INTERNATIONAL UNIVERSITY LIAISON INDONESIA

BACHELOR'S THESIS

ANALYSIS OF RANGE AND ENDURANCE OF N-219 UNDER POTENTIAL ELECTRIC PROPULSION

By

Maheswara Sinatriyo

11201601010

Presented to the Faculty of Engineering

In Partial Fulfilment Of the Requirements for the Degree of

SARJANA TEKNIK

In

AVIATION ENGINEERING

FACULTY OF ENGINEERING

BSD City 15345

Indonesia

APPROVAL PAGE

ANALYSIS OF RANGE AND ENDURANCE OF N-219 UNDER POTENTIAL ELECTRIC PROPULSION

MAHESWARA SINATRIYO
11201601010
Presented to the Faculty of Engineering
In Partial Fulfillment of the Requirements for the Degree of

SARJANA TEKNIK
In
AVIATION ENGINEERING

FACULTY OF ENGINEERING

Triwanto Simanjuntak, Ph.D.	__________
Thesis Advisor I	Date
Dr. Eng. Ressa Octavianty	__________
Thesis Advisor II	Date
Tutun Nugraha, BASc, MASc, Ph.D.	__________
Vice Rector of Academic Affairs	Date

EXAMINERS APPROVAL PAGE

Dr. Ir. Erie Sandhita MsAe, DEA

Examiner 1 | Date

Dr. Eng. Ressa Octavianty

Examiner 2 | Date

Triwanto Simanjuntak, Ph.D.

Examiner 3 | Date

STATEMENT BY THE AUTHOR

I, hereby, declare that this submission is my own work and to the best of my knowledge, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at any educational institution, except where due acknowledgement is made in the thesis.

Maheswara Sinatriyo	
Student	Date

ABSTRACT

ANALYSIS OF RANGE AND ENDURANCE OF N-219 UNDER POTENTIAL ELECTRIC PROPULSION

by

Maheswara Sinatriyo

Triwanto Simanjuntak, Ph.D., Advisor

As carbon footprint continues to grow, electrical vehicle holds a high value for preventing further damages to the environment. Over the past few years EV has grown rapidly, even to the point where small fixed-winged propeller-driven aircraft were able to be operated under full electric system. But aircraft energy requirement increases exponentially by its total weight. Thus, this research tackle the ability of a 19 seater commuter aircraft to perform under full electric system modification, in which, the N-219 aircraft were made as the base model for this research for its flexibility. Therefore, in order to measure the capability of an all-electric N-219, a comparison between the maximum range and endurance performance of both aircraft is necessary. The airbreathing performance is calculated using Breguet formula for maximum range and endurance. A modification for the Breguet formula is necessarry to accommodate the Peukert's effect of a battery (Traub, 2011). To further enhance the analysis, the calculation were made using different scenario such as different payload, fuel/battery capacity, altitude, and airspeed. The result show that, an electric 19 seater aircraft is still very limited in terms of usability, since it can only achieve less than 300 km of range under half payload with current battery technology.

Keyword: *electric aircraft, electric vehicle, Peukert's effect*

ACKNOWLEDGEMENTS

The journey throughout this thesis would not be possible without a massive amount of support and assistance. First and foremost I would like to express my gratitude to god almighty Allah SWT. for all the blessing and opportunity that has been bestowed.

Many thanks to my Advisors Triwanto Simanjuntak, Ph.D., and Dr.Eng. Ressa Octavianty, whose knowledge and experience were crucial in formulating the research questions and methodology. Your informative criticism encouraged me to improve my thoughts and raise the quality of my work.

I would like to acknowledge the employee of Indonesian Aerospace who have helped me obtain vital data for this thesis. Without them it would not be possible to finish this thesis with accurate result.

In addition, I would like to thank my family for support and patience. Finally, to many of my friends, who have given me insightful perspective as well as helped me ease my mind outside of this research.

Contents

List of Figures

List of Tables

List of Abbreviations

EV	**E**lectric **V**ehicle
C0$_2$	**C**arbon di**O**xide
IATA	**I**nternational **A**ir **T**ransport **A**ssociation
MTOW	**M**aximum **T**ake **O**ff **W**eight
STOL	**S**ort **T**ake **O**ff **L**anding
VTOL	**V**ertical **T**ake **O**ff **L**anding
CG	**C**enter of **G**ravity
RPM	**R**otation **P**er **M**inute
IPCC	**I**ntergovernmental **P**anel on **C**limate **C**hange
Li-Ion	**Li**thium **Ion**
Li-Po	**Li**thium **Po**lymer
LiS	**Li**thium **S**ulfur
SAR	**S**earch **A**nd **R**escue
SHP	**S**haft **H**orse **P**ower
BHP	**B**rake **H**orse **P**ower
SFC	**S**pecific **F**uel **C**oefficient
FF	**F**uel **F**low Rate
ISA	**I**nternational **S**tandard **A**tmosphere

List of Symbols

p	Air pressure
T_M	Ambiance temperature
ρ	Air density
g	Acceleration of gravity
μ	Dynamic viscosity
p_0	Sea-level air pressure
T_{M0}	Sea-level ambiance temperature
ρ_0	Sea-level air density
g_0	Sea-level acceleration of gravity
γ	Air specifics heats ratio
T_{M1}	Temperature at H_1
H_1	Geopotential height
h	Geometrical height
m	Gas molecular mass
T	Absolute temperature
λ	temperature gradient
R_e	Reynolds number
β	Constant of $1.458 \times 10^{-6} kgs^{-1}m^{-1}K^{-1/2}$
S_C	Sutherland's constant
ΔT	Temperature deviation
R_a	Specific gas constant
R_u	Universal gas constant

$\vec{F}$	sum of all resultant external forces applied to the body of the aircraft
$\vec{V}$	linear velocity vector of the body's center of gravity relative to an inertial reference frame
M	Aircraft's mass
c_s	Speed of sound
D	Drag
T	Thrust
T_{static}	Static thrust
W	Aircraft's weight
V	Velocity/airspeed
F_A	Force dimension/Aerodynamic Force
C_R	Flight curvature radius
t	Time
M_o	Moment
M_{CG}	Moment at center of gravity
C_M	Coefficient of the nondimensional moment
$\bar{c}$	Factor of the length
c	Chord length
c_l	Airfoil lift coefficient
c_d	Airfoil drag coefficient
c_{mg}	Geometric chord
A	Aspect ratio
b	Wing span
S	Wing area
ϵ	Geometric twist
Λ	Sweep angle
Γ	Dihedral angle

C_L	Finite wing lift coefficient
C_{Lmax}	Maximum lift coefficient
C_D	Finite wing drag coefficient
C_{Dp}	Profile drag coefficient
C_{Di}	Induced drag coefficient
C_{D0}	Zero-lift drag coefficient
M_a	Mach number
μ	Angle of mach
M_{crit}	Critical mach number
D_n	Component drag
D_w	Wing drag
D_{min}	Drag minimum
Φ	Efficiency factor of the wing
α_{eff}	Efficiency
α_i	Induce angle of attack
V_∞	Freestream velocity
α	Angle of attack
θ	Angle of pitch
γ	Flight-path angle
RC	Rate of climb
RC_{max}	Maximum rate of climb
P_a	Power available
P_r	Power required
P_{rmin}	Minimum power required
P_c	Excess power
V_{Dmin}	Airspeed at minimum drag
V_{Pmin}	Airspeed at minimum power
k	Drag factor

