McGill University

MECH 532

MECH 532 Design Project

Amro Al Baali
260629840

Felix C.-Lamy
260617874

Mohammed Shalaby
260607172

Course instructor: Professor Meyer Nahon

April 11, 2019

Abstract

In this project, our goal was to design a lightweight, battery-operated aircraft and to analyze its performance and stability characteristics. Mission requirements were provided, from which we had to determine manoeuvres that the aircraft would need to perform. The aerodynamic analysis was performed using XFLR5 and XFOIL, from which we have done the airfoil selection, wing and tail geometry design, trim conditions, static stability analysis, and the analysis of the aircraft dynamic modes. Furthermore, a CAD model of our design was built to determine its geometric and mass characteristics.

This report will explain our methodology and design process, as well as intermediate stages and goals set for the completion of our design. Clear design targets from which potential designs were evaluated will be presented.

Furthermore, the total endurance of the aircraft for the selected flight profile and design were calculated, as well as the static and dynamic stability of the aircraft. The methodology behind the these calculations will be presented.

Finally, an overview of the final design along with relevant 3D and 2D views of the design will be presented, with a discussion of possible future improvements in our design.

Contents

1 Introduction

1.1 Aerodynamic Requirements

- The aircraft must be designed to maximize endurance.
- Fly at cruise speed of 20 m/s.
- Fly at an altitude of 100 m.
- Aircraft muse be statically stable.
- Able to recover when disturbed from steady conditions.
- Allow adequate range of manoeuvres without stalling.

1.2 Geometric Requirements

- The aircraft must fit inside a 2x2x0.5 m box.
- Must use a T-tail empennage.
- Wing must be a high wing configuration.
- Symmetric tail airfoils.

1.3 Propulsion Requirements

- Motor: Tornado T7 4215 680 kV.
- Propeller: APC SlowFly 11x7.
- Battery: Turnigy Nano-Tech Plus 2200 mAh 4S LiPo.

1.4 Structural And mass Constraints

- Must carry a 1.0 kg cube shaped steel payload, ($\rho = 8050\ \mathrm{kg/m^3}$).
- Wing and tail components must not be hollow surfaces.
- Built using EPP foam, ($\rho = 30\ \mathrm{kg/m^3}$).
- Minimum thickness of 5 mm for fuselage sections.
- Any component than constitutes less than 20% of the total weight can be approximated as a point mass.
- Maximum wing loading of $6\ \mathrm{kg/m^2}$.

2 Literature Review

2.1 Component Specifications

2.1.1 Motor - Tornado T7 4215

- Assume input voltage of 14.8 V.
- Assume motor efficiency of 90% ($\eta_{\text{motor}} = 0.9$).
- Mass = 0.215 kg.

2.1.2 Propeller - APC SlowFly 11x7

The aircraft must be designed using an APC SlowFly 11x7 propeller. The performance characteristics of the propeller were obtained from the UIUC Propeller Database[1] [1, 2]. From the figures below, we can see that the maximum efficiency is achieved at the highest RPM (6000 RPM) and that the maximum thrust produced is approximately 0.11.

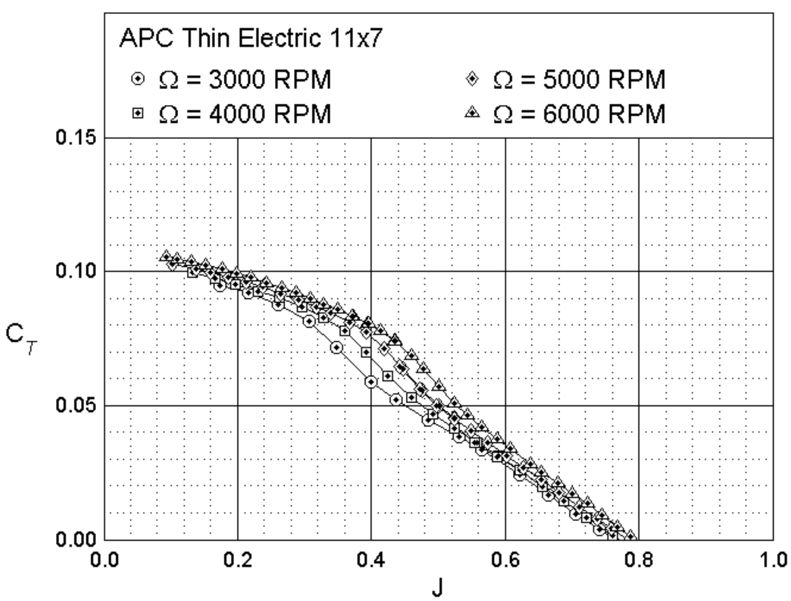


Figure 1: Coefficient of thrust of APC SlowFly 11x7.

[1]Link: https://m-selig.ae.illinois.edu/props/propDB.html

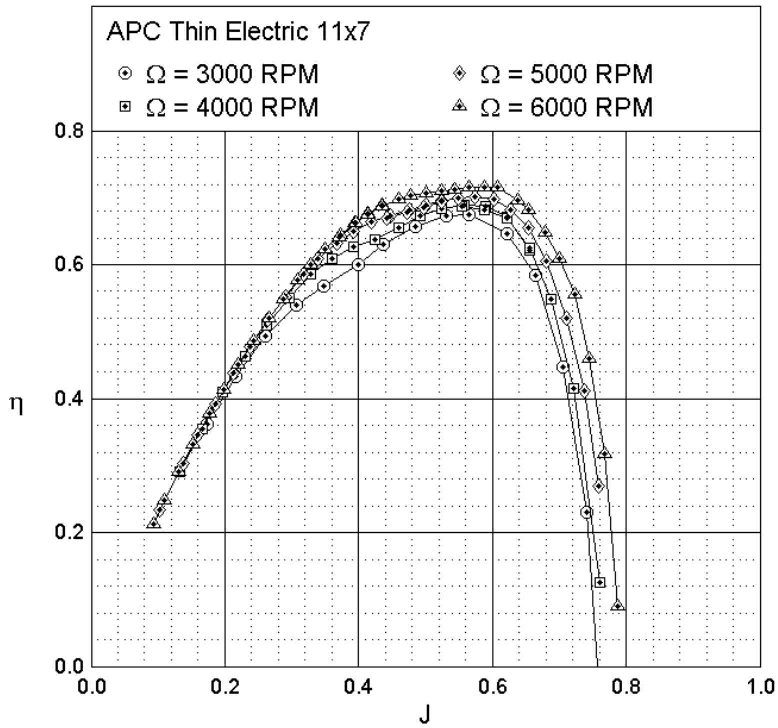


Figure 2: Propeller efficiency of APC SlowFly 11x7.

It is critical to have the above data for the aircraft performance simulations since the efficiency of the propeller varies greatly between various levels of thrust. Below are two equations used in the endurance calculations.

$$C_{\mathrm{t}} = \frac{\text{thrust}}{\rho * \eta^2 * D^4}, \tag{1}$$

$$J = \frac{v}{\eta * D}. \tag{2}$$

2.1.3 Battery - Turnigy Nano-Tech Plus

The battery is constructed from four LiPo cells connected in series (4S).
Specs

- $V_{\text{nom}} = 14.8$ V.
- Battery capacity. $Q_0 = 2.2$ Ah.
- Mass = 0.265 kg.

2.2 Performance Analysis

2.2.1 Propulsion Analysis

The energy calculations related to the battery State of Charge can be implemented using Coulomb counting. With this method, the current draw from the battery is integrated in time and assuming the airplane started the flight at 100% SOC, the remaining capacity in the battery can be estimated.

The current draw from the battery is the power required from the propeller, including propeller efficiency and motor efficiency, divided by the nominal voltage of the battery.

$$I_{\text{batt}} = Power/V_{\text{nom}}, \tag{3}$$

$$Power = \frac{T_{\text{prop}} * v}{\eta_{\text{prop}}\ \eta_{\text{motor}}}. \tag{4}$$

The battery capacity through the flight is calculated using

$$Q = Q_0 - \int_{t_1}^{t_2} I_{\text{batt}} \mathrm{d}t. \tag{5}$$

2.2.2 Takeoff

Equation of motion in x direction

$$T - D - \mu(mg - L) - mg\sin\theta = mA_x. \tag{6}$$

2.2.3 Climb

Equations of motion parallel to V

$$T - D - mg\sin\theta = m\frac{\mathrm{d}V}{\mathrm{d}t}. \tag{7}$$

Equations of motion perpendicular to V

$$L = mg\cos\theta. \tag{8}$$

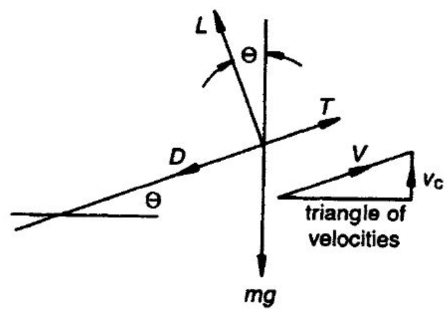


Figure 3: Forces acting on aircraft in climb flight

The total energy spent in climb phase can be calculated as follows.

$$E_{\text{climb}} = \frac{T_{\text{prop}} * V}{\eta_{\text{prop}}\ \eta_{\text{motor}}} * t_{\text{climb}}, \tag{9}$$

$$T = (\frac{1}{2}\rho A C_{\text{D}} V^2 + mg\sin\theta), \tag{10}$$

$$t_{\text{climb}} = \frac{\Delta H}{V\sin\theta}, \tag{11}$$

$$E_{\text{climb}} = \frac{(\frac{1}{2}\rho A C_{\text{D}} V^2 + mg\sin\theta)}{\eta_{\text{prop}}\ \eta_{\text{motor}}} * \frac{\Delta H}{\sin\theta}. \tag{12}$$

2.2.4 Steady Flight

Equations of motion in x direction

$$T - D = m\frac{\mathrm{d}V}{\mathrm{d}t} = 0, \tag{13}$$

$$T = D, \tag{14}$$

$$D = \frac{1}{2}\rho A C_{\text{D}} V^2. \tag{15}$$

Equations of motion in y direction

$$L = mg, \tag{16}$$

$$L = \frac{1}{2}\rho A C_{\text{L}} V^2. \tag{17}$$

The total energy spent in steady flight phase can be calculated as follows.

$$E_{\text{SF}} = \frac{T_{\text{prop}} * V}{\eta_{\text{prop}}\ \eta_{\text{motor}}} * t_{\text{SF}}, \tag{18}$$

$$T = \frac{1}{2}\rho A C_{\text{D}} V^2, \tag{19}$$

$$t_{\text{SF}} = \frac{Distance}{V}, \tag{20}$$

$$E_{\text{SF}} = \frac{\frac{1}{2}\rho A C_{\text{D}} V^2}{\eta_{\text{prop}}\ \eta_{\text{motor}}} Distance. \tag{21}$$

(22)

2.2.5 Banked Turn

Equations of motion in horizontal direction

$$L \sin\phi = m\frac{V^2}{R}. \tag{23}$$

Equations of motion in vertical direction

$$L \cos\phi = mg. \tag{24}$$

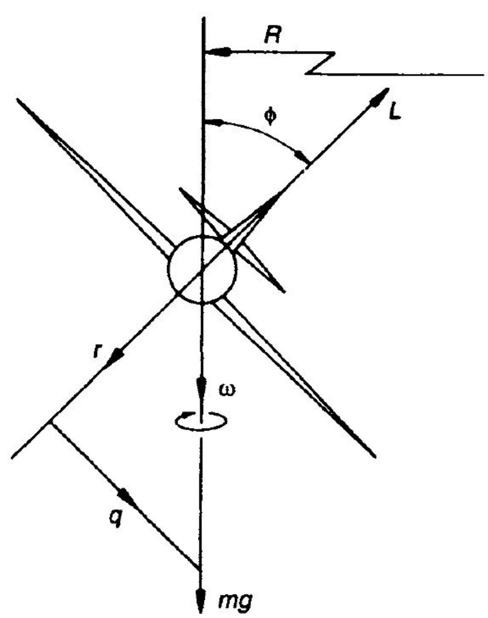


Figure 4: Aircraft in correctly banked turn.

2.2.6 Glide

Equation of motion parallel to V.

$$D = mg\sin\gamma. \tag{25}$$

Equation of motion perpendicular to V.

$$L = mg\cos\gamma. \tag{26}$$

In order to maximize endurance, the sink rate (V_s) must be minimized.

$$V_\mathrm{s} = \sqrt{\frac{2mg}{\rho S}}\frac{C_\mathrm{D}}{(C_\mathrm{L}^2 + C_\mathrm{D}^2)^{3/4}}. \tag{27}$$

2.3 Static Stability Analysis

2.3.1 Center of Gravity

Static stability revolves around the positioning of the center of gravity such that the aircraft is both statically stable and maneuverable. To do so, two main parameters must be taken into consideration

1. Neutral Point

$$h_\mathrm{n} = h_0 + \bar{V}_\mathrm{T}\frac{a_1}{a}\left(1 - \frac{\mathrm{d}\epsilon}{\mathrm{d}\alpha}\right). \tag{28}$$

2. Maneuver Point

$$h_\mathrm{m} = h_0 + \bar{V}_\mathrm{T}\left[\frac{a_1}{a}\left(1 - \frac{\mathrm{d}\epsilon}{\mathrm{d}\alpha}\right) + \frac{a_1\rho S l_\mathrm{T}}{2m}\right]. \tag{29}$$

where h_0 represents the aerodynamic centre and $h_i = x_i/\bar{c}$. These parameters are shown graphically in Fig. 5.

