	MH 114
	SD7037
	NACA 4415
	Eppler 182
	SA7038
	Clark Y
	S1223

2.6

Shortlisted Airfoil Combinations

1. NACA 0006 and NACA 4412
2. NACA 0006 and Eppler 205
3. NACA 0006 and SA7038
4. NACA 0009 and NACA 4412
5. NACA 0009 and Eppler 205
6. NACA 0009 and SA7038
7. NACA 0012 and NACA 4412
8. NACA 0012 and Eppler 205
9. NACA 0012 and SA7038
10. NACA 0006 and S1223
11. NACA 0009 and S1223
12. NACA 0012 and S1223

Selected Airfoil Combination

NACA 0006 and NACA 4412

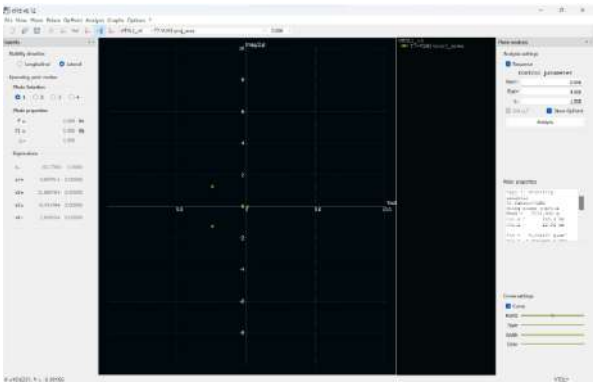
This airfoil combination was the most suitable for various parameters such as L/D ratio, balanced A.O.A, design Cl, and cruise velocity. The values of these parameters for this airfoil combination are mentioned below:-

- Cl max = 1.85
- Design Cl = 0.295
- Cruise velocity = 17.3 m/s
- Balanced A.O.A = -0.6
- L/D ratio = 13.879

2.2 OVERVIEW OF STABILITY ANALYSIS

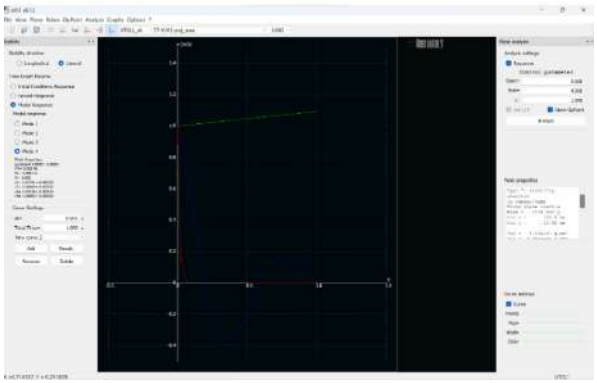
There are 2 Longitudinal modes: Short Period and Phugoid and 3 Lateral modes: Roll Damping, Dutch Roll, and Spiral

The stability analysis was conducted after verifying the stability prerequisites, first with zero gain. One point should appear in the polar graphs: the AOA at Cm=0 (checked from the log file). This point should lie in the Type-2 analysis, which was successfully observed. 3 ways of observing the stability are 3D view, root locus graph, and time response analysis. Root locus graph: More negative values mean more damping (as the real axis corresponds to the damping factor), and the imaginary axis corresponds to the frequency of oscillations. Positive values would mean undamped and unstable. Therefore, more values to the left correspond to more stability. Modes lying on the real axis are non-oscillatory. Airfoil doesn't have much influence on stability, but mass, inertia, and geometry do.



2.7

Time response analysis: various flight parameters vs. time were plotted. Longitudinal parameters: fluctuations in horizontal (u) and vertical (w) speed, pitch rate (q) and pitch angle (theta). Lateral parameters: lateral velocity variation (v), roll rate (p), yaw rate (r) and bank angle (phi). Modal response analysis was performed for various modes. The initial conditions and forced response analysis were studied. It was observed that the plots displayed the typical behaviors for their respective modes.



2.8

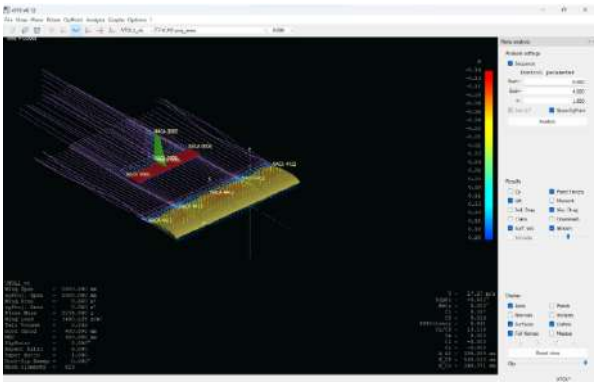
Stability mode notation of XFLR5:
Longitudinal:-
Mode 1: Short Period
Mode 2: Short Period
Mode 3: Phugoid
Mode 4: Phugoid Lateral:-

Mode 1: Roll Damping
Mode 2: Dutch Roll
Mode 3: Dutch Roll
Mode 4: Spiral Ideal characteristics of graphs for various modes:-

Short Period: high frequency, well-damped
Phugoid: slow, lightly damped, stable, or unstable
Roll Damping: high frequency, well-damped
Dutch Roll: faster and more damped than Phugoid
Spiral: non-oscillatory, slow, generally unstable

Sensitivity analysis: The influence of inertia was analyzed on stability by running a stability analysis, but this time by adding gain to the required inertia parameter. Sequence analysis was conducted, the log file was checked, and the eigenvalues (which correspond to points on the root locus graph) were compared to understand the influence of that inertia parameter on the damping of the various modes. The polar plots, as well as the root locus graphs, were also studied. Control Derivatives: The influence of various control parameters, such as flap deflection, on the stability

was observed. The motions that were being affected by that parameter and those that were not were verified.



2.9

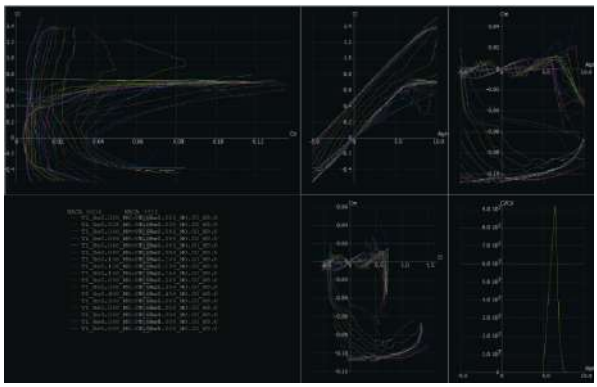
The idea behind performing stability analysis is to find optimal values for various flight parameters such as tail distance, control surface dimensions, etc., for which the various flight modes are stable to obtain minimum time response characteristics. The variations of the various aerodynamic and flight parameters in the context of the Dutch roll were primarily analyzed. The reasons for not including the other modes in our analysis are as follows:-

- Phugoid: This longitudinal stability mode, in general, is almost undamped/ slightly damped, and the design has already achieved a stable configuration in this regard.
- Short Period: This is a highly damped mode with a very short settling time, and therefore, the minuscule variations observed in the short period dynamics are not of analytical significance.
- Roll Damping: As is the case with the short period mode, the highly damped nature of the roll damping mode renders it unfeasible to perform comparative analysis for variations in flight parameters.
- Spiral: The observed spiral dynamics complied with the conventional flight dynamics norms. The spiral response was non-oscillatory and divergent. This introduces a tendency in the aircraft to sideslip away from its heading. However, a simple fix to this and other similar problems that involve introducing sideslipping in the aircraft, such as any unanticipated pressure gradient developing on the rudder due to the flow stream emerging from the fuselage, is to add a trim to the necessary control surfaces.

Dutch roll mode can be particularly sensitive to parameter changes like tail distance due to its inherent coupling between yaw and roll dynamics. Therefore, it provides clearer indications of how alterations in design features affect lateral stability compared to other modes.

2.3 X-FOIL DIRECT ANALYSIS

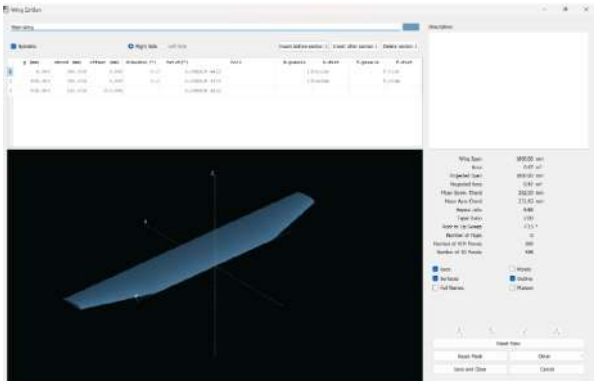
After finalizing the airfoil combination, a direct analysis of these airfoils was conducted. Plots of various aerodynamic plots, such as coefficient of lift (Cl), coefficient of drag (Cd), angle of attack (alpha), coefficient of pitching moment (Cm), and glide ratio (Cl/Cd) were generated for the selected airfoils: NACA 4412 and NACA 0006 using multi-threaded batch analysis for Reynold's number range from 20,000 to 5,000,000 and angle of attack range from -5 to 10 degrees.



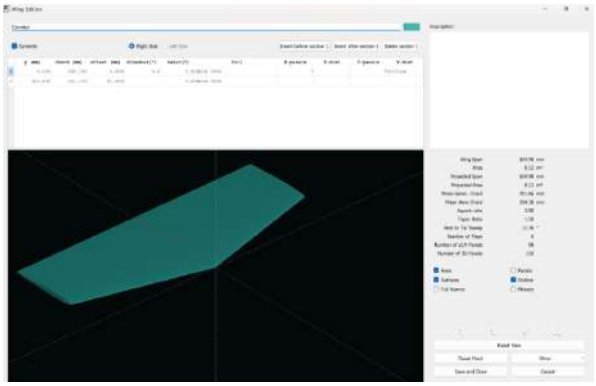
2.10

2.4 WING AND PLANE DESIGN

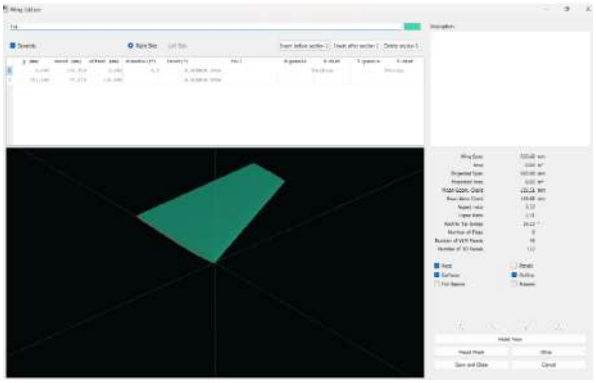
NACA 4412 has been used for the main wing, and NACA 0006 for the elevator and fin.



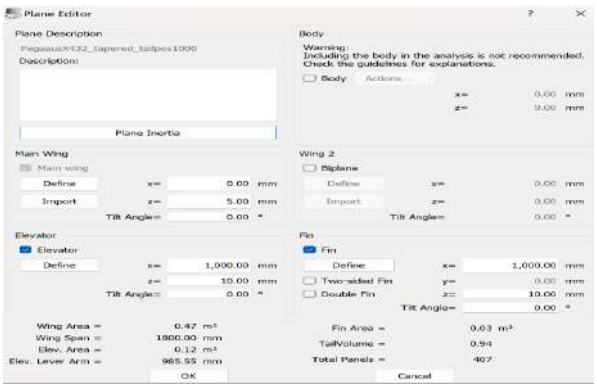
2.11



2.12



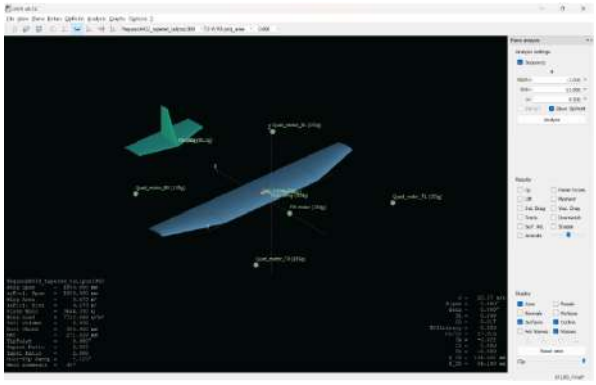
2.13



2.14

An almost negligible z elevation of 5 mm has been given to the main wing and 10 mm to the elevator and fin to prevent the Goldstein singularity error (division by zero error). Different elevations have been given to the main wing and the elevator and fin so that the airflow emerging from the main wing is not perturbed by the elevator and fin in the computational model making it possible to calculate the flow characteristics accurately.