η_j	Propulsive efficiency
η_p	Propeller efficiency
η_{tot}	Total efficiency
P_{br}	shaft horse power
C_P	Power coefficient
C_T	Thrust coefficient
$(C_L/C_D)_{max}$	Maximum lift drag coefficient
C_{LC}	Lift coefficient for maximum rate of climb
V_{max}	Maximum airspeed
W_f	Fuel weight
F	Fuel flow rate
R	Range
R_{max}	Maximum range
E	Endurance
E_{max}	Maximum endurance
c_p	Specific fuel consumption
t	Time/endurance
Rt	Battery hour rating
P_B	Power battery
V_o	Voltage
i	Current
n	Peukert's coefficient
C	Battery capacity

CHAPTER 1
INTRODUCTION

1.1 Background

Over the past few decades, aircraft has been a necessity and a vital component for the human race. It allows humankind to travel around the world quickly and with ease. Compared to other modes of transportation, an aircraft can travel further in a shorter amount of time. As new destinations open every year, more and more people are willing to travel using aircraft over time. Thus, the airline industries will keep on growing and producing and operating more aircraft, and as with most other transportation modes, aircraft uses fossil fuel, which is finite and unsustainable. Thus the growing aircraft industries come with a cost, an increase of harmful carbon emission, which is one of the causes of climate change.

1.1.1 Climate change

There are many things that cause climate change, and one of them is carbon dioxide (CO_2). Natural processes like respiration and volcanic eruption emit carbon dioxide, as do human activities such as deforestation, urban development, and fossil fuel combustion. Since the Industrial Revolution, humanity has boosted CO_2 levels in the atmosphere by 47%. (Jackson, 2021a)

We can already see the effect of global climate change on the environment. Things such as glaciers have receded, ice on rivers and lakes has broken up earlier, plant and animal ranges have altered, and trees have begun to bloom earlier. Scientists are convinced that global temperature will continue to increase over the next decades due to greenhouse gas emissions caused by human activities. A prediction that over the next century, the temperature will rise from 2.5 up to 10 degrees Fahrenheit was made by over 1,300 experts from the United States and other nations that make up the Intergovernmental Panel on Climate Change (IPCC). (Jackson, 2021b)

Even though the number of aircraft in the world is not that much compared to other vehicles, globally, commercial aviation accounts for around 2% to 3% of global carbon emissions (*Working Towards Ambitious Targets*, n.d.). It is heartbreaking to see that such an essential technology for humanity brings harm to the environment. But what if there is a way to reduce the damage? If we look at automotive industries, they are now racing on developing sustainable vehicles using renewable energy such as electricity and hydrogen. As a result, over the past few years, we have seen an increase in sustainable automotive around the globe, so much so that there is even the of Formula E.

1.1.2 Green technology

If greener technologies such as electric vehicles are implemented into aircraft, we would definitely see a reduction of global carbon emission from commercial aviation. Electric propulsion has several benefits over airbreathing propulsion; mainly, an aircraft can reduce carbon emission directly and indirectly by using electric propulsion. Electric propulsion can also have significantly less noise (depending on the design), which opens up the possibility of flying at noise ordinance time without disturbing the area, and also a more comfortable flight (Schäfer et al., 2019). Furthermore, electric propulsion design is very simple, thus could potentially reduce maintenance costs. Also, cost-wise using an electric motor as a means for aircraft propulsion could reduce the operating cost of an aircraft. Schäfer and colleagues also analyzed how the emission produced from electricity generation compares to direct jet engine combustion for a narrow-body aircraft with 740 km mission length, see Figure 1.1. Furthermore in the same analysis, a breakeven comparison of electricity price and jet fuel price is elaborated, see Figure 1.2.

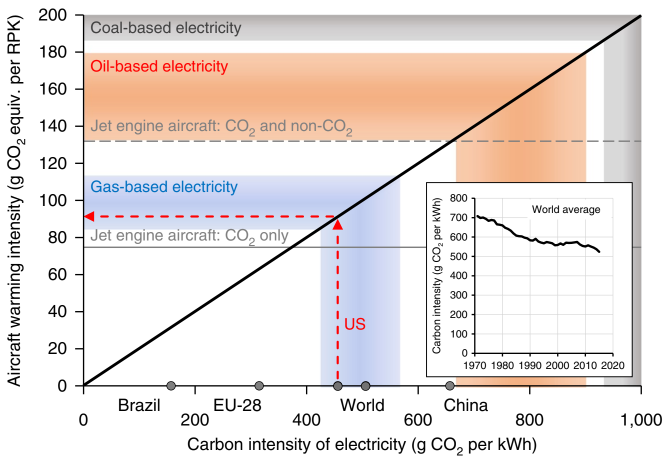


FIGURE 1.1: Warming intensity of a projected first-generation all-electric aircraft and of a current-generation jet engine aircraft versus carbon intensity of electricity. (Schäfer et al., 2019).

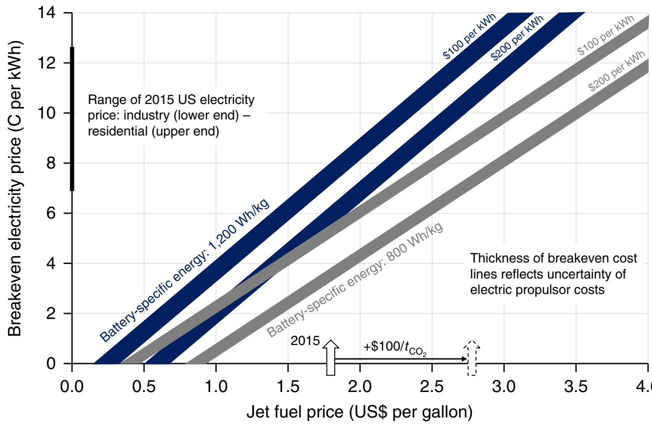


FIGURE 1.2: Break-even electricity price for a first-generation all-electric aircraft. (Schäfer et al., 2019).

But why are there not many or any electric aircraft in the world if it could benefit the environment? Well, the answer lies in the battery; batteries are very heavy compared to fossil fuel. Unlike a car which can benefit from extra weight for grip and traction, an aircraft needs to be as light as possible to fly efficiently. Thus, when can we expect the technology to be implemented into aircraft? A few years? Decades? or maybe it is actually possible right now? Yes, it is possible right now, with limitations, of course.

1.1.3 Electric aircrafts

Actually, electric aircraft have been around for a while, but those electric aircraft are usually designed for a 1 to 2 seater and hobbyist aircraft due to battery technology limitations. Only recently a company named Eviation has been developing a nine-seater all-electric aircraft called Alice. The aircraft can fly up to 440 nmi (815 km) with a cruising speed of 220 kts (113 m/s). This is impressive for an all-electric aircraft, considering the energy density of a battery is roughly a tenth of fossil fuel. Furthermore, operating an Alice only costs 200$ per hour compared to 1200-2000 operating cost of similarly performing turboprop aircraft (Narishkin, 2020). This low operating cost is like opening the door to a new world because the airline industries have always had a low margin of profit, around 8% (Pearce, 2017). Moreover, the volatile price of oil does not help the operating cost of an airline. Therefore, with an all-electric aircraft, tickets price could be lower while maintaining or even an increase in the overall profitability of an airline.