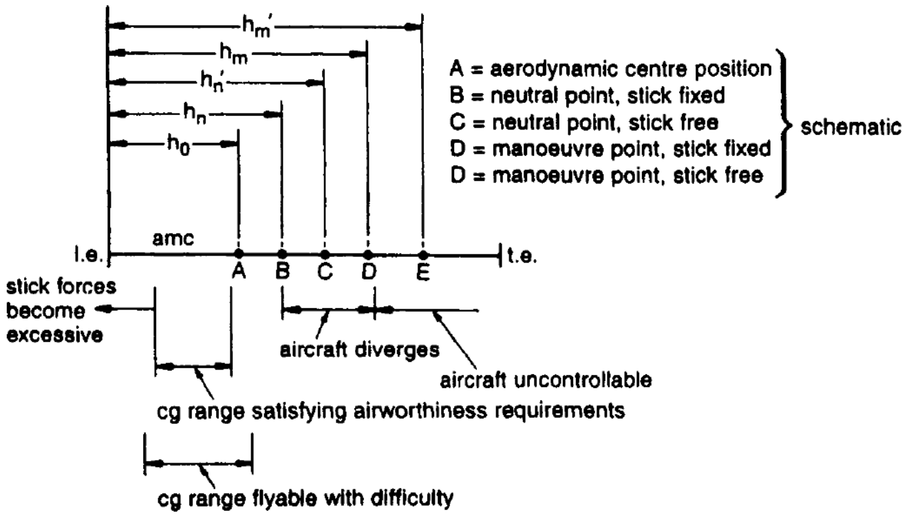


Figure 5: Schematic of various points of importance in static stability.

Consequently, the positioning of the center of gravity h allows us to define the neutral (static) and the maneuver margin as

$$H_{\mathrm{n}} = h_{\mathrm{n}} - h \tag{30}$$

$$H_{\mathrm{m}} = h_{\mathrm{m}} - h, \tag{31}$$

respectively, thus quantifying the distance between the center of gravity and those two crucial points. As Fig. 5 shows, the aft c.g. limit can be set to prevent static instability and uncontrollability.

Another variable dependent on the center of gravity is the amount of lift generated at steady, level flight. For the aircraft to be 'trimmed', C_{m} must be 0. Since

$$C_{\mathrm{m}} = C_{\mathrm{m}_0} + (h - h_0)\, C_{\mathrm{L}} - \bar{V}_{\mathrm{T}} \left[\frac{a_1}{a} \left(1 - \frac{\mathrm{d}\epsilon}{\mathrm{d}\alpha} \right) C_{\mathrm{L}} + a_1 \eta_{\mathrm{T}} + a_2 \eta + a_{\mathrm{e}} \beta \right], \tag{32}$$

C_{L} always varies with h at $C_{\mathrm{m}} = 0$, which in turn means that L varies with h since

$$L = \frac{1}{2} \rho V^2 S C_{\mathrm{L}}. \tag{33}$$

2.3.2 Stability Derivatives

More generally, the static stability of an aircraft can be determined by analyzing the stability derivatives. Stability derivatives essentially quantify the behaviour of forces and moments about the body axis of the aircraft as the vehicle accelerates in one of its 6 degrees of freedom. Static stability is determined by the signs of the different derivatives as shown in Tables 1-2, thus allowing the aircraft to return to its trimmed state when small perturbations occur.

Table 1: Longitudinal static stability conditions.

Stability derivative	**Stability condition**
X_u	< 0
X_w	> 0
X_q	> 0
$X_{\dot{q}}$	> 0
Z_u	< 0
Z_w	< 0
Z_q	< 0
$Z_{\dot{q}}$	< 0
M_u	< 0
M_w	< 0
M_q	< 0
$M_{\dot{q}}$	< 0

Table 2: Lateral static stability conditions.

Stability derivative	**Stability condition**
Y_v	< 0
Y_p	< 0
Y_r	> 0
L_v	< 0
L_p	< 0
L_r	> 0
N_v	> 0
N_p	< 0
N_r	< 0

2.4 Dynamic Stability Analysis

Dynamic stability is concerned with several open-loop aircraft dynamic modes, which characterize the behaviour of the aircraft over time. The different modes can be split into two categories; longitudinal and lateral. The eigenvalues associated with the different modes can be approximated from the stability derivatives mentioned earlier.

2.4.1 Short Period Mode

The short period mode is a longitudinal mode that is fast and heavily damped, seen in fluctuations of the pitch angle. The short-period poles are approximately equal to

$$\lambda_{\text{short}} = \frac{Z_w + M_q}{2} \pm \sqrt{\left(\frac{Z_w + M_q}{2}\right)^2 - M_q Z_w + M_q Z_q}. \tag{34}$$

2.4.2 Phugoid Mode

The phugoid mode is another longitudinal mode with a long period and light damping. The mode results in a response about the pitch angle and the forward velocity. The phugoid poles are approximately equal to

$$\lambda_{\text{phugoid}} = -\frac{Z_u X_q - X_u Z_q}{2Z_q} \pm \sqrt{\left(\frac{Z_u X_q - X_u Z_q}{2Z_q}\right)^2 + \frac{g Z_u}{Z_q}}. \tag{35}$$

2.4.3 Roll Mode

The roll mode is a first order lateral mode with a fast response, and results mainly in rolling with some sideslip. The rolling pole is approximately equal to

$$\lambda_{\text{rolling}} = L_p \; . \tag{36}$$

2.4.4 Spiral-divergence Mode

The spiral-divergence mode is also a first order lateral mode but with a very slow response, and mainly causes a change in heading. This mode could also be mildly unstable at high altitudes. The spiral-divergence pole is approximately equal to

$$\lambda_{\text{spiral}} = \frac{N_r L_v - N_v L_r}{L_v} \; . \tag{37}$$

2.4.5 Dutch-roll Mode

The dutch-roll mode is the only second order lateral mode resulting in rolling and yawing. This mode is usually the most concerning lateral mode for controller design/piloting. The dutch-roll poles are approximately equal to

$$\lambda_{\text{dutch roll}} = \frac{Y_v + N_r}{2} \pm \sqrt{\left(\frac{Y_v + N_r}{2}\right)^2 - (Y_v N_r + N_v Y_r)} \; . \tag{38}$$

One condition to ensure that the response of the aircraft converges rather than diverges is to have all poles in the left-hand plane; i.e., negative eigenvalues. If that is the case, then the time to half amplitude of the response for pole λ_i is

$$t_{\text{half}} = \frac{0.693}{\mathrm{Re}\left\{\lambda_i\right\}}; \tag{39}$$

otherwise, if the response diverges, the time to double amplitude is

$$t_{\text{double}} = \frac{0.693}{\mathrm{Re}\left\{\lambda_i\right\}}. \tag{40}$$

3 Mission Requirements

3.1 Flight Profile

The aircraft is designed to maximize endurance. It will fly at a cruise speed of 20 m/s and an altitude of 100 m. It must also takeoff and land from the same airport. Therefore, the aircraft will need to takeoff, climb to an altitude of 100 m, cruise at a speed of 20 m/s in steady flight for the longest possible time. Approximately halfway through the flight, the aircraft will need to perform a banked turn to make its way back to the airport. It will then accomplish a second steady flight period followed by a descent (glide) and landing.

Before approaching the design of the aircraft, it is also important to develop the theory to optimize the aircraft states (Speed, Flight path angle, Angle of Attack) in climb, steady flight and glide in order to maximize the performance potential (endurance) of each design iteration (i.e., various airfoil characteristics). Those flight states will likely vary between various designs thus will need to be optimized to each design iteration. The following section will explain our methodology for optimizing flight manoeuvres for endurance.

3.2 Maneuvers Optimization for Endurance

3.2.1 Climb

The first manoeuvre optimized was the climb flight. Our goal was to minimize the energy spent in climb. Our method consisted in numerically searching for the best flight states in climb. The total energy spent in climb was calculated for a range of speed, climb angles and angle of attacks. The flight states resulting in the lowest energy spent were selected. Our goal was to save as much energy as possible for the steady flight phase, as our aircraft will be optimised to that phase of flight.

$$
\begin{aligned}
\text{flight states} &= [\text{speed}, \text{climb angle}, \text{Angle of Attack}]^{\mathsf{T}} && (41)\\
&= [V, \theta, \alpha]^{\mathsf{T}}, && (42)
\end{aligned}
$$

$$E_{\text{climb}} = \frac{\left(\frac{1}{2}\rho A C_{\mathrm{D}} V^2 + mg\sin\theta\right)}{\eta_{\text{prop}}\,\eta_{\text{motor}}} * \frac{\Delta H}{\sin\theta}. \tag{43}$$

The aircraft performance characteristics, namely C_{L} vs α and C_{D} vs α were implemented using the MATLAB function `interp1`, with the output from XFLR.

While doing the flight states optimization, we found that it was critical to calculate a minimum Angle of attack for a given speed and angle of climb. Some combinations of the first two flight states were also impossible to achieve thus the energy result was discarded. Therefore, only realistic cases, taking into account how much lift the aircraft could produce at that speed, were kept.

3.2.2 Takeoff

For the takeoff, no optimization of the flight states was made. However, we used the climb speed as the rotation speed for calculating the ground run. A forward Euler simulation was created where the propeller thrust, drag and friction drag vary non-linearly with speed. Thus, the MATLAB function `interp1` was used for calculating C_L, C_D, and C_t every step of the simulation.

3.2.3 Steady Flight

For the steady flight, only one of the flight states had to be optimized, namely the angle of attack. Both the aircraft speed and flight path angle were decided by our mission requirements ($V = 20$ m/s and $\theta = 0$). The angle of attack did not require an optimization, since it would simply need to be sufficient to provide the required lift for steady level flight. The angle of attack at steady flight was in a way "baked in" our aircraft design, and more precisely the airfoil characteristics.

In order to maximize endurance in steady flight, the aircraft needs to fly at minimum power condition.

$$P_r = \left(\frac{D}{L}\right) LV \tag{44}$$

$$= \left(\frac{C_\mathrm{D}}{C_\mathrm{L}}\right)(mg)\frac{2mg}{\rho S C_\mathrm{L}} \tag{45}$$

$$= \left(\frac{C_\mathrm{D}}{C_\mathrm{L}^{3/2}}\right)\sqrt{\frac{2}{\rho S}}(mg)^{3/2}. \tag{46}$$

$$\tag{47}$$

Thus, to minimize the power output, the metric $\frac{C_\mathrm{D}}{C_\mathrm{L}^{3/2}}$ needs to be minimized. We will refer to this metric as the steady flight endurance metric.

3.2.4 Banked Turn

For the banked turn, we decided to keep the same flight states as steady flight except for the angle of attack. Thus, the aircraft performs the manoeuvre at the same speed and flight path angle, but increases the angle of attack in order to maintain altitude. Furthermore, after analysis, the radius of turn was chosen to be approximately 150 m, as it was calculated that higher turn radius would lead to minimal gains in terms of energy.

3.2.5 Glide

For the last phase of the flight, namely the glide or descent, the flight states were optimized to minimize the aircraft's sink rate. The equation for the sink rate is:

$$V_{\mathrm{s}} = \sqrt{\frac{2mg}{\rho S}} \frac{C_{\mathrm{D}}}{(C_{\mathrm{L}}^2 + C_{\mathrm{D}}^2)^{3/4}}. \tag{48}$$

Assuming that $C_{\mathrm{L}} \gg C_{\mathrm{D}}$, we can drop the C_{D}^2 and we get

$$V_{\mathrm{s}} = \sqrt{\frac{2mg}{\rho S}} \frac{C_{\mathrm{D}}}{C_{\mathrm{L}}^{3/2}}. \tag{49}$$

Thus, minimizing $\frac{C_{\mathrm{D}}}{C_{\mathrm{L}}^{3/2}}$ will also minimize the sink rate and thus maximize endurance. Therefore, for given aircraft lift and drag characteristics, the optimal angle of attack to minimize the metric $\frac{C_{\mathrm{D}}}{C_{\mathrm{L}}^{3/2}}$ is found using the MATLAB function `polyfit and interp1` and the lift and drag tabulated data from XFLR.

4 Design and Analysis

4.1 Overview

Our design methodology was optimized around our goal of maximizing endurance, with a focus on steady flight.

The very first step in our design process was to estimate the mass of our aircraft. A simple spreadsheet was built to summarize the mass of all components and calculate the total mass. This allowed us to make an initial estimate of the lift required from the airfoils during the flight.

The second design step consisted in selecting the wing airfoil, with the goal of optimizing our steady flight endurance metric to maximize endurance. Preliminary design parameters were set up at this point.

The third step in our design was setting up a simulation of the complete aircraft mission (takeoff, climb, steady flight, banked turn, steady flight, glide) with the goal of calculating its endurance. Even thought the takeoff time is not included in the endurance, it was necessary to calculate the energy spent accelerating on the runway.

Once the performance aspects of the potential design were set up, the fourth step was to analyze the static stability of the aircraft using XFLR5. This step allowed us to verify and optimize the geometrical parameters of the airplane. These parameters include the placement of the CoG by moving the different components, such as the battery and the payload. In addition, XFLR5 was used to generate the aerodynamic behaviour of the full aircraft structure.