2.5 MASS DISTRIBUTION

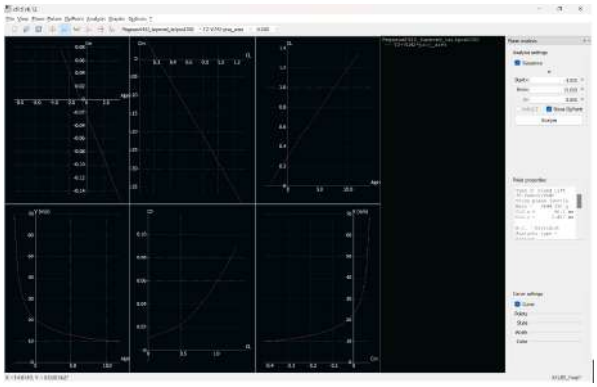


2.15

Mass (g)	x (mm)	y (mm)	z (mm)	Description
155.000	-575.000	-650.000	0.000	Quad_motor_TL
155.000	-575.000	650.000	0.000	Quad_motor_TR
155.000	725.000	-650.000	0.000	Quad_motor_BL
155.000	725.000	650.000	0.000	Quad_motor_BR
250.000	-200.000	0.000	0.000	Fixed Wing motor
2,353.000	75.000	0.000	0.000	Miscellaneous

2.16

A Type-2 (fixed lift) analysis has been performed on this wing and plane setup using the ring vortex (VLM2) method and wing planform projected on the xy plane. A sequential analysis from the angle of attack -3 to 15 degrees with increments of 0.5 degrees has been conducted.



2.17

The aerodynamic plots of the various polar properties have been obtained. The stability prerequisites have been verified as the Cm vs alpha slope is negative indicating that the plane is positively stable and the lift at zero Cm is also positive. The neutral point has been calculated from the Cm vs. alpha curve as it is the point where this curve becomes horizontal. Using hit and trial, the sequential analysis was run enough times to obtain the neutral point. The neutral point has been identified to be 196 mm from the leading edge of the main wing. Considering a 15 percent static margin, the position of CG has been calculated to be 155 mm from the leading edge.

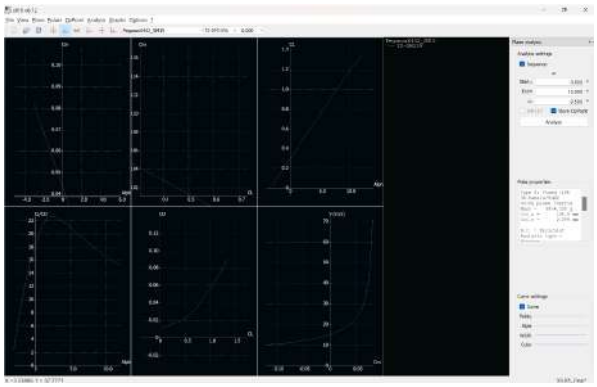
2.6 NEW MASS DISTRIBUTION

Mass (g)	x (mm)	y (mm)	z (mm)	Description
155.000	-495.000	-650.000	0.000	Quad_motor_TL
155.000	-495.000	650.000	0.000	Quad_motor_TR
155.000	805.000	650.000	0.000	Quad_motor_BR
155.000	805.000	-650.000	0.000	Quad_motor_BL
250.000	-200.000	0.000	0.000	FW motor
2,353.000	146.000	0.000	0.000	Miscellaneous
250.000	155.000	0.000	0.000	Payload

2.18

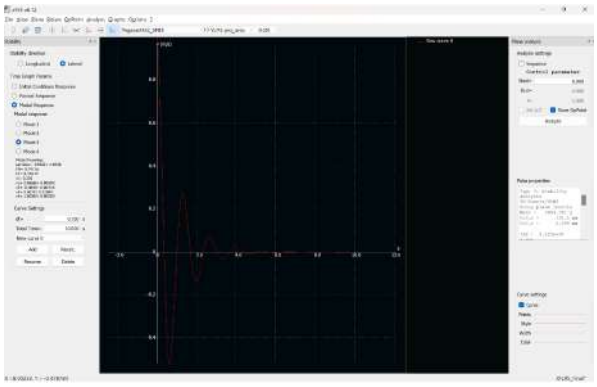
On performing the type 2 sequential analysis for this 15 percentage static margin setup, the following plots were obtained: In conclusion, the final parameters obtained from the XFLR5 analysis are:

- Balanced angle of attack = 3.2 degrees
- Design Cl = 0.3
- Cdmin = 0.01
- Cruise velocity = 15 m/s
- L/D ratio = 15.069



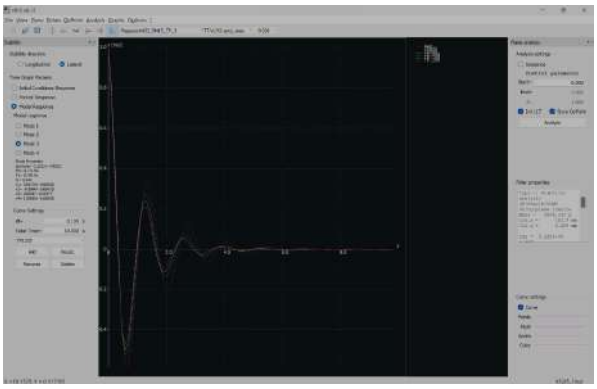
2.19

The stability prerequisites remain satisfied. On performing type 7 stability analysis, the expected root locus plots and time response graphs for the longitudinal and lateral stability modes were obtained. A very small Dutch roll time of 6 seconds was obtained, indicating the strong lateral stability and the self-stabilizing nature of the plane.



2.20

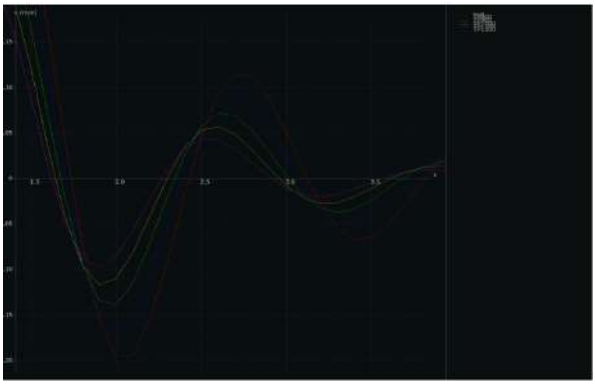
The Dutch roll time response analysis for different tail positions (800, 900, 1000, 1100, and 1200 mm from the main wing) have been compared. Here are the obtained results:-



2.21

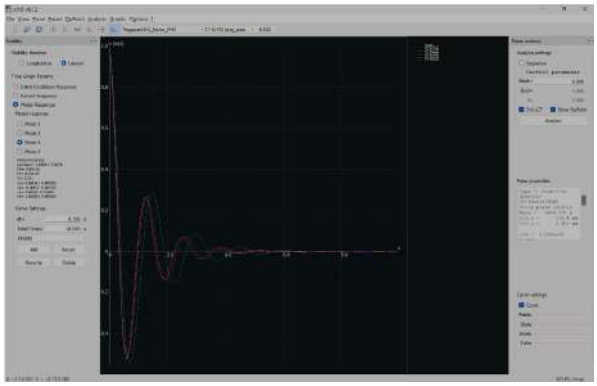
Legend:-

- TP800 - red
- TP900 - blue
- TP1000 - green
- TP1100 - yellow
- TP1200 - pink



2.22

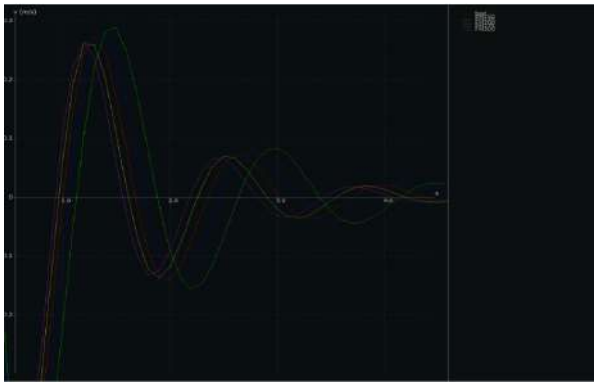
From the close-up of the comparison plots, it is clear that as the tail position increases, the Dutch roll time response curve dampens faster. However, after a certain point, the aircraft size is inadvertently increasing by increasing the tail position. Moreover, as the design includes a carbon fiber rod connecting the fuselage and main wing to the tail, increasing the tail distance beyond proportion leads to huge bending moments acting on the connecting rod, which will inevitably cause the rod to fail. To prevent this, an optimal value of the tail position (1000 mm from the leading edge of the main wing) has been chosen to maintain a balance between the stability and the structural strength of our design. The Dutch roll time response analysis for different fin heights (200, 225, 250, 275, and 300 mm) have been compared. Here are the obtained results:-



2.23

Legend:-

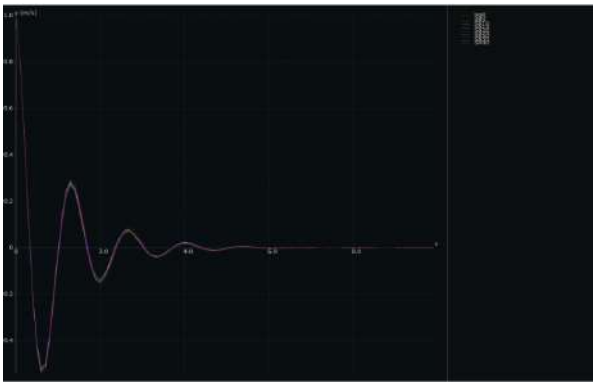
- FH250 - red
- FH225 - blue
- FH200 - green
- FH275 - yellow
- FH300 - pink



2.24

From the close-up of the comparison plots, it is clear that as the fin height increases, the Dutch roll time response curve dampens faster. As the fin height increases, the fin's loading conditions increase greatly due to the bending moments acting on the fin. Therefore, as perpetually increasing the fin height leads to its eventual failure due to the intense structural loading, an optimal fin height that ensures good stability characteristics

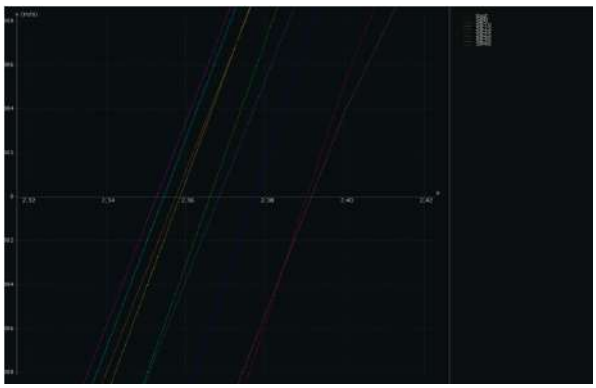
while also maintaining the structural integrity of our design has been identified. The Dutch roll time response analysis for different sweep angles of the fin (0, 5, 10, 15, 20, 25, 30, 35, and 40) has been compared. Here are the obtained results:-



2.25

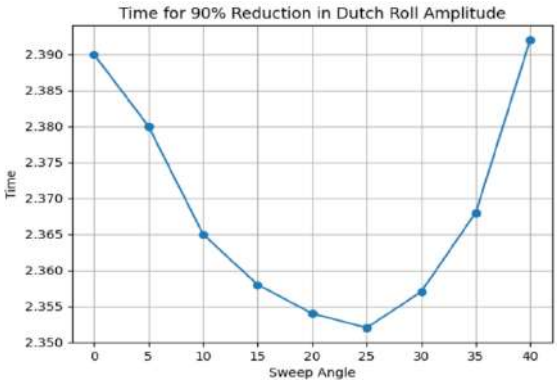
Legend:-

- SA0 - red
- SA5 - dark blue
- SA10 - green
- SA15 - yellow
- SA20 - light blue
- SA25 - purple
- SA30 - brown
- SA35 - medium blue
- SA40 - pink



2.26

From the close-up of the comparison plots, it is observed that the Dutch roll time response dampens faster as the sweep angle increases, but only till 25 degrees. Once the sweep angle exceeds 25 degrees, the time required for Dutch roll damping increases. Therefore, a sweep angle of 25 degrees has been chosen for the fin to achieve minimum dutch roll damping time.

