Eliot Ducksbury

Design of a Fixed-Wing Aircraft to Fly on Saturn's Moon, Titan

This report is submitted in partial fulfillment of the requirements for MEng Aeronautics and Astronautics, Faculty of Engineering and Physical Sciences, University of Southampton

2021.04.29

Word Count: 9911

Supervisors:
Professor Bharathram Ganapathisubramani
Doctor Sean Symon

Declaration

I, Eliot James Ducksbury, declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research. I confirm that:

1. This work was done wholly or mainly while in candidature for a degree at this University;
2. Where any part of this thesis has previously been submitted for any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. None of this work has been published before submission.

Acknowledgements

I would like to thank my supervisor, Professor Bharathram Ganapathisubramani[1], for engaging with the idea of investigating fixed-wing flight on Titan, helping me define the project scope, and plan my research accordingly. I would like to thank my supervisor, Doctor Sean Symon[2], for providing weekly reviews of my plans and gathered data. I would like to thank Charles Dhenin[3] for sharing his understanding of the XFLR5 application and helping me troubleshoot related issues.

[1] Professor, Experimental Fluid Mechanics, Faculty of Engineering and Physical Sciences
[2] Lecturer, Department of Aeronautics and Astronautics, Faculty of Engineering and Physical Sciences
[3] Part II Undergraduate Student, Department of Aeronautics and Astronautics, Faculty of Engineering and Physical Sciences

Abstract

In 2027 NASA plans to launch its Dragonfly mission. The octocopter will fly in Titan's atmosphere, landing at different locations on the surface to take samples. On Titan, the surface gravity is seven times smaller than Earth's, requiring lower lift for flight and reduced structural loads; the sea-level air is four times as dense and of three times less viscosity, resulting in flights of greater Reynolds numbers. The atmospheric profile is generated using data from NASA's Huygens probe and the design principles to produce an optimised aircraft for this environment are found.

The aircraft flight and payload requirements are based off those from the Dragonfly mission. A standard aircraft design process is followed and analysed using the inviscid flow software, XFLR5. An IPython notebook is used to store and analyse iterative development parameters. An XML file conversion method is developed to transfer aircraft geometry from the IPython notebook into XFLR5. The aircraft is analysed in straight-and-level cruise, with the speed of this condition itself optimised, and in static and dynamic stability conditions.

A complete aerodynamic design of a fixed-wing aircraft is produced. It is found that a minimised surface area is required to reduce the dominant viscous drag component. The efficiency of the configuration is calculated as significantly greater than Dragonfly, providing a much greater range and endurance. The fixed-wing aircraft is found to be reliable in static and dynamic stability. However, the roll mode is heavily damped such that the attitude control system is expected to require increased power.

Table of Contents

List of Abbreviations and Acronyms

Symbol	Description
GUI	Graphical User Interface
RTG	Radioisotope Thermoelectric Generator
HASI	Huygens Atmospheric Structure Instrument
NASA	National Aeronautics and Space Administration (of the USA)
CAD	Computer Aided Design
NACA	National Advisory Committee for Aeronautics (of the USA)
STOL	Short Take-Off and Landing
CFD	Computational Fluid Dynamics
LSA	Light Sport Aircraft
VII	Viscous-Inviscid Interaction
FSMF	Forward Semi-Minor Fraction
Config	Configuration

All symbols are defined in the text at the first appearance.

Introduction

There is great public and scientific interest in exploring extra-terrestrial bodies which can support life and contain atmospheres that can help to explain the evolution of the solar system. The atmosphere and surface features of Saturn's moon, Titan, may have been formed by similar geological processes to those on Earth [1]. Therefore, on-site studies of Titan could lead to an improved understanding of Earth's evolutionary process.

In 2027, NASA plans to launch its "Dragonfly" mission [2]. The rotorcraft will arrive in approximately 2034 and begin sampling atmospheric and surface properties. It will cruise at an approximate speed of $10\ \mathrm{ms}^{-1}$ and altitude of $500\ \mathrm{m}$ [3]. The range of each flight varies between a few kilometres up to $60\ \mathrm{km}$. Each flight will contain loiter stages where potential landing sites are observed.

Rotorcraft allow for access to landing sites that are inaccessible to fixed-wing aircraft and can hover at altitude to take a reading. However, an aircraft can be designed for short take-off and landing (STOL) conditions and to loiter in a local area for measurements. In general, rotorcraft are much less efficient than a fixed-wing aircraft designed for a similar flight profile [4] and are comparably limited in range and service ceiling. Therefore, it is surprising that a fixed-wing configuration has not been strongly considered for the mission.

Titan's atmospheric and gravitational conditions allow for different design principles. Universal aerodynamics and flight mechanics laws still apply but dimensionless terms take values very different to those experienced together on Earth. This project seeks to design a conventional fixed-wing aircraft capable of flying in these conditions. It is expected that these different conditions will result in different optimal lifting surface parameters than a comparable aircraft in Earth's atmosphere. The surface gravity is seven times less [3], reducing the required lift and the structural loads experienced. The sea-level gas is four times as dense, and of three times lower viscosity. This results in an increased Reynolds number [5] and is expected to require design for reduced boundary layer drag. The speed of sound is 60% lower than on Earth, such that at higher altitudes, transonic effects are expected to become significant.

Available data is used to produce an atmospheric profile. These conditions are then used to determine a flight envelope and stored in an IPython notebook. These parameters are then used directly in aerodynamics calculations in the same notebook, allowing updates in conditions to immediately update the same calculations. Conventional knowledge is used to select an initial configuration for the conditions presented by the known operating requirements. Multiple wings are designed to meet these initial targets, first using analytical calculations in the IPython notebook. A tail is then designed to balance each wing, also by calculations in the IPython notebook. To gain more accurate aerodynamic results, a panel-based interpolator of 2D flow solutions, XFLR5 is used. To allow rapid analysis of a change in aircraft geometry, an aircraft to XFLR5 readable file converter is developed and implemented in the IPython notebook. The configurations are iterated until they meet the initial lift and moment requirements. The results are studied to determine the proportion of each type of drag on Titan and this knowledge is applied to selecting the best characteristics of each configuration to combine into a single aircraft. The geometry of the aircraft is then used to select more efficient flight conditions, then the geometry is improved to meet these new conditions. A simple power analysis is conducted to compare the aircraft to Dragonfly. Finally, the static and dynamic stability of the aircraft is analysed, with possible improvements studied.

As a consequence of this process, computational design tools are developed. These are useful in the design of any aircraft, not just one designed for Titan. These are made available in the Appendix.

1 Mission Conditions and Requirements

The design of a fixed wing aircraft requires several parameters to compare against, specifically atmospheric profiles and aircraft planform dimensions. In addition, many parameters are dependent on others, meaning a change in one value can cascade into changes in many other parameters. For example, a change in cruise speed will change the lift requirements of the aircraft and therefore its geometry. It is therefore beneficial to make use of a computational tool, where values can automatically update if another that they depend on does too.

Multiple methods of implementation are available. Spreadsheets allow an intuitive relationship between cells to be visually created, automatic value highlighting to instantly indicate whether a design is feasible, and easy comparison of many configurations in a list. However, implementing complex equations becomes obscured by cell references and syntax requirements. A simple GUI computer program allows all equations to be encapsulated in code, which refers to parameters with identifying names. Through this method, values can be entered then only relevant calculations made upon a button press, protecting the designer from overwriting code. At each stage of design, the code may be updated with new calculations and results saved through file writing. However, program development is likely to divert time away from aircraft design to bug fixes and does not keep an active log of design progress unless specifically programmed. Alternatively, an IPython notebook allows serial entry and overwriting of variables and functions, while maintaining a document style commentary through embedded code blocks and markdown. At any point, previous parameters can be input to a function, the result recorded. Additionally, the same benefits of meaningful identifiers is shared with the GUI approach. Furthermore, Python has many modules available that allow fast analysis, for example, graphical plotting. However, a single notebook hosting the many iterations of design can become long, making it difficult to find previous function declarations, for example.

Therefore, a shared approach was used. An IPython notebook was used as described above in combination with a spreadsheet for directly comparing designs, without having to scroll or re-reference a previous iteration to view its parameters and results. The design of an aircraft is determined by its mission profile – what it's carrying and to where. This is decided and estimated in Section 1.1 then the IPython notebook is made use of to determine the atmospheric conditions in Section 1.2.

1.1 Scientific Mission and Payload Requirements

The mission objectives are considered to be similar to those of NASA's Dragonfly mission [3] – taking readings including spectrometry, pressure, temperature, and humidity – at different locations on the surface near the equator. Similar targets shall exist for the fixed-wing aircraft. The range shall be 40km and the flight duration no longer than an hour, flying at an altitude selected within the region of 0.5-4km, the selection of this will depend on the best operating conditions for the design. The aircraft must be able to land between dunes spaced 3km apart, where each is about 150m tall at a 5-degree slope.

To select a landing site, the aircraft will be required to loiter at 500m so that it can photograph and collect data on the site before descending. This will require high lift at a low speed, similar to landing requirements. One site is photographed before turning around and landing at a previously photographed site.

The Dragonfly payload is described in [3], with each component being unique to the module however, are similar to those already in use on current missions, such as the Mars rover, thus allowing the geometry and mass of these components can be estimated. The properties of these components are described in Table 1.1. An additional mass of 5kg is added to account for other components, for example, communications, processors, batteries, and internal structures which hold these components in place. This defines a total payload requirement of $m_p = 101.5\,\mathrm{kg}$. These mass and dimension estimates are used in Section 2.2 to determine the payload configuration and lift requirements. This then dictates the wing and fuselage sizing in Sections 2.3 and 2.6.

Table 1.1: Payload masses and geometries estimated from similar components

Component	Mass (kg)	Dimensions (cm)
Radioisotope thermoelectric generator (RTG) **[6]**	45+5(fuel)	64 Ø x 66
Mass spectrometer **[7] [8]**	45	35 x 45 x 26
Geophysics and meteorology package	Negligible	External and negligible
Camera suite **[9]**	1.5	18 x 16 x 3
Inertial measurement unit	Negligible	Negligible

1.2 Atmospheric and Planetary Conditions

The atmospheric conditions are calculated from known values in the following sections and are summarised in Table 1.2.

Table 1.2: Atmospheric and planetary properties of Titan

Symbol	Property	Value	Units
$\boldsymbol{g_0}$	Surface-level gravitational acceleration	1.354	$\mathrm{m\ s^{-2}}$
$\boldsymbol{p_{atm}}$	Surface-level atmospheric pressure	1.47×10^5	Pa
$\boldsymbol{T_{atm}}$	Surface-level atmospheric temperature	94	K
$\boldsymbol{\rho_{atm}}$	Surface-level atmospheric density	5.4	$\mathrm{kg\ m^{-3}}$
$\boldsymbol{R}$	Gas constant	290	$\mathrm{J\ kg^{-1}\ K^{-1}}$
$\boldsymbol{\gamma}$	Specific heat ratio	1.40	
$\boldsymbol{\mu}$	Atmospheric viscosity	6×10^{-6}	$\mathrm{Pa\ s}$
$\boldsymbol{a_0}$	Surface-level speed of sound	195	$\mathrm{m\ s^{-1}}$

The Huygens Probe measured temperature and pressure [10] in Titan's lower atmosphere, allowing the density to be calculated from this data through the ideal gas law [11], producing an error of 2.8% [12]. This is a small error, such that designing to these conditions will cause very little difference in performance than predicted. The density profile is given by [13], showing an approximately linear relationship between the surface density of 5.4 $\mathrm{kgm^{-3}}$ and the density at 100km altitude of several times lower magnitude. The precision of these data is typically to three significant figures. The density profile of the altitude up to 10km using ideal gas relations is shown in Figure 1.1. This allows the cruise conditions to be defined, which is significantly less dense than the sea-level take-off condition.

Using this profile, the Reynolds number at any altitude within this region can be computed. The design take-off and cruise speed are expected to be of the order of magnitude of $V = 10\ \mathrm{ms^{-1}}$ and the main wing root chord length of approximately $c_r = 1\ \mathrm{m}$, therefore producing Reynolds numbers approaching $Re = 10^7$ in both take-off and cruise conditions. Consequently, towards the trailing the aircraft's surfaces, the flow is considered turbulent [14]. These turbulent conditions allow the aircraft to create lift at high angles of attack without stalling [15], which can be made use of in the take-off and landing conditions. The skin friction caused by this turbulence is therefore identified a significant cause of the drag produced by the aircraft and the wing must be designed with its reduction as a primary target.

With the same HASI data, the speed of sound with altitude can be calculated within the flight limits, as shown in Figure 1.2. It is expected to be inefficient to fly at transonic speeds due to the mass requirement of a propulsion system able to deliver enough power and the short range of the flight dictates that a very small period of time would be spent flying at this speed, before having to decelerate for landing . However, the aircraft's propeller tips may approach the speed of sound and lose efficiency. Therefore, the selection of the propeller speed and diameter must consider this.

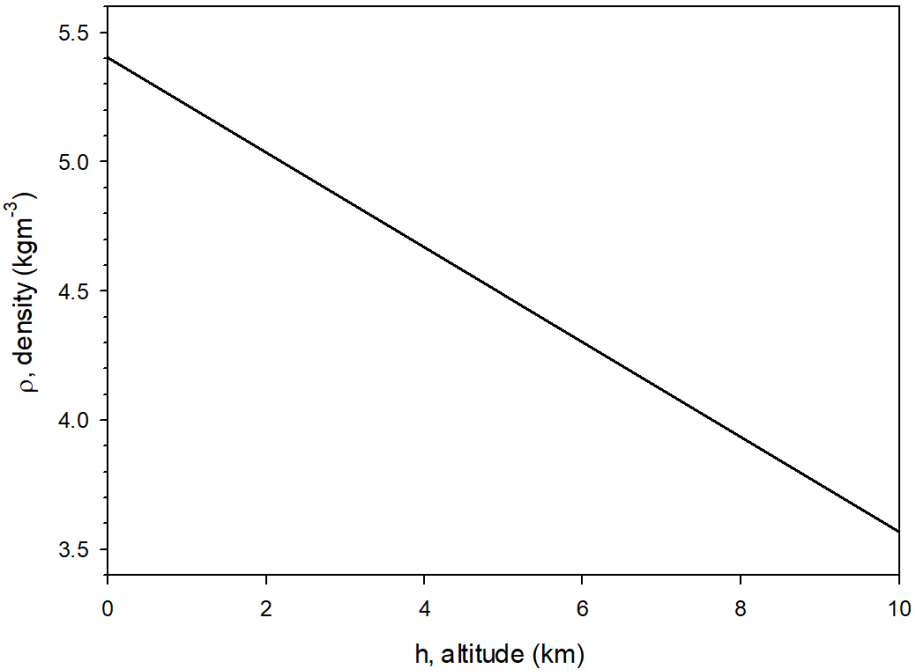


Figure 1.1: Density profile with altitude up to 10km as measured by HASI data [13]

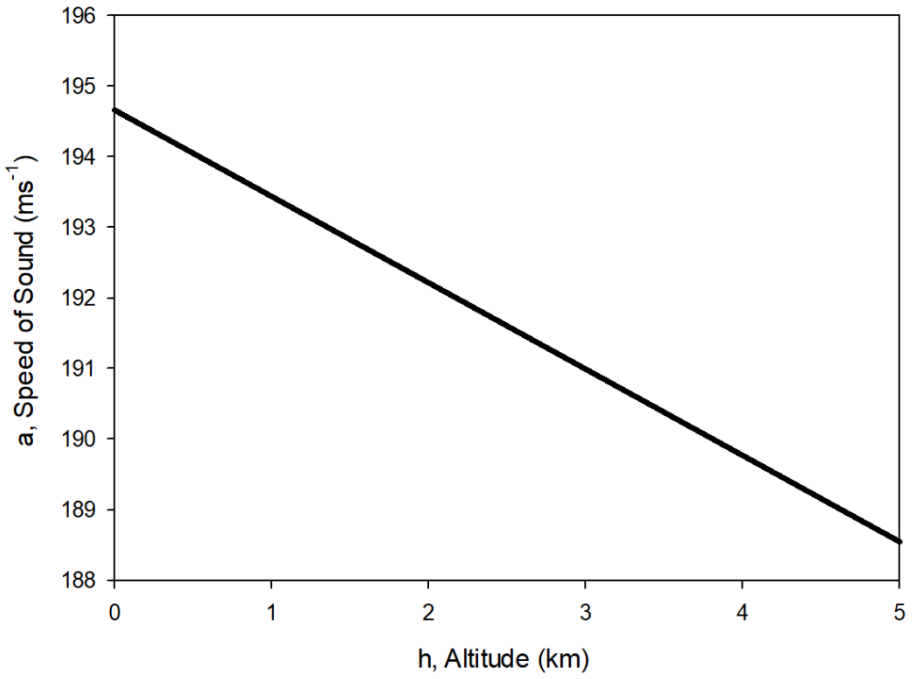


Figure 1.2: Speed of sound with altitude profile up to 5km as measured by HASI data [13]

By Newton's law of universal gravitation, the difference in gravitational acceleration between surface-level and 5km altitude is 0.46%. Therefore, during the aircraft's flight, the gravitational acceleration is assumed as a constant $g = 1.354\ \mathrm{m\ s^{-2}}$. This error is very small such that it has a negligible effect on aircraft performance.

At low altitudes, predicted wind speeds are $V_w = 2\ \mathrm{ms^{-1}}$ at maximum. Given that the order of magnitude of expected flight speed is greater than the wind speed, the flow is modelled as steady and the direction of flight is not limited by wind direction. At the take-off and landing stages, the wind shear near to the dunes may have been a concern. However, the wind speed is low enough for this effect to be considered negligible.