In the fifth step, we proceeded with evaluating the dynamic stability of the aircraft and the designing the control surfaces to meet the design targets. In addition, different twist and dihedral angles were considered and their effect on the dynamic stability and aerodynamics were analyzed.

Finally, a last round of analysis was performed with the updated geometry, mass and control surface area in order to confirm the performance and stability of our final aircraft design.

4.2 Design Targets

- Maximize endurance, with goal of reaching 1 hour total endurance.
- Minimize mass, with goal of $mass_{a/c} < 2$ kg.
- Maximum wing loading of 6 kg/m^2.
- Target roll acceleration of 10 deg/s.
- Target pitch acceleration of 10 deg/s.
- Statically stable aircraft with a static margin between 10% and 20%.
- Dynamically stable aircraft with all poles in the CLHP.

- Phugoid mode target half amplitude time of 30 seconds.
- Dutch roll mode target half amplitude time of 15 seconds.

4.3 Airfoil Selection

To select our airfoil, we used our own MATLAB script combined with XFOIL to simulate all NACA four digit series airfoils. From the full results data, we then selected five best airfoil candidates using an objective function.

The objective function's goal was to minimize our endurance metric at our approximate operating required coefficient of lift in steady flight. Equation 17 with an initial mass estimate was used to calculate the approximate C_{L} required at steady level flight.

$$C_{\mathrm{L}} = \frac{mg}{\frac{1}{2}\rho S V^2}, \tag{50}$$

$$Metric = \frac{C_{\mathrm{D}}}{C_{\mathrm{L}}^{3/2}}, \tag{51}$$

$$L = 0.2 \text{ m}, \tag{52}$$

$$V = 20 \text{ m/s}, \tag{53}$$

$$\mu = 1.562 \cdot 10^{-5} \text{ at } 25°\mathrm{C}, \tag{54}$$

$$\text{Re} = vL/\mu \approx 250\ \ 000, \tag{55}$$

$$\text{Mach} = u/c \approx 0.058. \tag{56}$$

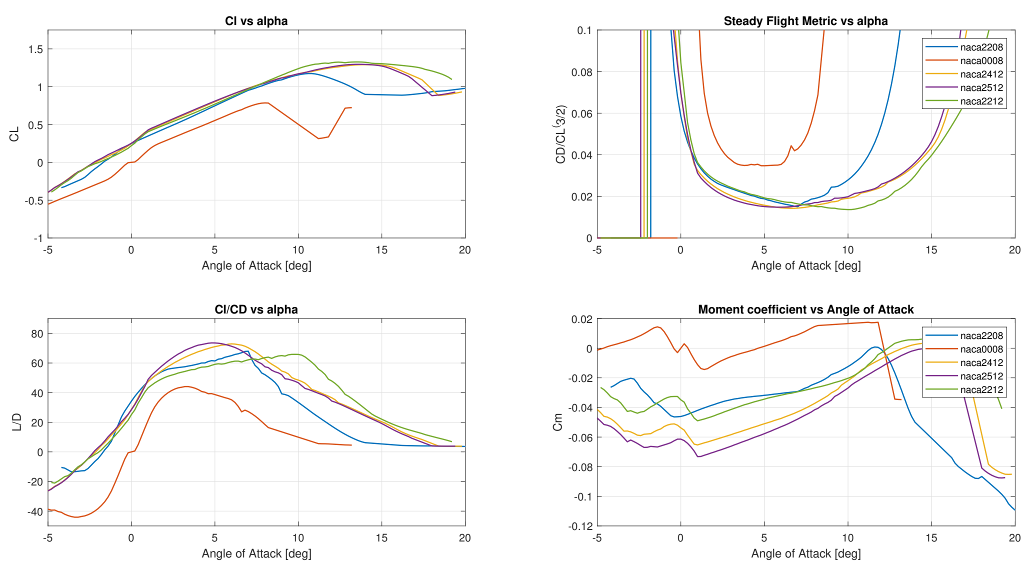


Figure 6: Airfoil selection key performance metrics.

From our results, we can see that the two best candidates are the NACA 2208 and the NACA 2512. The NACA 2208 efficiency is particularly good at around $\alpha = 0$, both in terms of L/D and our endurance efficiency metric. The disadvantages of NACA 2208 compared to NACA 2512 are its lower peak C_{L}, lower stall angle and higher C_{m}. Therefore, the next steps in our design process will allow us to decide what airfoil is better for our application.

Another advantage of the NACA 2208 is its lower cross-sectional area. The wing will be more than 2 m long therefore it could have a significant effect on the total aircraft mass. Thus, the choice of a NACA 2208 would help us achieve our design target of minimizing mass.

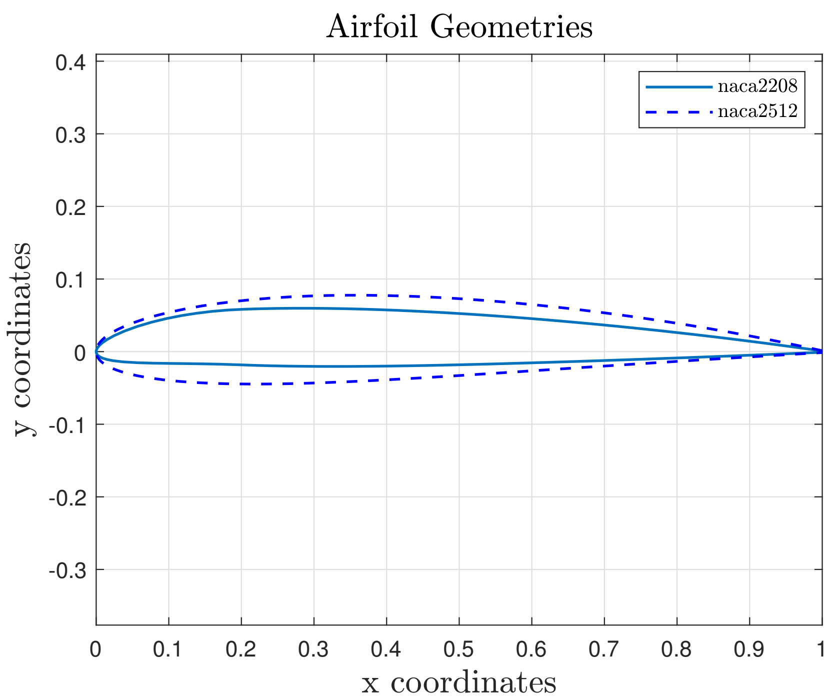


Figure 7: NACA 2208 and NACA 2512 geometries.

Therefore, from our airfoil analysis, we have decided to go forward with our two candidates: NACA 2208 and NACA 2512. The next steps in the analysis will allow us to select the best airfoil.

4.4 Endurance Analysis

To allow initial simulation of the aircraft performance, preliminary design parameters were set up at this point. These parameters, as will be discussed in more detail in Section 5.1, include but are not limited to the dimensions of the fuselage, main wing, horizontal stabilizer and the fin as well as the positioning of the different components, such as the battery, motor, payload and the main wing. One constraint to be taken into consideration through the design is the maximum wing loading of $6\ \mathrm{kg/m^2}$. The goal was to keep the chord small enough to maximize the aspect ratio, but big enough to satisfy this requirement. It can be shown that this requirement is achieved by satisfying the following equation:

$$\text{mass of wing} = 12 \cdot \text{chord of wing} - \text{sum of masses of other components.} \tag{57}$$

Each design iteration was simulated through the full flight profile, thus through takeoff, climb, steady flight, banked turn and glide.

4.4.1 Takeoff

As mentioned in Section 3.2.2, the takeoff was calculated with a forward-Euler numerical integration, using (6). The total energy for takeoff was recorded in order to know how much energy was left for the following manoeuvres. See simulation code in Appendix A for further details.

4.4.2 Climb

As mentioned in Section 3.2.2, the flight states for the climb were optimized to minimize energy. Once the flight states were calculated, the total time for climb could be calculated using the speed and angle of climb. The total energy required for climb was calculated using (9).

4.4.3 Banked Turn

The banked turn endurance and total energy was calculated prior to calculating the steady flight phase. This was done in order to calculate the total energy left for the steady flight. The banked turn calculations were done using equations from Section 2.2.5.

4.4.4 Steady Flight

For calculating the steady flight endurance, the total energy available for steady flight was first calculated. Thus, the energy required to takeoff, climb and turn was subtracted from the initial battery energy. Furthermore, a 5% safety margin was included in the energy calculation. Therefore, the plane would land with 5% battery capacity.

$$\Delta E_{\mathrm{SF}} = \Delta E_{0_{\mathrm{batt}}} - \Delta E_{\mathrm{takeoff}} - \Delta E_{\mathrm{climb}} - \Delta E_{\mathrm{banked_turn}} - \Delta E_{\mathrm{safety}} \tag{58}$$

$$\Delta E_{\mathrm{safety}} = 0.05 \Delta E_{0_{\mathrm{batt}}} \tag{59}$$

The steady flight total endurance was then calculated using the following equation:

$$Endurance = \frac{E_{\mathrm{SF}} \eta_{\mathrm{prop}}\ \eta_{\mathrm{motor}}}{(\frac{1}{2}\rho A C_{\mathrm{D}} V^2) * V} \tag{60}$$

4.4.5 Glide

For the last phase of the flight, namely the glide or descent, the flight states were optimized to minimize the aircraft's sink rate.

$$V_{\mathrm{s}} = \sqrt{\frac{2mg}{\rho S}} \frac{C_{\mathrm{D}}}{(C_{\mathrm{L}}^2 + C_{\mathrm{D}}^2)^{3/4}}. \tag{61}$$

Assuming that $C_{\mathrm{L}} \gg C_{\mathrm{D}}$, we can drop the C_{D}^2 and we get

$$V_{\mathrm{s}} = \sqrt{\frac{2mg}{\rho S}} \frac{C_{\mathrm{D}}}{C_{\mathrm{L}}^{3/2}}. \tag{62}$$

Once the optimal sink rate was calculated, the total endurance of the glide was calculated.

4.5 Static Stability Analysis

As mentioned in Section 2.3.1, the CoG position is determined by evaluating the required lift at steady level flight. The goal is that at $C_{\mathrm{m}} = 0$, the aircraft travels naturally at the required C_{L} without flap inputs which result in increased drag.

Steady level flight is simulated using MATLAB, where the required C_{L} is calculated using (17) and (33), based on the different aircraft geometries and parameters as well as the constant flight conditions. Consequently, with the help of XFLR5, different CoG positions are analyzed until we converge iteratively to a CoG position that provides the required C_{L} at $C_{\mathrm{m}} = 0$. Fig. 8 shows one such case, where the goal was to find the CoG position that results in $C_{\mathrm{L}} = 0.24$ at $C_{\mathrm{m}} = 0$; therefore, the red line was the chosen one. The desired CoG position was achieved by moving the payload, which constitutes $\approx 50\%$ of the weight of the plane.

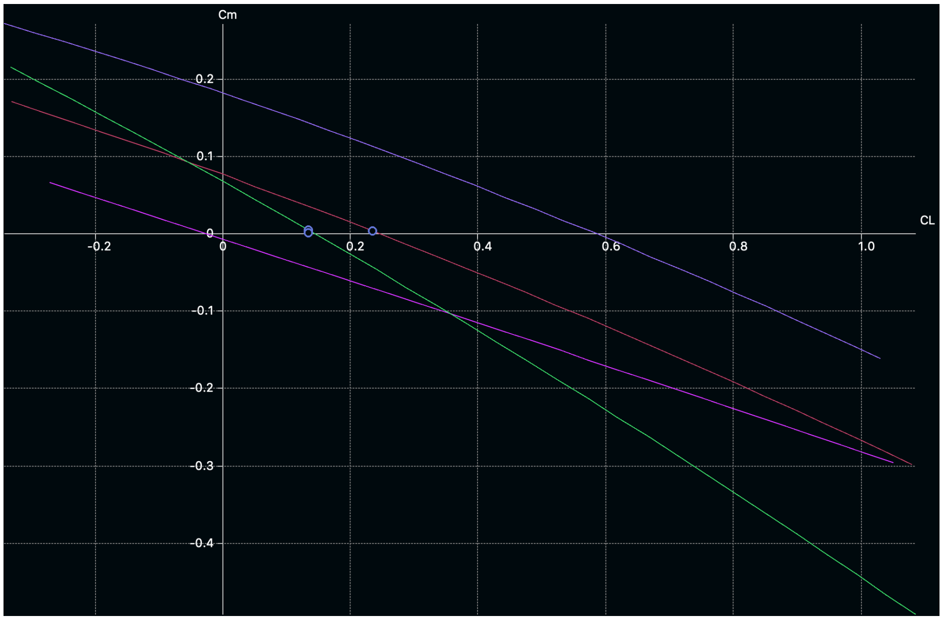


Figure 8: C_{m} vs. C_{L} for different CoG positions.

The next step is then to calculate the static margin based on (30), with the only requirement that the neutral point is aft the CoG, to ensure static stability. This also means that the positive maneuver margin requirement can be ignored for now, as the former requirement ensures the latter one. The last step of the static stability analysis, which is to be performed only for the final design, is to compute and compare the different stability derivatives to the requirements mentioned in Table 1 and Table 2.