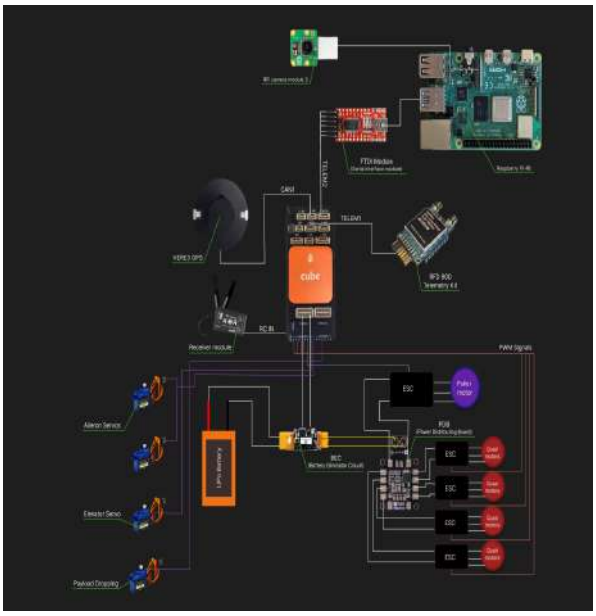


2.27

3 ELECTRONICS

(Adopted from *Design, Modeling, and Hybrid Control of a QuadPlane*)

3.1 ELECTRONIC DEVICES



3.1

Name	Use	Specifications
Holybro pixhawk cube	To be used as a flight controller to control the Quad-rotor as well as its systems. Pixhawk cube/ was selected owing to on-board processing power, low-level control, programmable firmware, failsafe and integrated sensors for state estimation.	32bit ARM® STM32H753 Cortex®-M7, ICM 20649 integrated accelerometer/gyro, MS5611 barometer on base board, 14x PWM servo outputs (8 from IO, 6 from FMU, S.Bus servo output,32 bit STM32F103 failsafe co-processor
TF 1-d lidar	To be used for correction in altitude estimation is done by the flight controller to improve accuracy	Accuracy – ±6cm, Frame Rate is 10 – 1000Hz, Communication Interface – UART, Supply Voltage – 5V
HEX HERE3 GPS	Used for global pose estimation i.e to tell Pixhawk its position	receiver - u-blox high precision GNSS modules (M8P-2), ICM20948 imu sensor, position accuracy - 3D FIX: 2.5 m / RTK: 0.025 m, protocol -DroneCAN 1Mbit/s
Telemetry (RFD 900x Radio Modem Bundle)	Allows for two-way communication with the ground station, the given telemetry pair was selected considering the greater range and maximum compatibility with Pixhawk	New Processor, ARM 32-bit core. Air data rate: 500kbit/s. frequency Range: 902 - 928 MHz. Long range >40km. Temp. Range: -40 to +85 deg
FrySky X8R Receiver	Is the perfect receiver for the FrySky Taranis, with its built-in XJT Module or any of the D8 systems to take advantage of the new digital Smart Port sensors. Additionally, it supports 8 standard servo outputs	16 channels, operating range - >=1.5km, 4- 10V operating voltage
Raspberry Pi 4B	The RPi 4B is a companion computer used for the purpose of running autonomous scripts stored as python file. Allows GCS interface with RPi camera module 3 to get live video feed if needed.	4 GB RAM 2.4 GHz CPU speed 64-bit quad-core Cortex-A72 processor
Raspberry Pi camera module 3	The camera module is needed to get the live video data and sending it to the python program, that constantly checks for the target location at which the payload has to be dropped.	SONY IMX708 12 mega pixel 120 degree FOV
MN501-S 240 KV	Used for the quad motors	prop diameter - 14 inches thrust (100% throttle) - 3.891 kg motor weight ~ 170g
AT3520 720 KV	Used for the puller motor	prop diameter - 14 inches thrust (100% throttle) - 4.316 kg motor weight ~ 219g
Tower Pro SG90 Servo	Controlling the control surfaces	Torque provided - 1.2 kg-cm at 4.4 operating voltage
FTDI UART to TTL module	Used for establishing serial communication between pixhawk and RPi	v2.0 USB

3.2

3.2 BATTERY LIFE

The above table mentions all the important electronic components used in the UAV, which essentially drain out a considerable amount of current for their operation. The table also mentions the respective average current and time of operation, which helps to find out the total capacity of charge used by the respective component.

S. No	Electronic component	Average current (A)	Time of operation (s)
1	4x MN4012 480Kv	10.9 *4 (85% throttle)	~150
2	1x AT3520 720Kv	42.17 (80% throttle)	t-150
3	Raspberry pi 4B+ Camera module	3.7 (Maximum Value)	t
4	RFD 900 Telemetry	0.8	t
5	Here 3+	0.21	t

3.3

The above table mentions all the important electronic components used in the UAV, which essentially drain out a considerable amount of current for their operation. The table also mentions the respective average current and time of operation, which helps us find out the total capacity of charge used by the respective component.

These values help us estimate the total capacity of the battery we need to use and also give us the lifetime of the UAV on a full battery charge.

$$I_i * t_i = C_i \tag{3}$$

$$\sum_i C_i = C_b \tag{4}$$

$$\frac{C_b}{I} = T \tag{5}$$

Where,

- C_i = individual component capacity cost
- C_b = total battery capacity
- I = UAV average current
- T = lifetime

Thus, the equation for the total capacity of the battery is given by,

C_b = 1820 + 11.7(t-150) + 3.7t + 0.8t + 0.21t

The UAV is expected to operate at least for 20 minutes, with all the functionalities, thus, taking t = 20 * 60 = 1200s, we get:

C_b = 1820 + 11.7*1050 + 3.7*1200 + 0.8*1200 + 0.21*1200
C_b = 19757 mAh

Considering available options in the market, the battery choice of 22000 mAh, 4s lipo battery would be the most efficient option.
This battery choice would give us the following equation:

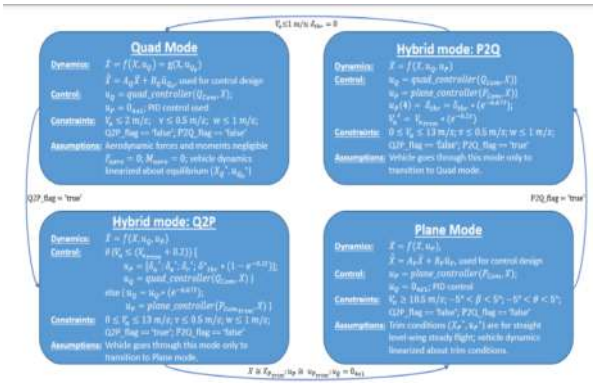
22000 = 1820 + 11.7(t-150) + 3.7t + 0.8t + 0.21t
22000 = 65 + 16.41t
21935 = 16.41t
T = 1336 seconds 22 minutes

These values help to estimate the total capacity of the battery needed and also give the lifetime of the UAV on a full battery charge.

4 CONTROLS

4.1 SYSTEM MODELING AND CONTROL

The QuadPlane was developed to operate in three modes: (i) Quad mode treating the vehicle as a quadrotor, (ii) Plane mode treating the vehicle as a fixed-wing aircraft with quad motors off, and (iii) Short-term transition modes Q2P (Quad to Plane) and P2Q (Plane to Quad). A four-state hybrid automaton governs vehicle controller behavior. Transitions are initiated by setting Q2P or P2Q flags while in Quad or Plane mode, respectively. In Quad mode, the vehicle operates at hover and slow airspeeds allowing aircraft aerodynamic effects to be neglected. In-Plane mode, vertical thrust motors are turned off and thus neglected. During Q2P and P2Q transitions, both Quad and Plane mode control effectors are used.



4.1 : QuadPlane Hybrid System Controller Model.

Nonlinear, six degrees of freedom, body axes equations model QuadPlane dynamics in all modes:

$$\dot{X} = f(X, u_P, u_Q) \quad (1)$$

4.2

where

$$X = \begin{bmatrix} x & y & z & u & v & w & \phi & \theta & \psi & p \\ q & r \end{bmatrix}^T,$$

$$u_P = \begin{bmatrix} \delta a & \delta e & \delta r & \delta t & h_r \end{bmatrix}^T,$$

$$u_Q = \begin{bmatrix} m_1 & m_2 & m_3 & m_4 \end{bmatrix}^T.$$

4.3

1)Navigation Equations describe the dynamics of the position vector in the inertial (world) frame:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R_b^i / \text{for ld} * \begin{bmatrix} u \\ v \\ w \end{bmatrix}; \quad R_b^i = \begin{bmatrix} c\theta c\psi - c\phi s\psi & -c\theta c\phi - s\phi s\psi & c\psi s\theta \\ c\theta s\psi + c\phi c\psi & -c\theta s\phi + c\phi c\psi & s\theta s\psi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}$$

4.4

2) Force Equations describe the dynamics of the vehicle velocity vector in the body frame:

$$\dot{u} = rv - qw - g\sin\theta + \frac{F_x}{M}$$
$$\dot{v} = -ru + pw + g\sin\phi\cos\theta + \frac{F_y}{M}$$
$$\dot{w} = qu - pv + g\cos\phi\cos\theta + \frac{F_z}{M}$$

4.5

3) Euler angle kinematics are given by:

$$\dot{\phi} = p + \tan\theta(q\sin\phi + r\cos\phi)$$
$$\dot{\theta} = q\cos\phi - r\sin\phi$$
$$\dot{\psi} = \frac{q\sin\phi + r\cos\phi}{\cos\theta}$$

4.6

4) Moment Equations describe angular rate derivatives in the body frame:

$$\dot{p} = \frac{1}{\Gamma}[J_{xz}[(J_x - J_y + J_z)qr - (J_z(J_z - J_y) + J_{xy})qr + J_z l + J_{xz} n]]$$
$$\dot{q} = \frac{1}{J_y}\left[(J_z - J_x)pr - J_{xz}(p^2 - r^2) + mr\right]$$
$$\dot{r} = \frac{1}{\Gamma}\left[[(J_x - J_y)J_x + J_{xz}^2]pq - J_{xz}[(J_x - J_y + J_z)qr + J_x l + J_{xn}]\right]$$

4.7

Equations (2) through (5) are linearized about equilibrium points uniquely identified for each mode.

4.2 PLANE MODEL

The plane model considers the QuadPlane as a conventional aircraft with forward thrust and aerodynamic surface forces and moments; quad control inputs uQ = [0 0 0 0]T . Airspeed is constrained to be higher than wing stall speed. Lift, drag, and pitching moment on each surface are given by:

$$L_i^f = \frac{1}{2}\rho V^2 a S C_L; \quad D_{\text{rag}} = \frac{1}{2}\rho V^2 a S C_D; \quad M_{\text{oment}} = \frac{1}{2}\rho V^2 a c \bar{C}_M$$

4.8

Aerodynamic forces were calculated in the wind frame and then transformed into the body frame:

$$A|_{bf} = R_{wind/bf} A|_{wf}; \quad R_{wind/bf} = \begin{bmatrix} cos\alpha cos\beta & cos\alpha sin\beta & -sin\alpha \\ sin\beta & cos\beta & 0 \\ sin\alpha cos\beta & -sin\alpha sin\beta & cos\alpha \end{bmatrix}$$

4.9

where A—bf denotes force or moment vector in the body frame and A—wfx denotes the vector resolved in the wind frame. Forces and moments acting on each surface was summed with forward motor thrust and torque to compute the total forces and moments. Eqs. 2 to 5 are used to update the system state. The The plane controller is based on dynamics linearization to obtain equilibrium or trim conditions. Trim conditions are denoted by states X* and input commands u*P. Eq. 1 then becomes f(X*, u*P) = 0. The aircraft trim state is defined by airspeed (V*a), flight path angle (y*), and coordinated turn radius (R*). In this paper, a reference trim state was calculated for constant airspeed (V*a = 11 m/s), zero climb rate (y* = 0) along a straight path (R* = inf), using methods from [?, ?, ?] with results shown in Appendix A. Eqs. (2)-(5) were linearized with Taylor series perturbations about this trim state with states newX = X-X* and inputs newuP = uP-u*P. The linearized equations are split into longitudinal and lateral equations given by:

$$\dot{X}_{\text{lat}} = A_{\text{lat}} X_{\text{lat}} + B_{\text{lat}} u_{\text{lat}} \quad \dot{X}_{\text{lon}} = A_{\text{lon}} X_{\text{lon}} + B_{\text{lon}} u_{\text{lon}}$$

where

$$X_{\text{lat}} = \begin{bmatrix} \bar{v} & \bar{p} & \bar{r} & \bar{\phi} & \bar{\psi} \end{bmatrix}^T, \quad u_{\text{lat}} = \begin{bmatrix} \bar{\delta}_a & \bar{\delta}_r \end{bmatrix}^T;$$

$$X_{\text{lon}} = \begin{bmatrix} u & \bar{w} & \bar{q} & \bar{\theta} & \bar{z} \end{bmatrix}^T, \quad u_{\text{lon}} = \begin{bmatrix} \bar{\delta}_e & \bar{\delta}_t & h_r \end{bmatrix}^T.$$

Values for A_{lat}, B_{lat}, A_{lon}, and B_{lon} are given in Appendix A.

4.10

Values for Alat, Blat, Alon, and Blon are given in Appendix A. The Plane mode feedback controller must track the commanded airspeed Vca, altitude hc = -zc, and heading angle yc while maintaining zero sideslip bc = 0 in the absence of ambient wind. Successive loop closure [?] is applied over the linearized dynamics. For lateral dynamics, a Roll Attitude Hold (RAH) loop tracks a reference roll command pc with a Proportional Integral Derivative (PID) controller over da. A PI Course Hold loop tracks commanded heading angle pc as input and generates commanded roll angle pc which then feeds into the RAH loop. A PI Sideslip Hold loop over dr is defined.
To control longitudinal dynamics, a Pitch Attitude Hold (PAH) loop is used to track a reference pitch command (Oc) using a PD controller and providing perturbed elevator deflection de as an output. The Altitude Hold loop assumes that the airspeed is held fairly constant and altitude can be adjusted by commanding a perturbed reference pitch angle Oc. To get the actual pitch command, the trim pitch value is added to this value, i.e., Oc = O*c +O*. The Altitude Hold loop uses a PI controller to ensure that no steady-state error propagates through the inner PAH loop. The Airspeed Hold loop is a standalone PI control loop that controls the vehicle's airspeed. The loop takes commanded airspeed Vca as an input and generates the required perturbed throttle command de as an output. Equations for all six control loops are given by: where p and q denote perturbed

$$\begin{aligned} \tilde{\phi}^c &= k_{p_\phi}(\phi^c - \psi) + \frac{k_{i_\phi}}{s}(\psi^c - \psi) & \tilde{\theta}^c &= k_{p_z}(z^c - z) + \frac{k_{i_z}}{s}(z^c - z) \\ \tilde{\delta}_a &= k_{p_\phi}(\phi^c - \phi) + \frac{k_{i_\phi}}{s}(\phi^c - \phi) - k_{d_\phi}\tilde{p} & \tilde{\delta}_e &= k_{p_\theta}(\theta^c - \theta) - k_{d_\theta}\tilde{q} \\ \tilde{\delta}_r &= k_{p_\beta}(\beta^c - \beta) + \frac{k_{i_\beta}}{s}(\beta^c - \beta) & \tilde{\delta}_{thr} &= k_{p_V}(V_a^c - V_a) + \frac{k_{i_V}}{s}(V_a^c - V_a) \end{aligned} \tag{9}$$

4.11

roll and pitch rates, respectively. Actual Plane control inputs, uP , are obtained by adding trim input commands to perturbed inputs, i.e., newuP = uP + u*P, and nonlinear dynamics are used to simulate the system. To prevent control input saturation, proportional gains in RAH and PAH loops are tuned such that a maximum error of 5 degrees in roll or pitch causes maximum aileron or elevator deflection.