2 Conceptual Design

In the previous section, the mass and geometry of the payload have been found. This allows the lift requirement to be estimated and the payload layout to be configured with respect to reducing fuselage size and delivering a suitable static margin. Alongside a mission definition, the relationship of altitude with density and speed of sound have been found, allowing a decision on flight altitude and velocity to be made. With respect to Earth's forms of drag, boundary layer viscosity is predicted to be greater. Therefore, the size of the aircraft must be minimised to reduce the impact of this drag. These results are used in this section to produce multiple aircraft designs.

2.1 Initial Configuration Analysis

Selecting an initial configuration design is possible when referencing Table 2.1. As noted in Section 1.2, the dominant form of drag is expected to be boundary layer turbulence from the high Reynolds number present in the atmosphere. It is therefore hypothesised that the most efficient wing design will be of high aspect ratio, with as a thin a chord as structurally possible, though this will increase parasitic drag and structural mass from supporting the long cantilever wing. It is important to try multiple configurations to test this hypothesis.

Table 2.1: Design choice mapped against performance parameters. Arrows indicate whether the design choice increases or decreases the performance parameter. Colours indicate whether this is desired behaviour, with green indicating positive effects.

	Stall Speed	Control in Stall	Top Cruise Speed	Structural Mass	Parasitic Drag	Elliptical Distribution	Induced Drag	Runway Distance	Longitudinal Stability	Roll Stability	Yaw Stability	Manoeuvrability	Tailplane Influence
Wing Loading	↑		↑	↑					↑			↓	
Aspect Ratio				↑	↑		↓	↓		↑		↓	
Sweep											↑	↓	↑
Taper				↓		↑	↓						↑
Twist		↑				↑	↓						
Wing Height										↑		↓	↑
Dihedral										↑	↑	↓	

Referring to Table 2.1, an elliptically tapered, untwisted, small area, high aspect ratio planform is identified as ideal for the operating conditions and mission profile identified in Section 1. The elliptical wing delivers the reduced induced drag identified in Table 2.1. However, is difficult to manufacture and design in CAD tools. To ensure that these difficulties are limited, the wing is untwisted because twist is also difficult to design and manufacture. As noted in Table 2.1, this choice reduces control in stall. To offset this loss, the longitudinal stability reduces the dependency of the aircraft of control in stall. The high aspect ratio reduces induced drag but increases the parasitic drag. This is offset by reducing the area of the wing while maintaining the same aspect ratio, thereby reducing the surface area available for shear forces. A high aspect ratio increases the structural mass required to support the outboard sections. This is offset by the elliptical taper, which is thicker at the root and less massive at the tips. On Earth, the maximum typical aspect ratio a beam wing structure can support is $\mathrm{AR} = 9$ [16]. It would be possible to increase the aspect ratio given the lighter structural weight on Titan and the use of support structures could be employed. However, the behaviour of wings at higher aspect ratios is less known – especially the flutter damping on Titan – and it is unnecessary to increase when a reasonable design may be achievable at this value.

The aircraft is expected to operate in well below transonic conditions such that standard incompressible aerofoil theory applies. The high aspect ratio reduces the dependency of a high sectional lift coefficient such that a symmetric aerofoil can be used at a low setting angle of attack or a slightly cambered aerofoil at almost zero setting angle of attack.

The high aspect ratio provides greater roll stability as does a high wing, shown in Table 2.1. A high wing is selected in this initial configuration design such that if the aspect ratio should be reduced, roll stability is still provided. If the first few iterations result in an aircraft that is too stable, anhedral can be added to improve manoeuvrability. Should the reduction in the yaw-to-roll coupling stability of adding anhedral [17] become prevalent, sweep is identified by Table 2.1 as being able to offset this.

The tailplane shall have an elliptical planform (for both the horizontal and vertical stabilisers) of high aspect ratio to minimise induced drag. The effectiveness of the horizontal stabiliser is reduced when it is affected by the downwash of the main wing, which has a high vertical position. The downwash of the main wing is proportional to sectional lift coefficient and inversely proportional to the ellipticity and aspect ratio [18]; where the former is low and the latter is high from the selected configuration. Therefore, the influence of downwash is expected be limited. This allows the tailplane to be at the same height as the main wing. If downwash is found to be significant, it is necessary to have a high horizontal stabiliser, which increases the structural mass required to withstand the additional moment from weight and drag contributions.

Table 2.2: Summary of main wing and tailplane planform and configuration design choices made in Section 2.1.

Property	Main Wing	Tailplane
Aspect Ratio	High	High
Sweep	None	None
Taper	Elliptical	Elliptical
Twist	None	None
Height	High	Main Wing Height
Dihedral	None	None

The initial configuration design is summarised in Table 2.2. It is important that the wing height is firmly decided. Repositioning the payload completely changes static margin calculations and has cascading effects for wing design. In the IPython notebook, the position of each of the payload components is parametric. This allows the centre of mass to be recalculated upon moving these components, saving the time of manual recalculations and allows the testing of multiple iterations faster. Therefore, the payload layout is established in Section 2.2.

2.2 Initial Sizing

The photography mission requirements are described in Section 1.1. Therefore, the initial cruise altitude is selected as $h_{cruise} = 1\ \mathrm{km}$ because significantly changing altitude to $h_{loiter} = 500\ \mathrm{m}$ throughout the short flight would waste energy due to repeated climbing and descending. The initial cruise speed V is estimated from the range $R = 40\ \mathrm{km}$ and time $t = 60^2\ s$ requirement stated in Section 1.1. Estimating cruise as 75% of the flight time, the cruise speed V can be estimated by Eq. (2.1) as,

$$V = \frac{1}{0.75}\frac{R}{t} \tag{2.1}$$

This results in $V = 14.8\ \mathrm{m\ s^{-1}}$, which is $4.8\ \mathrm{m\ s^{-1}}$ faster than the estimated best range speed of Dragonfly [3], allowing this aircraft to conduct a similar mission in two-thirds of the time.

The payload geometries defined in Table 1.1 are placed within a fuselage configuration in Figure 2.1. This diagram allows an approximate payload centre-of-mass to be determined, assuming that all components are of uniform density, which will cause a slight error. If the density of components were to vary so much, the denser side should be positioned towards the front of the aircraft to increase static stability. The positive x direction denotes increasing distance from the front of the aircraft. Excluded from the figure are the additional masses of unspecified components, which are assumed to be point masses which fit 5cm behind the RTG, resulting in a centre of mass estimate that has a slight forward error. The largest component is placed towards the front such that it is at the thickest point of an overlayed shape, which is compact and possible to form an approximately axial (teardrop) aerofoil (the lowest drag shape possible for storing the payload), the actual geometry of the fuselage is expected to deviate from this shape where it is necessary for ground clearance. There is plenty of space remaining for unaccounted-for components, control mechanisms, and structural members.

The centre of mass $\mathrm{C.O.M.}$ (shown in purple in Figure 2.1) therefore has the coordinates from the (red) reference point. Each axis, generically d, has the centroid position $\bar{d}$ described by Eq. (2.2),

$$\bar{d} = \frac{\sum m_i d_i}{\sum m_i}, \tag{2.2}$$

as the sum of all component masses m_i and the distance d_i of its centroid from the reference point. Using Eq. (2.2), the payload centre of mass is found as Eq. (2.3),

$$\mathrm{C.O.M.}\,(\bar{x}, \bar{y}, \bar{z}) = (62.4, 0, -30.5)\ \mathrm{cm}. \tag{2.3}$$

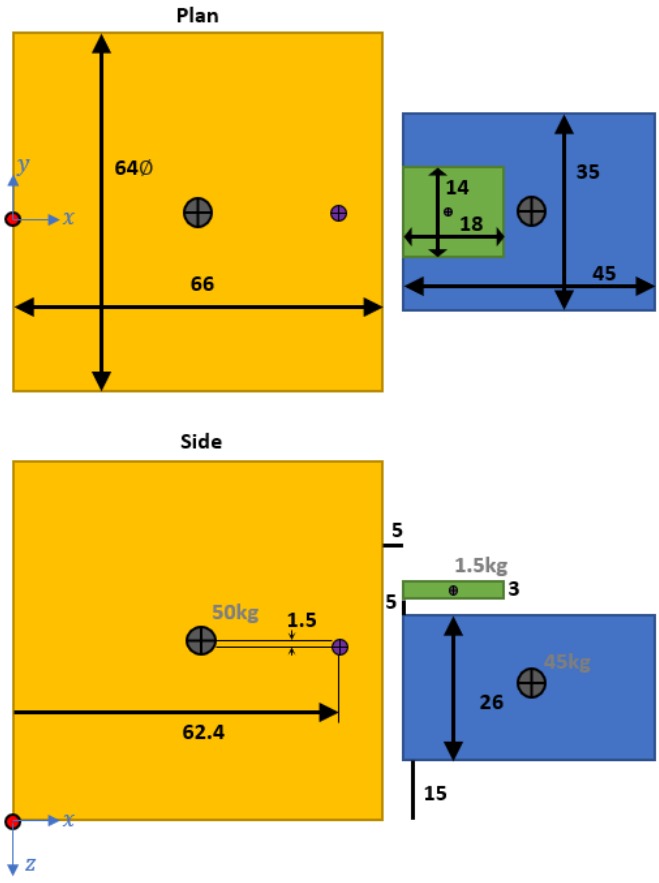


Figure 2.1: Fuselage payload configuration with dimensions (in cm) and masses (in kg). The reference point on each view is shown by a red dot, the calculated centre of mass is shown by the purple crossed circle.

For greater static longitudinal stability, the centre of mass of the overall aircraft should be further forwards of the centre of lift. The contribution of the wing and tail will shift the overall centre of mass aft of the payload centre of mass. It should be noted that the structure required to support the wings need only support 7 times less weight than an analogous aircraft on earth. The negative lift of the tail in trimmed flight shifts the centre of net lift forwards of that of the wing. Using an initial safety margin, the centre of lift of the wing is positioned and sized. The centre of mass of the payload is estimated to be at 54% of the payload length. Taking all of the previous statements into account, the centre of lift should be positioned at approximately 70-75% of the payload length. Figure 2.2 shows the initial design placement of the main wing centre of lift. The centre of lift is significantly clear of payload components such that its structure can be supported.

The fuselage should not dictate the chord length of the, more aerodynamically important, wing but should provide a guideline for geometric limits, such as thickness and chord length. The profile for the fuselage is selected based on wing positioning and payload geometry in Section 2.5.

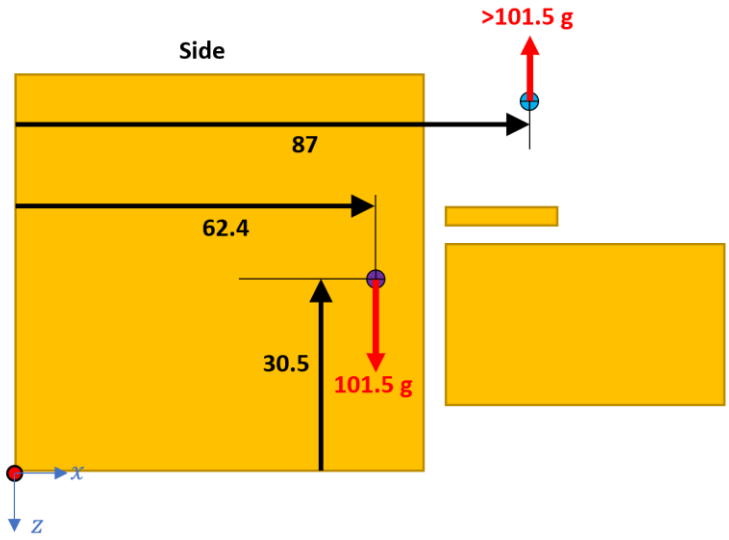


Figure 2.2: Main wing centre of lift (blue crossed circle) overlayed on the reduced detail payload geometry

For an aircraft on Titan, it is difficult to estimate an initial structural weight of the aircraft because no aircraft data have been published for this application. To proceed with the design, some estimation must be made. This could produce a large error in the magnitude of forces required. Using [19], a light sport aircraft (LSA) is most applicable to compare against. To arrive at a reasonable initial estimate, the structural (empty weight) of these aircraft can be linearly interpolated for the payload mass. The LSA's structural mass is interpolated as 112 kg. The engine mass is subtracted, a small engine for a LSA is estimated as 80 kg [20]. No motor mass will be added as it is considered negligible. The structural mass can be lighter as the weight required to support will be seven times less due to Titan's gravitational acceleration. However, to include a safety margin, the structural mass estimate is half of the LSA's. This yields an estimated structural mass of $m_s = 16$ kg. After adding the payload mass estimated in Section 1.1 of $m_p = 101.5$ kg, the total mass of the aircraft is estimated as $m = 117.5$ kg. The product of total aircraft mass and Titan's gravitational acceleration yields a straight and level lift force requirement of $L = 159$ N.

The lift is governed by Eq. (2.4),

$$L = \frac{1}{2}\rho V^2 S C_L, \tag{2.4}$$

where S is the main wing area and C_L is the lift coefficient of the main wing. This equation assumes that the main wing produces nearly all of the required lift, assuming that the lift of the tailplane is negligible. This yields

a product of wing area and lift coefficient $SC_L = 0.279\ \mathrm{m}^2$ requirement for cruise conditions. Compare the Titan lift requirement to that of the same mass on Earth, $L_\oplus = 1.15\ \mathrm{kN}$. For the same cruise speed, in an air density of $\rho_\oplus = 1.225\ \mathrm{kg\,m}^{-3}$, the Earth requirement is $SC_L|_\oplus = 8.59\ \mathrm{m}^2$. This means that the product of wing area and lift coefficient must be thirty times larger on Earth. This result is desirable because Section 2.1 emphasised the need for a small wing area to reduce viscous drag.

2.3 Main Wing

The main wing was studied using XFLR5, a three-dimensional interpolation of Xfoil results. This uses a viscous-inviscid interaction (VII) method. This considers the flow to be in two distinct regions – an inviscid flow away from the surface and a viscous boundary layer close to the surface [21]. This method is appropriate because viscous drag is expected to be of significant interest. This method is less accurate once the boundary layer transitions to turbulent flow [22]. Recall that turbulence is expected in these conditions as predicted by Section 1.2.

Optimisation of an aerofoil provides very little improvement in comparison to optimisation of the planform and configuration as shown by the effects identified in Table 2.1. Therefore, a simple comparison is made between two different cambered aerofoils, NACA 1212 and 2412. Both are analysed at the expected operating Reynolds number of $Re = 10^7$. A low angle of attack was expected due to the low lift requirement so the aerofoil of less camber is preferred due to being able to produce a lower lift without having to be set to a negative incidence, yet is still slightly cambered such that lift is still able to be generated without an angle of attack.

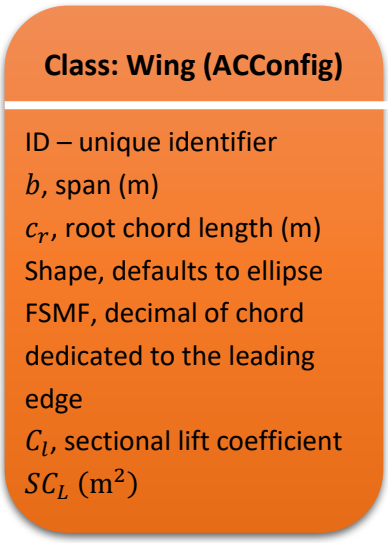


Figure 2.3: Visual representation of the wing class properties, where a unique ID representation is generated, and SC_L is calculated from the other parameters.

Many different wing configurations are tried, so a class is created in the IPython Notebook to store all properties of each, as shown in Figure 2.3. Appendix B.1 provides the constructor code. Upon creating an instance of this class, the properties are used to calculate the corresponding SC_L and create a graphical figure of the wing planform, as demonstrated in Figure 2.4. The figure is used to identify whether the expected wing has been generated, allowing input verification, and if it appears to be an unreasonable design, for instance, if it is of excessive span. The chord length c at any span position y of an elliptical wing is given by Eq. (2.5),

$$c(y) = c_r \sqrt{1 - \left(\frac{2}{b} y\right)^2}, \tag{2.5}$$

where c_r is the chord length at the wing root. The method used to calculate the SC_L in this first stage is simply the product of area S and infinite span lift coefficient C_l as calculated by Xfoil. The area is the integral of the chord length along the span b, as shown in Eq. (2.6),

$$SC_L = 2C_l \int_0^{\frac{b}{2}} c(y)\,\mathrm{d}y\,. \tag{2.6}$$

Table 2.2 identifies the need for a high aspect ratio. To evaluate this claim, three planforms are produced and analysed: high aspect ratio, low aspect ratio (both at zero setting angle of attack), and a medium aspect ratio with a setting angle of attack. The iterations are shown in Table 2.3, with those achieving an SC_L within 5% error being selected. The plots of the selected planforms are shown in Figure 2.4, with all appearing as reasonable designs. All three planforms consist of an untwisted NACA 1212 aerofoil with a forward semi-minor fraction FSMF of 0.25. The first iterations are for developing a high aspect ratio wing, achieved at ID = 002. The second set are for a low aspect ratio wing, achieved at ID = 006. The final set are for the wing with an incidence angle of $\alpha_s = 0.5°$, achieved at ID = 008.