But the thing about Eviation Alice is that the aircraft is designed from the ground specifically for an all-electric aircraft. Then what about other aircraft that are already operating in the world? There is a solution for that, retrofit. Retrofitting an aircraft from a fossil fuel-based system into a fully electric system could be one of the several solutions for greener flights. magniX, an electric motor

FIGURE 1.3: Eviation Alice (Alcock, 2020).

company that designs electric motor for aircraft, has been retrofitting aircraft since 2019. The company has retrofitted a De Havilland Beaver seaplane that Harbour Air operates and a 208B Cessna Grand Caravan into an all-electric aircraft. The Cessna B208 is a nine-seater single-engine aircraft, which initially utilizes Pratt & Whitney Canada PT6A-114A with 497kW power and can fly as far as 1,982 km. The retrofitted B208, also known as the e-Caravan, operates using magniX magni500, a 560kW electric engine. magniX claimed that the e-Caravan could carry 4-5 passengers up to 160 km with reserve power using 2019 battery technology. Furthermore, the e-Caravan claimed to only use 6$ USD worth of electricity for a 30 minutes test flight (Garrett-Glaser, 2020).

Schäfer and colleagues published a first-order study of an all-electric aircraft's potential to decrease overall environmental consequences in Nature Energy (Schäfer et al., 2019). They explained that if a battery technology that is sufficiently advanced is available, an all-electric regional aircraft that can reach up to 1100 km has a tremendous potential to minimize negative environmental effects, be cost-competitive, and have superior noise performance. Based on Schäfer analysis, Viswanathan and Knapp visualized Figure 1.5. Furthermore, this figure indicates

FIGURE 1.4: Cessna B208 eCaravan by magniX and AeroTEC (Calderwood, 2020).

that smaller aircraft such as Cessna B208 would benefit more from electrification rather than bigger aircraft such as Airbus A320 or Boeing 737. Figure 1.5 indicates at what specific energy can a battery sufficiently supply power to operate different types of aircraft. Thus a battery specific energy that allows a Cessna B208 e-Caravan to operate might not be even able to provide enough energy for a regional aircraft even if the battery's volume/mass were to scale up. Schäfer also explained that a narrow-body aircraft with more than 100 passenger capacity might rely on a much-advanced technology such as lightweight high-temperature superconducting electric motors for better cooling and higher power-to-weight ratio than standard electric motors.a

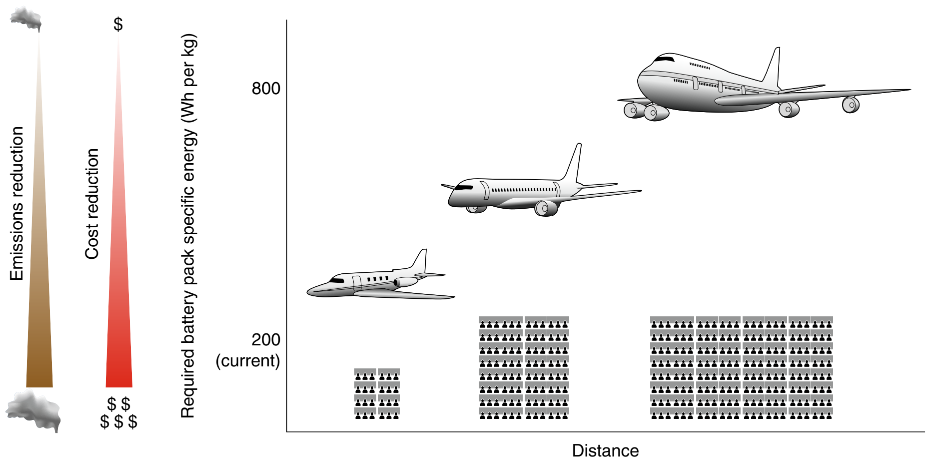


FIGURE 1.5: Electric aircraft scaling (Viswanathan & Knapp, 2019).

1.1.4 Batteries advancement

The success of Eviation and magniX shows that battery technology is mature enough to be used for electric aircraft. Furthermore, with all the development of electric vehicles, humanity is exposed to the benefits of greener technologies; thus, more and more people are open to buying EVs and greener vehicles. This indirectly opens up a market for EVs and, in turn, demands better batteries; thus, more companies are willing to spend more into researching and developing a better battery for commercial use.

Nowadays, Lithium-Ion is probably the most widely used battery type. Lithium-Ion batteries can be found almost everywhere, from electronic devices such as laptops, phones, e-cigarettes to EVs. This is because Lithium Ion has the highest volumetric and gravimetric energy densities for commercial batteries at 700 Wh/L and 265 Wh/kg. Schäfer and colleagues calculated the battery's exact energy demands considering weight is a serious challenge for air travel. Thus if we compare current battery technology with Figure 1.5 we can see that we are limited to smaller aircraft; Furthermore, to achieve full electrification for an aircraft such as Airbus

A320 or Boeing 737 that can do at least 1100 km mission, we need a battery specific energy that is up to 5 times higher than current battery packs (Schäfer et al., 2019). However, there is actually another battery type that had almost double the specific energy of Lithium Ion, which is the Lithium-Sulfur battery. Lithium-Sulfur has 1000 Wh/L and 500 Wh/kg of volumetric and gravimetric energy density, but because the technology is still very new, there are not many applications yet. See Table 3.8 for battery specifications.

1.1.5 N-219

If we look at the previous example of electric aircraft, we found that those aircraft are for nine seaters and lower. Furthermore, an Article by Viswanathan and Knapp and an Analysis by Schäfer and colleagues explained that it is impossible for narrow-body aircraft to be electrified using current battery technology. Thus, we want to see the effect of aircraft scaling for an aircraft that is a niche bigger than both Eviation Alice and the Cessna e-Caravan.

The Indonesian Aerospace's N-219 is the perfect candidate for this task, and it is a 19 seater multi-purpose aircraft. N-219 was created to increase people's economic prosperity and maintain defense and security in remote places. The N-219 Nurtanio is the latest-generation multi-purpose aircraft with the biggest cabin cross-section in its class, a proven and economic engine, sophisticated avionics system, fixed tricycle landing gear, and a broad cargo door to make altering aircraft configurations easier. As a result, N-219 Nurtanio is intended to give operators with economical and technical advantages. Aside from passenger transport, the N-219 Nurtanio can be configured to handle a variety of mission requirements, including medevac, and search and rescue (SAR), troop transport, reconnaissance, cargo transportation (Aerospace, n.d.-a).

The N-219 aircraft has a certification for basic CASR 23, commuter category.

FIGURE 1.6: N-219 (Meszaros, 2019).

The powerplant for this aircraft is Two Pratt & Whitney Canada PT6A-42, and attached to it is the Hartzell 4-Blade Metal Propeller. The engine has a 625 kW (850 shp) power output @2000 rpm while only weighing around 159 kg (350 lbs). N-219 has a high wing configuration, and this helps with engine clearance from the ground because N-219 is designed to be able to take off and land on an unpaved runway. This also includes short Take-Off and Landing (STOL) operation, with a Take-Off distance of 435 m (1230 ft) and Landing distance of 509 m (1670 ft), Figure 1.7. This shows the capabilities of N-219 for reaching and traversing remote areas (*N219 SOLUTION FOR FRONTIER GATEWAY*, 2018).