4.3 QUAD MODEL

The Quad model considers the QuadPlane purely as a quadrotor and assumes that the wing/tail aerodynamic forces and moments are negligible. To support this assumption, vehicle airspeed is constrained (Fig. 4) to be low such that aerodynamic surface effects are negligible. Plane mode control inputs are set to uP = [0 0 0 0]T . The forces and moments acting on the vehicle in Quad mode originate from the quad motors only and are given by:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -(m_1 + m_2 + m_3 + m_4) \end{bmatrix};$$
$$\begin{bmatrix} l \\ m \\ n \end{bmatrix} = \begin{bmatrix} L_{Q/2} \times (-m_1 - m_2 + m_3 + m_4) \\ L_{Q/2} \times (m_1 - m_2 - m_3 + m_4) \\ \gamma_{mot} \times (m_1 - m_2 + m_3 - m_4) \end{bmatrix}$$

4.12

where ymot was found experimentally. To simulate Quad dynamics, rcpilot parallel inner loop control logic was reused. Reference outer loop states, uQo = [zr er pr Or Pr] T , were input directly into the dynamics function, altering Eq. (1) to Eq. (11) for this mode using Eq. (12) to compute commanded motor forces from uQo within function g(): where MQUa is the mixing matrix for the

$$\dot{X} = g(X, u_{Q_o}) \quad (11)$$
$$u_Q = M_{Quad} * u_{Q_o} \quad (12)$$

4.13

QuadPlane in Quad mode, given in Appendix A. To ensure that quad motors are not saturated with a large uQo input value, the system in Eq. (11) is linearized about equilibrium point (X* , u o Q) and linearized as:

$$\dot{\bar{X}} = A_Q \bar{X} + B_Q \bar{u}_{Q_o} \quad (13)$$

4.14

where AQ and BQ are linearized system model matrices; X and u o Q are the perturbed state and control inputs. PID controllers computed reference outer loop state such that motor commands are not saturated. Another PID loop tracked reference airspeed which output commanded pitch angle. PID control is given by:

$$\bar{st}_{ref} = (k_{P_{st}} + \frac{k_{i_{st}}}{s} - s k_{D_{st}})(st^c - st) \quad (14)$$

4.15

where stc represents commanded state value, stref is the corresponding element of uQo and st represents the actual value of the state being tracked.

4.4 TRANSITION MODEL

The transition model combines Quad and Plane dynamics and controllers. The QuadPlane must execute Quad-toPlane (Q2P) and Plane-to-Quad (P2Q) transitions over a range of airspeeds (0 m/s ¡ Va ¡ 13 m/s) that are controllable in either Quad and Plane modes with constraints. Q2P mode smoothly transitions from hover to the aircraft trim state. P2Q mode smoothly transitions from the aircraft trim state back to Quad mode.
In Q2P mode, an airspeed higher than the trim velocity (V c a = V * a + 2 m/s) is tracked using the Quad mode controller with aircraft control surfaces at their trim values and the forward motor ramping up to its trim state. This prevents the quad motor from dropping to zero thrust, which otherwise occurs as the vehicle approaches trim airspeed and can cause the vehicle to destabilize. Once an airspeed threshold is reached, 11.2 m/s for this analysis, the vehicle switches to the Plane mode controller and tracks the trim state, while the quad motors ramp down to zero.
For P2Q transition, the commanded airspeed (V c a) ramps down from aircraft trim value (V * a) to 1 m/s which is within the Quad mode flight envelope. The strategy adopted is to use the vertical motors and elevator to pitch up, reducing forward airspeed, while maintaining level altitude. At each time step, the Quad airspeed controller calculates O c , which is tracked using the Plane PAH controller. Lateral Plane controllers and the generated P c are used in uQ. O c then feeds into the Quad controller. Once airspeed drops below 1 m/s, the hybrid automaton switches to Quad mode.

4.5 RESULTS

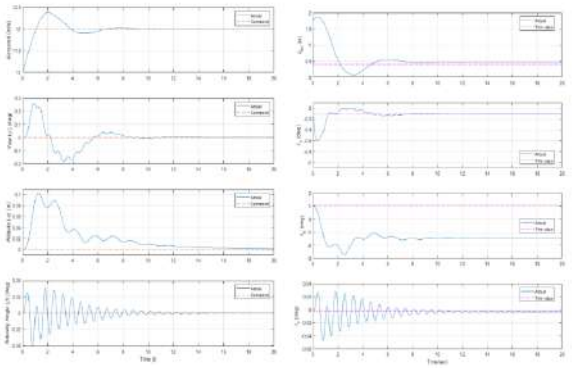


Figure 2: Plane Mode - Command Tracking (left) and Control Inputs (right).

4.16

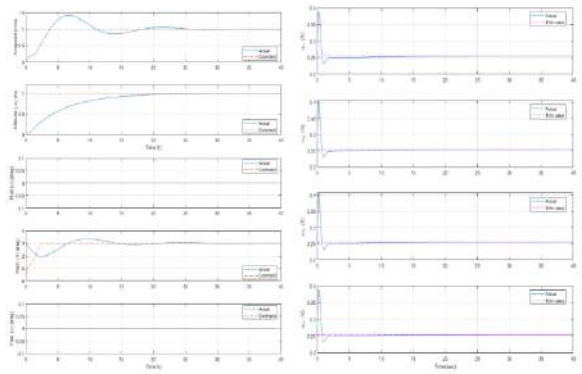


Figure 3: Quad Mode - Command Tracking (left) and Control Inputs (right).

4.17

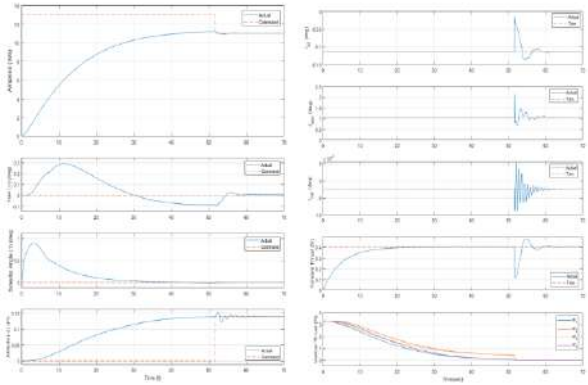


Figure 4: Q2P Transition - Command Tracking and Control Inputs.

4.18

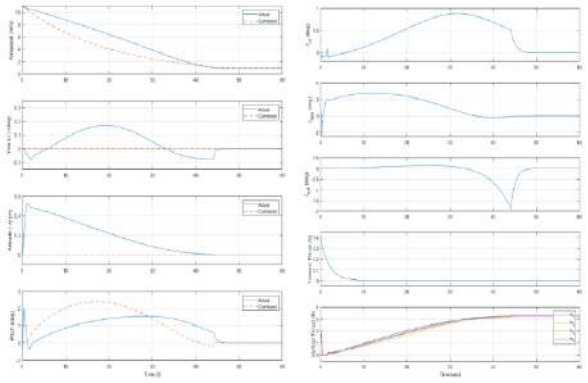


Figure 5: P2Q Transition - Command Tracking and Control Inputs.

4.19

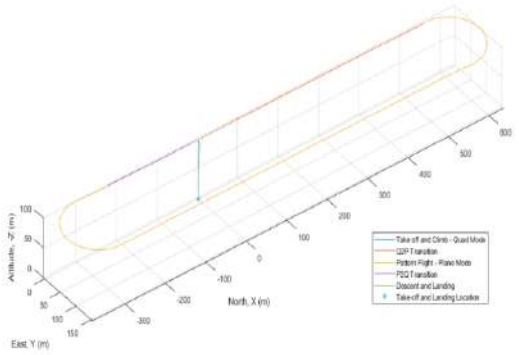


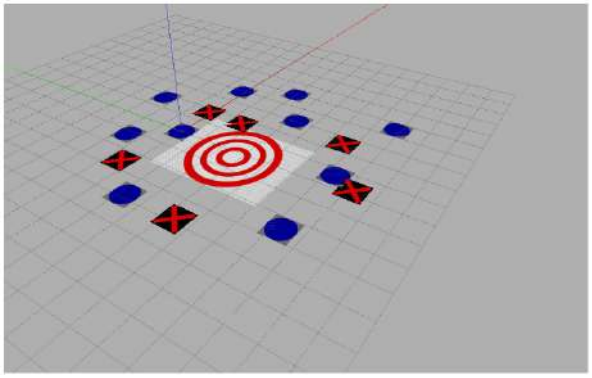
Figure 6: Full-Envelope Flight Trajectory: Take-off from $(0, 0, 0)$; Transitions and Pattern Flight at 100 m Altitude; and Landing at $(0, 0, 0)$.

4.20

5 TARGET DETECTION

5.1 ENVIRONMENT

The environment consists of a Red Bullseye Target surrounded by decoys like Red X's and Blue Circles. The goal is to detect the Bullseye Target from the camera feed of the Drone and to find the coordinates of the center of the Bullseye for package delivery.



5.1

5.2

5.2 PROCEDURE

- **Processing the Captured Image:** The camera feed is converted to BGR from BGRA. The converted feed is thresholded for Red color detection by cv2.inRange() with:
  Lower Bound = np.array([0, 94, 0])
  Upper Bound = np.array([179, 255, 255])
  After thresholding, circles are detected using Hough Transform. This gives us the center of the bullseye target since it is the only red Circle in the frame.
  Although the thresholding eliminates most of the blue circles, to ensure that only the bullseye is detected and not the blue circles, red contours are detected in the thresholded feed with cv2.findContours(). This returns a bounding box around the red Bullseye and also the Red X's.
  If the center of the detected circle and the center of the detected Bounding Box is within a threshold, the bullseye is detected, and the center of the Bounding Box is used as the center of the Bullseye. This is done to eliminate the possibility of falsely detecting the Red X's and Blue Circles as Bullseye.

- **Distance and Coordinate estimation of detected bullseye from the camera:** Once the center of the Bullseye is detected, its coordinates relative to the camera are found using the following formulae:

$$P_{ixelToMeter} = \frac{2 * H_{Camera} * Tan(\frac{\theta_{FOV}}{2})}{W_{idth}}$$

$$x_{center} = \frac{W_{idth}}{2}$$

$$y_{center} = \frac{H_{eight}}{2}$$

$$x = (x_{center} - x_{cameracenter}) * P_{ixelToMeter}$$

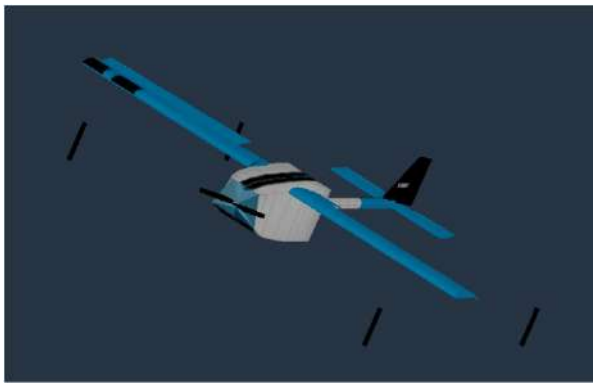
$$y = (y_{center} - y_{cameracenter}) * P_{ixelToMeter}$$

$$z = -H_{camera}$$

6 XPLANE

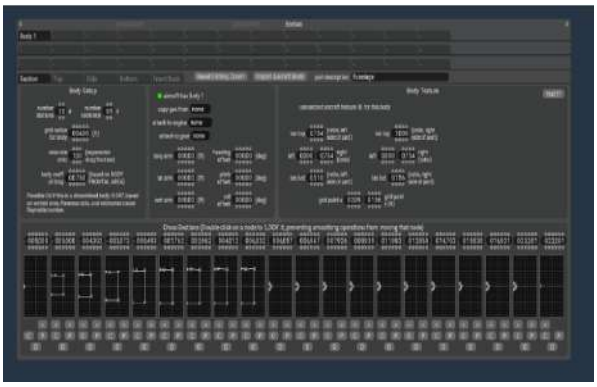
6.1 XPLANE OVERVIEW

X-Plane serves as a virtual environment where we can simulate the specifications and requirements of our drone models without the need for physical prototypes.