Table 2.3: Initial configuration manual iteration records, colour coding shows those that produced errors less than 5% for target SC_L

Target SC_L (m^2): 0.2791

ID	b (m)	c_r (m)	α_s (°)	C_l	SC_L (m^2)	SC_L % difference	Comments
000	2.8	0.7	0.00	0.1065	0.1639	-41.3	Initial wing created.
001	5.6	0.7	0.00	0.1065	0.328	17.5	Found to be too slender. The aspect ratio is reduced in the next iteration.
002	5.0	0.7	0.00	0.1065	0.293	4.98	**High AR configuration selected.**
003	3.0	1.2	0.00	0.1065	0.301	7.85	Too much lift is generated. The root chord and span are reduced in the next iteration
004	2.9	1.0	0.00	0.1065	0.243	-12.9	The area is reduced too much. Only the span is increased in the next iteration to limit turbulent boundary layer length.
005	3.0	1.0	0.00	0.1065	0.251	-10.1	The lift is increased, span is increased in the next iteration.
006	3.2	1.0	0.00	0.1065	0.268	-4.00	**Low AR configuration selected.**
007	2.8	0.7	0.50	0.1633	0.251	-10.1	Planform with a non-zero α_s. Lift is too low so span is increased in the next iteration.
008	3.0	0.7	0.50	0.1633	0.269	-3.62	**$\alpha_s \neq 0$ configuration selected.**

After three basic wing planforms are designed, all are updated to more accurate SC_L values using finite wing theory. The lift coefficient is now calculated by Eq. (2.7),

$$C_L = \frac{(\alpha_s - \alpha_0)}{\left(1 + \frac{\alpha_0}{\pi \mathrm{AR}}\right)(1+\tau)} \frac{\mathrm{d}C_L}{\mathrm{d}\alpha}, \tag{2.7}$$

and then multiplied by area. The zero-lift angle of attack α_0 and lift curve slope $\frac{dC_L}{d\alpha}$ *for* the NACA 1212 aerofoil are obtained in Xfoil. This equation assumes that the setting angle of attack is not above or approaching the stall angle of attack, such that the lift curve slope is linear, which is true for angles of attack less than 10°. The

lift efficiency factor τ is approximately zero for an elliptical wing. Each wing is again iterated until it satisfies the target SC_L within 5% error. This iteration exclusively involves increasing setting angle of attack, keeping the planform geometry the same.

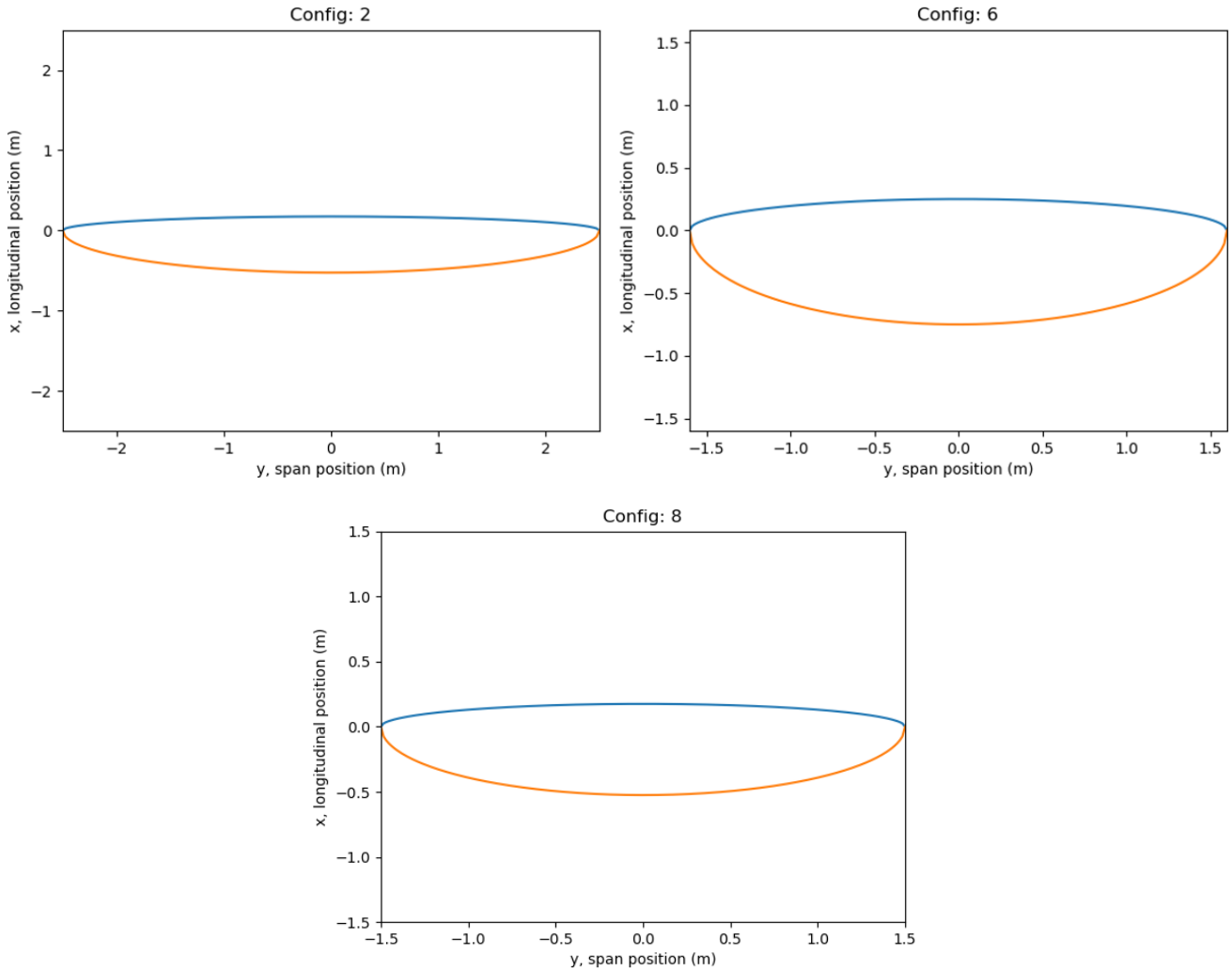


Figure 2.4: Plan views of the three successful wing planforms. The blue and orange curves are the leading and trailing edges respectively. Each wing is centred on (0,0). The configuration ID above each planform corresponds to the ID in Table 2.3.

2.4 Tailplane

With multiple wings generated, an aircraft class is developed in the computational tool to store multiple instances of the wing (*ACConfig*) class. This class relationship is shown in Figure 2.5. Appendix B.2 provides the constructor code. This also stores the relationships between them for example, the distance l between the main wing and horizontal tail aerodynamic centres. This is expected to be of a few metres. Therefore this distance was defined as $l = 3$ m; from this, a target $S_T C_{L_T}$ of the horizontal stabiliser can be defined by expanding the moment balance equation given by [23] into Eq. (2.8),

$$C_{M_0} + C_W(h - h_0) - C_{L_T} S_T \frac{l}{S\bar{c}} = 0, \tag{2.8}$$

where C_{M_0} is the coefficient of moment about the aerodynamic centre of the main wing, C_W is the weight coefficient, and C_{L_T} and S_T are the lift coefficient and planform area of the tail respectively. With a wing, weight, and distance l defined, an aircraft's target value of $S_T C_{L_T}$ can be determined. To this end, the wing

class is updated to calculate its moment coefficient and mean aerodynamic chord. Instantiating an aircraft class with a wing, mass, and distance l, automatically produces this target. This allows new instances of the wing class to created until a tailplane that meets this target within 5% error is found. For example, the high aspect wing, wing 2, was calculated to require $S_T C_{L_T} = -0.0180\ \mathrm{m}^2$, thereby requiring a horizontal stabiliser of negative camber or angle of attack.

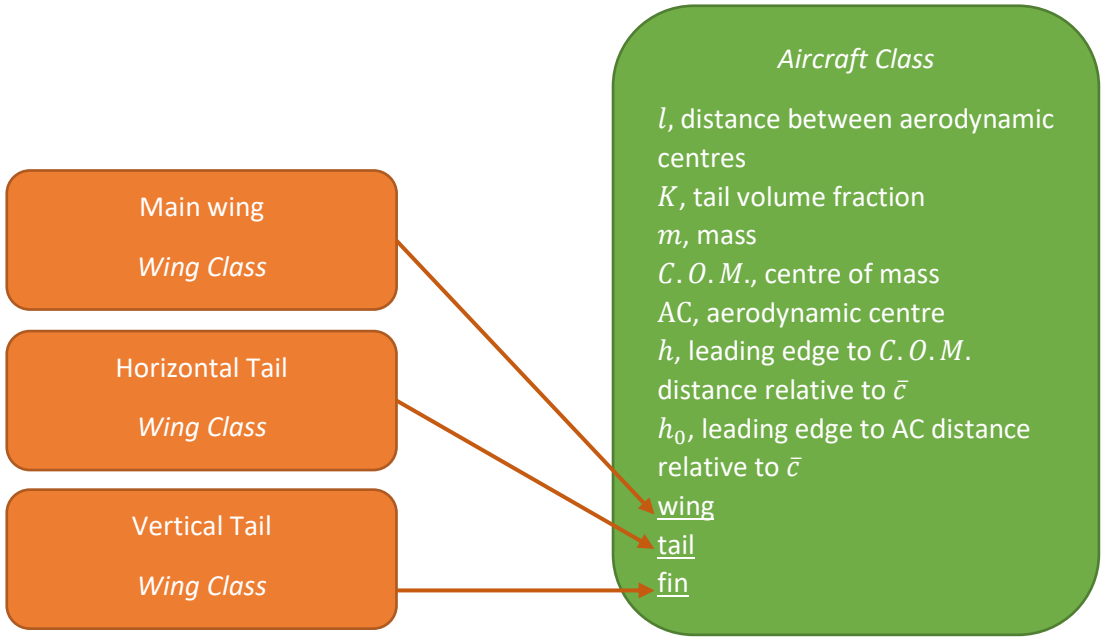


Figure 2.5: How instances of the wing class are linked by the aircraft class in the IPython notebook

Two aerofoils are considered, a NACA -1212, which is an inversion of the cambered main wing, hence the minus sign; and a symmetric NACA 0012. The NACA 0012 is selected based on the availability of data in computational tools, such as XFLR5. Using an inverted aerofoil is deemed to add unnecessary complexity to the semi-automated process. The iterations for the tail aiming to balance wing 2 are shown in Table 2.4. A more negative angle is required for symmetric aerofoil to approach the same $S_T C_{L_T}$ as the cambered aerofoil, which is expected. Note how sensitive the lift is to angle of attack changes when compared to a typical aircraft on Earth. Comparing iterations 0 and 1, with the same area, a change of $\Delta\alpha_T = 1°$ halves the lift coefficient. This highlights the need for precisely actuated control surfaces. A similar iteration of tailplane design is executed for wings six and eight. All wings are given an equal aerodynamic centre separation of $l = 3\ \mathrm{m}$.

Table 2.4: Iteration of surface area times lift coefficient for the tailplane corresponding to wing 2. An error less than 5% is reached in iteration 7, the 17th instance of the wing class.

Wing 2 Target $S_T C_{L_T}$ (m^2): -0.01798

ID	b_T, span (m)	c_{T_r}, root chord (m)	Aerofoil (NACA)	α_{T_s}, setting angle (°)	$S_T C_{L_T}$ (m^2)	$S_T C_{L_T}$ % difference
0/9	1.6	0.4	-1212	-1	-0.0788	338
1/10	1.6	0.4	-1212	0	-0.0383	113
2/11	1.2	0.4	-1212	0	-0.0262	45.7
3/12	1	0.4	-1212	0	-0.0204	13.5
4/13	1	0.3	-1212	0	-0.017	-5.45
5/15	1	0.3	0012	-0.2	-0.0036	-80.0
6/16	1	0.3	0012	-0.8	-0.0144	-19.9
7/17	1.2	0.3	0012	-0.8	-0.0183	1.78

2.5 Fuselage

To design a fuselage with the lowest drag, the maximum thickness should be kept to a minimum. Therefore, the thickest location on the fuselage should line up with the thickest payload location it has to encase. It is also important to ensure that the wing can be supported by the fuselage, where enough space is provided internally. Figure 2.6 shows the current geometric requirements of the fuselage. The length between aerodynamic centres for all configurations is given by Section 2.4 as $l = 3\text{m}$. The distance between the aerodynamic centre and location of maximum thickness is calculated as 0.246m in Section 2.2.

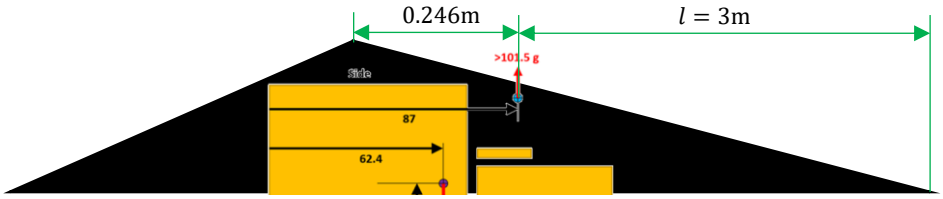


Figure 2.6: Triangular representation of fuselage control points (leading edge, trailing edge, and maximum thickness locations), overlayed with payload geometry (showing only the top half of both). On the maximum thickness and trailing edge x-ordinates are defined. The LE x-ordinate and maximum thickness y-ordinate are yet to be defined.

The only positions that remain to be defined are the maximum thickness height and the leading edge position. If the leading edge and maximum thickness are too close to the payload, the fuselage skin will not cover the payload correctly. If these points are too far away from the payload, unnecessary extra drag is added to the aircraft. It is important that the fuselage is at a zero-lift angle in the cruise condition to minimise induced drag. This can be achieved by developing a symmetric fuselage and setting it to no angle of incidence.

The fuselage profile is defined using an aerofoil profile. A NACA 4-digit library [24] is imported to the IPython notebook to instantly generate geometries and test that they are valid for the payload requirements. A function to overlay the main payload component with the fuselage surface is created and is simply tested with different four-digit aerofoils and lengths. Appendix B.3 provides the code for this function. If the red body intersects the blue fuselage, it is clear that the profile lacks the required length or thickness. The final fuselage produced by this method is shown in Figure 2.7. This is a NACA 0020 aerofoil with a chord length $c_b = 4.5\text{ m}$. This body is thick enough to fit the payload and allow extra space for any support structures and minor payload components.

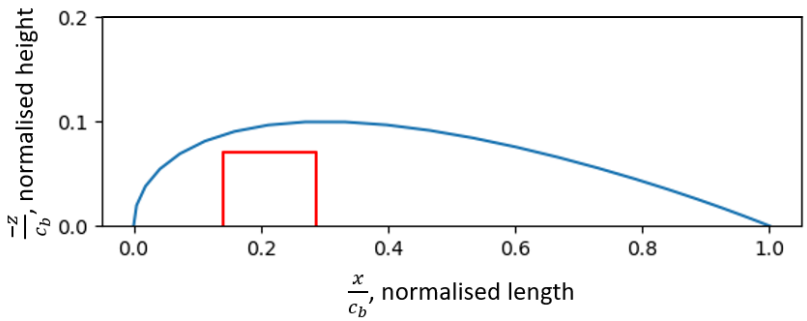


Figure 2.7: Fuselage 2D cross-section scaled to unit length overlayed with the thickest payload geometry

3 Iterative Design using XFLR5

The conceptual design is now complete, with assumptions and analytical calculations made to determine the aircraft speed and geometry for three different configurations. A more accurate prediction of lift and therefore moment balance is made using computational methods. Drag is yet to be calculated. The design choices for reducing drag, in reference to Table 2.1, need validating. The operating speed selected can be refined to minimise drag for the designed aircraft geometry. To this end, Section 3 iteratively updates the aircraft geometry and speed to improve its efficiency. This process is described by Figure 3.1. Geometry and speed decisions are tested in XFLR5, then the resulting lift, drag and pitching moment data are used to inform new geometry and speed choices.

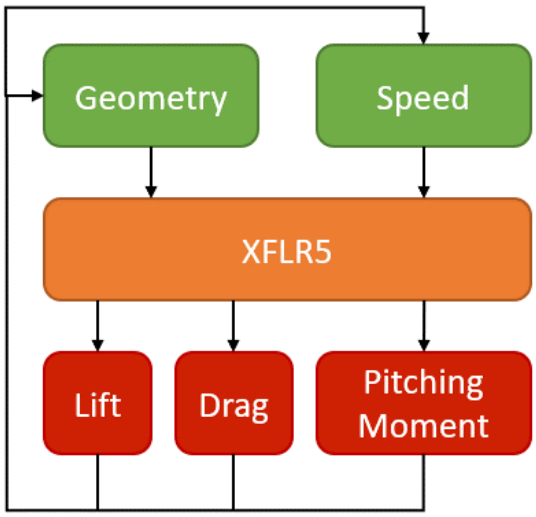


Figure 3.1: Iterative development plan for XFLR5. Lift, drag, and pitching moment results are used to inform new geometry and speed selections.

3.1 Aircraft Analysis

The use of wing analytical aerodynamics theory is widely used to inform design decisions before more refined computational studies are conducted. As described in Section 2.3, the flow can be modelled as inviscid and therefore vortex lattice methods are applicable [25]. XFLR5 allows for the use of multiple panelled wing sections for all major components, using vortex lattice methods, VII, and accounts for downwash effects. It should be recalled that the objective is not to find an accurately designed aeroplane but to find an aeroplane that has accounted for the different aerodynamic conditions on Titan.