The N-219 are able to achieve a maximum cruising speed of 210 kts (108 m/s) with an economical cruise speed of 170 kts (87.5 m/s). Figure 1.8 shows the range achievable by N-219 depending on its load, and as can be seen, the N-219 can reach 480 nmi with a 19 passenger load, and if the payload were reduced, it could even reach up to 828 nmi, almost doubling the range of the maximum passenger configuration. This also shows that the N-219 can reach further if operated within its 170 kts economical cruise speed.

Since N-219 is designed to be flexible and versatile, it is equipped with a quick

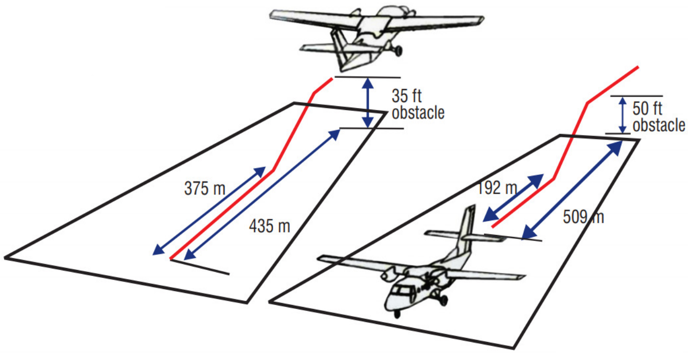


Figure 1.7: N-219 Take-Off and Landing illustration (*N219 SOLUTION FOR FRONTIER GATEWAY*, 2018).

change configuration. For example, several of the N-219 cabin configurations include foldable seats; this design makes it easier to change the aircraft configuration. That is why the N-219 also provides a wide cargo door that can load D2 containers. Furthermore, to add to the already flexible characteristic of N-219, Indonesian Aerospace planned on making an amphibian version of the N-219. This plan was proposed to increase the reachability of the N-219 aircraft further. With an amphibian version of N-219, the aircraft could reach even the remotest areas, such as a small island that has a lot of hills thus cannot build any runways and only reachable using water (*Pesawat N219 Melengkapi Jam Terbang Sebelum Diproduksi Massal: Ekonomi*, 2019).

1.1.6 Aircraft performance

The capacity of an aircraft to do particular tasks that make it useful for specific reasons is referred to as performance. For example, it is critical for pilots that fly in and out of short, unimproved airfields to know and understand the performance and limitations of an aircraft take-off and landing capabilities. There are many things that are categorized as a performance of an aircraft and more that affect it.

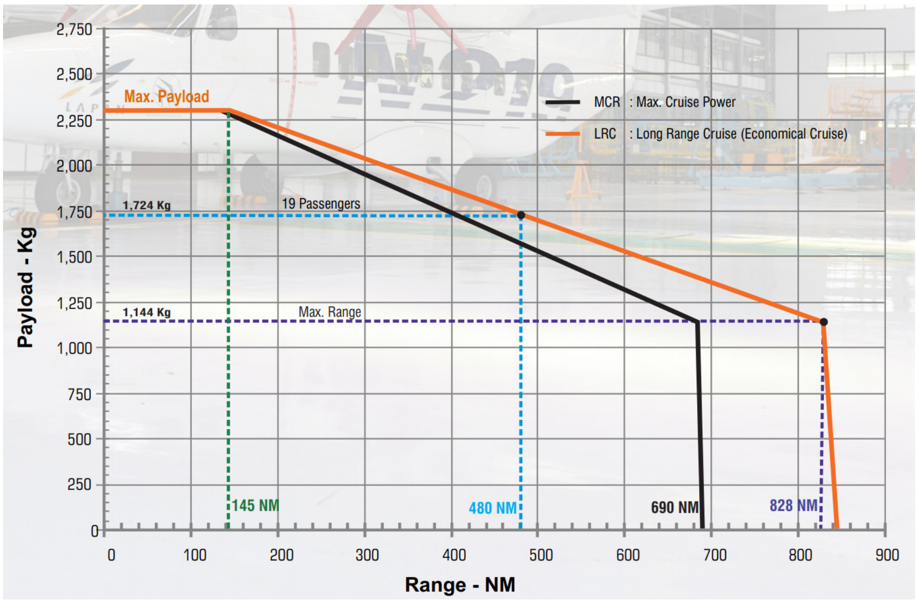


FIGURE 1.8: N-219 Payload Range (*N219 SOLUTION FOR FRONTIER GATEWAY*, 2018).

The takeoff and landing distances, stability, ceiling, payload, fuel economy, speed, rate of climb, maneuverability, range, and endurance are the significant parameters most impacted by performance. The predominance of one or more of these elements determines aircraft differences and explains the high level of specialization observed in modern aircraft (*Pilot's Handbook of Aeronautical Knowledge*, 2016).

Most of the aircraft flight took place in a cruising state, and the aircraft's range and endurance are primarily restricted by the fuel (for an airbreathing engine) or battery (for an electric motor). The aircraft's range and endurance can be calculated using Breguet's equation, but generally, a Breguet equation is tailored for airbreathing engine calculation. For battery-powered aircraft, the weight does not change over time, but the battery's voltage changes with the battery charge state. The reduction in battery voltage will force the battery to supply more current to maintain the same amount of power. Thus by drawing more current, specifically more current than the battery rated capacity, the battery will lose its efficiency. In other words, with an increasingly higher current, the usable capacity of the battery will be reduced; this phenomenon is also known as Peukert's law. (Traub, 2011)

1.2 Problem Statement

Now with all the known benefits of green technology, imagine how much better would it be if the N-219 and other similar aircraft could fully fly using renewable energy? or perhaps it might be worse since we know that battery volumetric and gravimetric energy density is very small compared to fossil fuel energy density, but how much does it really affect the performance? Unfortunately, we do not fully know yet about the effect of electrification on a commuter type of aircraft. One thing for sure is that retrofitting an existing aircraft from an airbreathing engine into a fully electric system is not an easy task. A modification on such a large scale would pose problems ranging from fittings, weight distribution, aerodynamic behavior, performance capability, safety, and many more. But before considering other things, we must first find out about the aircraft's performance after retrofitting, specifically for the maximum range and endurance performance. Because if it turns out that the aircraft could not even fly after retrofitting, why bother considering other factors. That is why this thesis challenges the idea of how the performance of the N-219 aircraft would be affected by the electrification of the N-219 aircraft, and what paramters affect the performance of the electric N-219 aircraft. Furthermore, how would the electric N-219 aircraft compare with the original N-219 in terms of maximum range and endurance.

1.3 Research Purpose

Knowing the estimated range and endurance of an electric N-219 could potentially open up the possibility of realizing the idea, especially if we are able to analyze it under various condition such as different altitude, airspeed, and even different battery properties. Suppose that we are able to know and understand the effect and behavior of retrofitting a commuter-type aircraft into an all-electric aircraft. We could figure out the next step of making an electric commuter aircraft a possibility in that case. Let's say an all-electric plane can achieve one-fifth of the airbreathing counterpart range and endurance, and then from here, we can figure out how we can increase the range or when can we expect the performance of an all-electric counterpart to match with the airbreathing counterpart? By comparing both results, we can determine whether we should start electrifying commuter aircraft with current technology or wait for more.