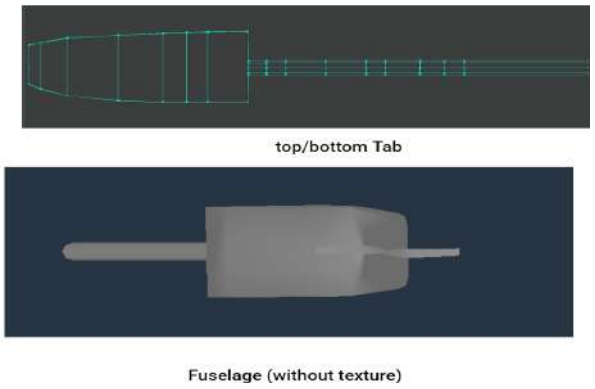
6.1

6.2 FUSELAGE



6.2

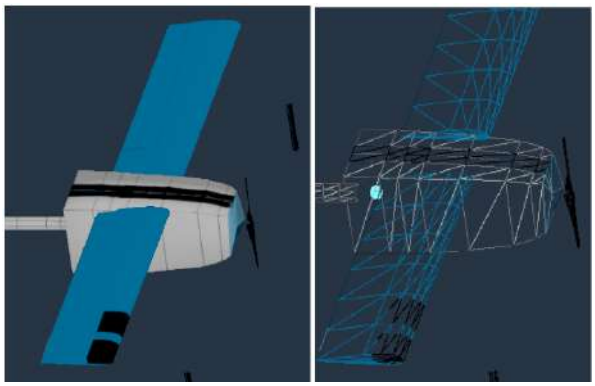
- SHAPING :
  All locations in Plane Maker are defined relative to a fixed, arbitrary point; the reference point. Firstly the number of sections and the body radius are defined that are needed to shape the fuselage. A rough outline of its shape is created using relevant options in the Sections tab. The fuselage was a new model, so the body had to be made from scratch.
- TEXTURE :
  For fine-tuning the painted texture on the aircraft, the Body Texture box is used. This creates a png of the mesh, which can be edited in any painting software, and it colors the body according to the need.



6.3

6.3 WINGS :

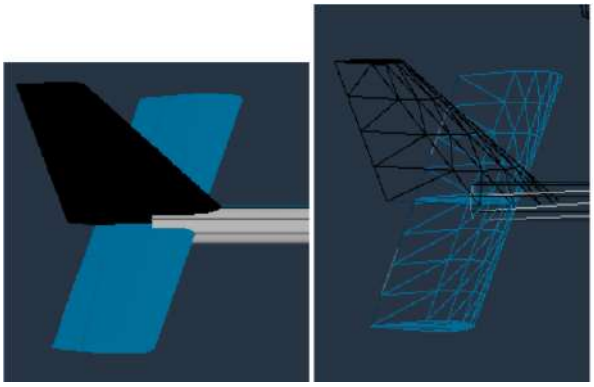
Shaping the wings: Wings in Plane Maker are composed of individual wing sections. A very simple wing might comprise a single wing section, while a very complex wing might comprise four or more wing sections. For each wing section we can have control surfaces added, such as ailerons, elevators, or flaps.



6.4

6.4 TAILS :

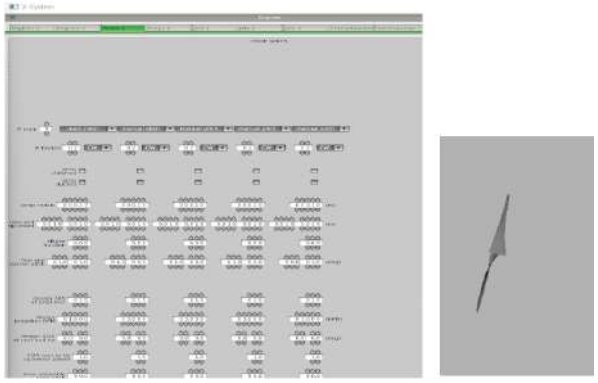
The tail consisted of horizontal and vertical stabilizers, which contained the control surfaces of the plane, which are the elevator and rudder used for pitch and yaw motions. The control surfaces were added the same way as the wings.



6.5

6.5 PROPELLERS :

The number of blades can be defined independently for each propeller. In the quadplane, 2 blades have been used and the prop radius was set. The prop engine has the option to set the weight of the prop relative to the mass of the aluminum. The ratio of the mass of the propeller to the mass of aluminum was taken to be 0.13, which is a standardized setting. As we are using electric motors for VTOL, we used the electric engine.

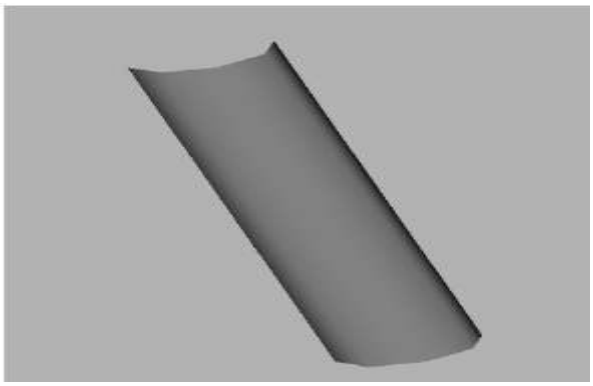


6.6

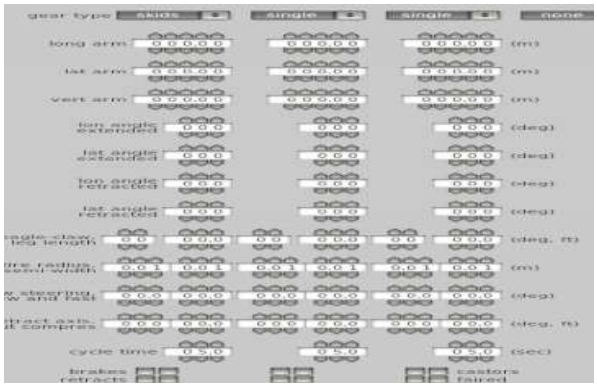
6.6 SKIDS :

The landing gear needed was a skid below the plane, and it is naturally at an equal distance from the center of mass to ensure a safe landing. Landing gears come in a variety of configurations, ranging from simple metal skids to a single wheel, to groups of many wheels. Any landing gear needs to have its position on the aircraft specified, and if the gear is retractable, it must have a retracted position that is different from its extended position. The gear also must have a size—both its tire size and its strut length.

At the top of the Gear Loc tab is the gear type parameter. The drop-down menu contains a wide array of wheel (or skid) configurations. A lateral wheel configuration arranges the wheels side-by-side, while a truck configuration arranges them in rows. A "long" wheel configuration arranges them one in front of another. A skid gear type was added, which is just a rod. Its orientation and position can be set along with the number of gears needed and their size.



6.7



6.8

6.7 WEIGHT AND BALANCE :

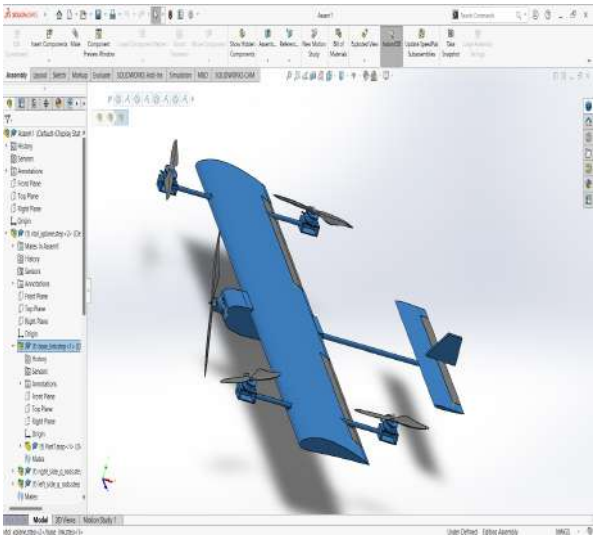
. Setting the CG: The aircraft's center of gravity (CG) is set using only the longitudinal and vertical parameters of the standard location controls; that is, it has only a distance behind and above the reference point. In X-Plane, the user may move the center of gravity forward or aft. In light of this, three longitudinal positions are defined for the center of gravity. The vertical position of the center of gravity stays constant no matter how the CG is moved. The empty weight and maximum weight are set. The empty weight is the weight without the payload.

Reference document: link

7 GAZEBO

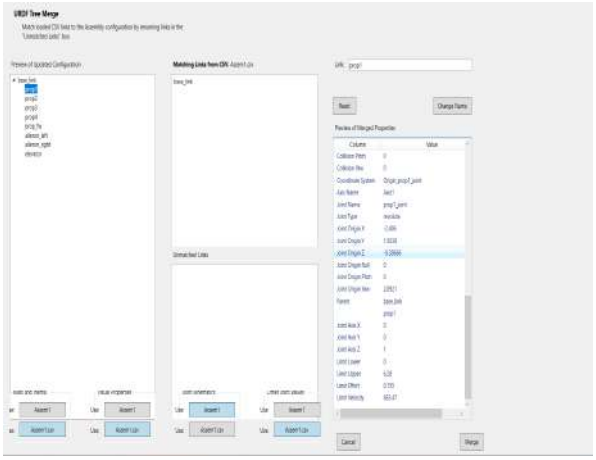
7.1 DESIGNING VTOL MODEL FOR GAZEBO

The model was made in Autodesk Fusion 360. The base links, propellors, and control surfaces were exported individually as step files so that we could use them in SolidWorks. SolidWorks was used for the final assembly as URDF exporter plugin was used to create URDF files of our VTOL.

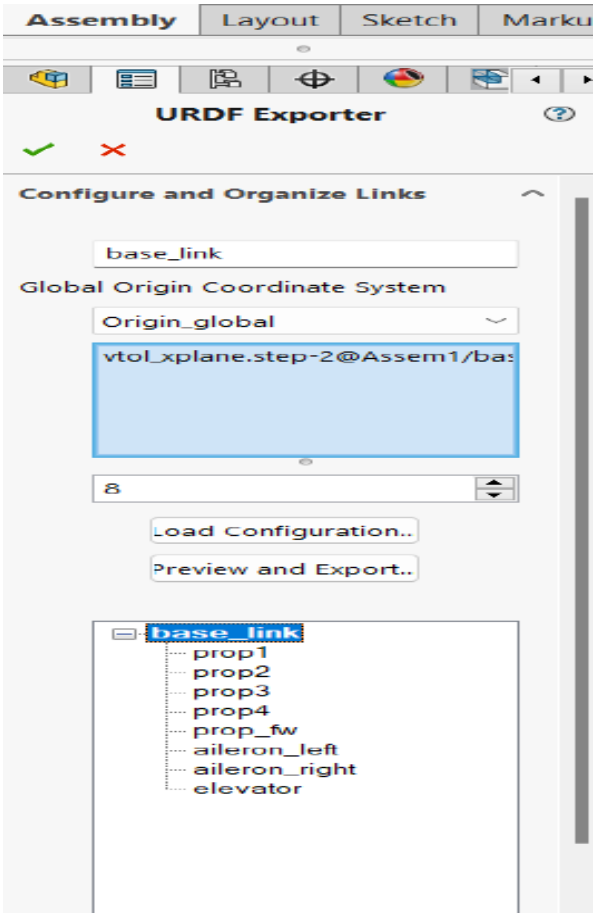


7.1

Subsequently, the final assembly was done in Solidworks, defining each joint properly, assigning names to each joint, and specifying base links and child links properly. There is 1 base link and 8 child links. 5 joints of the base link were with propellors, 2 joints of the base link with ailerons, and 1 joint of the base link with the elevator. Then, masses were defined for every link and the limits of each revolutionary joint were defined.Subsequently, the model was exported along with its mesh files.