For an aircraft to be analysed in its straight and level condition, multiple stages of calculations must be completed. First, the lift, drag, and moment data for any aerofoil sections used must be obtained. This includes any expected Reynolds number and angle of attack of a wing section in the aircraft layout. Data for these are collected by sampling at specified intervals for these parameters within a specified domain. The associated lift, moment, and drag data is obtained by interpolating within the domain. The maximum expected Reynolds number is determined in Section 1.2 to approach $Re = 10^7$ and the minimum requirement, given the small finite chords of the wingtips, is found to be $Re = 5 \times 10^4$. The maximum setting angle of attack for a lifting surface is expected to be less than $\alpha = 5°$, which defines the bounds for the angle of attack domain. The smaller the change in value before sampling for aerofoil data, the more precise the lift, drag, and moment data but the longer the compute time.

A short study is carried out to determine the necessary sample spacing of angle of attack and Reynolds number for appropriate precision with reduced computing time. The approximate computing time experienced is 10 minutes, though this is expected to vary on different machines. Reduction of this time is not essential so only

three samples of each parameter are taken to get an indication of the effects of each. It is expected that a smaller sample spacing for either parameter will cause the measured lift coefficient to be more precise. A more precise lift coefficient does not necessarily correlate to a larger or smaller lift coefficient value, just the uncertainty has decreased. It is also expected that precision will be negligibly affected by angle of attack sample spacing due to the linear nature of the lift curve before the stall condition is reached.

Figure 3.2 plots calculated lift coefficient against varying angle of attack and Reynolds number sample spacing. All varied Reynolds number spacings were measured at a fixed angle attack spacing of $\Delta\alpha = 0.25°$, all of the varied angle of attack spacings were measured at a fixed Reynolds number spacing of $\Delta\mathrm{Re} = 250$.

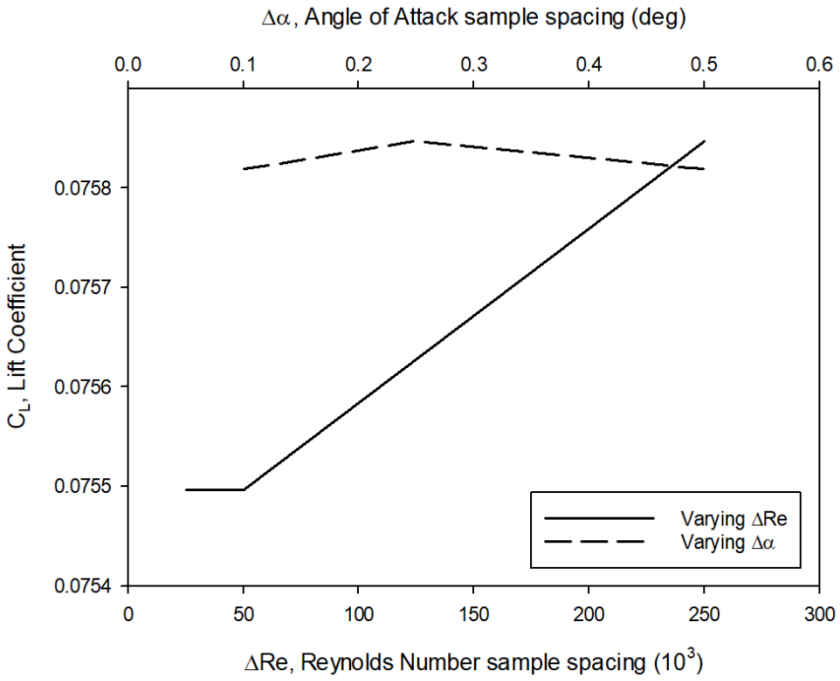


Figure 3.2: Effect of angle of attack and Reynolds number sample spacing on measured lift coefficient. Reynolds number spacing is shown to have the greatest impact.

These results indicate that, as expected, the angle of attack sample spacing have little effect on precision relative to Reynolds number spacing. Therefore a coarse sample spacing for angle of attack can be selected, at $\Delta\alpha = 0.5°$. The range of C_L shown in Figure 3.2 changes by less than 0.0005. Given that the order of area is $S \approx 10^1\ \mathrm{m}^2$, a coarse sample spacing for Reynolds number has little effect on precision of the key parameter, SC_L, which determines whether a design is considered within error for meeting the lift requirements. Therefore a relatively coarse Reynolds number sample spacing is selected as $\Delta\mathrm{Re} = 100{,}000$.

In the IPython notebook, aircraft classes are populated with their associated wings and tails, as shown in Figure 2.5. To define an initial vertical stabiliser geometry, the planform of the horizontal stabiliser is used. A method is developed to split an aircraft into its individual component XML files which XFLR5 can use to represent the geometry. This process is explained in Appendix B.4. This allows the quick switching out of components when repeatedly updating parameters, therefore saving the time of having to manually calculate and input elliptical section data. Each aircraft is assigned a unique identification letter:

A. Wing 2, Tail 17
B. Wing 6, Tail 20
C. Wing 8, Tail 22

The wings cannot be represented as perfectly elliptical because the tips would be of zero length, causing a Reynolds number of zero which XFLR5 cannot interpolate. Therefore the end sections are given a slight finite length.

Each aircraft is analysed using XFLR5's (Type 1) fixed speed analysis at $V = 14.81\ \mathrm{ms}^{-1}, \mathrm{M} = 0.08, \mathrm{Re} = 9.324 \times 10^{6}, \alpha = 0°, \rho = 5.22\ \mathrm{kgm}^{-3}, \nu = 6 \times 10^{-6}\ \mathrm{m}^{2}\mathrm{s}^{-1}$, using the ring vortex (VLM2) method with viscous effects included. For every aircraft, the mass is set to 117.5 kg acting at the same centre of mass, calculated in Section 2.2, regardless of the geometry.

Table 3.1 shows the series of iterations for the straight and level condition, experimenting with parameters until the aircraft is within error of the SC_L target. Drag components are split into induced C_{D_i} and viscous C_{D_v}. The wing planforms remain constant throughout this process, only the angle of attack α and tailplane height are changed. This demonstrates the advantages and disadvantages of each planform. The changes and notes on the results are shown in the comments column.

It is found that the overall SC_L of the aircraft was overestimated by theory when compared to the results of the XLFR5 cruise analysis. This varies with wing planform; for example, aircraft A, with the greatest aspect ratio and area, produces 25% less lift than predicted by analytical calculations; aircraft C has a smaller area and aspect ratio, producing 38% less lift than previously calculated. Aircraft B, with the smallest aspect ratio and area, initially has an error of 89% less lift than predicted and the wing's angle of attack cannot be set to a reasonable setting angle to meet the lift requirement before being close to the stall regime. Therefore, it is concluded that the wing area is too small to meet the requirements.

Both aircraft A and C are able to meet the SC_L requirement at a suitable angle of attack. Aircraft C has a greater lift-to-drag ratio despite having a higher drag coefficient that aircraft A. This is due to the lower wing area of this aircraft, causing less viscous drag. 94% of the drag coefficient of aircraft A is due to viscous drag, compared to only 80% on aircraft C. This supports the hypothesis, made in the Introduction, that low-drag design is dictated by minimising boundary layer viscosity. Therefore surface area must be reduced, as predicted by Section 2.1. The aircraft at the beginning of this iteration stage have a lift-to-drag ratio of approximately 12 and end at approximately 16. If the low viscous drag advantages of aircraft C's low area and low induced drag advantages of aircraft A's high aspect ratio can be combined, then it is expected that the lift-to-drag ratio can be increased.

3.2 Combined Geometry

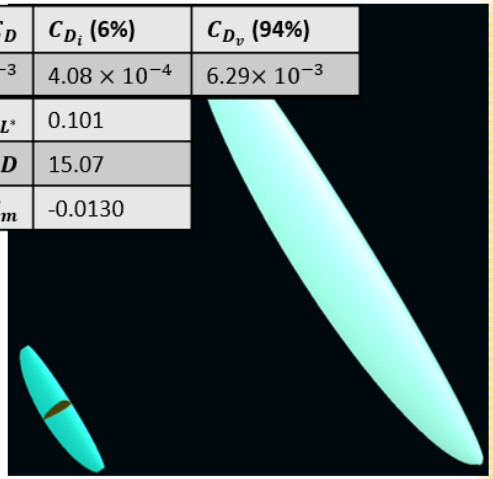
As identified in Section 3.1, the beneficial characteristics of aircraft A and C are combined to produce an improved aircraft. The principle of low area, increased setting angle of attack to compensate, and high aspect ratio of $AR \approx 9$ are followed. The setting angle is set to $\alpha_s = 2°$ as this is significantly below the stall angle, allowing for effective climbing and gives a moderate $C_L = 0.2$. The required area can then be found from the SC_L target as $S = 1.35\ \mathrm{m}^2$.

Basing this new aircraft off aircraft A, the root chord and span are scaled down while maintaining an aspect ratio of approximately 9. The main wing and horizontal tailplane of this aircraft are then refined iteratively to increase lift-to-drag ratio and minimise the moment coefficient. The combined aircraft compared to the predecessor aircraft is shown in Figure 3.3, with a resulting lift-to-drag ratio of 21.3, which is a 21% improvement over Aircraft C's. The viscous drag proportion from aircraft C is maintained, resulting in a lower overall drag.

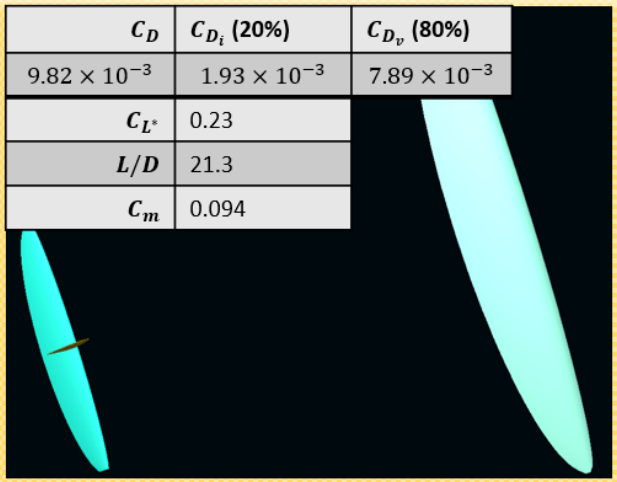
Table 3.1: Three aircraft configurations through a series of iterations using XFLR5. In all aircraft, generated lift is significantly below analytical predictions. Aircraft B is unable to generate the required lift without modifying its planform.

	C_L $\times 10^{-1}$	C_D $\times 10^{-3}$	C_{D_i} $\times 10^{-4}$	C_{D_V} $\times 10^{-3}$	C_m $\times 10^{-1}$	L/D	SC_L $\times 10^{-1}$	ΔSC_L (%)	Comments (α is given in °)
A	0.758	6.548	2.43	6.305	-0.068	11.58	2.08	-25.4	Aircraft A is found to have 30% less lift than predicted. α of the main wing is increased in the next iteration.
A	1.01	6.696	4.08	6.288	-0.130	15.07	2.77	-0.67	Wing 2 at $\alpha = 0.5°$, this generates the required lift.
B	0.604	7.372	4.50	6.922	0.483	8.19	0.305	-89.1	Higher C_{D_i} ∵ lower AR, higher C_{D_v} ∵ larger c, the next iteration increases α.
B	0.886	7.763	8.51	6.911	0.545	11.41	0.447	-84.0	α = 1.0, SC_L is still low, the next iteration uses a larger α.
B	2.01	1.088	3.90	6.981	0.781	18.51	1.02	-63.6	α = 3.0, even at this angle, lift is still too low. The α is increased close to the stall angle in the next iteration.
B	4.84	3.032	2.21	8.242	1.28	15.95	2.44	-12.6	α = 8, this still generates too little lift. α cannot be increased any further. Tail height is increased in the next iteration to decrease the influence of downwash.
B	2.12	1.14	N/A (unsuccessful planform)			18.57	1.07	-61.7	α = 3.0, tail height = 0.5m. L/D has increased but negligible impact on lift produced. Aircraft B is therefore an unviable configuration.
C	1.06	8.803	8.35	7.968	1.02	12.03	1.74	-37.6	C_{D_V} is high despite the short span, long chord causes this due to the high Re. α is increased in the next iteration.
C	1.25	9.100	1.13	7.970	1.11	13.68	2.05	-26.7	$\alpha = 1.4$. Linear interpolation of $SC_L(\alpha)$ for this aircraft yields a requirement of $\alpha = 2.1$. This next iteration's wing is set to this angle.
C	1.68	10.02	2.00	8.019	0.132	16.74	2.76	-1.24	$\alpha = 2.1$. This achieves the required lift.

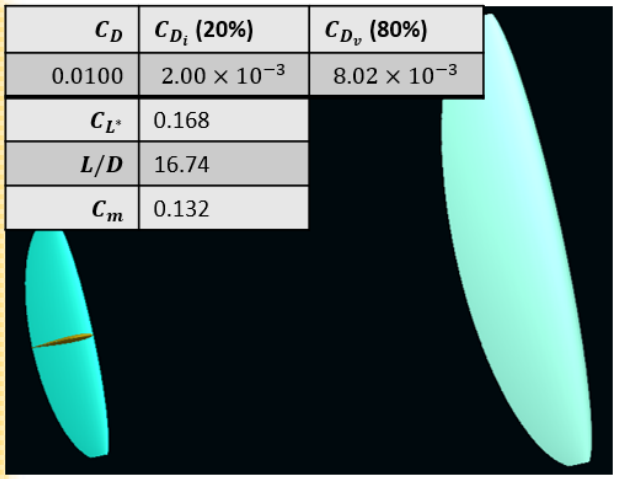
C_D	C_{D_i} (6%)	C_{D_v} (94%)
6.70×10^{-3}	4.08×10^{-4}	6.29×10^{-3}
C_{L^*}	0.101	
L/D	15.07	
C_m	-0.0130	



C_D	C_{D_i} (20%)	C_{D_v} (80%)
9.82×10^{-3}	1.93×10^{-3}	7.89×10^{-3}
C_{L^*}	0.23	
L/D	21.3	
C_m	0.094	



C_D	C_{D_i} (20%)	C_{D_v} (80%)
0.0100	2.00×10^{-3}	8.02×10^{-3}
C_{L^*}	0.168	
L/D	16.74	
C_m	0.132	



Aircraft A	Wing	Elevator	Fin
Shape	Ellipse	Ellipse	Ellipse
b (m)	5.0	1.2	1
c_r (m)	0.7	0.3	0.3
Forward Semi-Minor Fraction	0.25	0.25	0.25
NACA Aerofoil	1212	0012	0012
α (°)	0.5	-0.8	0.0
AR	9.09	5.09	4.24
S (m^2)	2.75	0.28	0.24
l (m)	3.0		

The Aircraft	Wing	Elevator	Fin
Shape	Ellipse	Ellipse	Ellipse
b (m)	3.5	1.7	1.6
c_r (m)	0.5	0.24	0.225
Forward Semi-Minor Fraction	0.25	0.25	0.25
NACA Aerofoil	1212	0012	0012
α (°)	2.0	-0.7	0.0
AR	8.91	9.02	9.05
S (m^2)	1.37	0.32	0.28
l (m)	3.0		

Aircraft C	Wing	Elevator	Fin
Shape	Ellipse	Ellipse	Ellipse
b (m)	3.0	1.6	(1.6)
c_r (m)	0.7	0.4	0.4
Forward Semi-Minor Fraction	0.25	0.25	0.25
NACA Aerofoil	1212	0012	0012
α (°)	2.1	-0.5	0.0
AR	5.45	5.09	5.09
S (m^2)	1.65	0.50	0.50
l (m)	3.0		

Figure 3.3: Combination of aircraft A and C with increased L/D, and maintained 80:20 viscous drag ratio from aircraft C. Aircraft A is on the left and C on the right, the resulting aircraft is highlighted in gold in the centre.

3.3 Improved Flight Condition Selection

The initial cruise velocity is determined by a simple assumption in Section 2.2. Now that an efficient aircraft geometry has been defined, a drag-velocity analysis can be conducted to determine a cruise velocity of lower drag for the aircraft. This analysis assumes no fuselage and only consists of lifting surfaces, with drag only consisting of viscous and induced components. For the aircraft to maximise range, it must fly at the minimum drag speed [26].