1.4 Problem Scope

Estimating an aircraft's range and endurance requires a lot of foundation work. There are many variables to collect and calculate before one can go on assessing a range and endurance of an aircraft, let alone calculating an electric aircraft that have a different principle than fossil fuel-based aircraft. In addition, a seemingly small thing can affect the aircraft's performance in many ways. Take the atmosphere as an example; the atmosphere of the earth varies within regions and heights. Thus an exact same aircraft tested in America can have a different performance than an aircraft tested in Australia. This is why it is crucial to account for atmospheric properties using the International Standard Atmosphere (ISA) as a fundamental aspect.

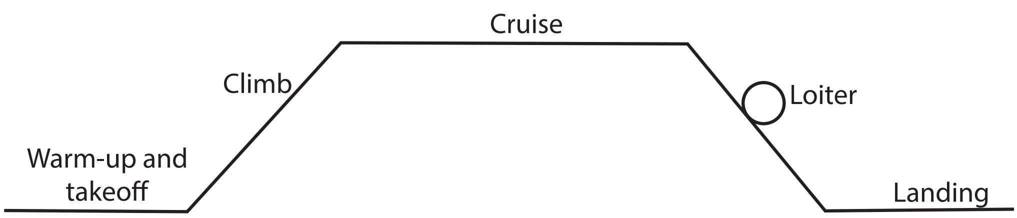


FIGURE 1.9: Typical aircraft mission profile (Meszaros, 2019).

In general, for an aircraft to reach from one point to another, the aircraft needs to ascend/climb, then cruise, finally descend, and land when the aircraft is near the destination, see Figure 1.9. According to boeing, most of aircraft flight time (for a 1.5-hour flight) is on cruise (*Statistical Summary of Commercial Jet Airplane Accidents*, n.d.). But, in terms of range coverage, cruising takes about 90% of distance covered for a medium-range flights. Furthermore, the ratios of maximum landing weight with MTOW of all classification ranges are within 71% to 93% (Loftin, 1980).Thus, it is safe to estimate the range and endurance of an aircraft using only cruising state performance. Furthermore, Breguet's equation, a most common equation to estimate an aircraft's range and endurance, mostly only accounts for the aircraft's cruising state and does not include climbing and descend state.

Although there are many other commuter aircraft are out there, it is better to keep the variable as minimal as possible. Thus this thesis will solely focus on estimating the range and endurance of electric and airbreathing N-219 in the hope that the result will also be relevant for similarly spec aircraft.

The problem that occurs when comparing electric N-219 and airbreathing N-219 is that there are not any exact engine but electric. We already know the airbreathing N-219 engines and its specification, but what about the electric motor for the electric N-219? It is very unlikely to find an electric motor that has the exact same specification of 625 kW of power with 153 kg of weight; it might be possible, but only if one would go through the trouble of designing an electric motor that performs similarly pays someone to. Thus, at best, we can assume that there is exactly the same performing electrical motor with several attributes borrowed from a close enough performing electric motor.

magniX also produces a commercial electric motor for aircraft. Currently, the company has two models, which are the magni250 and magni500. The bigger one, magni500 only capable of producing a continuous power of 560 kW (750 shp) with

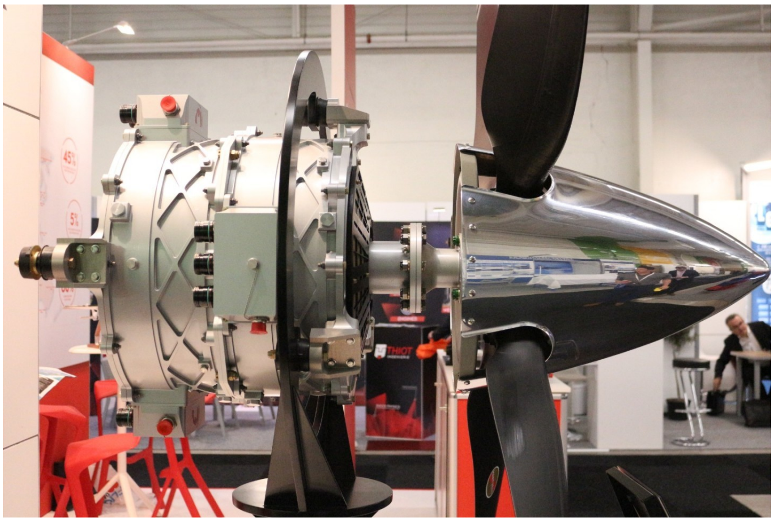


FIGURE 1.10: magni500 with propeller (Lavars, 2019).

a 133 kg weight, which falls short if we compare it to the Pratt & Whitney Canada PT6A-42. But, there are other important parameters of magni500 that cannot be replicated: voltage and motor efficiency. Thus in this thesis, the electric motor will be assumed to be as heavy and as powerfull as the PT6A, but with the voltage and efficiency of magni500.

As mentioned earlier, retrofitting and aircraft modification can pose a lot of complications. For example, things like different engine shape and placement can affect the aerodynamics, with fossil fuel being a fluid have a different effect when maneuvering compared to a solid material such as batteries, and much other stuff that can affect the performance. But, the feasibility analysis starts form cruising performance. Thus, everything about the N-219 is assumed to be the same except mentioned.

TABLE 1.1: N-219 weight component

Airbreathing Version	Electric Version
Structure + Engine + Payload + Fuel	Structure + Engine + Payload + Battery

In general the N-219 total weight makes up from 4 major components, Table

1.1. For both airbreathing and electric N-219, the structure and engine weight are fixed. But, the payload and fuel/battery varies depending its configuration.

TABLE 1.2: N-219 configurations

Configuration	Payload	Fuel/Battery Weight	Fuel/Battery Volume	Total Weight
Airbreathing A	1100 kg	1600 kg	1975 L	7030 kg
Electric A	1100 kg	1600 kg	632 L	7030 kg
Airbreathing B	2100 kg	644 kg	795 L	7030 kg
Electric B	2100 kg	644 kg	255 L	7030 kg
Electric C	1100 kg	4997 kg	1975 L	10 603 kg
Electric D	2100 kg	4997 kg	1975 L	11 603 kg

Since aircraft flies with different payload, we must take into account the variety of payload and battery/fuel capacity. Thus there will be several N-219 configurations that are classified by its payload capacity and battery/fuel capacity, Table 1.2. For this thesis, we take into account that 1 person with luggage takes up around 100 kg of weight each even though in reality the number might be lower. Thus the configurations are as follows:

1. **Airbreathing A**, 11 PAX (1100 kg payload) with 1975 L AVTUR (1600 kg). This configuration maximize the fuel capacity to its limit, which is 1600 kg in weight or around 1975 L in volume. Thus in order to maintain a total weight below MTOW, the payload needs to be less than 1144 kg, thus in terms of passenger it can only hold up to 9 passengers, with 2 pilots at a total of 11.