4.6 Dynamic Stability Analysis

The main requirement for dynamic stability is that the eigenvalues for the different modes are negative. Some modes (mainly the spiral mode) can be mildly unstable; however, this is not favourable. In addition, to be able to compare different design iterations, (39) and (40) can be used to better understand how the response converges over time. By running a stability analysis, XFLR5 was utilized to compute all the poles for the different planes. XFLR5 also computes the time response of the associated states for every mode, which were also considered.

4.7 Control Surfaces Analysis

Standard control surfaces were implemented in the design of this aircraft. An aileron, an elevator and a rudder were designed on the main wing, horizontal stabilizer and fin respectively. There are three variables determining the influence of the control surfaces on the response of the plane. These are

1. % chord,

2. width of flap, and
3. angle of flap.

The transition between the ground run and the climb requires an elevator input, while the transition between steady level flight and the banked turn requires an aileron input. The associated target accelerations are 10 deg/s, as specified in Section 4.2. Therefore, the required moments in both conditions can be calculated as

$$M_\eta = \dot{h}_2 = I_{yy} \cdot 10 \frac{\pi}{180}, \tag{63}$$

$$L_\xi = \dot{h}_1 = I_{xx} \cdot 10 \frac{\pi}{180}. \tag{64}$$

5 Design Iterations

5.1 Preliminary Design

5.1.1 Overview

The goal of our first design iteration was to finalize the airfoil selection for our wing (NACA 2208 vs NACA 2512). In order to do so, we first estimated the mass of the aircraft. Once we had a weight estimate, we simulated the endurance of both wing configurations through the full flight profile. The wing associated with NACA 2512 airfoil was significantly heavier (1.8 times heaver) than the wing associated with NACA 2208 airfoil, which in turn gave lower endurance (16% less endurance) thus NACA 2512 airfoil airfoil was dismissed.

5.1.2 Geometry

For the preliminary design, the component masses were estimated using conservative mathematical equations. For example, the main wing was assumed to be a cuboid of length b, width $\bar{c}$, and thickness t_{max}. Thus the total weight was expected to be an overestimate. A more accurate estimate was obtained in Design Iteration II.

Furthermore, the component positions were arbitrarily chosen. More accurate positioning was done in the Final Design.

Table 3: Preliminary design component masses (estimates).

Component	**Mass** (kg)
Main Wing	0.665
Fuselage	0.092
Fin	0.032
Stabilizer	0.121
Propulsion	0.480
Wheels	0.050
Payload	1.000
Total	**2.441**

Table 4: Preliminary design aircraft dimensions.

Dimension	**NACA 2208**	**NACA 2512**
Wing span	2000 mm	2000 mm
Chord	203 mm	250 mm
Horizontal stabilizer span	600 mm	600 mm
Horizontal stabilizer MAC	110 mm	110 mm
Fin span	250 mm	250 mm
Fin MAC	139.5 mm	139.5 mm
Wing dihedral	0.0°	0.0°

Table 5: Preliminary design component positions relative to aircraft nose.

Dimension	**Value**
C.G.-x	326.6 mm
C.G.-Z	17.6 mm
Neutral Point-x	399.4 mm
Wing LE-x	274.1 mm
Wing LE-z	70.8 mm
H. stabilizer root LE-x	1403.5 mm
H. stabilizer root LE-z	248.5 mm
Fin root LE-X	1343.8 mm
Fin root LE-z	0.0 mm
Payload-x	363.7 mm
Payload-z	15.4 mm
Battery-x	—
Battery-z	—
Motor-x	—
Motor-z	—

5.1.3 Performance

From our endurance analysis, our configuration with the NACA 2208 has a 19% greater endurance compared to the NACA 2512 endurance. This performance difference is significant and is mainly due to the difference in airfoil performance and aircraft mass. The only potential disadvantage of the NACA 2208 airfoil is its low C_{m}. Therefore, further stability analysis will be required to finalize the selection.

Table 6: Design Iteration I performance.

Wing Configuration	**Endurance** (sec)
NACA 2208	45.18
NACA 2512	37.96

5.1.4 Static Stability

Only the static margin is presented for this subsection. The stability derivatives of the final design are presented in Subsection 6.4.

$$\frac{x_{\text{C.G.}} - x_{\text{N.P.}}}{\bar{c}} = \frac{|326.6 - (399.4)|}{203.4} \tag{65}$$
$$= 35.8\%. \tag{66}$$

5.1.5 Dynamic Stability

Table 7 summarizes the dynamic response of the aircraft. The spiral mode was unstable so that was the main focus in Design Iteration I. Furthermore, the position of the payload and other components was not carefully assessed at this stage of the design.

Table 7: Dynamic stability.

Mode	$\mathrm{Re}\{\lambda\}$	**Stable mode**	$t_{1/2}$ (sec)
Phugoid	-0.0176	✓	39.375
Short period	-13.46	✓	0.051485884
Dutch roll	-26.48	✓	0.026170695
Sprial	0.04798	✗	-14.44351813
Roll convergence	-26.48	✓	0.026170695

5.2 Design Iteration I

5.2.1 Overview

The goal of Design Iteration I was to stabilize the aircraft as the preliminary design gave unstable spiral mode. The aircraft was stabilized by adding 5.0° dihedral to the main wings.

5.2.2 Geometry

The component masses, dimensions, and positions are identical to the ones estimated in the Preliminary Design. The only difference to the Preliminary Design is the addition of the dihedral to the main wing.

The aircraft masses were the same as in the Preliminary Design. Refer to Table 3 for complete masses details.

Table 8: Design iteration I aircraft dimensions.

Dimension	Value
Wing span	2000 mm
Chord	203.4 mm
Horizontal stabilizer span Horizontal stabilizer MAC	mm 110 mm
Fin span	500 mm
Fin MAC	139.5 mm
Wing dihedral	0.0°

Table 9: Design iteration I component positions relative to aircraft nose.

Dimension	Value
C.G.-x	169 mm
C.G.-Z	7.0 mm
Neutral Point-x	**558.0 mm**
Wing LE-x	274.1 mm
Wing LE-z	0.0 mm (no fuselage)
H. stabilizer root LE-x	1403.5 mm
H. stabilizer root LE-z	500 mm
Fin root LE-X	1343.8 mm
Fin root LE-z	0.0 mm
Payload-x	50.0 mm
Payload-z	0.0 mm
Battery-x	10.0 mm
Battery-z	0.0 mm
Motor-x	0.0 mm
Motor-z	0.0

5.2.3 Performance

Table 10: Design Iteration I performance.

Manoeuvre	**Time** (sec)	**Capacity** (As)
Takeoff	8.8	61.1
Climb	54.87	270.45
Steady Flight	1217.5	3561
Banked Turn	23.56	69.36
Steady Flight	1217.6	3561
Glide	191.25	0.00
Total	**45.09 min**	**–**

5.2.4 Static Stability

Only the static margin is presented for this subsection. The stability derivatives of the final design are presented in Section 6.4.

$$\frac{x_{\text{C.G.}} - x_{\text{N.P.}}}{\bar{c}} = \frac{|169.0 - (558.0)|}{203.4} \tag{67}$$

$$= 191\%. \tag{68}$$

5.2.5 Dynamic Stability

Table 11 summarizes the dynamic response of the aircraft; all the modes are stable.

Table 11: Dynamic stability.

Mode	$\mathrm{Re}\{\lambda\}$	**Stable mode**	$t_{1/2}$ (sec)
Phugoid	-0.0172	✓	40.29069767
Short period	-13.36	✓	0.051871257
Dutch roll	-26.35	✓	0.02629981
Sprial	-0.03366	✓	20.58823529
Roll convergence	-2.049	✓	0.0049

5.3 Design Iteration II

5.3.1 Overview

The goal of Design Iteration II was to generate a detailed CAD of the aircraft in order to get a more accurate component weights. This design iteration also involved changing the aircraft dimensions such as the aircraft's total length and the fin span. Furthermore, the payload's position was tuned to maintain the aircraft's stability.

5.3.2 Geometry

The component masses were updated after modelling the aircraft using a CAD software (Autodesk Inventor®). The aircraft dimensions such as the total length were updated to stabilize the aircraft.

Table 12: Design iteration II component masses.

Component	**Mass** (kg)
Main Wing	0.147
Fuselage	0.051
Fin	0.009
Stabilizer	0.010
Propulsion	0.480
Wheels	0.050
Payload	1.000
Total	**1.747**

Table 13: Design iteration II aircraft dimensions.

Dimension	**Value**
Wing span	2000 mm
Chord	203.4 mm
Horizontal stabilizer span	600 mm
Horizontal stabilizer MAC	110 mm
Fin span	500 mm
Fin MAC	139.5 mm
Wing dihedral	$5.0°$

Table 14: Design iteration II component positions relative to aircraft nose.

Dimension	**Value**
C.G.-x	169 mm
C.G.-Z	7.0 mm
Neutral Point-x	74.0 mm
Wing LE-x	274.1 mm
Wing LE-Y	35.0 mm
H. stabilizer root LE-x	1403.5 mm
H. stabilizer root LE-z	500 mm
Fin root LE-X	1343.8 mm
Fin root LE-z	0.0 mm
Payload-x	50.0 mm
Payload-z	0.0 mm
Battery-x	10.0 mm
Battery-z	0.0 mm
Motor-x	0.0 mm
Motor-z	0.0

5.3.3 Performance

Table 15: Design Iteration II performance.

Manoeuvre	**Time** (sec)	**Capacity** (As)
Takeoff	3.64	31.9
Climb	42.93	191.48
Steady Flight	1283.5	3617.5
Banked Turn	23.56	66.57
Steady Flight	1283.5	3617.5
Glide	228.32	0.0
Total	**47.69 min**	**–**

5.3.4 Static Stability

Only the static margin is presented for this subsection. The stability derivatives of the final design are presented in Section 6.4.

$$\frac{x_{\text{C.G.}} - x_{\text{N.P.}}}{\bar{c}} = \frac{169.0 - 74.0}{203.4} \tag{69}$$

$$= 46.7\%. \tag{70}$$

5.3.5 Dynamic Stability

Table 16 summarizes the dynamic response of the aircraft; all the modes are stable. However, the phugoid mode takes too long to dissipate. In the final design, the payload position was relocated and the chord length adjusted to get better stability performance.

Table 16: Dynamic stability.

Mode	$\mathrm{Re}\{\lambda\}$	**Stable mode**	$t_{1/2}$ (sec)
Phugoid	-0.02335	✓	296.8
Short period	-39.73	✓	0.017
Dutch roll	-7.971	✓	0.087
Sprial	-0.04188	✓	16.54
Roll convergence	-140.6	✓	0.0049

6 Final Design

6.1 Overview

The final design was a wrap up to Design Iteration II. The aircraft's total weight was recalculated in Design Iteration II using the CAD 3D model. The more-accurate recalculated weight (1.747 kg) was lighter than the initial estimate (2.441 kg), so further aerodynamic analysis was carried to choose the appropriate chord length while keeping the wing loading under $6\ \mathrm{kg/m^2}$. The stability analysis was carried again to ensure that the aircraft is statically and dynamically stable and has appropriate responses.

6.2 Geometry

The final design aircraft dimensions are summarized as in Table 18 and the positions of the components relative to the aircraft nose are summarized in Table 19. Note that a drawing of the detailed dimensions is provided in Appendix C. Fig. 9 shows the CAD of the aircraft.

Table 17: Final design component masses.

Component	**Mass** (kg)
Main Wing	0.102
Fuselage	0.051
Fin	0.009
Stabilizer	0.010
Propulsion	0.480
Wheels	0.050
Payload	1.000
Total	**1.702**

Table 18: Final design aircraft dimensions.

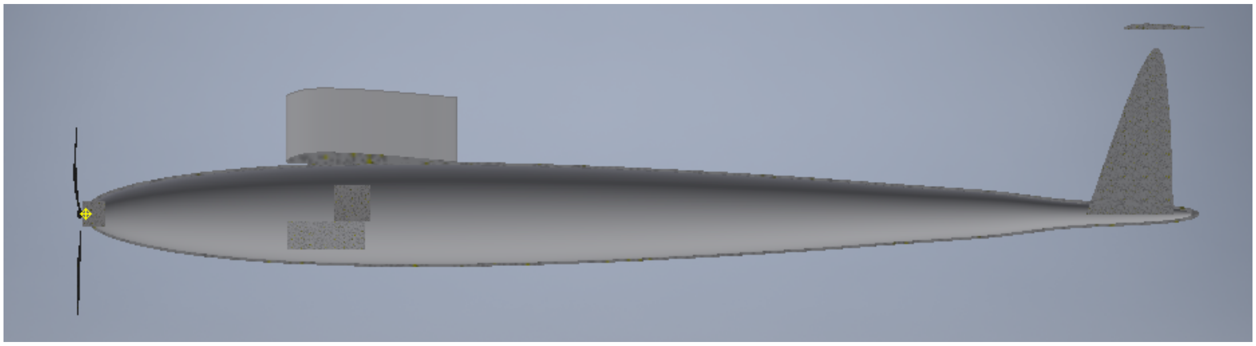
Dimension	**Value**
Wing span	2000 mm
Chord	141 mm
Horizontal stabilizer span	600 mm
Horizontal stabilizer MAC	110 mm
Fin span	250 mm
Fin MAC	139.5 mm
Wing dihedral	$5.0°$

Table 19: Final design component positions relative to aircraft nose.