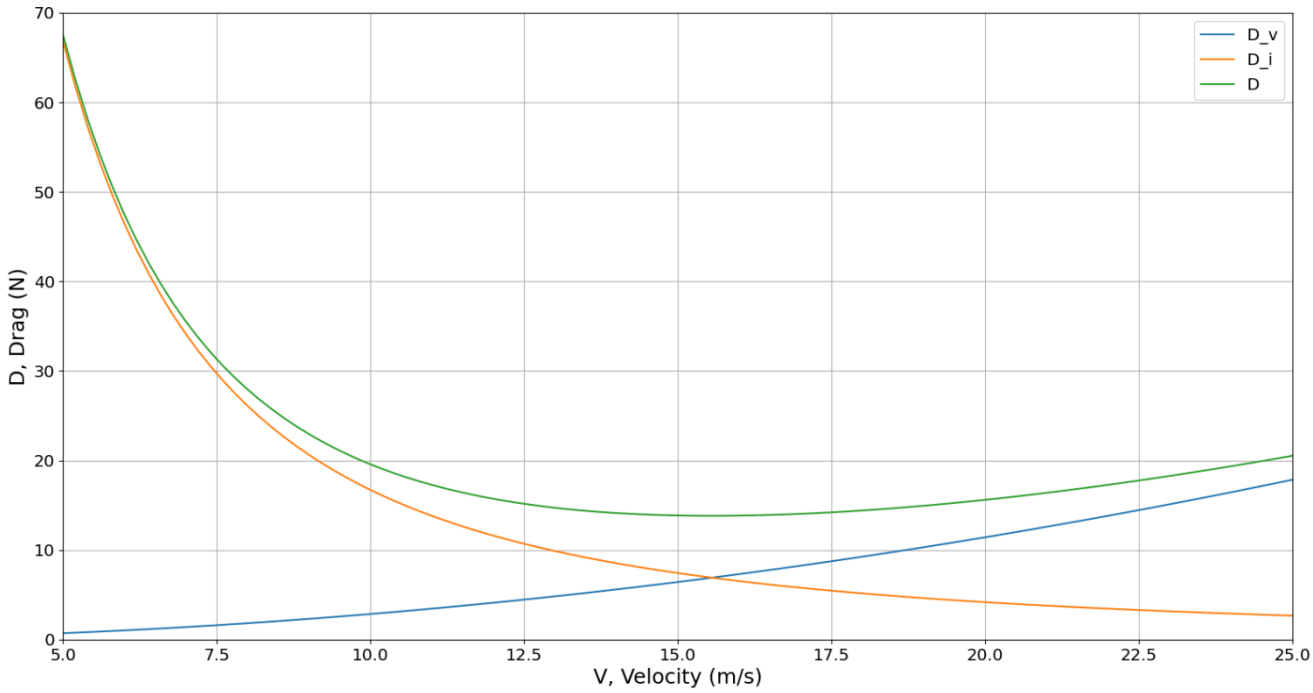


Figure 3.4: Drag-Velocity plot for the aircraft's lifting surfaces, the key identifies the individual drag components and the total drag curve, which is at a minimum where the two components intercept at $15.55\ \mathrm{ms}^{-1}$.

Figure 3.4 is produced following the programming method in [27], where the minimum drag speed is found by summing the two drag components at a range of speeds then identifying the minimum point. Figure 3.4 identifies the minimum drag speed as $V = 15.55\ \mathrm{ms}^{-1}$. The aircraft geometry can now be re-optimised for this new flight condition, as illustrated in Figure 3.1. This is only a slight speed increase of 5.0%, resulting in very little adjustment required It is predicted that once a new optimised geometry is created, the change in cruise speed for maximum range is negligible and therefore will not necessary to re-optimise again for that new speed.

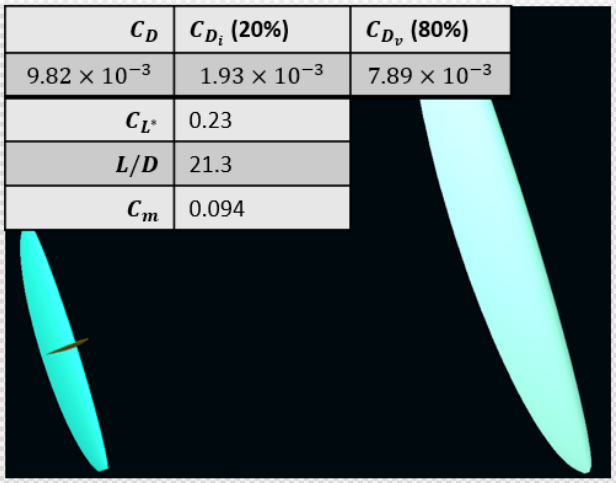
3.4 Improved Geometry

Due to the higher velocity, the SC_L required to produce the same lift is reduced. If the angle of attack remains fixed at $\alpha_s = 2°$, the lift coefficient remains at approximately $C_L = 0.2$. This allows a surface area reduction of 9.5%, which is expected to reduce viscous drag even further. The aspect ratio must be maintained to ensure that induced drag remains low.

Additionally, the tail is adjusted to reduce the moment coefficient to as close to zero as possible. The tailplane is set to two different angles α_{T_s} and the moment coefficient C_m measured by the XFLR5 fixed speed analysis. The C_m-α_{T_s} relationship is assumed to be linear and the tailplane angle which produces no overall moment is interpolated. The selected tailplane angle is given to a precision of $0.1°$, despite the sensitivity demonstrated, to limit production costs.

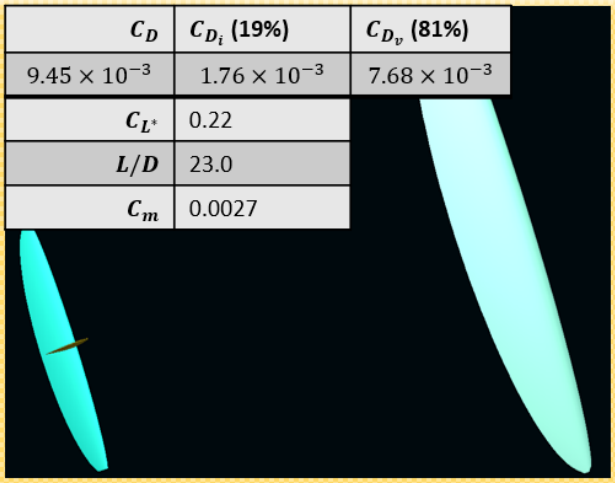
Figure 3.5 shows the resulting aircraft geometry and resulting lift, drag, and moment. At the minimum drag speed, the two drag components are equal, yet the viscous drag remains the most significant. Figure 3.4 shows that viscous drag increases with speed, implying that the aircraft is flying above its minimum drag speed.

C_D	C_{D_i} (20%)	C_{D_v} (80%)
9.82×10^{-3}	1.93×10^{-3}	7.89×10^{-3}
C_{L^*}	0.23	
L/D	21.3	
C_m	0.094	



The Aircraft	Wing	Elevator	Fin
Shape	Ellipse	Ellipse	Ellipse
b (m)	3.5	1.7	1.6
c_r (m)	0.5	0.24	0.225
Forward Semi-Minor Fraction	0.25	0.25	0.25
NACA Aerofoil	1212	0012	0012
α (°)	2.0	-0.7	0.0
AR	8.91	9.02	9.05
S (m^2)	1.37	0.32	0.28
l (m)	3.0		

C_D	C_{D_i} (19%)	C_{D_v} (81%)
9.45×10^{-3}	1.76×10^{-3}	7.68×10^{-3}
C_{L^*}	0.22	
L/D	23.0	
C_m	0.0027	



The Aircraft	Wing	Elevator	Fin
Shape	Ellipse	Ellipse	Ellipse
b (m)	3.2	1.7	1.6
c_r (m)	0.45	0.24	0.225
Forward Semi-Minor Fraction	0.25	0.25	0.25
NACA Aerofoil	1212	0012	0012
α (°)	2.0	0.1	0.0
AR	9.08	9.02	9.05
S (m^2)	1.24	0.32	0.28
l (m)	3.0		

Figure 3.5: Left: previous aircraft geometry parameters and resulting coefficients. Right: optimised aircraft at new minimum drag speed. The surface area has been reduced and the aspect ratio has been approximately maintained.

3.5 Power Plant

A simple assumption of the power requirement is made by assuming that cruise power is used for the entire flight. This causes an error because Section 2.2 estimated that about 25% of the flight is not spent in cruise. It is important to note that flight will not be the only source of power usage, with significant proportions of power being dedicated to communications [3]. The power P required for a single flight is given by Eq. (3.1),

$$P = DV = \frac{1}{2}\rho V^3 S C_D, \tag{3.1}$$

where D is the total drag of the aircraft and ρ is the density in the cruise condition. Substituting the appropriate values from Figure 3.5 into Eq. (3.1) produces a power requirement of $P = 115$ W. The power output of the RTG is estimated as $P_{RTG} = 70$ W [3]. Therefore batteries are required to store charge that will be used on a flight.

Recall from Section 1.1 that a single flight is 40 km of range in 1 hour. On Titan, a single day-night cycle is 384 hours [3], with half of that time being in sunlight, when missions can be conducted. Overnight the battery is charged, with the rest of the day assumed to be using only the RTG's power for communications and other

power intensive tasks. The battery energy density is expected to be $\epsilon = 3.60 \times 10^5$ J Kg^{-1} after degradation from the Earth-Titan journey [3]. Therefore, the total flight time per Titan solar day t^* can be estimated from the battery mass m_b and flight power requirement P in Eq. (3.1),

$$t^* = \frac{m_b \epsilon}{P - P_{RTG}}. \tag{3.2}$$

Section 1.1 budgeted 5 kg for additional components, including the battery. This permits a maximum $t^* = 11.1$ hours of flying per solar day. This time will be reduced if the additional mass budget is assigned to other components. By comparison, Dragonfly is expected to fly 60 km for 2 hours with a battery of six times more mass [3]. Per unit battery mass, the designed fixed wing aircraft is found to have 44 times the range of and 33 times more endurance than Dragonfly. This suggests that a fixed-wing aircraft is more suitable as a research vehicle on Titan. It is expected that as the mass assumptions are refined, this efficiency will reduce but remain greater than Dragonfly's.

4 Stability

The entire aircraft geometry is defined in the previous section. It is shown that the aircraft is suitable for unperturbed flight. Each component of the geometry must now be declared with a mass to consider the exact moment balance and oscillatory response of the aircraft. The positioning of mass with respect to the centre of mass, is related to the aircraft's resistance to movement. In this Section 4.1 considers the static response to longitudinal perturbations, Section 4.2 determines the oscillatory response of the aircraft to perturbances in all directions, and Section 4.3 conducts a study on the use of anhedral to improve the dynamic response.

Using the mass fractions of the overall structural mass of a mid-sized turboprop aircraft [28], an initial estimate of component mass fractions can be determined. The wing mass also includes the mass of the engine nacelles. The total structural mass is estimated as 16kg in Section 2.2, which can be distributed in proportion to these fractions. Table 4.1 contains the initial structural mass estimates for each component, allowing XFLR5 to estimate the inertial matrix using the geometry provided for a given aircraft configuration. Errors will arise from the exact volume distribution of the aircraft, for instance, if the wing area is increased but its thickness and the tailplane area remain constant, its mass fraction would go up and those of the other components would reduce; however, this method considers all masses as fixed so loses accuracy as the "rubber aircraft" gets stretched away from the model mid-sized turboprop.

Table 4.1: Initial component structural mass estimates

Component	Structural Mass %	Absolute Mass (kg)
Wing	59	9.44
Horizontal Tail	6	0.96
Vertical Tail	3	0.48
Fuselage	33	5.28

4.1 Static Stability

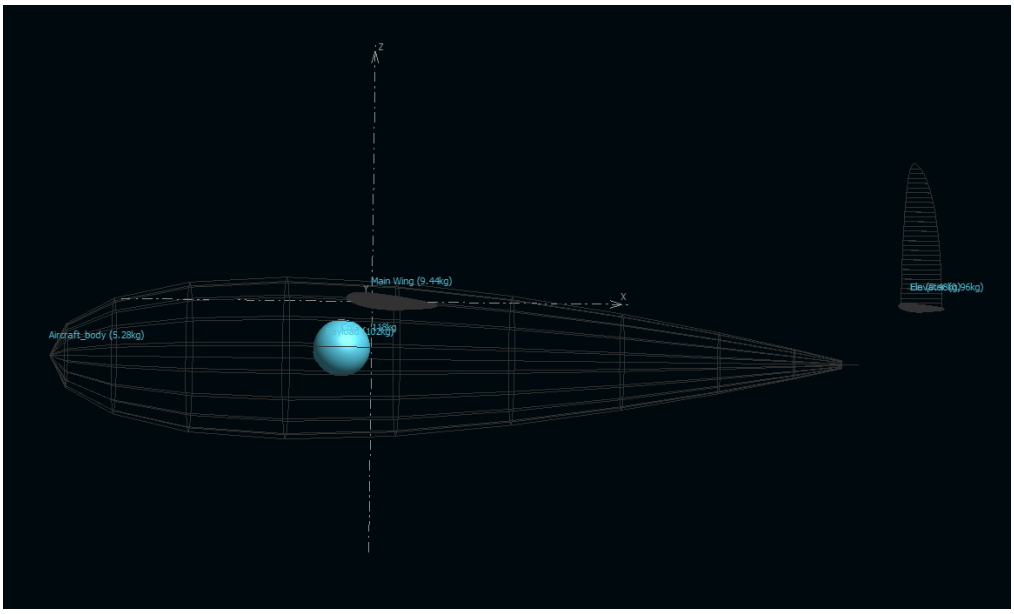


Figure 4.1: Wireframe of components in XFLR5 with masses overlayed. The centre of mass is shown behind the payload point mass (blue sphere) and in front of the aerodynamic centre.

Figure 4.1 displays the calculated centre of mass on a wireframe of the aircraft's XFLR5 representation. The centre of mass is, as predicted by Section 2.2, aft of the payload centre of mass and forward of the

aerodynamic centre. This means that the designed downwards pitch tendency of the aircraft is maintained. The restorative pitch tendency is confirmed by extending the fixed-speed analysis to a domain of $-15° \leq \alpha \leq 15°$. Recall from Section 3.1 that angle of attack is sampled with $0.5°$ spacing. Figure 4.2 shows the moment coefficient as a function of aircraft incidence. The negative gradient implies longitudinal static stability, where an increase in aircraft nose-up pitch increases the downwards restoring moment. The curve ends before $15°$ of incidence is achieved because the stall occurs somewhere between 14 and 14.5 degrees. The stall angle is identified as the minimum angle at which XFLR5 cannot converge. This is a suitable angle, similar to that expected on Earth.

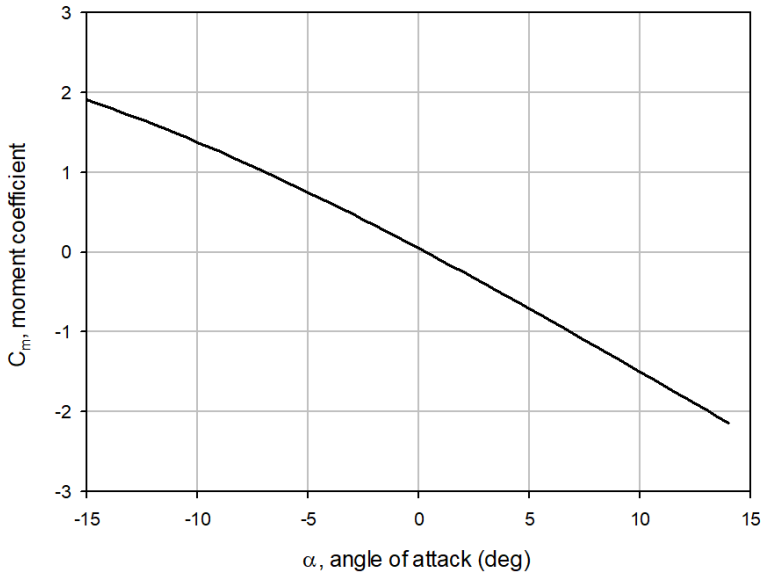


Figure 4.2: Moment coefficient plotted against angle of attack. Negative gradient shows stability.

4.2 Dynamic Stability

Results for dynamic modes can be found through XFLR5 in a (Type 7) stability analysis. The aircraft is trimmed to its zero-moment condition, the inertial matrices are populated using mass and geometry data, stability derivatives are calculated, then the state matrices are defined, finally the state matrices are solved to produce eigenvalues. These eigenvalues describe the oscillatory response of each mode. To contextualise these results, they can be compared against the U.S. military's flying quality specifications, MIL-F-8785C [29]. For the aircraft performing its mission in the main flight (non-terminal) phase, it is identified as a *Category B* craft. As a light observation aircraft, it is identified as *Class I*. Identifying the class and category allow the dynamic stability requirements to be defined. For each dynamic mode, three levels of requirements are given, with level 1 being the most appropriate for this type of aircraft. The definition of each level in [29] are reproduced in Table 4.2 for reference. It should be noted that achieving level 3 is acceptable however will result in increased controller workload for remaining stable and manoeuvring the aircraft. The greater the level of a mode, the greater the power usage to remain stable or manoeuvre through it.

Table 4.2: Flying quality dynamic stability level descriptions, reproduced from [29].

Level	Description
1	Flying qualities clearly adequate for the mission Flight Phase.
2	Flying qualities adequate to accomplish the mission Flight Phase, but some increase in pilot workload or degradation in mission effectiveness, or both, exists.
3	Flying qualities such that the airplane can be controlled safely, but pilot workload is excessive, or mission effectiveness is inadequate, or both. The Category B Flight Phase can be completed.

The requirements for each dynamic mode are given in Tables 4.3 through 4.7. XFLR5 automatically processes each eigenvalue λ to provide the appropriate information required by these specifications, through the methods described in [30]. The eigenvalue is separated into real σ and imaginary components in Eq. (4.1),

$$\lambda = \sigma + i\omega. \tag{4.1}$$

The imaginary component of each eigenvalue is that mode's damped natural angular frequency ω. The damping ratio ζ is described by Eq. (4.2),

$$\zeta = \left| \frac{\sigma}{\sqrt{\sigma^2 + \omega^2}} \right|. \tag{4.2}$$

The time constant of a mode is described by Eq. (4.3),

$$\tau = \left| \frac{1}{\sigma} \right|, \tag{4.3}$$

and the time T_2 for an oscillation to double in amplitude, analogous to half-life, is given by Eq. (4.4),

$$T_2 = \left| \frac{\ln 2}{\sigma} \right|. \tag{4.4}$$

Table 4.3: Short period oscillation (SPO) damping ratio limits for a category B aircraft.

Level	**ζ, damping ratio limit**	
	Minimum	Maximum
1	0.30	2.00
2	0.20	2.00
3	0.15	*No limit*

Table 4.4: Phugoid damping and half-life requirements for the aircraft.