2. **Electric A**, 11 PAX (1100 kg payload) with 632 L battery (1600 kg) This configuration has the same payload as configuration **Airbreathing A**. But instead of AVTUR, this configuration uses Lithium Ion battery of the same weight (1600 kg). Since Lithium Ion has a much different density compared AVTUR, the total volume of the battery is only 632 L.

3. **Airbreathing B**, 21 PAX (2100 kg payload) with 795 L AVTUR (644 kg) The second airbreathing configuration maximize the payloads, which are 19

passengers and 2 pilots. This payloads have a remainder weight of around 644kg before it reaches MTOW, thus the rest of the remainder weight is filled by fuel. This inherently will give out significantly different result than configuration **Airbreathing A**.

4. **Electric B**, 21 PAX (2100 kg payload) with 255 L battery (644 kg) This configuration has the same payload as configuration **Airbreathing B**. But instead of AVTUR, this configuration uses Lithium Ion battery of the same weight (644 kg). Since Lithium Ion has a much different density compared AVTUR, the total volume of the battery is only 255 L.

5. **Electric C**, 11 PAX (1100 kg payload) with 1975 L battery (4997 kg) Similar to configuration **Electric A**, but instead of the same fuel/battery weight as **Airbreathing A**, this configuration has the same fuel/battery volume as configuration **Airbreathing A**. Thus, with its 1975 L battery, the battery weight is 4997 kg.

6. **Electric D**, 21 PAX (2100 kg payload) with 1975 L battery (4997 kg) Similar to configuration **Electric B**, but instead of the same fuel/battery weight as **Airbreathing B**, this configuration has the same fuel/battery volume as configuration **Airbreathing B**. Thus, with its 1975 L battery, the battery weight is 4997 kg.

1.5 Thesis Structure

This thesis will be branched into five parts, which are:

1. Introduction: The first part of this thesis will introduce the topics and the problems around them. This chapter emphasizes the possibility of electric aircraft and its benefits, including general information about electric engines, batteries, and the N-219 aircraft.

2. Literature review: This part will be about all the formulas and theories behind the thesis. This part is very detailed, including the final form of the equation and the basic form from where it begins. These parts explain International Standard Atmosphere (ISA), Equation of motions, Aerodynamics, Cruise performance, and a modified Breguet's equation for electric aircraft.

3. Research methodology: The workflow and structure of this thesis is explained intricately within this section. From researching about the effect of electrification of aircraft, data taking of N-219 aircraft, and how the equation works will be described in great detail within this chapter.

4. Results and discussion: The result from range and endurance calculation of both electric and airbreathing N-219 will be compared and analyzed within this section.

5. Conclusion and recommendation: The final part will be about concluding the result of this research, answering the problem statement of the thesis, and giving advice about the analyzed result.

CHAPTER 2
LITERATURE REVIEW

2.1 The Atmosphere

2.1.1 Atmospheric Layer

The properties of the atmosphere play a significant role in defining the performance of an aircraft. The atmosphere's properties, such as temperature, pressure, density, and viscosity, vary based on height. The atmosphere is divided into five parts based on their height; those are the troposphere, the stratosphere, the mesosphere, the thermosphere, and the exosphere.

The troposphere is the lowest part of the atmosphere and expands to about 11 km above sea level. In this atmosphere layer, a phenomenon known as weather occurs; the temperature in this atmosphere layer also decreases 6.5 °C per kilometer. This part of the atmosphere is also where most terrestrial aircraft operate.

The stratosphere is divided into two parts, the lower stratosphere, and the upper stratosphere. The lower stratosphere lies just above the troposphere and up to 20 km above sea level. The temperature in these atmospheric layers is constant thoroughly. In the upper stratosphere, in this part of the atmosphere, the temperature starts to increase by 1 °C per kilometer from 20 km to 32 km and 2.8 °C per kilometer from 32 km to 50 km up to 0 °C.

The mesosphere stretch just shy from upper stratosphere up to 90 km, the temperature then again change by -2.8 °C per kilometer from 50 km 71 km and

−2 °C per kilometer from 71 km to 90 km up to −90 °C.

2.1.2 Standard atmosphere

In a real-life scenario, the atmosphere never stays the same. There will always be a variation of atmosphere properties at any given moment in time, be it small or large. Thus, since an aircraft's performance depends on the atmosphere's condition, there must be a generalization of the atmosphere condition so that a comparison between two aircraft or more can be relevant.

If the air in the atmosphere is assumed to be a perfect gas and that the atmosphere is not moving in reverence to the Earth, a standardization for computation such as temperature, pressure, density, viscosity, and other properties with respect to altitude can be formulated.

TABLE 2.1: Primary constants values
(Ruijgrok, 2009)

Sea-level pressure	$p_0 = 101325$ N/m^2
Sea-level temperature	$T_{M0} = 288.15\,\text{K}(15\,°\text{C})$
Sea-level density	$\rho_0 = 1.225$ kg/m^3
Acceleration of gravity at sea level	$g_0 = 9.80665$ m/s^2
Universal gas constant	$R_a = 8314.32$ J/K kmol
Ratio of specific heats of air	$\gamma = c_p/c_v = 1.4$

As mentioned in 2.1.1 temperature change differently in the different atmospheric layer. The calculation for the temperature can be calculated using

$$T_M = T_{M1} + \lambda(H - H_1) \tag{2.1}$$

In which T_{M1} is the temperature at H_1 or the lower barrier of the atmospheric layer (e.g., 0 km for the troposphere, 11 km for the lower stratosphere), and λ is the temperature gradient dT_M/dH which varies with each atmospheric layer.

The pressure of the atmosphere decreases as one goes higher within the atmosphere. This is expressed by the equation

$$p - (p + dp) - \rho g dh = 0 \;\; or \;\; dp = -\rho g dh \tag{2.2}$$

where:

- p is pressure,
- ρ is density,
- h is geometrical altitdue,
- g is the acceleration of gravity.

The Equation (2.2) can be derived multiple times to get the pressure equation for a non-isothermal layer. A non-isothermal layer is characterized by the change in temperature according to the height. When Equation (2.2) is inserted by

$$\frac{p}{\rho} = \frac{R_a}{m} T_M,$$

yields the following

$$dp = -\frac{mp}{R_a T_M} g dh \;\; or \;\; \frac{dp}{p} = \frac{mg}{R_a T_M}, \tag{2.3}$$

where

- R_a denoted as the universal gas constant,
- m identified as the molecular mass of the gas,
- T_M is the absolute temperature.

When Equation (2.3) is integrated from sea level into a certain altitude h, it will yields

$$\int_{p_0}^{p} \frac{dp}{p} = \int_{0}^{h} \frac{Mg}{R_a T_M} dh, \quad \text{and} \tag{2.4}$$

$$ln \frac{p}{p_0} = \int_{0}^{h} \frac{Mg}{R_a T_M} dh. \tag{2.5}$$

Geopotential altitude H is the height that considers the change of gravity with latitude and altitude, which defined by the following equation,

$$H = \int_{0}^{H} dH = \frac{1}{g_0} \int_{0}^{h} g dh, \tag{2.6}$$

with g_0 being the acceleration of gravity at sea level.