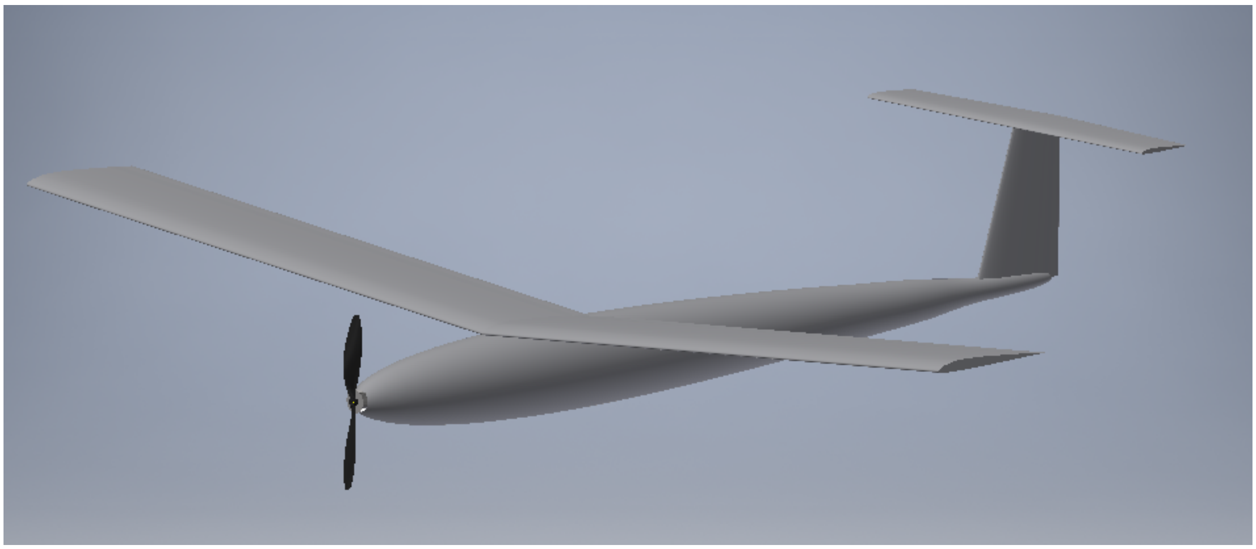
Dimension	Value
C.G.-x	326.6 mm
C.G.-Z	17.6 mm
Neutral Point-x	431.1 mm
Wing LE-x	274.1 mm
Wing LE-z	70.8 mm
H. stabilizer root LE-x	1403.5 mm
H. stabilizer root LE-z	248.5 mm
Fin root LE-X	1343.8 mm
Fin root LE-z	0.0 mm
Payload-x	363.7 mm
Payload-z	15.4 mm
Battery-x	325.0
Battery-z	-10.0
Motor-x	0.0
Motor-z	0.0

Final design inertia tensor:

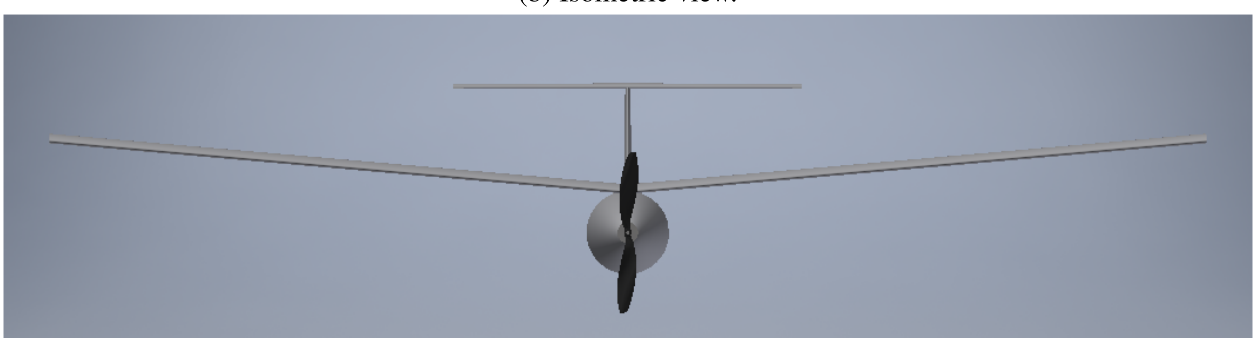
$$\mathbf{I} = \begin{bmatrix} 0.03538 & 0.00000 & -0.00641 \\ 0.00000 & 0.07752 & 0.00000 \\ -0.00641 & 0.00000 & 0.11012 \end{bmatrix} \mathrm{kg/m^2}. \tag{71}$$



(a) Section view.



(b) Isometric view.



(c) Front view.

Figure 9: Aircraft final design CAD views.

6.3 Performance

Table 20: Final Design performance.

Manoeuvre	**Time** (sec)	**Capacity** (As)
Takeoff	4.06	42.56
Climb	37.41	187.21
Steady Flight	1367.6	3615.9
Banked Turn	23.56	62.46
Steady Flight	1367.6	3615.9
Glide	226.76	0.0
Total	**50.38 min**	**–**

6.4 Static Stability

Table 21: Final aircraft longitudinal static stability.

Stability derivative	**Value**	**Stability condition**	**Condition satisfied**
X_u	-0.072051	< 0	✓
X_w	0.61032	> 0	✓
Z_u	-1.4537	< 0	✓
Z_w	-24.335	< 0	✓
Z_q	-5.3959	< 0	✓
M_w	-3.112	< 0	✓
M_q	-4.9238	< 0	✓

Table 22: Final aircraft lateral static stability.

Stability derivative	**Value**	**Stability condition**	**Condition satisfied**
Y_v	-1.5511	< 0	✓
Y_p	-0.79228	< 0	✓
Y_r	1.8096	> 0	✓
L_v	-1.0134	< 0	✓
L_p	-5.7214	< 0	✓
L_r	0.74547	> 0	✓
N_v	1.5105	> 0	✓
N_p	-0.21208	< 0	✓
N_r	-1.8829	< 0	✓

Table 23: Dynamic stability.

Mode	$\mathrm{Re}\{\lambda\}$	**Stable mode**	$t_{1/2}$ (sec)
Phugoid	-0.0221	✓	31.6
Short period	-38.82	✓	0.018
Dutch roll	-9.13	✓	0.076
Sprial	-0.03727	✓	18.6
Roll convergence	-164.2	✓	0.0042

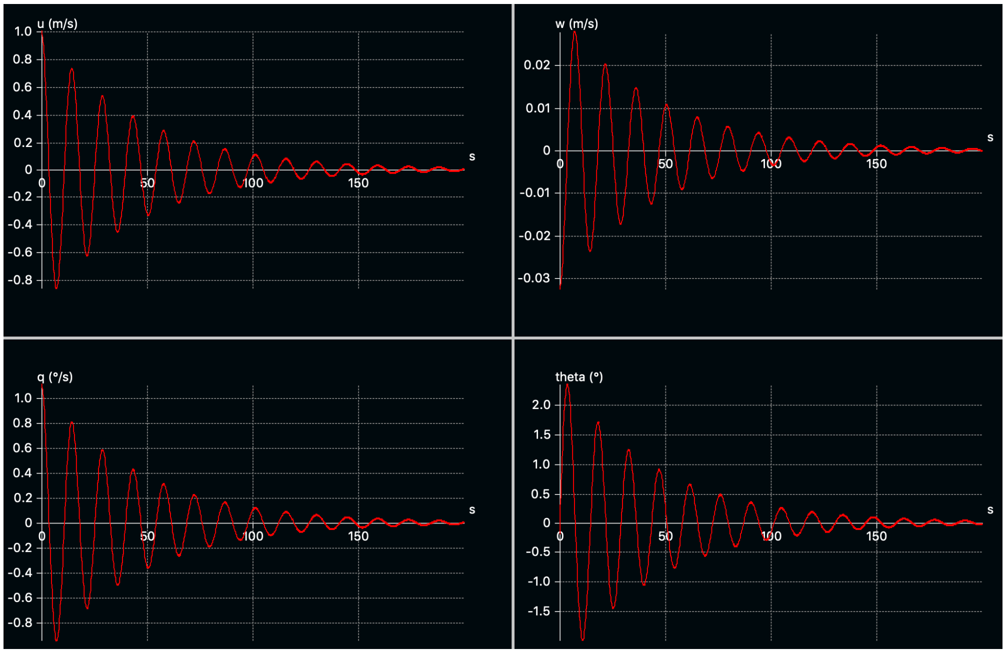


Figure 10: Phugoid mode response.

6.5 Dynamic Stability

Table 23 summarizes the dynamic response of the aircraft. The phugoid mode half-amplitude time is close to the sought-after target of 30 seconds. The aircraft is stable in all modes, maybe "too" stable in some cases. Figures 10-14 show the dynamic responses of the system; they are within a reasonable range. It should be noted that these are the natural responses of the aircraft; better performance can be achieved if a controller is used.

6.6 Control Surfaces

As mentioned in Section 4.7, the required moments must first be obtained to be able to design the control surfaces. Using the inertia tensor as specified in (71) as well as (63) and (64), we can

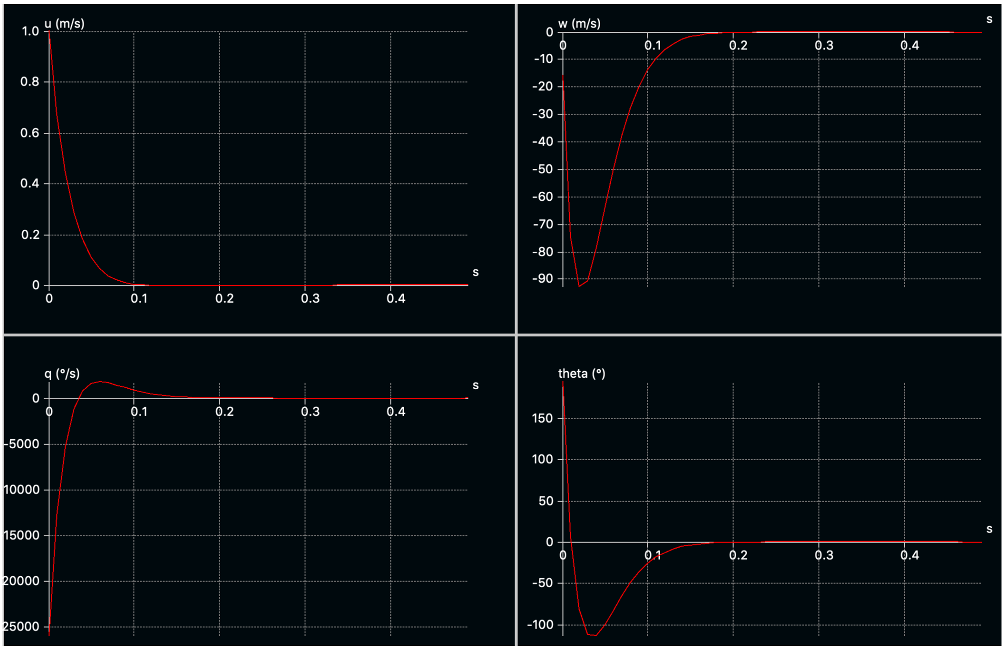


Figure 11: Short period mode response.

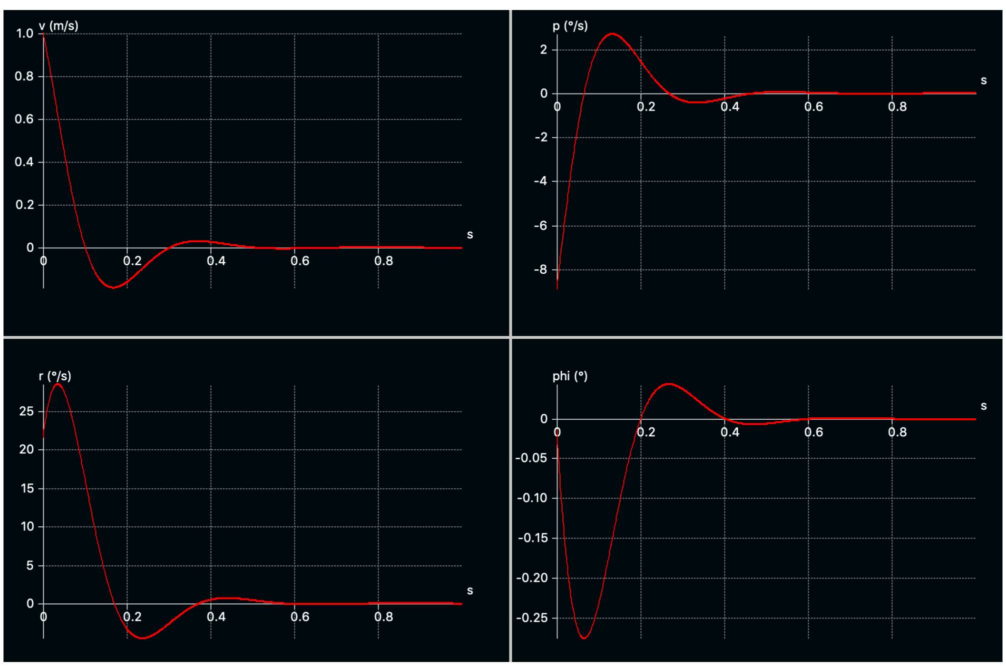


Figure 12: Dutch roll mode response.