Level	**Parameter**	**Minimum Value**
1	ζ	0.04
2	ζ	0
3	T_2	55 s

Table 4.5: Roll mode time constant requirements for each response level.

Level	**Maximum τ, time constant (s)**
1	1.4
2	3.0
3	10

Table 4.6: Real eigenvalue component and damping ratio requirements for the Dutch roll mode.

Level	**Minimum σ $(\mathbf{rad\ s^{-1}})$**	**Minimum ζ**
1	0.15	0.08
2	0.05	0.02
3	*No minimum*	0

Table 4.7: Half-life requirements for the spiral mode.

Level	Minimum T_2 (s)
1	20
2	8
3	4

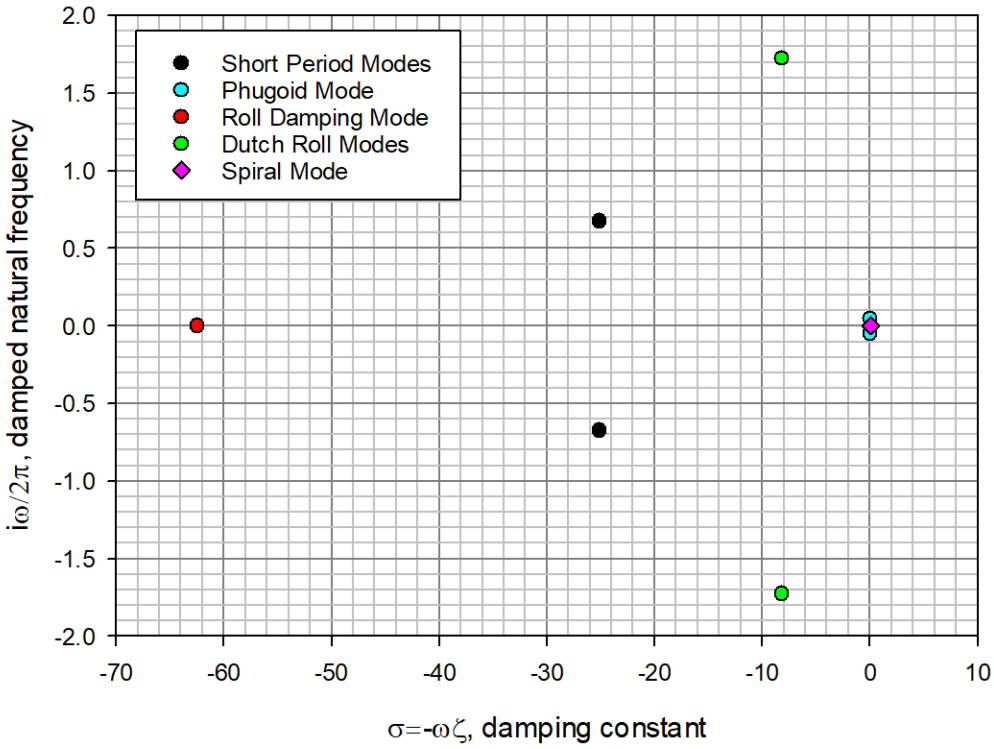


Figure 4.3: Root-locus plot of the eigenvalues representing dynamic modes. The more negative the real component of an eigenvalue (left on the horizontal axis), the heavier its damping. A positive real component represents instability for that mode. The greater the magnitude on the imaginary axis, the greater the mode's natural frequency.

The aircraft's response parameters for each mode are listed in Table 4.8, with the corresponding level, as described by Table 4.2. Primarily, the results demonstrate that the designed aircraft is dynamically stable and manoeuvrable to a safe degree because all but one dynamic mode satisfy the requirements of level 1. The damping ratio of the short period mode is above the minimum required for level 1. The phugoid mode is damped to level 2 requirements. This damping is much lighter than the short period mode, as expected for a phugoid mode. Figure 4.3 shows the low frequency of these oscillations, with small imaginary components. For this mode, the level 2 response indicates that infrequent control adjustments are required. This can be easily handled by the flight computer. Therefore, no further action is required to address longitudinal modes.

Table 4.8: The aircraft's response to dynamic modes in terms of parameters used to compare against MIL-F-8785C. The corresponding level is given, with all modes achieving a level 1 response except the short period mode.

Dynamic Mode	Result(s)	Level
Short Period	$\zeta = 0.986$	1
Phugoid	$\zeta = 0.0299$	2
Roll	$\tau = 0.0160$ s	1
Dutch roll	$\sigma = 8.192$ (rad s^{-1}) $\zeta = 0.603$	1
Spiral	$T_2 = 21.89$ s	1

All lateral modes meet level 1 requirements, meaning that stability is adequate. However, the roll mode time constant is almost two orders of magnitude less than the maximum permitted by Table 4.5, resulting in an aircraft which is unresponsive to aileron inputs. This is illustrated in Figure 4.3, where the eigenvalue of the roll mode has a large negative real component. Recall from Table 2.1 the three identified causes of increased lateral dynamic stability. The high wing was chosen for the increased stability and placing the payload with a lower centre of mass. The aspect ratio has been integral to reducing induced drag as demonstrated in Figure 3.1. The remaining parameter for lateral dynamic stability is dihedral. Adding anhedral – a negative dihedral angle on the wing – is expected to reduce the damping ratio in the roll mode and does not compromise previous design considerations therefore is the most appropriate configuration change to improve roll manoeuvrability.

4.3 Anhedral for Roll Manoeuvrability

Table 4.9: Lateral dynamic response parameters for a given anhedral angle. The corresponding response level for each parameter is given.

	Results(s)		
Anhedral angle (°)	Roll	Dutch Roll	Spiral
0	$\tau = 0.016\ \mathrm{s}$	$\sigma = 8.192\ (\mathrm{rad\ s^{-1}})$ $\zeta = 0.756$	$T_2 = 21.89\ \mathrm{s}$
Levels	1	1	1
5	$\tau = 0.016\ \mathrm{s}$	$\sigma = 8.238\ (\mathrm{rad\ s^{-1}})$ $\zeta = 0.585$	$T_2 = 8.003\ \mathrm{s}$
Levels	1	1	2
10	$\tau = 0.016\ \mathrm{s}$	$\sigma = 8.344\ (\mathrm{rad\ s^{-1}})$ $\zeta = 0.568$	$T_2 = 5.162\ \mathrm{s}$
Levels	1	1	3
15	$\tau = 0.011\ \mathrm{s}$	$\sigma = 8.510\ (\mathrm{rad\ s^{-1}})$ $\zeta = 0.552$	$T_2 = 5.162\ \mathrm{s}$
Levels	1	1	3

Wings with anhedral angles up to $15°$ at $5°$ spacing are defined in the IPython notebook then imported to XFLR5 and analysed on the same aircraft in a stability analysis. Table 4.9 provides the lateral mode response, with the corresponding level, to increased anhedral angle. The roll and Dutch roll modes remain too heavily damped, with only minute changes, while compromising spiral mode stability. If the anhedral angle is increased past $15°$ it is expected that the roll response remains heavily damped. This is likely due to the high structural mass of the wing. To improve the lateral manoeuvrability, mass should be distributed closer to the central axis of the aircraft. The area of the wing can be reduced with an increased angle of attack, so long as there is a safety margin preventing the stall angle of attack being reached in a simple manoeuvre. More accurate lateral mode stability predictions can be made following a geometry- and material-based mass estimate as opposed to the assumptions made from available data in Table 4.1. Following this, lightweighting exercises can be used to distribute mass closer to the fuselage of the aircraft. Therefore the final aircraft's wings, based on these data, will have no anhedral angle. Therefore the geometry of the final aircraft is given by Figure 3.5 (right) and displayed in a three-view diagram in Figure 4.4. The overall findings of final aircraft and process are presented in Section 5.

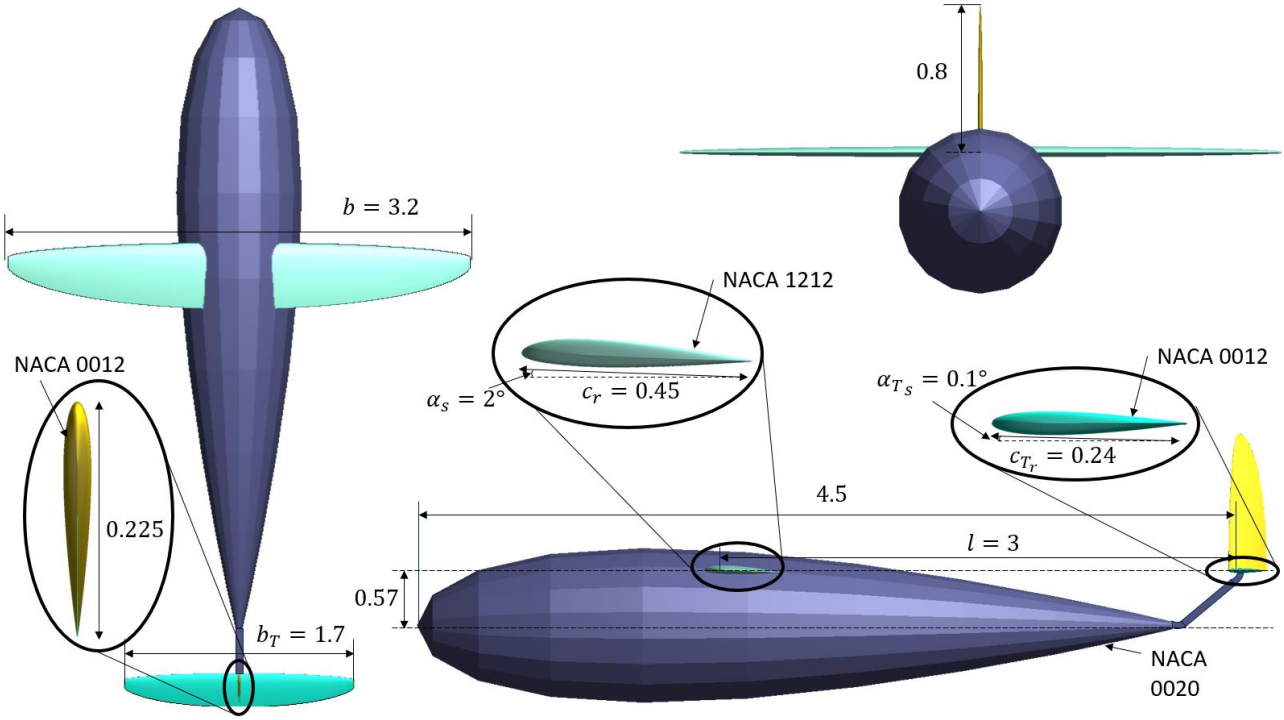


Figure 4.4: Three-view diagram of the final aircraft configuration. All dimensions in metres. All aerodynamic surfaces are elliptical, with a forward semi-minor fraction of 0.25. Tubing connects the end of the fuselage to the tailplane, allowing control cable connections.

5 Conclusions and Further Work

The differences between flying conditions on Earth and Saturn's Moon, Titan, are investigated. The design principles adopted due to these conditions are explored by designing a fixed-wing aircraft to fly on the surface of Titan. A fixed-wing aircraft was hypothesised to be more efficient for sampling different surface sites than a rotorcraft. A suitable aircraft is designed for this mission to determine if a fixed-wing aircraft is feasible and more efficient.

An IPython notebook is developed to take geometry and flight condition inputs and uses them to predict lift and moment characteristics of a given aircraft. It stores multiple aircraft instances and is capable of updating predictions without requiring all parameters to be restated. XFLR5 is used to obtain more accurate estimates of lift and moments as well as drag and stability. The IPython notebook converts defined aircraft into files which are imported into XFLR5. The lift, drag, and stability predictions are used to define new geometry and flight conditions.

An aircraft is designed to the necessary precision to understand the key design factors unique to Titan. The reduction of induced drag by use of a high aspect ratio is still important but less so than in Earth's atmosphere as shown by the 80:20 ratio of viscous to induced drag. Therefore, an increased lift coefficient, thus increasing the induced drag coefficient, is a valid trade-off for decreasing the area to reduce viscous drag. This is in contrast to the typical avoidance of increased lift coefficient due to the squared proportionality between it and induced drag. To lift the same mass on Titan as Earth, at the same speed, the product of surface area and lift coefficient required is thirty times less. For a given lift coefficient, this allows the surface area to be reduced by thirty times, reducing the viscous drag experienced.

This aircraft configuration is estimated to deliver 44 times more range and 33 times more endurance than the Dragonfly rotorcraft. This power requirement of a flight is expected to increase as mass and drag estimates are refined. Therefore, this efficiency is expected to reduce. Nevertheless, the efficiency improvement of a fixed-wing configuration over a rotorcraft is demonstrated, suggesting that a fixed-wing aircraft should be strongly considered for a future Titan exploration mission.

In comparison to an aircraft on Earth, the lift-to-drag ratio of 23 is high. However, this does not include the drag of interfaces between components, the fuselage body, or other components, such as landing gear. Furthermore, drag estimates become less accurate once the boundary layer transitions to turbulence due to the utilisation of the vortex panel method in XFLR5. Therefore, this ratio is expected to reduce and become similar to a lift-to-drag ratio on Earth. Due to the 60% lower speed of sound, transonic effects were expected to be considered. However, the selected flight profile results in a Mach number less than 0.1. This allows drag due to compressibility to be ignored.

The aircraft is shown to be statically and dynamically stable. It is able to operate at $\alpha = \pm 14°$ without stalling, allowing climb and descent for the loiter and take-off and landing phases of the mission. Despite achieving recognised dynamic performance standards, the roll modes are heavily damped, resulting in extra flight computer effort to manoeuvre the aircraft. The low lift requirement results in sensitive control surfaces. It is found that for a change of $0.1°$, the tailplane moment is increased by a factor of 6.6. This requires a more precise, and therefore expensive, manufacturing process. The flight control computer must also be precise when deflecting a surface.

The accuracy of atmospheric information used for this model was limited to those measured by the HASI experiments detailed in Section 1.2. The Dragonfly mission itself will collect more accurate atmospheric data which can be used to refine the design of a fixed-wing aircraft to fly under these conditions.

The outputs of an XFLR5 analysis are manually entered into a spreadsheet which calculates the percentage difference in lift requirement as well as lift-to-drag ratio. An advantage of the IPython notebook is the speed

at which an aircraft can be imported to XFLR5. The overall design process speed can be increased by the XFLR5 analysis being launched by the IPython notebook and the results loaded and processed all in one file.

The lower drag flight speed selected in Section 3.3 is found to not be the minimum drag speed. This inaccuracy is caused by the analytical calculations used in the method. A more accurate minimum drag speed can be found by using XFLR5's fixed angle of attack (Type 4) analysis, sweeping through a range of speeds to find that with the least drag. Flying at this speed will reduce the power usage of a single flight and allow more missions to occur in a Titan solar day.

The designed aircraft should be re-simulated in XFLR5 under Earth's atmospheric conditions to determine the proportions of each drag component. Despite the aircraft not being able to produce enough lift for a flight on Earth, the proportion of each drag component should be compared to the results on Titan to determine how much increase in viscous drag is caused by Titan's atmospheric conditions.

Sections 3.5 and 4.3 identify the need for improved mass estimates. A computer-aided design (CAD) model of the aircraft should be developed. With materials defined, this can be used to determine the mass of each component. This will modify the eigenvalues obtained in Section 4.2, this allows the design to be changed to increase manoeuvrability if necessary. The updated component masses will provide a more accurate structural weight estimate, requiring the lifting surfaces to be scaled or rotated to produce the necessary lift. The CAD model allows the geometry to be loaded into a more accurate CFD tool than XFLR5, producing improved drag estimates. The power requirements can then be recalculated from the updated drag force.

The unique sensitivity of control surfaces requires a unique design and controller to precisely fly the aircraft. A study of control sizing should be conducted. The flight control system of the aircraft must be so precisely defined that it requires a detailed study.

References

[1] R. M. C. Lopes, M. J. Malaska and A. M. Schoenfeld et. al., "A global geomorphologic map of Saturn's moon," *Nature Astronomy,* vol. 4, pp. 228-233, 2020.

[2] NASA, "Dragonfly Launch Moved to 2027," 25 September 2020. [Online]. Available: https://www.nasa.gov/feature/dragonfly-launch-moved-to-2027.

[3] R.D. Lorenz, E.P. Turtle, J.W. Turtle et. al., "Dragonfly: A Rotorcraft Lander Concept for Scientific Exploration at Titan," *John Hopkins APL Technical Digest,* vol. 34, no. 3, pp. 374-387, 2018.

[4] D. F. Anderson and S. Eberhardt, "Table 8.1: Comparison of the Specifications of a Cessna 172 and a Bell-47G Helicopter," in *Understanding Flight, Second Edition*, McGraw-Hill, 2010, p. 238.

[5] Y. A. Cengel, J. M. Cimbala and R. H. Turner, "Reynolds Number," in *Fundamentals of Thermal-Fluid Sciences, Fifth Edition*, New York, McGraw-Hill, 2017, p. 530.

[6] National Aeronautics and Space Administration, "Multi-Mission Radioisotope Thermoelectric Generator (MMRTG)," [Online]. Available: https://mars.nasa.gov/internal_resources/788/.

[7] P. Mahaffy, "SAM - Mars Curiosity Rover Instruments," National Aeronautics and Space Administration, [Online]. Available: https://mars.nasa.gov/msl/spacecraft/instruments/sam/. [Accessed 03 12 2020].