When Equation (2.6) is combined with Equation (2.5) will resulted in the following relationship geopotential altitude H and air pressure

$$\ln \frac{p}{p_0} = - \int_{0}^{H} \frac{m g_0}{R_a T_M} dH, \tag{2.7}$$

and if we insert Equation (2.1) into Equation (2.7) the following equation is obtained

$$\ln \frac{p}{p_1} = - \int_{H_1}^{H} \frac{g_0}{R \left[T_{M1} + \lambda (H - H_1) \right]} dH = - \frac{g_0}{R\lambda} ln \left[T_{M1} + \lambda (H - H_1) \right] \Big|_{H_1}^{H} \quad \text{or}$$

$$\ln \frac{p}{p_1} = - \frac{g_0}{R\lambda} ln \frac{T_{M1} + \lambda (H - H_1)}{T_{M1}}. \tag{2.8}$$

The variation with altitudes will result in

$$\frac{p}{p_1} = \left[1 + \frac{\lambda (H - H_1)}{T_{M1}} \right]^{-\frac{g_0}{R\lambda}}. \tag{2.9}$$

Repeating the same derivation method for density ratio result in

$$\frac{\rho}{\rho_1} = \frac{pT_{M1}}{p_1 T_M} = \left[1 + \frac{\lambda(H - H_1)}{T_{M1}}\right]^{-\left[\frac{g_0}{R\lambda}+1\right]}. \tag{2.10}$$

For an isothermal atmospheric layer such as the lower stratosphere, the equation for pressure ratio and density ratio differs. Direct integration of Equation (2.7) between the lower boundaries of the layer H_1 and the height H result in

$$\ln\frac{p}{p_1} = -\int_{H_1}^{H} \frac{g_0}{RT_{M1}} dH = -\frac{g_0}{RT_{M1}}(H - H_1) \;\; or \tag{2.11}$$

$$\frac{p}{p_1} = e^{-\frac{g_0}{RT_{M1}}(H-H_1)}. \tag{2.12}$$

and for density

$$\frac{\rho}{\rho_1} = \frac{p}{p_1}.$$

$$\frac{\rho}{\rho_1} = e^{-\frac{g_0}{RT_{M1}}(H-H_1)}. \tag{2.13}$$

The function above requires knowing the geopotential altitude, which is a height that considers the change of gravity due to altitude and latitude. To better understand the function of geopotential altitude, we first must establish the relationship between the geometrical altitude with the acceleration of gravity. The following expression can be obtained by applying Newton's law of gravitation.

$$\frac{g}{g_0} = \frac{R_e^2}{(R_e h)^2} \tag{2.14}$$

Then, a combination of Equation (2.14) and (2.6) yields

$$\int_0^H dH = \int_0^h \frac{R_e^2}{(R_e + h)^2} dh = R_e^2 \int_0^h \frac{dh}{(R_e + h)^2}$$

$$H = R_e^2 \int_0^h -d(R_e + h)^{-1} = \frac{R_e h}{R_e + h}, \quad \text{or}$$

$$h = \frac{R_e H}{R_e - H}. \tag{2.15}$$

The viscosity μ can be obtained by the following

$$\mu = \frac{\beta T_M^{3/2}}{T_M + S}, \tag{2.16}$$

where:

- β is a constant of $1.458 \times 10^{-6} kgs^{-1}m^{-1}K^{-1/2}$
- S_C is the Sutherland's constant which is 110.4 K
- T_M is the absolute temperature

The dynamic viscosity coefficient at sea level μ_0 is $1.7894x10^{-5}kg/ms$. The kinematic viscosity is a ratio of dynamic viscosity over density, $v = \frac{\mu}{\rho}$. The value speed of sound c_s can be calculated using

$$c_s = \left(\gamma \frac{R_a}{m} T_M\right)^{1/2} = (\gamma R T_M)^{1/2} \tag{2.17}$$

at sea level, the value of c is 340.294 m/s.

2.1.3 Off-standard atmosphere

A standard atmosphere (ISA+0) is generalized using base temperature 15 °C at sea-level. Since different deviation in the temperature ΔT_M affects the performance of the aircraft, the equation must consider the temperature difference, thus

$$T_M = T_{ISA} + \Delta T_M, \tag{2.18}$$

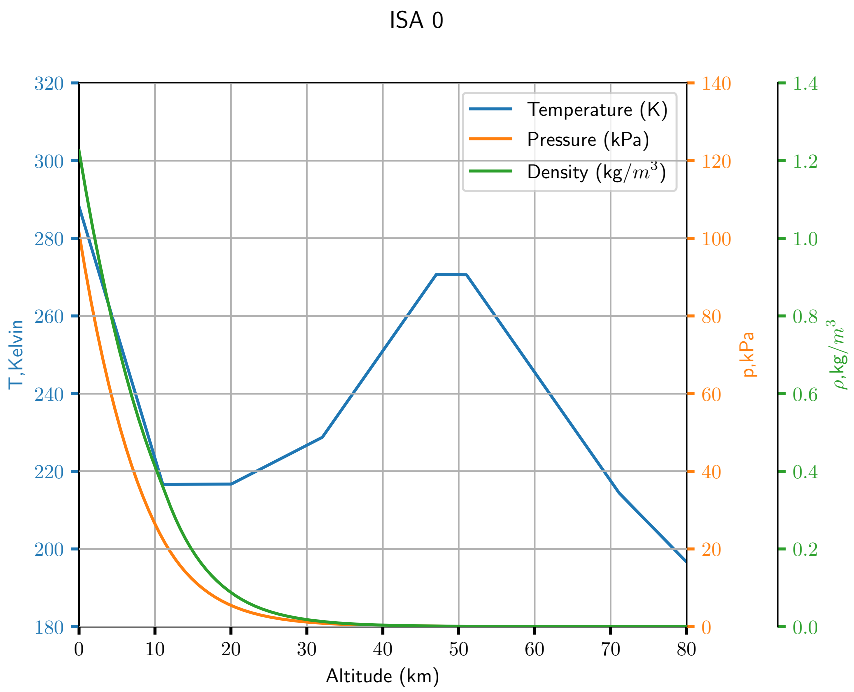


FIGURE 2.1: Variation of pressure, density and temperature In the International Standard Atmosphere

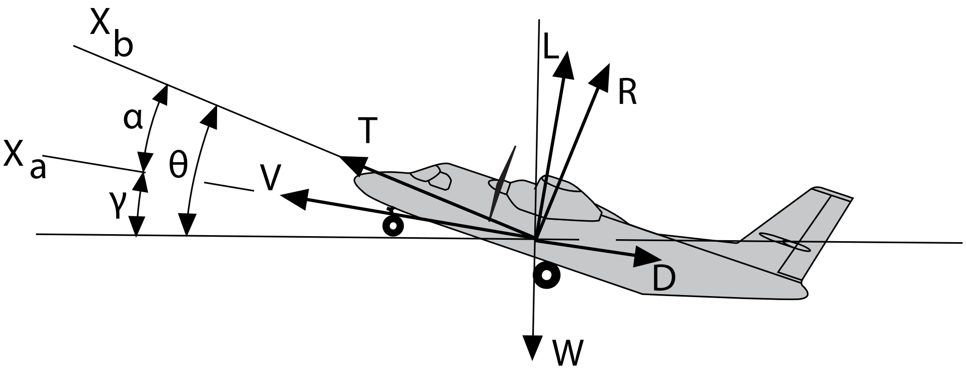


FIGURE 2.2: Free body diagram

where T_{ISA} is the base temperature at standard atmosphere, which is 15 °C or 288.15 K at sea-level. Although the temperature change in off-standard atmosphere, the pressure remain constant. The density can be calculated using perfect gass law

$$\rho = \frac{p}{(RT_M)}, \tag{2.19}$$

where R_u is the specific gas constant, $287.053 J/(kg\ K)$.