7.2



7.3

7.2 CREATING MODEL'S SDF FILE AND WORLD FILE

Creating a world file was easy; our model was to be included, so whenever the world file was launched, the model was already spawned in it. Firstly, the URDF file and the mesh files were copied and

pasted into our package. The mesh files were in STL format, so they were first converted to dae format, as loading STL files in the gazebo gave an error. Also, an SDF file was needed for the model instead of URDF, so the URDF file was converted into an SDF file. Then all the required plugins were added, one of which was LiftDragPlugin.

```
<plugin name="elevator" filename="libLiftDragPlugin.so">
  <a0>-0.2</a0>
  <cla>4.752798721</cla>
  <cda>0.6417112299</cda>
  <cma>-1.8</cma>
  <alpha_stall>0.3391428111</alpha_stall>
  <cla_stall>-3.85</cla_stall>
  <cda_stall>-0.9233984055</cda_stall>
  <cma_stall>0</cma_stall>
  <cp>-0.5 0 0</cp>
  <area>0.01</area>
  <air_density>1.2041</air_density>
  <forward>1 0 0</forward>
  <upward>0 0 1</upward>
  <link_name>base_link</link_name>
  <control_joint_name>
    elevator_joint
  </control_joint_name>
  <control_joint_rad_to_cl>-4.0</control_joint_rad_to_cl>
</plugin>
```

7.4

```
<plugin name="rotor_1_blade_1" filename="libLiftDragPlugin.so">
  <a0>0.3</a0>
  <alpha_stall>1.4</alpha_stall>
  <cla>4.2500</cla>
  <cda>0.10</cda>
  <cma>0.00</cma>
  <cla_stall>-0.025</cla_stall>
  <cda_stall>0.0</cda_stall>
  <cma_stall>0.0</cma_stall>
  <area>0.002</area>
  <air_density>1.2041</air_density>
  <cp>0.084 0 0</cp>
  <forward>0 1 0</forward>
  <upward>0 0 1</upward>
  <link_name>rotor_1</link_name>
</plugin>
```

7.5

Then, other necessary details like the IMU sensor and GPS and other things related to Ardupilot were added to our file to facilitate operating our VTOl using the MavProxy console and map.

```
<link name='standard_vtol/imu_link'>
  <pose>0 0 0 0 0 0</pose>
  <inertial>
    <pose>0 0 0 0 0 0</pose>
    <mass>0.015</mass>
    <inertia>
      <ixx>1e-05</ixx>
      <ixy>0</ixy>
      <ixz>0</ixz>
      <iyy>1e-05</iyy>
      <iyz>0</iyz>
      <izz>1e-05</izz>
    </inertia>
  </inertial>
  <sensor name="imu_sensor" type="imu">
    <pose>0 0 0 3.141593 0 0</pose>
    <always_on>1</always_on>
    <update_rate>1000.0</update_rate>
  </sensor>
</link>
<joint name='standard_vtol/imu_joint' type='revolute'>
  <child>standard_vtol/imu_link</child>
  <parent>base_link</parent>
  <axis>
    <xyz>1 0 0</xyz>
    <limit>
      <lower>0</lower>
      <upper>0</upper>
      <effort>0</effort>
      <velocity>0</velocity>
    </limit>
    <dynamics>
      <spring_reference>0</spring_reference>
      <spring_stiffness>0</spring_stiffness>
    </dynamics>
    <use_parent_model_frame>1</use_parent_model_frame>
  </axis>
</joint>
```

7.6

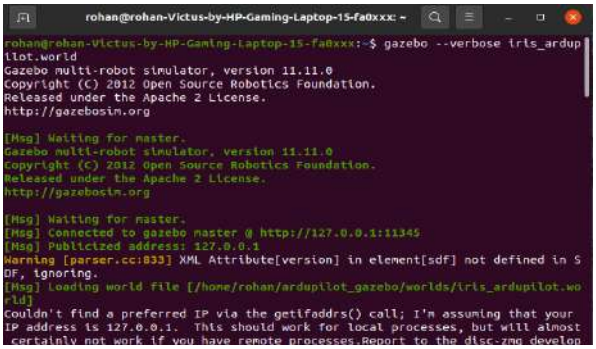
```
<plugin name="arducopter_plugin" filename="libArduPilotPlugin.so">
  <fdm_addr>127.0.0.1</fdm_addr>
  <fdm_port_in>9002</fdm_port_in>
  <fdm_port_out>9003</fdm_port_out>
  <!--
      Require by APM :
      Only change model and gazebo from XYZ to XY-Z coordinates
  -->
  <modelXYZToAirplaneXForwardZDown>0 0 0 3.141593 0 0</modelXYZToAirplaneXForwardZDown>
  <gazeboXYZToNED>0 0 0 3.141593 0 0</gazeboXYZToNED>
  <imuName>standard_vtol/imu_link::imu_sensor</imuName>
  <connectionTimeoutMaxCount>5</connectionTimeoutMaxCount>
  <control channel="0">
      <multiplier>1</multiplier>
      <offset>-0.5</offset>
      <type>POSITION</type>
      <p_gain>2.5</p_gain>
      <i_gain>0</i_gain>
      <d_gain>0</d_gain>
      <i_max>0</i_max>
      <i_min>0</i_min>
      <cmd_max>1</cmd_max>
      <cmd_min>-1</cmd_min>
    <jointName>left_elevon_joint</jointName>
  </control>
  <control channel="1">
      <multiplier>1</multiplier>
      <offset>-0.5</offset>
      <type>POSITION</type>
      <p_gain>2.5</p_gain>
      <i_gain>0</i_gain>
      <d_gain>0</d_gain>
      <i_max>0</i_max>
      <i_min>0</i_min>
      <cmd_max>1</cmd_max>
      <cmd_min>-1</cmd_min>
    <jointName>right_elevon_joint</jointName>
  </control>
```

7.7

In this process, a big mistake was identified while converting the URDF file to sdf file, as in both systems, the joint poses are defined relative to different types of origins. Thus, the poses and inertia had to be rectified accordingly.

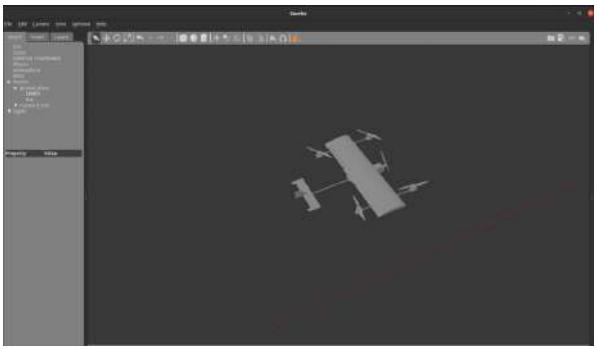
7.3 LAUNCHING THE WORLD FILE AND CONSOLE

The world file was launched using the command " gazebo –verbose iris_ardupilot.world ".

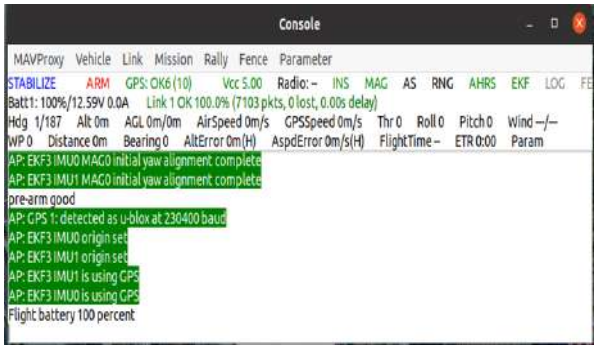


7.8

7.11

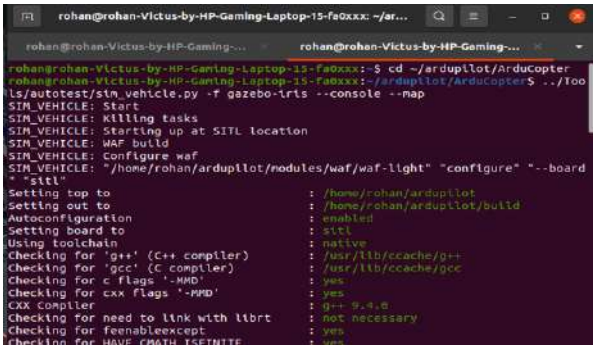


7.9



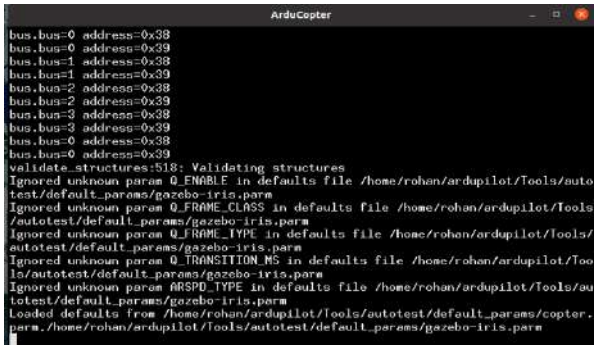
7.12

The command " sim_vehicle.py -f gazebo-iris –console –map " was run next.



7.10

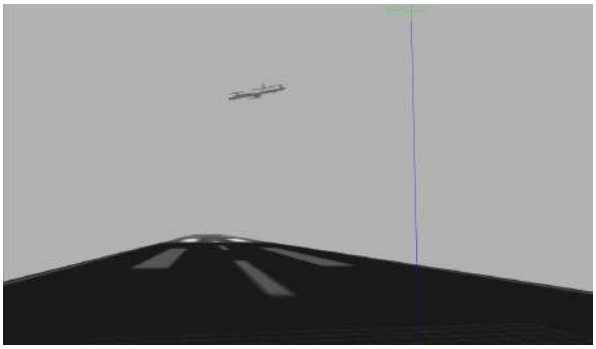
3 windows named "map," "console," and "ArduCopter" were obtained.



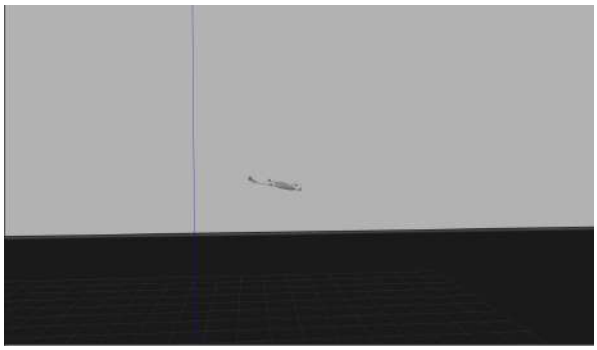
7.13

7.4 TESTING VTOL

The VTOL was then tested in guided mode by typing the command "mode GUIDED" into the terminal.

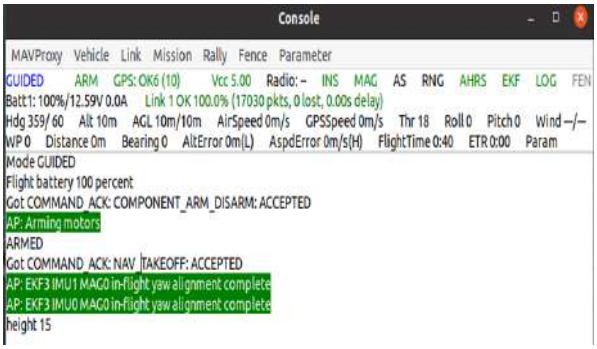


7.14

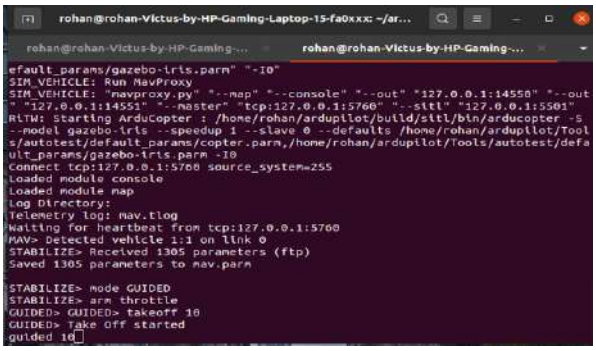


7.18

After taking it to an altitude of 10 m, it was taken to a waypoint in GUIDED mode.



7.15



7.16

7.17

8 AUTOMATION

8.1 REQUIREMENT

The competition required us to automate the flight process, given the waypoints. The task will include taking off in VTOL mode, transitioning into fixed-wing mode, and approaching the waypoint trajectory. Along the trajectory, an RPi camera module is being used to detect marks on the ground and identify the targeted drop location, followed by dropping the payload at the detected location.

For automation, an onboard companion computer Raspberry Pi 4B is being used, which helps the purpose of automating the flight mission by using a Python script, along with ROS communication with the flight controller incorporating the camera module and transmitting back the camera feed to the GCS (if required).

8.2 PIPELINE

The Pixhawk orange cube flight controller takes in and sends messages of Mavlink type. A Mavlink message sends messages serially between the ground control station and the vehicle in Mavlink packets. The message follows a particular format, and messages of specifically those formats are identified by the vehicle and the GCS.

Python is used as our programming language to follow an automation script, which is stored in our Raspberry Pi. Now, even though our Raspberry Pi has built a successful serial connection with the flight controller, it will still not be able to run the Python script normally. The flight controller needs the data in and out messages to be of Mavlink type, while the Python script trivially does not send the messages in Mavlink format. Now, there are 2 available options to solve this problem:

- Altogether remove the RPi computer and program the flight controller microprocessor, which can be done via programming in LUA. LUA is a low-level programming language that can be used to access the microchip input outputs and other functionalities directly.

- The second and more popular way is to use a Python to Mavlink converter to convert the messages from the RPi into Mavlink message type. Pymavlink is a Python library that can do so.

The second method can be used to implement simple Python scripts. For more complex functions, ROS is used, giving us access to different services and topics that might be needed for calculating motion planning, computer vision, etc. Again, a direct use of ROS will not be possible for the above-mentioned reasons.

MAVROS is a packaging around ROS that facilitates using Mavlink messages and ROS to communicate with the flight controller.