[8] F. Goesmann et. al., "The Mars Organic Molecule Analyzer (MOMA) Instrument: Characterization of Organic Material in Martian Sediments," *Astrobiology,* vol. 17, no. 6-7, pp. 655-685, 2017.

[9] Malin Space Science Systems, "ECAM Imaging Systems Brochure," 2013. [Online]. Available: http://www.msss.com/brochures/ecam.pdf.

[10] NASA, "HASI Data," [Online]. Available: https://atmos.nmsu.edu/PDS/data/hphasi_0001/DATA.

[11] T. Mäkinen, "Processing the HASI measurements," *Advances in Space Research,* vol. 17, no. 16, pp. 219-222, 1996.

[12] R. Jacobsen, R. Stewart, R. McCarty and H. Hanley, "Thermodynamical properties of Nitrogen from the fusion line to 3500 R (1944 K) for pressures to 150,000 psia (10342 x 10^5 N/m^2).," *National Bureau of Standards,* no. NBS Technical Note 648, 1973.

[13] M. Fulchignoni, F. Ferri, F. Angrilli et. al., "In situ measurements of the physical characteristics of Titan's environment," *Nature,* vol. 438, no. 785-791, 2005.

[14] R. C. J.J. Bertin, "Table 2.2: Reynolds Number Regimes and Characteristics for an Airfoil," in *Aerodynamics for Engineers*, Harlow, UK, Pearson Education, 2014, p. 72.

[15] D. F. Anderson and S. Eberhardt, "Vortex Generators," in *Understanding Flight, Second Edition*, McGraw-Hill, 2010, pp. 70-71.

[16] A. Da Ronch, "2.2.2 General Requirements," in *Aircraft Structural Design and Aeroelasticity*, University of Southampton, version 2.9, pp. 16-17.

[17] J. W. Kim, "Yaw-to-Roll Coupling Effect," in *Lecture 2.5: Lateral/Directional Stability of SESA 2025: Mechanics of Flight*, University of Southampton, 2019/20, pp. 5, 8-9.

[18] J. W. Kim, "Tail plane angle of attack: Influence of downwash," in *Flight Mechanics Lecture 1.1: Equations of motion and tailplane equation*, University of Southampton, 2020, p. 12.

[19] S. Gudmundsson, "Table 4-2: Typical Properties of Aircraft Based on Class," in *General Aviation Aircraft Design: Applied Methods and Procedures*, Waltham, Elsevier, 2014, p. 83.

[20] Continental Aerospace Technologies, "Continental 200 Series AvGas Engine," [Online]. Available: http://continental.aero/engines/200.aspx. [Accessed 18 12 2020].

[21] Z. Hu, Wing Aerodynamics Lecture 4.6: Viscous Inviscid Interation method, University of Southampton, 2021.

[22] "XFLR5 Theoretical Background Part IV: Limitations and Shortcomings," [Online]. Available: http://www.xflr5.tech/xflr5.htm. [Accessed December 2020].

[23] J.-W. Kim, "Aircraft Static Stability," in *Mechanics of Flight Lecture 2.1*, University of Southampton, 2020, p. 8.

[24] M. Robijns, "NACA.py," Delft University of Technology, Available on Github, 2015.

[25] Z. Hu, "Vortex Filament," in *Vortex Lattice Method, Wing Aerodynamics Lecture Slides*, University of Southampton, 2021, p. 16.

[26] A. Forrester, "Range and Endurance: Electric Aircraft," in *Flight Mechanics Made Clear*, University of Southampton, 2019, pp. 47-49.

[27] A. Forrester, "Drag," in *Flight Mechanics Made Clear*, University of Southampton, 2019, pp. 25-31.

[28] O. Dababneha and T. Kipouros, "Table 1: Mass fractions of passenger aircraft as percentage of MTOM," in *A Review of Aircraft Wing Mass Estimation Methods*, Aerospace Science and Technology Vol 72, 2017, pp. 256-266.

[29] U.S. Department of the Air Force, Military Specification: Flying Qualities of Piloted Airplanes, Washington, D.C., 1980.

[30] A. Deperrois, "Stability Analysis with XFLR5," November 2010. [Online]. Available: http://www.xflr5.tech/docs/XFLR5_and_Stability_analysis.pdf.

A. Tabulated Aircraft Iterations, Changes and Results

Table A.1: Wing iterations as recorded in the spreadsheet. Those without lift results are shown in Table A.4 by the combined lift of the aircraft. Green highlighting in the ID column indicates an accepted wing for a base of the next pass. The colour coding of the SC_L percentage difference column indicates whether the lift generated by the wing is within 5% error of the target lift.

							Target SC_L:	0.2791	
ID	wing span	wing root chord	fwd sm fraction	Foil	setting alpha	Dihedral (deg	sectional lift	wing SC_L	SC_L diff (%)
000	2.8	0.7	0.25	NACA 1212	0.00		0.1065	0.1639	-41.27553
001	5.6	0.7	0.25	NACA 1212	0.00		0.1065	0.328	17.520602
002	5	0.7	0.25	NACA 1212	0.00		0.1065	0.293	4.9802938
003	3	1.2	0.25	NACA 1212	0.00		0.1065	0.301	7.8466499
004	2.9	1	0.25	NACA 1212	0.00		0.1065	0.243	-12.93443
005	3	1	0.25	NACA 1212	0.00		0.1065	0.251	-10.06808
006	3.2	1	0.25	NACA 1212	0.00		0.1065	0.268	-3.977069
007	2.8	0.7	0.25	NACA 1212	0.50		0.1633	0.251	-10.06808
008	3	0.7	0.25	NACA 1212	0.50		0.1633	0.269	-3.618775
									-100
									-100
									-100
Second Pass (Finite Wing Theory)									
002	5	0.7	0.25	NACA 1212	1.00			0.494	76.997492
002	5	0.7	0.25	NACA 1212	0.50			0.367	31.494088
002	5	0.7	0.25	NACA 1212	0.20			0.291	4.2637048
006	3.2	1	0.25	NACA 1212	0.20			0.216	-22.60838
006	3.2	1	0.25	NACA 1212	0.30			0.235	-15.80079
006	3.2	1	0.25	NACA 1212	0.40			0.254	-8.993192
006	3.2	1	0.25	NACA 1212	0.50			0.273	-2.185597
008	3	0.7	0.25	NACA 1212	1.00			0.264	-5.410247
008	3	0.7	0.25	NACA 1212	1.10			0.277	-0.752418
									-100
									-100
Third Pass (XFLR5)									
002	5	0.7	0.25	NACA 1212	0.50				-100
006	3.2	1	0.25	NACA 1212	1.00				-100
006	3.2	1	0.25	NACA 1212	3.00				-100
008	3	0.7	0.25	NACA 1212	1.40				-100
008	3	0.7	0.25	NACA 1212	2.10				-100
									-100
Forth Pass (Combined A/C Refinement)									
002.001	4	0.56	0.25	NACA 1212	0.60			0.251	-10.10391
002.002	4	0.56	0.25	NACA 1212	0.70			0.267	-4.263705
002.003	2.85	0.4	0.25	NACA 1212	1.40			0.194	-30.49086
002.004	2.85	0.4	0.25	NACA 1212	2.00			0.243	-12.86277
002.005	3.2	0.45	0.25	NACA 1212	2.00			0.307	10.032247
002.006	3	0.42	0.25	NACA 1212	2.00			0.269	-3.618775
002.007	4	0.56	0.25	NACA 1212	2.00				
002.008	3.5	0.5	0.25	NACA 1212	2.00				
002.009	3.2	0.45	0.25	NACA 1212	2.00				
			0.25	NACA 1212					
9	3.2	0.45	0.25	NACA 1212	2.00	-5.00			
10	3.2	0.45	0.25	NACA 1212	2.00	-10.00			
11	3.2	0.45	0.25	NACA 1212	2.00	-15.00			

Table A.2: Horizontal stabiliser iterations and design parameters. Refer to Table A.1 for colour coding rules. Cross reference ID numbers with Table A.4 for results of use on an aircraft in XFLR5. For the ID, the lower number represents the iteration of horizontal stabiliser, the larger number is used in Table A.4 and is the identification of that instance of the wing class.

ID	wing span	wing root chord	fwd sm fraction	Foil	setting alpha	height	S_T_C_L_T
To Balance: Wing 2							
0/9	1.6	0.4	0.25	NACA - 1212	-1	0	-0.0788
1/10	1.6	0.4	0.25	NACA - 1212	0	0	-0.0383
2/11	1.2	0.4	0.25	NACA - 1212	0	0	-0.0262
3/12	1	0.4	0.25	NACA - 1212	0	0	-0.0204
4/13	1	0.3	0.25	NACA - 1212	0	0	-0.017
5/15	1	0.3	0.25	NACA 0012	-0.2	0	-0.0036
6/16	1	0.3	0.25	NACA 0012	-0.8	0	-0.0144
7/17	1.2	0.3	0.25	NACA 0012	-0.8	0	-0.0183
To Balance: Wing 6							
8/18	1.4	0.4	0.25	NACA 0012	-0.8	0	-0.0273
9/19	1.5	0.4	0.25	NACA 0012	-0.8	0	-0.02988
10/20	1.5	0.4	0.25	NACA 0012	-1	0	-0.03736
To Balance: Wing 8							
11/22	1.6	0.4	0.25	NACA 0012	-0.5	0	-0.0203
						0	
Aircraft A Pass 2							
7/17	1.4	0.3	0.25	NACA 0012	-0.8	0	
Aircraft B Pass 2							
10/20	1.5	0.4	0.25	NACA 0012	-1	0.5	-0.03736
Combined Aircraft Pass							
7/17	2.1	0.3	0.25	NACA 0012	-0.4		-0.01826
7/17	2.1	0.3	0.25	NACA 0012	-0.5		-0.02282
17.001	1.7	0.24	0.25	NACA 0012	-0.5		-0.0148
17.002	1.7	0.24	0.25	NACA 0012	-0.4		-0.01185
17.003	1.7	0.24	0.25	NACA 0012	-0.3		-0.00889
17.004	1.5	0.21	0.25	NACA 0012	-0.4		-0.00916
17.005	1.6	0.225	0.25	NACA 0012	-0.4		-0.0105
17.006	1.7	0.24	0.25	NACA 0012	-0.7		-0.0207
17.007	1.7	0.24	0.25	NACA 0012	0.7		
17.008	1.7	0.24	0.25	NACA 0012	0.0		
17.009	1.7	0.24	0.25	NACA 0012	0.1		

Table A.3: Vertical stabiliser iterations. Refer to Table A.4 for the results of use of an aircraft. ID convention explained in Table A.2.

ID	wing span	wing root chord	fwd sm fraction	Foil
0/14	1	0.3	0.25	NACA 0012
1/21	1.5	0.4	0.25	NACA 0012
14.001	1.7	0.24	0.25	NACA 0012
14.002	1.6	0.225	0.25	NACA 0012

Table A.4: Iterations of aircraft tested with XFLR5. Colour coding is explained in the caption of Table A.1. The wing, horizontal and vertical stabiliser ID's correspond with those in Tables A.1 through A.3.

ID	V	wing	horiz	vert	l	S	AR	CL	CD	Cdi	CDv	Cm	CL/CD	SCL
A	14.81	2	17	14	3.00	2.7475	9.0990	0.075819	0.006548	0.000243	0.006305	-0.00679	11.57896	0.208313
A	14.81	2	17	14	3.00	2.7475	9.0990	0.100902	0.006696	0.000408	0.006288	-0.01299	15.069	0.277228
A	14.81	2	17	14	3.00	2.7475	9.0990	0.098642	0.00679	0.000397	0.006392	-0.00096	14.52754	0.271019
B	14.81	6	20	21	3.00	0.5043	3.9270	0.060384	0.007372	0.000450	0.006922	0.048277	8.190993	0.030452
B	14.81	6	20	21	3.00	0.5043	3.9270	0.088593	0.007763	0.000851	0.006911	0.054514	11.41221	0.044677
B	14.81	6	20	21	3.00	0.5043	3.9270	0.201406	0.010883	0.003902	0.006981	0.078143	18.50648	0.101569
B	14.81	6	20	21	3.00	0.5043	3.9270	0.483629	0.030317	0.022075	0.008242	0.127876	15.9524	0.243894
B	14.81	6	20	21	3.00	0.5043	3.9270	0.2117	0.0114				18.5718	0.10676
C	14.81	8	22	23	3.00	1.6434	5.2590	0.105904	0.008803	0.000835	0.007968	0.101696	12.03044	0.174043
C	14.81	8	22	23	3.00	1.6434	5.2590	0.12446	0.0091	0.00113	0.00797	0.110826	13.67692	0.204538
C	14.81	8	22	23	3.00	1.6434	5.2590	0.167723	0.01002	0.002001	0.008019	0.131873	16.73882	0.275636
Geometry combined														
A	14.81	2	17	14	3.00	2.7485	9.0960	0.095127	0.007102	0.000354	0.006748	0.017997	13.3944	0.261457
A	14.81	2.002	17.001	14.001	3.00	1.7530	8.7660	0.110153	0.007725	0.00484	0.007241	0.023041	14.25929	0.193098
A	14.81	2.006	17.005	14.002	3.00	0.9862	8.7650	0.2034	0.010402				19.55393	0.200593
A	14.81	2.007	17.006	14.002	3.00	1.7530	8.7660	0.210695	0.009086	0.001736	0.00735	0.034167	23.18897	0.369348
A	14.81	2.008	17.006	14.002	3.00	1.3740	8.9160	0.204521	0.009607	0.001635	0.007972	0.094198	21.28875	0.281012
A	14.81	2.008	17.007	14.002	3.00	1.3740	8.9160	0.232247	0.009822	0.001934	0.007888	-0.09157	23.64559	0.319107
New flight speed														
A.101	15.55	2.009	17.007	14.002	3.00	1.2322	8.7520	0.199681	0.009355	0.001608	0.007747	0.143718	21.34484	0.246047
A.102	15.55	2.009	17.008	14.002	3.00	1.2322	8.7520	0.215109	0.009429	0.001739	0.007689	0.020305	22.81355	0.265057
A.103	15.55	2.009	17.009	14.002	3.00	1.2322	8.7520	0.217313	0.009447	0.001764	0.007683	0.002674	23.00339	0.267773

B. IPython Notebook Functions of Interest

B.1. Wing (ACConfig) Class

To plot an elliptical wing, as in Figure 2.4, given its root chord (_c_root), span, and forward semi-minor fraction (_fwd_sm_fraction_), the following function can be used. Note that the NumPy library has been imported as "np".

```
from matplotlib import pyplot as plt
%matplotlib

def drawEllipticalWing(_c_root, _span, _fwd_sm_fraction, _gr_title = ""):
    _K_fore = _c_root * _fwd_sm_fraction
    _K_aft = _c_root * (1-_fwd_sm_fraction)

    _ys = np.linspace(-_span/2, _span/2, 200)
    _c_fores = np.zeros(200)
    _c_afts = np.zeros(200)

    _c_fores = _K_fore*(1-(2/_span *_ys )**2)**0.5
    _c_afts = -_K_aft*(1-(2/_span *_ys )**2)**0.5

    plt.figure()
    plt.plot(_ys, _c_fores)
    plt.plot(_ys, _c_afts)
    plt.xlabel('y, span position (m)')
    plt.ylabel('x, longitudinal position (m)')
    plt.title(_gr_title)
    if(_span/2 > _c_root):
        plt.axis([-_span/2, _span/2, -_span/2, _span/2])
```

With a simple function for determining chord length at a span position within the ellipse,

`cElliptical(_y, _c_root, _span, _fwd_sm_fraction)`,

the infinite wing's SC_L value can be found using the following function.

```
from scipy.integrate import quad
#Integrating for SC_L
def SC_LfromWing(_c_l, _shape, _args=(), _draw=False, _gr_title = ""):
    _SC_L = 0
    if(_shape == "ellipse"):
        if(_draw):
            drawEllipticalWing(_args[0], _args[1], _args[2], _gr_title)
        _span = _args[1]
        _integral = quad(cElliptical, 0, _span/2, _args)[0]
        _SC_L = 2 * _c_l * _integral

    return _SC_L
```

This function encapsulates the elliptical plotting method, such that the designer can chose to see a plot at the same time as calculating a new wing's SC_L value. Here, the arguments `_args` are the latter three parameters of the elliptical chord length function. The foundations of the wing class are then constructed.

```
global_config_id = -1
def incrimentConfigID():
    global global_config_id
    global_config_id += 1
    return global_config_id

class ACConfig:
    _id = 0
    span = 0.0
    c_root = 0.7
    shape = "ellipse"
    fwd_sm_fraction = 0.0
    c_l = 0.0
    SC_L = 0.0

    def __init__(self, span, c_root, fwd_sm_f, c_l, draw_graph = False):
        self._id = incrimentConfigID()
        self.span = span
        self.c_root = c_root
        self.fwd_sm_fraction = fwd_sm_f
        self.c_l = c_l
        self.SC_L = self.updateSC_L(draw_graph)

    def updateSC_L(self, draw_graph = False):
        self.SC_L = SC_LfromWing(self.c_l, self.shape, (self.c_root, self.span,␣
↪self.fwd_sm_fraction), draw_graph, "Config: " +str(self._id))
        return self.SC_L
```