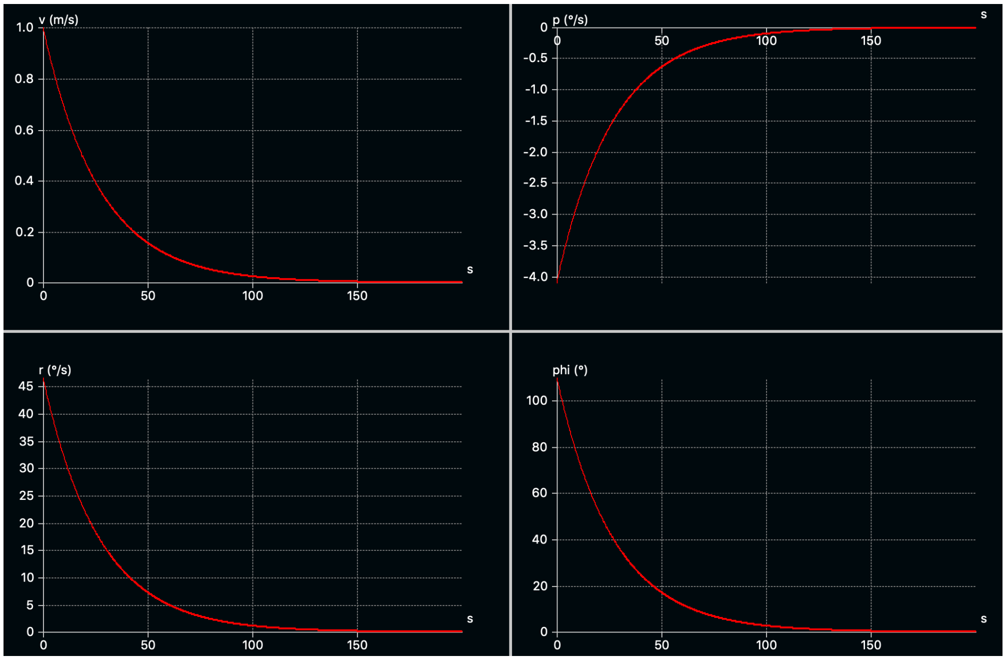


Figure 13: Spiral mode response.

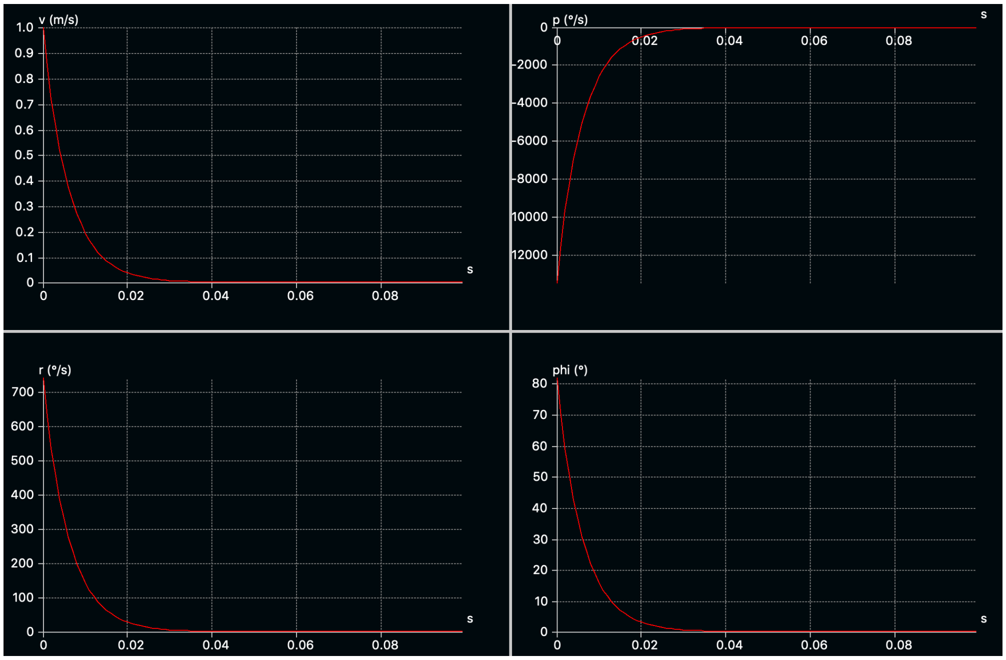


Figure 14: Roll mode response.

calculate the required moments to be

$$M_\eta = \dot{h}_2 = I_{yy} \cdot 10\frac{\pi}{180} = 0.01353\ \mathrm{N} \cdot \mathrm{m}, \tag{72}$$

$$L_\xi = \dot{h}_1 = I_{xx} \cdot 10\frac{\pi}{180} = 0.006175\ \mathrm{N} \cdot \mathrm{m}. \tag{73}$$

By setting a required $5.0°$ flap angle for both maneuvers, the other two parameters were designed iteratively using XFLR5. It was assumed that this should be sufficient to give the aircraft some level of control over the different maneuvers and enough buffer zone to vary the inputs. The parameters are specified in detail in Table 3 and shown in Fig. 15. The rudder parameters, however, were arbitrary, as there are no specific requirements from the rudder but to overcome disturbances.

Table 24: Control surfaces final design parameters.

Lifting Surface	**% chord**	**Width** (m)	**Required Angle** (°)
Aileron	85	0.72	5
Elevator	80	0.15	5

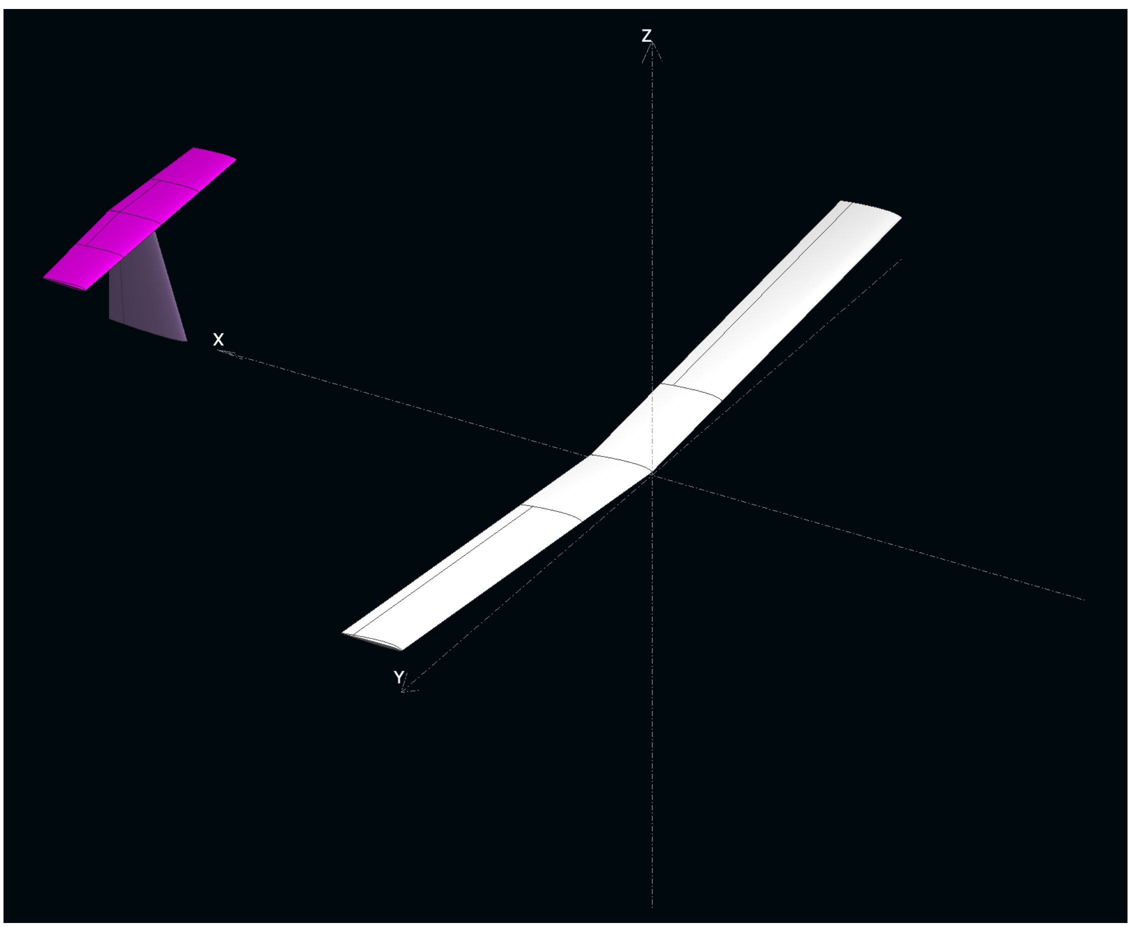


Figure 15: Final design with flaps in XFLR5.

7 Conclusion and Future Work

In this project, the aircraft had to be designed with a single objective in mind being endurance while maintaining constraints in different disciplines. The design had to be done iteratively as it was challenging to keep track of all disciplines at once. XFLR5 and XFOIL were used to run the analysis which included airfoil selection, wing and tail geometry design, and stability analysis. A 3D CAD model was built to determine the geometric characteristics of the components and to get a good estimate of the aircraft's mass and inertia tensor. By the end of the project, we got an aircraft with a decent endurance (50.4 minutes) that is stable and satisfies the geometric constraints. That being said, the aircraft can still be improved and further optimized.

Future work can be done on the symmetrical airfoils used on the tailplane, though they may not affect the performance of the aircraft. The effect of wing tweaks such as adding a wing sweep and winglets can be investigated.

Appendices

A Code - Endurance Calculations

```
function [outFlight] = calcFlight(plane,climb_states)
%calcFlight simulates the flight with a given climb states and steady
%flight distance.

% Take off
flight.takeoff = Takeoff(plane,climb_states(1));

% Climb
flight.climb = Climb(plane, climb_states);

% 180 deg Banked Turn
radius = 150;
flight.banked_turn = BankedTurn(plane,radius);

% Steady Flight
flight.steady_flight.deltaQ_SF = plane.batt.totalQ -
 plane.batt.safetyQ - flight.takeoff.deltaQ - flight.climb.deltaQ -
 flight.banked_turn.deltaQ ;
flight.steady_flight =
 SteadyFlight(plane,flight.steady_flight.deltaQ_SF);

% Descent
flight.descent = Descent(plane, plane.test.maxAltitude);

% flight metrics
flight.time = (flight.climb.time + flight.banked_turn.time +
 flight.steady_flight.time + flight.descent.time)/60;
flight.distance = (flight.climb.distance + flight.banked_turn.distance
 + flight.steady_flight.distance + flight.descent.distance)/1000;

% Output
outFlight = flight;
end
```

Published with MATLAB® R2018b

```
function [outTakeoff] = Takeoff(plane,climbV)
%This function simulates the takeoff of the airplane.

%initialization
dt = 0.01;
Simtime = 0:dt:20;
dx = zeros(1,length(Simtime));
Ax = zeros(1,length(Simtime));
v = zeros(1,length(Simtime));
power = zeros(1,length(Simtime));
Q = zeros(1,length(Simtime));
SOC = zeros(1,length(Simtime));

Q(1) = plane.batt.totalQ;
v(1) = 0.1;
SOC(1) = 100;

%forward euler
for n = 1:length(Simtime)
    if v(n) >= climbV
    takeoff.v = v(1:n);
    takeoff.SOC = SOC(1:n);
    takeoff.Ax = Ax(1:n);
    takeoff.power = power(1:n);
    takeoff.Q = Q(1:n);
    takeoff.SOC = SOC(1:n);
    takeoff.dx = dx(1:n);
    takeoff.distance = sum(dx(1:n));
    takeoff.time = dt*n;
    takeoff.deltaQ = Q(1)-(Q(n));
    break
    else
    %thrust
    CT = calcCT(plane,v(n));
    thrust = calcPropellerThrust(plane,CT);
    %drag
    drag = 0.5*plane.test.rho*plane.S*calcCd(plane,0)*(v(n)^2);
    %friction
    friction = plane.test.mu*((plane.m*9.81)-
(0.5*plane.test.rho*plane.S*calcCL(plane,0)*(v(n)^2)));
    %forward euler
    Ax(n) = (thrust-drag-friction)/plane.m;
    v(n+1) = v(n) + (Ax(n)*dt);
    dx(n) = v(n)*dt;
    %energy calcs
    power(n) = plane.m*Ax(n)*v(n)/plane.motor.eff;
```

```
end
```

Published with MATLAB® R2018b

```
function [outClimb] = Climb(plane,climb_states)
%Climb calculates climb flight

%Calcualte minimum thrust to climb and minimum CL
minCT = calcClimbCT(plane,climb_states);
minCL = (plane.m*9.81)/
(cosd(climb_states(2))*0.5*plane.test.rho*plane.S*((climb_states(1))^2));

%compare minimum CT and CL to aircraft capabilities
if minCT > max(plane.propeller.data(:,2)) || minCT <
 min(plane.propeller.data(:,2))
%     disp('Propeller cannot produce enough thrust');
    climb.flag = 1;
elseif minCL > max(plane.CL(:,2))
%     disp('Plane cannot produce enough lift');
    climb.flag = 1;
else
    %climb is possible, calculate output
    thrust = calcPropellerThrust(plane,minCT);
    power = thrust * climb_states(1) / calcEta(plane,minCT);
    time = plane.test.maxAltitude /
 (climb_states(1)*sind(climb_states(2)));
    deltaQ = power*time/(plane.motor.eff*plane.batt.Vnom);
    %output
    climb.thrust = thrust;
    climb.power = power;
    climb.time = time;
    climb.deltaQ = deltaQ;
    climb.flag = 0;
    climb.distance = climb_states(1)*cosd(climb_states(2))*climb.time;
end

outClimb = climb;
end
```