8.3 NAVIGATION AND MISSION CONTROL

"Waypoint Navigation" has been used as the current autonomous flight plan. Using the MAVlink extendable communication protocol, it is planned to execute our desired flight plan, which will be in the form of a Python script.
The Waypoint Navigation strategy works in three steps -

- **- The quadcopter will navigate around the given set of waypoints.**
  The GLOBAL-RELATIVE frame is used for this purpose as we can give the set of way points in the global frame as provided to us in text format
  - While navigating through each waypoint, the camera module will constantly look for the target location using computer vision. It will store the location of the target for use later.
  - After navigating around each waypoint, the quadcopter reaches the Payload Drop The point will be estimated by considering the wind and drop conditions such that the Payload reaches the Payload Release Point as close to the center as possible.
  - A list of waypoint messages is pushed to the mission plan, with a certain set of parameters defined for each message, including the location of the waypoint and the action we want to perform.
  - Each action is denoted by a different number that can be passed through the message as a command. A few examples of actions that the UA can perform at a particular waypoint include take-off, land, navigating to the waypoint, or loitering.
  - The MAVROS waypoint message allows to pass the coordinates of the required waypoint in a frame of reference decided by us. An acceptance radius can also be defined, representing the minimum distance that the UAV needs from the waypoint before it is

considered to have reached it. (7m as mentioned in the competition rules).
- Other additional parameters are used to decide the altitude, speed, and heading that the UAV should use to approach the waypoint, along with transition parameters for the transition between VTOL mode and fixed-wing mode or vice versa.
- These parameters can also be edited directly from the QGC (Q Ground Control) software.

- **Custom control for Static/Dynamic Drop**
  - After completing the previous task, the quadcopter will drop the payload depending upon its state (dynamically or statically) by considering environmental conditions. The mechanical design of the drone ensures smooth drop conditions without causing much disturbance.
  - A custom controller is used for this drop, written in Python. The controller tracks and completes the mission, starting with reaching the drop location, dropping the payload, and then going to the next task, essentially returning to the land location.

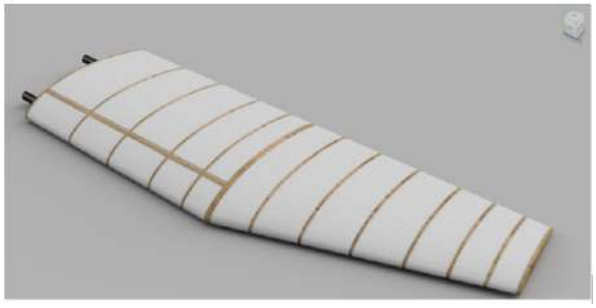
- **Endurance Run using Dynamic Decision Making**
  - The last task is executed by designing a smart decision-making strategy, which takes into account factors such as
  — Battery Life Remaining — Time of Mission Remaining
  - At every waypoint past a certain time threshold, our algorithm iteratively calculates the time it would take for our UA to visit- another waypoint and then return to the launch position. If this time is less than the amount of time left for the mission to end, the UA must return to the launch position immediately.
  - If the amount of time left allows for another waypoint to be visited, the UA continues its mission and repeats this decision-making process at the next point.
  - Since the algorithm may require the endurance task to be cut short due to time or battery constraints, we can end the waypoint mission by changing flight modes from "AUTO" to "GUIDED".

The flight control system has robust built-in safety and failsafe mechanisms in place. We have programmed a Geo-Fence Failsafe in regulation with the competition rules, which will ensure that the drone executes Loiter mode in case it escapes the GeoFence and then lands in the same place. Finally, failsafe mechanisms in the case of RC communication and Data-link loss is being worked on and will be integrated into the control system architecture.

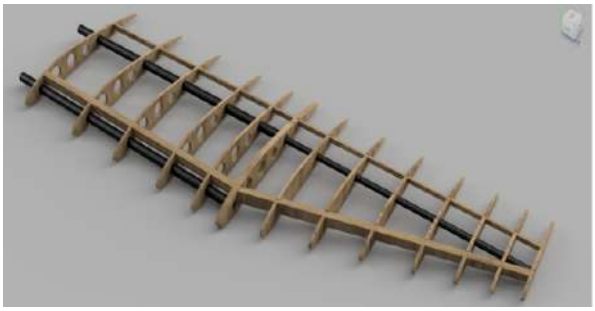
The whole stack for navigation and mission control, including waypoint navigation, dynamic decision-making protocols, and FTS algorithms, have been tested in simulation to confirm accuracy and proficiency. This flight stack will soon be tested on a smaller prototype quadplane. After the flight controls are successfully tested on the prototype, the algorithms will be run on the final manufactured quadplane to test the mission.

9 MANUFACTURING

9.1 DESIGN



9.1



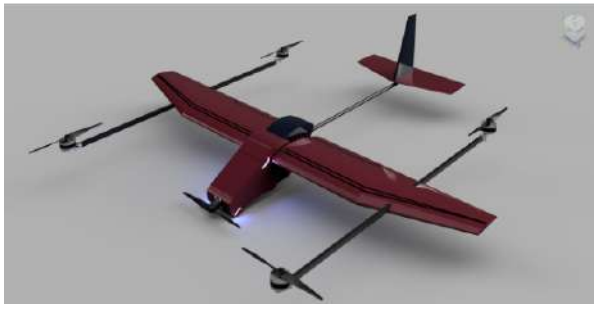
9.2

Plywood being more dense than Balsa, has been used in regions where there is a high stress concentration. The wing has been covered using a mylar sheet as it ensures a good aerodynamic surface finish over the wing. The space between the ribs has been covered using thermocol which acts as a good impact absorbent. Two spars have been used by us in the wing to add to the structural integrity of the wing. After considering various materials such as wood and aluminum, It was observed that carbon fiber possesses the highest strength-to-weight ratio among those materials, and hence we have chosen CF rods as the spars for the wing. The spacing between two adjacent ribs decreases as we move away from the fuselage. There are three main reasons behind increasing the rib density towards the tip of the wing:-

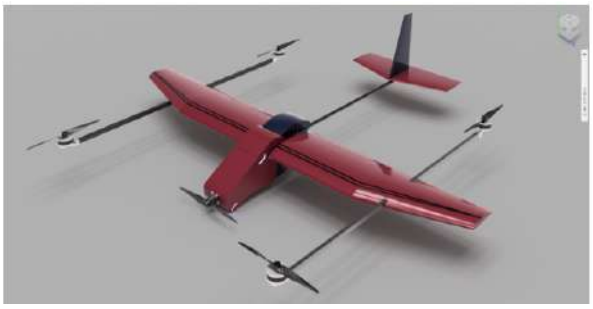
- Aerodynamic Considerations: Aerodynamically, decreasing rib spacing towards the tip can help mitigate the effects of wingtip vortices. Wingtip vortices are created due to the pressure difference between the upper and lower surfaces of the wing, particularly at the wingtips. Increasing rib density towards the tip can help minimize the adverse effects of these vortices by providing additional structural support and reducing the tendency for flow separation.

- Control of Wing Twist: Aerodynamic forces can induce wing twists, particularly under certain flight conditions, such as during maneuvers or in turbulent air. Increasing rib density towards the tip can help control wing twists by providing more structural rigidity in areas where twists are more likely to occur, thus improving overall aerodynamic performance and stability.

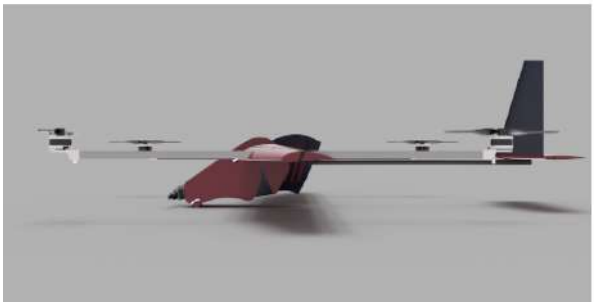
- Manufacturing and Assembly Considerations: Depending on the construction method used for the wing, such as composite layup or metal fabrication, varying rib density may simplify manufacturing and assembly processes. Increasing rib density towards the tip may allow for more precise control over wing shape and stiffness, facilitating manufacturing and ensuring consistent performance.



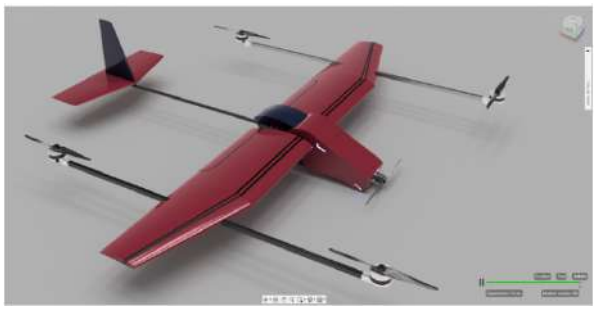
9.3



9.4



9.5

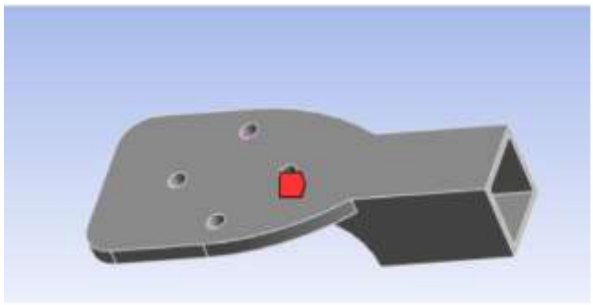


9.6

Next, comes the manufacturing of the fuselage. Carbon fiber had been chosen by us to make the fuselage as carbon fiber composites offer greater design flexibility as compared to conventional materials, allowing for the creation of complex shapes and streamlined aerodynamic profiles. This flexibility enables aerodynamic optimization, which can significantly reduce drag, ultimately enhancing the aircraft's performance and operational capabilities. It was ensured that the CF composite used is quite thin so that we could optimize the weight.

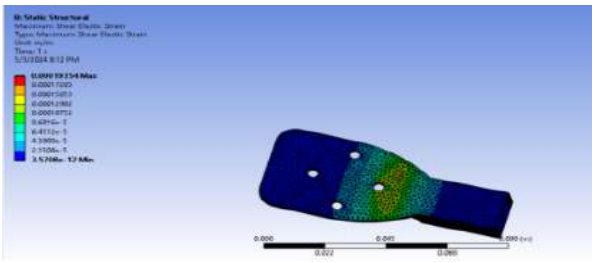
During the manufacturing of the prototype which we had developed for initial testing purposes, one challenge faced by us was to ensure the aerodynamic profile of the chosen airfoils NACA 4412 (for the wing) and NACA 0006 (for the horizontal and vertical tail). One of our primary objectives was to verify the results that we had obtained from XFLR5 and CFD simulations for which we had to ensure that we could accurately map the profile of the airfoil onto a physical construct to act as ribs for the prototype wing. Since thermocol was not used as the material for the prototype wing ribs, the most efficient solution to this challenge would be to develop a custom hot wire cutter styrofoam cutting machine. Using a regulated dual DC power supply, It enabled us to precisely control the temperature of the nichrome wire to achieve a good surface finish for all the ribs that we have cut from the parent thermocol block.

Mass optimization of motor mount: the CAD of the motor mount from Fusion 360 was imported as a step file into the Static Structural module of Ansys.



9.7

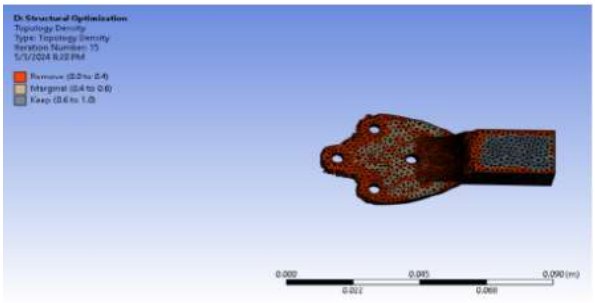
Once the step file had been used to define the geometry, the material of the motor mount was set to Aluminium. Following this, the mesh was generated by using appropriate sizing parameters. The locations of the 4 bolts were defined to be the points of application of force and the rectangular tube section was defined to be the fixed support. The solution for this model was then computed, and the maximum shear stress, maximum shear elastic strain, and the safety factor results were calculated.



Shear strain

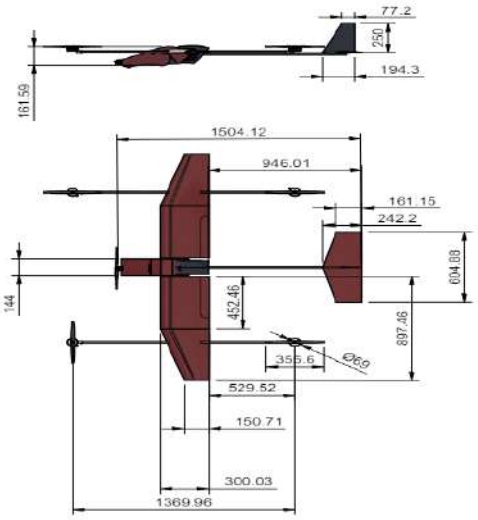
9.8

The safety factor came out to be 15 which suggested that we had a lot of scope to optimize the mass. We proceeded to perform the topology optimization and set the percent mass to retain as 50

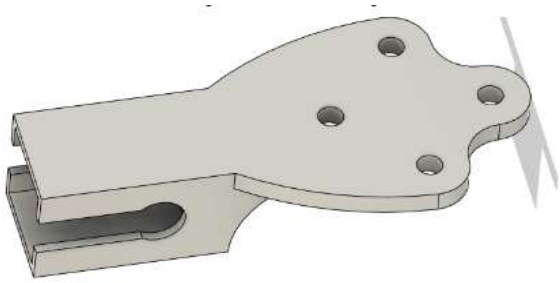


9.9 Topology density (50 percent except for boundaries)

Ansys removed the regions undergoing minimal stress near the area of the bolts and also created slits on two faces of the rectangular tube section. We then modeled the geometry suggested by the topology optimization of Ansys in Fusion 360 and imported the updated motor mount CAD back into Ansys to verify the mass optimization results.