In Sections 2.3 to 2.4, this class is updated to store and calculate moment, aspect ratio, and finite wing lift coefficient data.

```
#update the class to store moments
ACConfig.C_M_0 = 0.0 #Entire moment
ACConfig.C_m = 0.0 #Sectional moment
ACConfig.S = 0.0 #wing planform area
ACConfig.AR = 0.0 #aspect ratio
ACConfig.C_L = 0.0 #Lift coefficient
ACConfig.a_0 = 2 * math.pi #lift curve slope coefficient
ACConfig.z_0 = 1 #moment curve slope
ACConfig.alpha_zero = 0.0 #zero lift angle of attack (degrees)
ACConfig.alpha = 0.0 #current angle of attack
ACConfig.lift_efficient_factor = 0.0 #approximated as 0 for elliptical
```

```
def updateS(self):
    self.S = 2*quad(cElliptical, 0, self.span/2, (self.c_root, self.span, self.
 ↪fwd_sm_fraction))[0]

def C_M_0FromWing(_c_m, _shape, _S, _args=()):
    _C_M = 0.0
    if(_shape == "ellipse"):
        _span = _args[1]
        _integral = quad(cEllipticalSquared, 0, _span/2, _args)[0]
        _C_M = 2 * _span * _c_m / _S**2 * _integral
    return _C_M

def updateC_M_0(self):
    self.C_M_0 = C_M_0FromWing(self.C_m, self.shape, self.S, (self.c_root, self.
 ↪span, self.fwd_sm_fraction))

def updateAR(self):
    self.AR = self.span**2 / self.S

def updateC_L(self):
    #SC_L is based on sectional C_L, this better approximates using Aspect Ratio
    _C_L = (self.alpha - self.alpha_zero)*math.pi/180
    _C_L *= self.a_0 / (1+self.a_0/(math.pi * self.AR)*(1+self.
 ↪lift_efficient_factor))
    self.C_L = _C_L

def updateSC_LFinite(self):
    self.SC_L = self.C_L * self.S

def updateC_m(self):
    self.C_m = self.z_0*(self.alpha - self.alpha_zero)

ACConfig.updateS = updateS
ACConfig.updateC_M_0 = updateC_M_0
ACConfig.updateAR = updateAR
ACConfig.updateC_L = updateC_L
ACConfig.updateSC_LFinite = updateSC_LFinite
ACConfig.updateC_m = updateC_m
ACConfig.c_bar = 0.0

def updateMeanAerodynamicChord(self):
    #integrate c over y
    _args = (self.c_root, self.span, self.fwd_sm_fraction)
    _integral = quad(cElliptical, 0, self.span/2, _args)[0]
    _mac = 2 / self.span * _integral
    self.c_bar = _mac

ACConfig.updateMAC = updateMeanAerodynamicChord
```

An update all function is added to allow the inspection of any parameter after another has been updated.

```
def updateAll(self):
    self.updateS()
    self.updateAR()
    self.updateC_L()
    self.updateSC_LFinite()
    self.updateC_m()
    self.updateC_M_0()
    self.updateMAC()

ACConfig.updateAll = updateAll
```

In Section 4, the wing class is updated to store a dihedral angle.

```
#give wing a dihedral property in degrees, defaulting to 0
ACConfig.dihedral = 0
```

B.2 Aircraft Class

Multiple instances of the wing class created in Appendix B.1 are then stored by an aircraft class.

```
#New class Aircraft to hold instances of ACConfig (one for wing, one for tail)
#Also hold propeties that link the two:

global_AC_id = 64
def incrimentACID():
    #returns letter identifier
    global global_AC_id
    global_AC_id += 1
    AC_id = chr(global_AC_id)
    return AC_id

class Aircraft:
    _id = incrimentACID()
    K = 0 #S_T/S * l/c_bar #tail_volume_fraction
    mass_s = 16 #structural mass (kg)
    mass_p = 101.5 #payload mass
    mass = totalMass(mass_s, mass_p) #total mass
    COM = (0.624, 0, -0.305) #vector distance from front of payload (see doc)␣
↪to COM (x,y,z)
    CW = 0
    AC = (0.87, 0, 0.60)  #vector distance from front of payload to Aerodynamic␣
↪Centre
    h = 0 #LE→COM / c_bar
    h_0 = 0 #LE→AC /c_bar
    l = 0 #AC→AC_T
    wing = None
    tail = None
    horiz_moment_balance = 1024 #need to try and get this close to 0

    def __init__(self, _wing, _l, _tail):
        self.wing = _wing
        self.tail = _tail
        self.h = (self.AC[0] - self.wing.c_root * self.wing.fwd_sm_fraction -␣
↪self.COM[0])/self.wing.c_bar
        self.h_0 = self.wing.c_root * self.wing.fwd_sm_fraction / self.wing.
↪c_bar
        self.l = _l
        self.updateAll()
    def updateWeightCoefficient(self):
        global g_0
        global h_cruise
        global V_cruise
        _S = self.wing.S
        self.C_W = self.mass * g_0 * 2 / (densityFromAlt(h_cruise) *␣
↪V_cruise**2 * self.wing.S)
    def updateTailVolFraction(self):
        self.K = self.tail.S / self.wing.S * self.l / self.wing.c_bar
   def updateHorizMomentBalance(self):
       self.horiz_moment_balance = self.wing.C_M_0 + self.C_W*(self.h-self.
↪h_0) - self.tail.C_L * self.K

    def updateAll(self):
        self.wing.updateAll()
        self.tail.updateAll()
        self.updateWeightCoefficient()
        self.updateTailVolFraction()
        self.updateHorizMomentBalance()
```

The target tailplane $S_T C_{L_T}$ is then calculated for the aircraft.

```
def findTargetTailSC_L(self):
    S_T_C_L_T = self.wing.S * self.wing.c_bar / self.l * (self.wing.C_M_0 +␣
 ↪self.C_W*(self.h-self.h_0))
    return S_T_C_L_T

Aircraft.findTargetTailSC_L = findTargetTailSC_L
```

B.3 Fuselage Class

A NACA aerofoil creation library [24] is imported for creating the teardrop-shaped fuselage. This profile is then split into suction and pressure side surfaces such that it can be simplified to one side of the line of symmetry.

```
class Fuselage:
    foil = "NACA "
    fuse2d = [] #coords for entire 2d surface
    suction = [] #coords for top surface
    pressure = []
    chord = 0

    def __init__(self, naca_4, chord, N=24):
        self.fuse2d = naca.NACA4(naca_4, N)
        LE_index = int(len(self.fuse2d)/2) + 1
        self.pressure = self.fuse2d[LE_index:len(self.fuse2d)]
        self.suction = self.fuse2d[0:LE_index]
        self.foil += naca_4
        self.chord = chord
```

The side considered is the side with the tallest payload height to be sure that the maximum position is always covered. A class of any foil can be created and evaluated against the downscaled payload geometry. This geometry checking function returns whether the fuselage is legal – that is, does it completely cover the payload. Additionally, a plot is made with both the fuselage and payload profiles superimposed.

```
    def checkPayloadGeo(self, radius_h, radius_v, t_loc, top_taller = True,␣
↪plot = False):
        """
        <radius_h> = horizontal length occupied by payload forwards of thickest␣
↪payload point
        <radius_v> = vertical height occupied by payload upwards of fuselage␣
↪datum line
        <t_loc> = location of thickest payload geo as fraction of chord from LE
        [top_taller] = does payload stick out more at top or bottom

        returns: True if fuselage doesn't collide with payload
        """
        surf = self.suction
        if(not top_taller):
            print("w")
            surf = self.pressure

        fore_x = t_loc - radius_h / self.chord
        y = radius_v / self.chord
        aft_x = t_loc + radius_h / self.chord

        if(plot):
            pl.figure()
            xs = []
            ys = []
            for i in range(len(surf)):
                xs.append(surf[i][0])
                ys.append(surf[i][1])

            #box
            plx = []
            ply = []
```

```
            plx.append(fore_x)
            ply.append(0)

            plx.append(fore_x)
            ply.append(y)

            plx.append(aft_x)
            ply.append(y)

            plx.append(aft_x)
            ply.append(0)

            pl.plot(xs, ys)
            pl.plot(plx, ply, 'r')

            pl.ylim(0, 0.5)

        legal = True

        for coords in surf:
            if(coords[0] < t_loc):
                if(coords[0] > fore_x and coords[1] < y):
                    legal = False
                    break
            elif(coords[0] > t_loc):
                if(coords[0] < aft_x and coords[1] < y):
                    legal = False
                    break

        return legal
```

B.4 Converting an Aircraft to an XML File for Importing into XFLR5

```
import xml.etree.ElementTree as el

def createSimpleKVP(key, txt, parent):
    elem = el.Element(key)
    elem.text = txt
    parent.append(elem)
    return elem
```

The wing class is updated with important properties. All horizontal and vertical stabiliser instances of this class must be updated to have the corresponding type identifier, otherwise they will be treated as a main wing. Likewise, the aerofoil type must also be updated for that specific wing.

```
#method to be added to acconfig class
ACConfig.type = "mainwing" #can also be "horizontal stabiliser", "vertical␣
↪stabiliser"
ACConfig.finsym = False #true if this wing is a symmetric fin (reflected along␣
↪the z axis)
ACConfig.findouble = False #true if this wing is to be doubled (reflected along␣
↪y axis about centre-plane)
ACConfig.foil = "NACA 1212" #the foil of the entire wing (currently)
```

An aerofoil section can be created by the following method, which includes the number of panels to use in its three-dimensional plotting.

```
def createSection(parent, y, c, foil, x_sweep=0.000, dihedral=0.000, twist = 0.
↪000, x_panels = 6, x_distribution="COSINE", y_panels = 9, y_distribution =␣
↪"INVERSE SINE"):

    section = el.Element('Section')
    createSimpleKVP('y_position', str(round(y, 3)), section)
    createSimpleKVP('Chord', str(round(c, 3)), section)
    createSimpleKVP('xOffset', str(round(x_sweep, 3)), section)
    createSimpleKVP('Dihedral', str(round(dihedral, 3)), section)
    createSimpleKVP('Twist', str(round(twist, 3)), section)
    createSimpleKVP('x_number_of_panels', str(x_panels), section)
    createSimpleKVP('x_panel_distribution', x_distribution, section)
    createSimpleKVP('y_number_of_panels', str(y_panels), section)
    createSimpleKVP('y_panel_distribution', y_distribution, section)
    createSimpleKVP('Left_Side_FoilName', foil, section)
    createSimpleKVP('Right_Side_FoilName', foil, section)
    parent.append(section)
    return section
```

A wing object (main wing, elevator, and fin), can then be converted through the following method. This sweeps through the entire span, creating sections, in intervals defined by a desired resolution.

```
def wingToXML(self, resolution=50):
    explane = el.Element('explane')
    explane.set('version', "1.0")

    units = el.SubElement(explane, 'Units')
    ltm = el.SubElement(units, 'length_unit_to_meter')
    ltm.text = '1'
    mtk = el.SubElement(units, 'mass_unit_to_kg')
    mtk.text = '1'

    #default mainwing
    selfname = "wing"+str(self._id)
    selfred = '153'
    selfgreen = '254'
    selfblue = '227'
    selftype = 'MAINWING'
    selffin = 'FALSE'
    if(self.type == "horizontal stabiliser"):
        selfname = "horiz"+str(self._id)
        selfred = '20'
        selfgreen = '254'
        selfblue = '227'
        selftype = 'ELEVATOR'
    elif(self.type == "vertical stabiliser"):
        #this is a vert stabiliser
        selfname = "vert"+str(self._id)
        selfred = '254'
        selfgreen = '220'
        selfblue = '20'
        selftype = 'FIN'
        selffin = 'TRUE'
```

```
    wing = el.SubElement(explane, 'wing')
    name = el.SubElement(wing, 'Name')
    name.text = selfname
    _type = el.SubElement(wing, 'Type')
    _type.text = selftype
    color = el.SubElement(wing, 'Color')
    red = el.SubElement(color, 'red')
    red.text = selfred
    green = el.SubElement(color, 'green')
    green.text = selfgreen
    blue = el.SubElement(color, 'blue')
    blue.text = selfblue
    alpha = el.SubElement(color, 'alpha')
    alpha.text = '255'
```

```
    pos = el.SubElement(wing, 'Position')
    pos.text = '          0,          0,          0'
    tilt = el.SubElement(wing, 'Tilt_angle')
    tilt.text = '0.000'
    symm = el.SubElement(wing, 'Symetric')
    symm.text = 'true'
    fin = el.SubElement(wing, 'isFin')
    fin.text = selffin
    dbfin = el.SubElement(wing, 'isDoubleFin')
    dbfin.text = self.findouble
    symfin = el.SubElement(wing, 'isSymFin')
    symfin.text = self.finsym

    inertia = el.SubElement(wing, 'Inertia')
    volmass = el.SubElement(inertia, 'Volume_Mass')
    volmass.text = '0.000'
```

```
    sections = el.SubElement(wing, 'Sections')
    createSection(sections, 0.000, self.c_root, self.foil, -self.
↪fwd_sm_fraction*self.c_root, self.dihedral) #root
    dx = self.span/(2*resolution)
    y = dx
    while(y < self.span/2):
        if(self.shape == "ellipse"):
            c = cElliptical(y, self.c_root, self.span, self.fwd_sm_fraction)
            x_off = cFore(y, self.c_root, self.span, self.fwd_sm_fraction)
            if(c == 0.00):
                c = 0.001
            createSection(sections, y, c, self.foil, -x_off, self.dihedral)
        y+=dx
```

```
    with open(selfname+'.xml', 'wb') as file:
        file.write('<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE explane>'.
↪encode('utf-8'))
        el.ElementTree(explane).write(file, encoding='utf-8')

    print("Successfully created file "+ selfname +".xml")

ACConfig.toXML = wingToXML
```

Likewise a revolved aerofoil is converted to XML. It is important that this precise order is followed as there is an error in the XFLR5 source code that causes a crash when the degrees are defined after the frames.

```
def coordsToStr(coords):
    x = coords[0]
    x = round(x, 5)
    y = coords[1]
    y = round(y, 5)
```

```
    z = coords[2]
    z = round(z, 5)
    _str = str(x) + ", " + str(y) + ", " +str(z)
    return _str
```

The frame creation function produces the coordinates of an individual cross section of the fuselage. The resolution of this frame is a parameter.

```
def createFuselageFrame(x, y, parent, n=12):
    frame = el.Element('frame')

    n /= 2
    n = int(n)

    pos = [x, 0, 0]
    pos = coordsToStr(pos)
    createSimpleKVP('Position', pos, frame)

    points = []
    theta = math.pi / n
    for i in range(n+1):
        theta = math.pi / n
        theta *= i
        _sin = y * math.sin(theta)
        _cos = y * math.cos(theta)
        point = [x, _sin, _cos]
        point = coordsToStr(point)
        createSimpleKVP('point', point, frame)

    parent.append(frame)

    return frame
```

```
def fuselageToXML(self):
    explane = el.Element('explane')
    explane.set('version', "1.0")

    units = el.SubElement(explane, 'Units')
    ltm = el.SubElement(units, 'length_unit_to_meter')
    ltm.text = '1'
    mtk = el.SubElement(units, 'mass_unit_to_kg')
    mtk.text = '1'

    body = el.SubElement(explane, 'body')
    name = el.SubElement(body, 'Name')
    name.text = "Fuselage " + self.foil
    color = el.SubElement(body, 'Color')
    red = el.SubElement(color, 'red')
    red.text = '98'
    green = el.SubElement(color, 'green')
    green.text = '102'
    blue = el.SubElement(color, 'blue')
    blue.text = '156'
    alpha = el.SubElement(color, 'alpha')
    alpha.text = '255'

    desc = el.SubElement(body, 'Description')
    desc.text = 'fuselage'
    pos = el.SubElement(body, 'Position')
    pos.text = '0, 0 ,0'
    _type = el.SubElement(body, 'Type')
    _type.text = 'FLATPANELS'
    inertia = el.SubElement(body, 'Inertia')
    vmass = el.SubElement(inertia, 'Volume_Mass')
    vmass.text = '0.000'
```

```
    for i in range(len(self.suction)):
        x = self.suction[-i][0] * self.chord
        y = self.suction[-i][1] * self.chord
        createFuselageFrame(x, y, body, 18)

    xdeg = el.SubElement(body, 'x_degree')
    xdeg.text  = '3'
    hoop_deg = el.SubElement(body, 'hoop_degree')
    hoop_deg.text = '4'
    xpanels = el.SubElement(body, 'x_panels')
    xpanels.text = '19'
    hoop_panels = el.SubElement(body, 'hoop_panels')
    hoop_panels.text = '11'

    filename = str(name.text) + ".xml"

    with open(filename, 'wb') as file:
        file.write('<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE explane>'.
↪encode('utf-8'))
        el.ElementTree(explane).write(file, encoding='utf-8')

    print("Successfully create file "+ filename)
Fuselage.toXML = fuselageToXML
```

Then each of an aircraft's wing, elevator, fin, and fuselage can be converted to XML through an aircraft to XML method that calls the appropriate "to XML" function.

B.5 Miscellaneous Useful Functions

The total mass and the centre of mass for a list of components, each of which has its own mass and centre of mass can be found by:

```
def massAndCOM(_components):
    """
    Returns <tuple>( <float> total mass and <array-float>[x_bar, y_bar, z_bar] )
    Units: (kg, (m, m, m))
    Params:
        components - tuple of (mass, (x,y,z)), where coordinates are the component's centre of mass
    To access x coord: massAndCom(components)[1][0]
    Assumes constant density throughout component
    """
    total_m = 0.0
    total_md = [0, 0, 0]
    for comp in _components:
        total_m += comp[0]
        total_md[0] += comp[0] * comp[1][0]
        total_md[1] += comp[0] * comp[1][1]
        total_md[2] += comp[0] * comp[1][2]
    return (total_m, total_md)
```