Published with MATLAB® R2018b

```
function [outSteadyFlight] = SteadyFlight(plane,deltaQ)
%SteadyFlight calculates the time and capacity used to travel a
 distance at
%20 m/s.

%speed
v = plane.test.v;

%flight calculations
sflight.CL = (plane.m*9.81)/(0.5*plane.test.rho*plane.S*(v^2));
sflight.alpha = calcAlpha(plane,sflight.CL);
sflight.CT = calcClimbCT(plane, [v,0,sflight.alpha]);
sflight.thrust = calcPropellerThrust(plane,sflight.CT);
sflight.eta = calcEta(plane,sflight.CT);
sflight.power = sflight.thrust*v/(sflight.eta*plane.motor.eff);
sflight.distanceA = 0.5*deltaQ*v*plane.batt.Vnom/sflight.power;
sflight.distanceB = 0.5*deltaQ*v*plane.batt.Vnom/sflight.power;
sflight.distance = sflight.distanceA+sflight.distanceB;
sflight.deltaQ = deltaQ;
%time
sflight.time = (sflight.distance)/v;
%output
outSteadyFlight = sflight;
end
```

Published with MATLAB® R2018b

```
function [outBakedTurn] = BankedTurn(plane,R)
%BankedTurn calculates the total flight time and energy for performing
%banked turn.

v = plane.test.v;
banked_turn.banking_angle = atand((v^2)/(9.81*R));
banked_turn.CL = (plane.m*9.81)/
(0.5*plane.test.rho*plane.S*(cosd(banked_turn.banking_angle))*v^2);
w = v/R;
%time
banked_turn.time = pi/w;
%CD & alpha
banked_turn.CD = calcCd(plane,calcAlpha(plane,banked_turn.CL));
banked_turn.alpha = calcAlpha(plane,banked_turn.CL);
%propeller
banked_turn.CT = calcClimbCT(plane, [20,0,banked_turn.alpha]);
banked_turn.thrust = calcPropellerThrust(plane,banked_turn.CT);
banked_turn.eta = calcEta(plane,banked_turn.CT);
%energy calcs
banked_turn.power = banked_turn.thrust*v/
(banked_turn.eta*plane.motor.eff);
banked_turn.deltaQ = banked_turn.power*banked_turn.time/
plane.batt.Vnom;
banked_turn.distance = v*banked_turn.time;

%output
outBakedTurn = banked_turn;

end
```

Published with MATLAB® R2018b

```
function [outDescent] = Descent(plane,altitude)
%Descent calculates the descent flight phase parameters.

v = plane.test.v;
alpha_glide = calcAlphaGliding(plane);
CL = calcCL(plane,alpha_glide);
CD = calcCd(plane,alpha_glide);

descent.glide_path_angle = acos((0.5*plane.test.rho*plane.S*CL*(v^2))/
(plane.m*9.81));

% descent.vs = v*cosd(descent.glide_path_angle);
descent.vs = (((2*plane.m*9.81)/(plane.test.rho*plane.S))^0.5)*(CD/
((CL^2)+(CD^2)));
descent.glide_path_angle = rad2deg(CD/((CL^2+CD^2)^0.5));
descent.v = descent.vs/(sind(descent.glide_path_angle));
descent.vg = cosd(descent.glide_path_angle)*descent.v;
descent.time = altitude/descent.vs;
descent.distance = descent.vg*descent.time;
%output
outDescent = descent;

end
```

Published with MATLAB® R2018b

```
function [outPlane] = genPlane(DesignIter)
%genPlane generates a plan structure with all parameters for a given
 wing
%configuration.

%test parameters
plane.test.rho = 1.225;
plane.test.mu = 0.2;
plane.test.maxAltitude = 100;
plane.test.v = 20;

%propeller papremeters
load('apce11x7.mat');
plane.propeller.data = apce11x7;
plane.propeller.D = 0.2794;         %[m]
plane.propeller.n = 100;            %6000[rev/s]
plane.propeller.staticCT = 0.11;

%motor parameters
plane.motor.eff = 0.9;
% plane.motor.n = 1;

%battery parameters
plane.batt.Vnom = 14.8;     %nominal battery voltage [V]
plane.batt.totalQ = 7920;    %total battery capacity [As]
plane.batt.safetyQ = 0.05*plane.batt.totalQ;


%aircraft parameters
if strcmp(DesignIter,'1')
    load('DesignIteration1.mat');
    %Performance
    plane.CD(:,1) = DesignIteration1(:,1);
    plane.CD(:,2) = DesignIteration1(:,6);
    plane.CL(:,1) = DesignIteration1(:,1);
    plane.CL(:,2) = DesignIteration1(:,3);
    %structure
    plane.m = 2.441;
    plane.S = 0.4068;

elseif strcmp(DesignIter,'2')
    load('DesignIteration2.mat');
    %Performance
    plane.CD(:,1) = DesignIteration2(:,1);
    plane.CD(:,2) = DesignIteration2(:,6);
    plane.CL(:,1) = DesignIteration2(:,1);
    plane.CL(:,2) = DesignIteration2(:,3);
```

```
    %Performance
    plane.CD(:,1) = DesignIteration3(:,1);
    plane.CD(:,2) = DesignIteration3(:,6);
    plane.CL(:,1) = DesignIteration3(:,1);
    plane.CL(:,2) = DesignIteration3(:,3);
    %structure
    plane.m = 2.441;
    plane.S = 0.406;

elseif strcmp(DesignIter,'4')
    load('DesignIteration3.mat');
    %Performance
    plane.CD(:,1) = DesignIteration3(:,1);
    plane.CD(:,2) = DesignIteration3(:,6);
    plane.CL(:,1) = DesignIteration3(:,1);
    plane.CL(:,2) = DesignIteration3(:,3);
    %structure
    plane.m = 1.6;
    plane.S = 0.26;

else
    disp('Please enter a valid wing configuration');
end
outPlane = plane;
end
```

Published with MATLAB® R2018b

```
function [outClimbParam,outMapEnergy] = optimizeClimbparam(plane,
 map_param)
%optimizeClimbparam finds the best operating parameters for maximising
%climb endurance.

%map climb energy to operating parameters
mapEnergy =
 zeros(length(map_param.v),length(map_param.theta),length(map_param.alpha));
for l = 1:length(map_param.v)
    for m = 1:length(map_param.theta)
        for n = 1:length(map_param.alpha)
            climb_states = [map_param.v(l), map_param.theta(m),
 map_param.alpha(n)];
            climb = Climb(plane,climb_states);
            if climb.flag == 1
                mapEnergy(l,m,n) = 1E9;
                continue
            else
            mapEnergy(l,m,n) = climb.deltaQ;
            end
        end
    end
end

%find climbing states for minimum energy
[~,I] = min(mapEnergy(:));
[I,J,K] = ind2sub(size(mapEnergy),I);
climb_v = map_param.v(I);
climb_theta = map_param.theta(J);
climb_alpha = map_param.alpha(K);

%output
outClimbParam = [climb_v;climb_theta;climb_alpha];
outMapEnergy = mapEnergy;
end
```

Published with MATLAB® R2018b

```
function [outEnergyClimb] = calcEnergyClimb(plane, V, theta, alpha,
 total_climb)
%calcEnergyClimb calculates the total energy used for the place to

thrust = (0.5*plane.test.rho*plane.S*calcCd(plane,alpha)*(V^2)) +
 (9.81*plane.m*(sind(theta)));

outEnergyClimb = thrust*total_climb / sind(theta);

end
```

Published with MATLAB® R2018b

```
function [outAlpha] = calcAlphaGliding(plane)
%This function calculates the optimal angle of attack for gliding.

%calculate CD/CL^1.5
glide_metric_initial = plane.CD(:,2)./(abs(plane.CL(:,2)).^1.5);
m = 1;
for n = 1:length(glide_metric_initial)
    if plane.CL(n) < 0
        continue
    else
        glide_metric(m,1) = plane.CL(n,1);
        glide_metric(m,2) = glide_metric_initial(n);
        m = m+1;
    end
end

%fit data
p = polyfit(glide_metric(:,1),glide_metric(:,2),8);
x1 = linspace(glide_metric(1,1), glide_metric(m-1,1),150);
y1 = polyval(p,x1);

%plot fit
% figure
% plot(glide_metric(:,1),glide_metric(:,2),'o')
% hold on
% plot(x1,y1)

%find alpha to minimize metric
outAlpha = interp1(y1,x1,min(y1));
end
```

Published with MATLAB® R2018b

```
function [outAlpha] = calcAlpha(plane,CL)
%calcCD calculates the Coefficient of drag for the airplane at a given
%angle of attack.

outAlpha = interp1(plane.CL(:,2),plane.CL(:,1),CL);

end
```

Published with MATLAB® R2018b

```
function [outCL] = calcCL(plane,alpha)
%calcCL calculates the Coefficient of lift for the airplane at a given
%angle of attack.

outCL = interp1(plane.CL(:,1),plane.CL(:,2),alpha);

end
```

Published with MATLAB® R2018b

```
function [outCT] = calcClimbCT(plane, states)
%calcCT calculates the propeller coefficient of thrust required at
 steady state for
%a given speed, climb angle and angle of attack.

drag =
 0.5*plane.test.rho*plane.S*calcCd(plane,states(3))*(states(1)^2);

outCT = (drag + (plane.m*9.81*sind(states(2))))/
((plane.propeller.n^2)*plane.test.rho*(plane.propeller.D^4));

end
```

Published with MATLAB® R2018b

```
function [outCT] = calcCT(plane,v)
%This function calculates the propeller thrust coefficient for a given
%speed at 6000 rpm.

J = v / (plane.propeller.n*plane.propeller.D);

if J < min(plane.propeller.data(:,1))
    outCT = plane.propeller.staticCT;
else
    outCT =
 interp1(plane.propeller.data(:,1),plane.propeller.data(:,2),J);
end

end
```

Published with MATLAB® R2018b

```
function [outEta] = calcEta(plane,CT)
%Calculate propeller efficiency based on propeller Coeffient of thrust

if CT > max(plane.propeller.data(:,2)) || CT <
 min(plane.propeller.data(:,2))
    disp('CT outside of range');
    return
else
    outEta =
 interp1(plane.propeller.data(:,2),plane.propeller.data(:,4),CT);
end
end
```

Published with MATLAB® R2018b

```
function [outThrust] = calcPropellerThrust(plane,CT)
%calc Propeller Thrust calculates the thrust produced by the propeller
%based on CT and V.

outThrust =
 (plane.propeller.n^2)*plane.test.rho*(plane.propeller.D^4)*CT;

end
```

Published with MATLAB® R2018b

B Code - Airfoil Analysis

```
% MECH 532 Project

% This script is for building the airfoil database, using XFOIL
 results.

% Author: Felix C.Lamy

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
clc
clear all

% Test Structure
% test.airfoil = ["naca21015","NACA21016"] ;
airfoil_set = load('airfoilnameset1.mat');
test.airfoil = airfoil_set.str_all ;
test.reynolds_number = 250000;
test.mach = 0.058;
test.alpha = (-5:0.2:20);

% sweep through airfoil geometries

%loop through airfoils

for a = 122:length(test.airfoil)
    afoil_geo = char(test.airfoil(a));
    %loop through Reynolds numbers
    reynold_number = test.reynolds_number;

    %run xfoil
    [pol foil] =
 xfoil(afoil_geo,test.alpha,reynold_number,test.mach,'ppar n
 100','oper iter 600');

    %convert to character array
    Re = sprintf('Re%d', reynold_number);


    %generate database
    data.(afoil_geo).(Re).foil = foil;
    data.(afoil_geo).(Re).pol = pol;
end
```

Published with MATLAB® R2018b

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% MECH 532 Project Airfoil Analysis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
clc
clear all


% load data

load('airfoil_data.mat');

% calculate airfoil metrics

[airfoil_list, airfoil_metrics] = select_airfoils(data, 0.24, 0.26,
 10);

airfoil_metrics(airfoil_metrics>5) = 0;

% Airfoil Performance Subplot


font_size = 15;
line_size = 15;
line_width = 1.2;

airfoilA = char(airfoil_list(1));
airfoilB = char(airfoil_list(2));
airfoilC = char(airfoil_list(3));
airfoilD = char(airfoil_list(4));
airfoilE = char(airfoil_list(5));

% Plot CL vs Alpha
subplot(2,2,1)
plot(data.(airfoilA).Re250000.pol.alpha, data.
(airfoilA).Re250000.pol.CL, ...
    data.(airfoilB).Re250000.pol.alpha, data.
(airfoilB).Re250000.pol.CL, ...
    data.(airfoilC).Re250000.pol.alpha, data.
(airfoilC).Re250000.pol.CL, ...
    data.(airfoilD).Re250000.pol.alpha, data.
(airfoilD).Re250000.pol.CL, ...
    data.(airfoilE).Re250000.pol.alpha, data.
(airfoilE).Re250000.pol.CL,'Linewidth',line_width);
grid on;
```