9.11



9.10

The results obtained verified that we had reached the optimum state of mass optimization, as any further removal of mass would result in the formation of high-stress concentration regions between the bolt holes, resulting in the tearing failure of the motor mount.

10 FLIGHT TESTING

10.1 OVERVIEW

To ensure that all the controls are correctly set and the adjusted parameters contribute to a better transition and safer flight. Critically analyze the flight data and repeat iterations in mechanical and electrical components to eliminate the issues identified.

The Quadcopter prototype was tested at the Gymkhana grounds of IIT Bombay to check the execution of the transition from VTOL mode to fixed wing mode and back. The flight was manually controlled using telemetry implemented by TX16S Radiomaster transmitter and RC receivers. The controls assigned to the transmitter switches were:

10.1

To ensure safety, the complete testing was implemented in 3 stages: basic takeoff and landing in VTOL mode, a single transition from VTOL to fixed-wing mode and back, and a longer flight involving repetitive transitions between VTOL and fixed-wing modes, mainly QSTABILIZE AND FBWA.

Stage 1: Basic takeoff and landing in VTOL mode (13:07- 1 min 2 seconds)
A preliminary test was conducted to ensure proper mechanical fitting and motors were functioning per the telemetry commands. The motors were remotely armed using the transmitter, and the thrust was uniformly increased for takeoff in the VTOL mode. The quad motors were successfully tested and worked. However, the forward propeller was not tested. This test was successfully conducted, and no significant issues were identified.

10.2

Stage 2: Single transition from VTOL to fixed-wing mode and back (13:20- 1 min 43 seconds)
Having completed the first testing stage, the aim now was to check the transition from VTOL to fixed-wing mode and the working of the forward propeller, which was crucial to the mission moving forward. After the initial arming of the motors and takeoff, the quadplane was held at 6 m for about 10 seconds before transitioning into the FBWA mode. The expected outcome was a combined operation of quad motors and the forward propeller for 5 seconds (per the parameters Q-TRANSITION-MS and Q-ASSIST-SPEED), following which the quadplane solely operates in the fixed-wing mode.
The actual flight was reasonably successful, with the forward transition taking about 5 seconds, and the fixed-wing mode FBWA was smoothly implemented. The backward transition was subsequently implemented tried (FBWA to QLOITER). It was expected that the quadplane would eventually slow down and maintain its current location. However, during the actual flight, it was observed that the quadplane did not hold its position and began moving in abrupt directions. The quadplane was soon transitioned back into FBWA to ensure safety, and this problem was duly noted. After subsequent analysis, it was discovered that GPS was necessary to implement QLOITER mode due to its positional requirements. The quadplane was then missing a GPS, which caused QLOITER to fail.

GPS is important in QLOITER mode because it enables the quadplane to maintain its current location, heading automatically, and altitude. GPS plays a crucial role in providing accurate positioning data, which is essential for the quadplane to hold its position effectively when the control sticks are released. A good GPS lock and a solid

position estimate are vital for achieving optimal performance in QLOITER mode, allowing the pilot to control the quadplane's position using the control sticks. This reliance on GPS ensures that the quadplane can maintain stability and precision in its flight, enhancing its overall performance and usability.

A non-conventional transition was experimented with from FBWA to manual mode, resulting in the motors' sudden switching off, causing the quadplane to lose altitude and crash. This was not completely unexpected, so the altitude of this transition was kept to a minimum, causing the quadplane to sustain negligible damages and be ready for the next stage since the main aim of carrying out smooth transitions front and back from VTOL to fixed-wing was achieved.

10.3

Stage 3: A more extended flight involving repetitive transitions between VTOL and fixed-wing modes (13:40- 3 min 30 seconds)
After the success of the first two stages of testing, we sought to implement the third stage, simulating the path that the quadplane would be expected to follow in the final competition. The quadplane was maneuvered through a path with sharp turns and was transitioned between quad and fixed-wing modes five times to ensure the repeatability of the transition maneuver and a safe overall flight. Some additional problems that were not noticed in the previous stages were identified. Parameters like Q-TRANSITION-MS and Q-ASSIST-SPEED, initially set to default, were adjusted slightly to fine-tune the transitions and overall flight based on the observations from the flight data. No significant issues were identified in the electrical part until the motor clamp failed, and the motor was dismounted from its place, causing the quadplane to lose balance and crash. The motor clamp failed due to the melting of the PLA, which was used in its 3D printing. This was a breakthrough discovery, as two significant downsides were identified. Firstly, the material used for the clamp needed to be more suitable, and secondly, the heat generated by the electrical components should be managed.

To mitigate the problem of heat accumulation, several approaches were discussed and subsequently worked upon:

- **Allow for adequate ventilation and airflow:** Optimized spacing between heat-producing components promotes air convection rather than trapping hot air against surfaces. The heat-generating parts were placed to allow proper air circulation, causing heat to dissipate faster without accumulating at a point.
- **Strategic component placement:** The physical placement of components is crucial for managing heat accumulation. If the heat-generating ICs and regulators were placed together at the center of the dense boards, heat would accumulate, ultimately overheating the entire circuit. Dispersing these uniformly across the structure will prevent the formation of hot spots. Breaking up concentrated heat generation will avoid overload conditions in any localized area.
- **Use of heat-dissipating materials:** Specialized thermally conductive compounds can help transfer heat away from hot electronic parts to mitigate temperature rise. Applying thermally conductive paste or film between high-power components and their heat sinks improves surface contact and thermal transfer. Thermally conductive outer casings were used to promote outward radiation rather than internally trapping heat. Effective use of these types of targeted materials will aid heat rejection.
- **Position electronics away from heat sources:** Avoid operating electronics like radiators, ovens, burners, or other heat-producing equipment. We ensured plenty of open clearance space around all exterior surfaces for efficient outward heat radiation. Using electronic devices in hot environments would also add to the cooling burden; hence, the components were placed accordingly.
- **Active cooling systems:** Beyond passive measures, specialized active cooling systems

could have been required to protect exceptionally heat-sensitive devices or electronics that operate in hot ambient conditions. Options like thermoelectric coolers, heat exchangers, and liquid recirculation that offer precise temperature control to counter heavy thermal loads were considered. Still, the above measures proved sufficient to handle the heating issue.

10.4

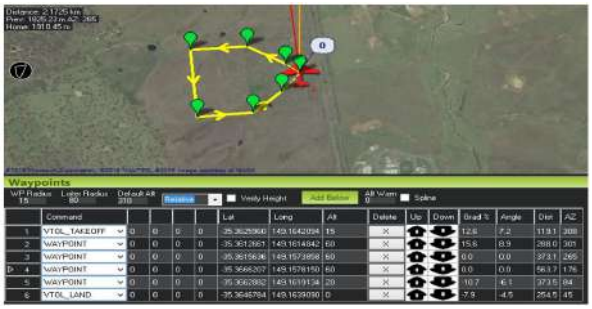
10.5

11 AUTOMATION

11.1 AUTOMATION TESTING

To test for automation of UAV, all we needed was to upload a simple mission on the Ardupilot mission planner since the flight was tested manually beforehand. A simple waypoint completion mission was required to test for the proper working of GPS and the flight controller in AUTO mode.

Before directly testing on the vehicle, it was necessary to test if the parameters set on the cube were correct for autonomous flights. This was achieved by using the Ardupilot mission planner SITL. The SITL within the mission planner can connect to a 'software version' of the cube, making it possible to connect it to the mission planner, alter the parameters, and simulate the mission.



11.1

The commands used in this case are a little different from a normal quadcopter. The transition happens automatically after taking off in VTOL mode. Unless absolutely given the command to do a VTOL transition, the UAV will complete the entire mission in fixed-wing mode and land in VTOL mode on receiving the land command.

After a successful testing on SITL, we moved on to testing on the actual UAV. The testing unfolded in the following steps:

1. GPS connection:
   Making a secure GPS connection was necessary for autonomous flight testing. At the initial stage, issues were identified with making a GPS connection and achieving a satellite lock. The issues are mentioned in detail below:

   - The GPS wasn't able to connect properly to the cube.
   - The here3 GPS initially showed a blinking blue light, which disappeared after some time. This essentially indicated that the connection between the GPS and cube was not established properly.

   Possible reasons for the above 2 problems and how we fixed it:

   - The GPS connectivity issue can happen because of high resistance offered in the signal wire often at soldering junctions. Using DMM indicated that the voltage available at the Pixhawk port was not getting to the GPS, possibly due to those high resistances.
   - The fixture was carefully using less solder material, enough to hold the wires together while not offering extra resistance to the signal.
   - Also, since a CAN port is being used to connect to the GPS, although the GPS-TYPE was set to Auto, it was quite unable to set up the GPS. Setting it to the desired type, DroneCAN, in this case, worked properly, and we were able to get a GPS lock in an open ground.

2. Initial conditions:

   When being operated in default settings of stream rate, before pushing the waypoints, the drone approaches the west direction unknowingly. The reason for this was found to be this:

   - Before pushing the waypoints, MAVROS takes origin to be at the global origin, which is present in Africa. This causes the UAV to move towards the east before the waypoints are pushed.
   - The slow stream rate is the reason why waypoints go slowly and give time to the drone to move toward the global origin. Thus, we increased the stream rate so that the waypoints get pushed at a much faster rate.

3. Safety:

   Firmware Safety Requirements:

   - The Pixhawk flight controller provides several safety features for drones.
   - These features include a motor arming safety mechanism during take-off and landing.

- The flight controller also has RTL, FailSafe, and GeoFence modes that respond to signal loss or motor failure scenarios.
- RTL mode can be programmed to return the drone to its launch location, while FailSafe mode ensures the drone's safety in emergencies.
- The GeoFence mode transmits the drone's current location to the operator or ground control station.
- A beeping alarm as soon as the drone leaves the geofence to alert everyone around the area
- To enhance drone safety, it is recommended to implement mechanical and electronic safety features such as propeller guards, fuses, emergency cut-off switches, and a return to-home feature

Operational Safety Requirements:

- Pre-flight checks on all components, including structural integrity and stability tests, are crucial for safe drone operation.
- Personal protective equipment such as insulating gloves and helmets can also contribute to ensuring the safety of the drone's operators and other personnel around the drone.
- Motor arming safety, RTL, FailSafe, and GeoFence modes can be customized to meet the specific requirements of different drone applications.

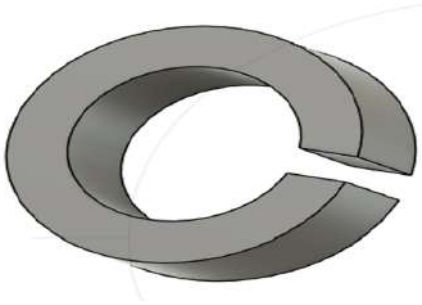
Design Safety Requirements:

- Structural Integrity- performing finite element analysis for the structure under static loads and analyzing vibrational frequencies of arms
- Stability, control, flight, and navigation performance- testing on simulations, followed by prototype testing

12 LEARNINGS

- **Relief cuts- instead of offset:**
  The basic idea conceived for a rod and clamp arrangement comprises a 3D-printed holder manufactured with a specific offset. This offset is often inaccurate, and the rod fits too tight into the holder or is too loose to stay in position. Even if it is very slightly loose, the angular allowance is scaled up by the factor of distance from the holder (d=r*theta), and the rod, along with the components placed, vibrates to a greater amplitude, causing high instability. The quadplane is unsafe to operate in this condition, and this problem must be addressed effectively to move forward.

  A practical solution to this problem is the use of the relief cut technique, which involves cutting a slit along the length of the holder, which expands as the rod is passed through, and contracts back to hold it tightly in position, owing to its elasticity. This technique was implemented successfully, effectively minimizing the persistent vibrations to an insignificant level.



L.1

- **Importance of kinetic and dynamic analysis:**
  Theoretically formulated ideas may not be kinetically viable, which might cause issues if not identified at the earlier stages, hindering progress significantly. Thus, it is worthwhile to perform kinetic and dynamic analysis of all the moving parts and re-iterate with each problem identified.
- **3D printing techniques:**
  The model to be 3D printed can be rendered in many possible orientations, and the perfect choice may only sometimes be the most basic one. Suppose the model is going to experience a tensile loading along its length. In that case, it may be suggested that the model be rendered horizontally because the tensile strength of the fibers is much larger than the force of attraction between adjacent layers. If rendered vertically, the layers might fail, which may cause problems subsequently. Instead, if a model is rendered horizontally, the tensile loading is opposed by the tensile strength of the PLA fibers, which has a far lesser chance of failing.

13 REFERENCES

- Ardupilot
- Prevention of overheating of power system
- Heat sink and their application in ESC's
- Design, Modeling and Hybrid Control of a QuadPlane