```
% Plot Efficiency Metric vs Alpha
subplot(2,2,2);
plot(data.(airfoilA).Re250000.pol.alpha,
 data.(airfoilA).Re250000.pol.CD./(abs(data.
(airfoilA).Re250000.pol.CL).^1.5), ...
    data.(airfoilB).Re250000.pol.alpha,
 data.(airfoilB).Re250000.pol.CD./(abs(data.
(airfoilB).Re250000.pol.CL).^1.5), ...
    data.(airfoilC).Re250000.pol.alpha,
 data.(airfoilC).Re250000.pol.CD./(abs(data.
(airfoilC).Re250000.pol.CL).^1.5), ...
    data.(airfoilD).Re250000.pol.alpha,
 data.(airfoilD).Re250000.pol.CD./(abs(data.
(airfoilD).Re250000.pol.CL).^1.5), ...
    data.(airfoilE).Re250000.pol.alpha,
 data.(airfoilE).Re250000.pol.CD./(abs(data.
(airfoilE).Re250000.pol.CL).^1.5),'Linewidth',line_width);
grid on;
xlabel('Angle of Attack
 [deg]','fontsize',font_size,'Interpreter','latex');
ylabel('${\it} Cl/
(Cd^{3/2})$','fontsize',font_size,'Interpreter','latex');
ylim([0 0.1]);
title('Steady Flight
 Metric','fontsize',font_size,'Interpreter','latex');
legend(airfoilA,airfoilB,airfoilC,airfoilD,airfoilE,'Interpreter','latex');
set(gca,'XMinorGrid','off','GridLineStyle','-','FontSize',line_size)


% Plot CL/CD vs Alpha
subplot(2,2,3);
plot(data.(airfoilA).Re250000.pol.alpha, data.
(airfoilA).Re250000.pol.CL./data.(airfoilA).Re250000.pol.CD, ...
    data.(airfoilB).Re250000.pol.alpha, data.
(airfoilB).Re250000.pol.CL./data.(airfoilB).Re250000.pol.CD, ...
    data.(airfoilC).Re250000.pol.alpha, data.
(airfoilC).Re250000.pol.CL./data.(airfoilC).Re250000.pol.CD, ...
    data.(airfoilD).Re250000.pol.alpha, data.
(airfoilD).Re250000.pol.CL./data.(airfoilD).Re250000.pol.CD, ...
    data.(airfoilE).Re250000.pol.alpha,
 data.(airfoilE).Re250000.pol.CL./data.
(airfoilE).Re250000.pol.CD,'Linewidth',line_width);
grid on;
xlabel('Angle of Attack
 [deg]','fontsize',font_size,'Interpreter','latex');
ylabel('L/D','fontsize',font_size,'Interpreter','latex');
```

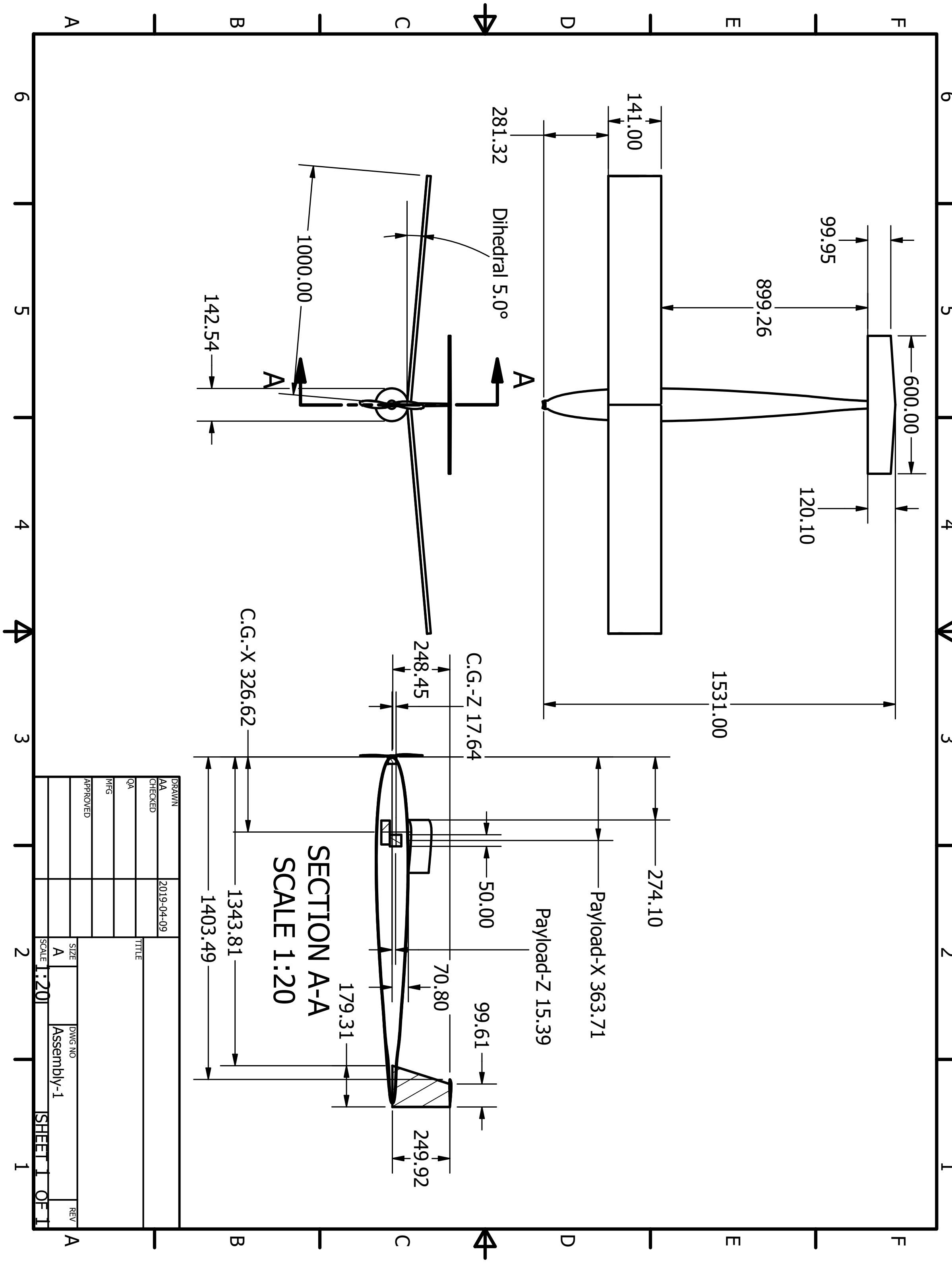
```
plot(data.(airfoilA).Re250000.pol.alpha, data.
(airfoilA).Re250000.pol.Cm, ...
    data.(airfoilB).Re250000.pol.alpha, data.
(airfoilB).Re250000.pol.Cm, ...
    data.(airfoilC).Re250000.pol.alpha, data.
(airfoilC).Re250000.pol.Cm, ...
    data.(airfoilD).Re250000.pol.alpha, data.
(airfoilD).Re250000.pol.Cm, ...
    data.(airfoilE).Re250000.pol.alpha, data.
(airfoilE).Re250000.pol.Cm,'Linewidth',line_width);
grid on;
xlabel('Angle of Attack
 [deg]','fontsize',font_size,'Interpreter','latex');
ylabel('Cm','fontsize',font_size,'Interpreter','latex');
title('Cm','fontsize',font_size,'Interpreter','latex');
set(gcf, 'Position', [70, 80, 1000, 650])
set(gca,'XMinorGrid','off','GridLineStyle','-','FontSize',line_size)

figure();
plot(data.(airfoilA).Re250000.foil.x(1:100,1),data.
(airfoilA).Re250000.foil.y(1:100,1),...
    data.(airfoilB).Re250000.foil.x(1:100,1),data.
(airfoilB).Re250000.foil.y(1:100,1),...
    data.(airfoilC).Re250000.foil.x(1:100,1),data.
(airfoilC).Re250000.foil.y(1:100,1),...
    data.(airfoilD).Re250000.foil.x(1:100,1),data.
(airfoilD).Re250000.foil.y(1:100,1),...
    data.(airfoilE).Re250000.foil.x(1:100,1),data.
(airfoilE).Re250000.foil.y(1:100,1),'--b','Linewidth',line_width)
legend(airfoilA,airfoilB,airfoilC,airfoilD,airfoilE,'Interpreter','latex');
xlabel('x coordinates','fontsize',font_size,'Interpreter','latex');
ylabel('y coordinates','fontsize',font_size,'Interpreter','latex');
title('Airfoil
 Geometries','fontsize',font_size,'Interpreter','latex');
grid on
axis equal


% Airfoil Performance Subplot - single Airfoil

airfoilA = char(airfoil_list(5));

% Plot CL vs Alpha
subplot(2,2,1)
plot(data.(airfoilA).Re250000.pol.alpha, data.
(airfoilA).Re250000.pol.CL,'Linewidth', 1.2);
grid on;
xlabel('Angle of Attack [deg]');
```

```
plot(data.(airfoilA).Re250000.pol.alpha,
 data.(airfoilA).Re250000.pol.CD./(real(data.
(airfoilA).Re250000.pol.CL.^1.5)),'Linewidth', 1.2);
grid on;
xlabel('Angle of Attack [deg]');
ylabel('CD/CL^(3/2)');
% ylim([0 0.05]);
title('Steady Flight Metric vs alpha');


% Plot CL/CD vs Alpha
subplot(2,2,3);
plot(data.(airfoilA).Re250000.pol.alpha,
 data.(airfoilA).Re250000.pol.CL./data.
(airfoilA).Re250000.pol.CD,'Linewidth', 1.2);
grid on;
xlabel('Angle of Attack [deg]');
ylabel('L/D');
title('Cl/CD vs alpha');
ylim([-50 90]);

% Plot CM vs alpha
subplot(2,2,4);
plot(data.(airfoilA).Re250000.pol.alpha, data.
(airfoilA).Re250000.pol.Cm,'Linewidth', 1.2);
grid on;
xlabel('Angle of Attack [deg]');
ylabel('Cm');
title('Moment coefficient vs Angle of Attack');
set(gcf, 'Position', [70, 80, 1000, 650])



% Plot airfoil

figure();
plot(data.(airfoilA).Re250000.foil.x(1:100,1),data.
(airfoilA).Re250000.foil.y(1:100,1),...
    data.(airfoilD).Re250000.foil.x(1:100,1),data.
(airfoilD).Re250000.foil.y(1:100,1),'--b','Linewidth',line_width)
legend(airfoilA,airfoilD,'Interpreter','latex');
xlabel('x coordinates','fontsize',font_size,'Interpreter','latex');
ylabel('y coordinates','fontsize',font_size,'Interpreter','latex');
title('Airfoil
 Geometries','fontsize',font_size,'Interpreter','latex');
grid on
axis equal
```

```
function [outList, outAFMetrics] = select_airfoils(data, minCL, maxCL,
 k)
%This function returns teh k best airfoils for the endurance metric
 between
%a minimum and a maximum CL.

list_airfoils = fieldnames(data);
afmetrics = zeros(length(list_airfoils),2);
selected_airfoils = zeros(10,2);

for m = 1:length(list_airfoils)
    %set airfoil geometry
    afoil_geo = char(list_airfoils(m));
    %find angles of attack where lift condition is satisfied
    indicesCL = find((data.(afoil_geo).Re250000.pol.CL > minCL) &
 (data.(afoil_geo).Re250000.pol.CL < maxCL));
    %calculate mean efficiency over that range of angle of attacks
    meanEff = mean(data.(afoil_geo).Re250000.pol.CL(indicesCL)./data.
(afoil_geo).Re250000.pol.CD(indicesCL));
    %calculate mean steady flight efficiency metric
    meanSFEff = mean(data.(afoil_geo).Re250000.pol.CD(indicesCL)./
(data.(afoil_geo).Re250000.pol.CL(indicesCL).^(1.5)));
    if meanSFEff < 0.01
        meanSFEff = 10;
    end
    %store metrics
    afmetrics(m,1) = meanEff;
    afmetrics(m,2) = meanSFEff;
end

[~,I] = mink(afmetrics(:,2),k);
selected_airfoils = (list_airfoils(I,1));

outAFMetrics = afmetrics;
outList = selected_airfoils;
end
```

Published with MATLAB® R2018b

C Aircraft Dimensions



References

[1] M. McCrink and J. W. Gregory, "Blade Element Momentum Modeling of Low- *Re* Small UAS Electric Propulsion Systems," in *33rd AIAA Applied Aerodynamics Conference*. Dallas, TX: American Institute of Aeronautics and Astronautics, Jun. 2015. [Online]. Available: http://arc.aiaa.org/doi/10.2514/6.2015-3296

[2] M. H. McCrink and J. W. Gregory, "Blade Element Momentum Modeling of Low-Reynolds Electric Propulsion Systems," *Journal of Aircraft*, vol. 54, no. 1, pp. 163–176, Jan. 2017. [Online]. Available: http://arc.aiaa.org/doi/10.2514/1.C033622

[3] J. B. Russell, *Performance and stability of aircraft*. London: Arnold, 1996, oCLC: 35878030.

[4] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton, NJ, USA: Princeton University Press, 2012.