2.2 Equations of Motion

2.2.1 Translational motion

The force used to propel an aircraft in any direction is equal to the weight of the aircraft times the desired acceleration. This implication is supported by Newton's second law of motion, which can be written as

$$\vec{F} = \frac{d(M\vec{V})}{dt}, \tag{2.20}$$

in this equation, the sum of all resultant external forces applied to the body of the aircraft is denoted as $\vec{F}$, while the linear velocity vector of the body's center of gravity relative to an inertial reference frame is referred to as $\vec{V}$, and the weight of the aircraft is known as M.

Realistically the body of an aircraft is flexible to some degree, in the sense that some of the relative positions of the various parts of the aircraft's structure change when affected by forces. But this adds complexity to the equation, and most often, simplifying by ignoring the deformations is beneficial and justified. Thus assuming the aircraft has a rigid body with constant mass, the Equation (2.20) morphed into

$$\vec{F} = M\frac{d\vec{V}}{dt} = M\vec{a}. \tag{2.21}$$

The equation for rotational motion of a rigid body is

$$\vec{M}_{cg} = \frac{d\vec{B}_{cg}}{dt}. \tag{2.22}$$

The above equation states that the external moment applied to a body relative to its center of gravity is equal to the time derivative of its angular momentum.

The translational motion of the rigid aircraft's constant mass will be defined within this chapter using the body axis system. In this case vector $\vec{F}$ becomes

$$\vec{F} = M\left(\frac{\delta\vec{V}}{\delta t} + \vec{\Omega}\times\vec{V}\right). \tag{2.23}$$

$\frac{\delta\vec{V}}{\delta t}$ is defined as the time derivative of the velocity vector in respect to the body axis system in the above equation.

2.2.2 Rotational motion

The sum of the angular momentum relative to the aircraft's center of gravity is

$$\vec{B}_{cg} = \int_M \vec{r}\times\left(\vec{\Omega}\times\vec{r}\right)dM. \tag{2.24}$$

The position vector of a mass element relative to the aircraft's center of gravity is denoted as vector $\vec{r}$. The equation above can be rewrite by utilizing the basic law of the vector triple product $(\vec{r} \times (\vec{\Omega}\vec{r}) = \vec{\Omega}(\vec{r} \cdot \vec{r})\vec{r}(\vec{\Omega} \cdot \vec{r}))$ as follows

$$\vec{B}_{cg} = \int_M \vec{\Omega}(\vec{r} \cdot \vec{r})dM - \int_M \vec{r}(\vec{\Omega}\vec{r})dM. \tag{2.25}$$

The dot product of Equation (2.25) in the condition that the coordinates are x,y,z are $\vec{r} \cdot \vec{r} = x^2 + y^2 + z^2$, and $\vec{\Omega} \cdot \vec{r} = px + qu + rz$. Which when integrated into Equation (2.25) become

$$\vec{B}_{cg} = \vec{\Omega} \int_M (x^2 + y^2 + z^2)dM - \int_M \vec{r}(px + qy + rz)dM.$$

2.2.3 Special Types of Flight

Although there are several special types of flight, in this thesis, we will only be discussing about **Steady straight nonsideslipping flight** as it is the state of flight that is mostly used on the cruise phase.

In this flight state, all variables in the lateral plane are zero. Thus β, C, S, and μ is equal to zero. Thus the forces acting on the aircraft are

$$\begin{aligned} -D + T\cos\alpha_T - W\sin\gamma = 0 \\ -L - T\sin\alpha_T + W\cos\gamma = 0 \end{aligned} \tag{2.26}$$

The aerodynamic of the aircraft change depend on the use of the controls. However, in this case of flight types, the lift, drag, and side force from the control surface are very small that it is arguably okay to neglect. It is also beneficial to ignore these previously contributed forces as the moment M_x,M_y, and M_z must be zero. Since this state of flight only focuses on translational motion, the equation is limited to forces only.

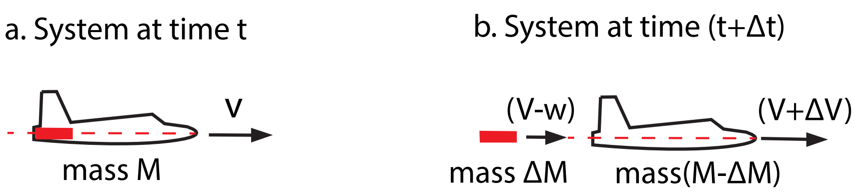


FIGURE 2.3: Ejection of airplane mass (Ruijgrok, 2009)

2.2.4 Translational equation for variable mass

When an aircraft operates, the power from the engine is provided by the fuel. As the aircraft fly, the fuel will be continuously consumed and thus gradually reduce the total mass of the aircraft.

To calculate the translational motion for a rigid aircraft, we first must formulate the equation by considering that it is flying straight in a horizontal plane and that has mass M and velocity V at time t with respect to the inertial frame of reference (Fig. 2.3a).

Since as time increases, the mass of the aircraft is reduced, and the reduction of mass increases the velocity; thus, at time $t + \Delta t$, the mass is $M - \Delta M$ and velocity is $V + \Delta V$ (Fig. 2.3b). The total external force acting on the whole mass system at time t is equal to the rate of change in linear momentum of the system, according to Newton's second equation of motion. We have

$$F = \lim_{\Delta t \to 0} \frac{(M - \Delta M)(V + \Delta V) + \Delta M(V - w) - MV}{\Delta t} = M\frac{dV}{dt} - w\frac{dM}{dt},$$

in the limit as Δt approaches zero. Thus

$$F + w\frac{dM}{dt} = M\frac{dV}{dt},$$

with $M\frac{dV}{dt}$ replaced as fuel flow rate $-m_f$ (the minus sign is because the mass is reduced) the equation become

$$F - m_f w = M\frac{dV}{dt}. \tag{2.27}$$

2.3 Aerodynamic Basis

2.3.1 Aerodynamic coefficients

Forces acting on a moving aircraft such as viscous forces and pressure forces produce moment M_{cg} and also the aerodynamic forces R. On the outer surface of the airplane, the shear stress generates viscous forces, and on the wing, along with the other aircraft's component, an asymmetric pressure occurs that creates pressure forces.

The fluctuation of the static pressure p along a streamline is given by the following equation according to Bernoulli's equation for compressible isentropic flow.

$$p_t = p\left[1 + \frac{\gamma - 1}{2}\frac{1}{\gamma}\frac{\rho}{p}V^2\right]^{\frac{\gamma}{\gamma-1}}. \tag{2.28}$$

The freestream total pressure p_t becomes the same as the static pressure when the velocity decreases until zero from the freestream value by the stagnation point on the wing's nose.

As seen in fig (2.4), the streamline flow along the upper wing's surface, the local static pressure decreases as the result of the increase in velocity. The pressure and the velocity keep decreasing and increasing respectively along the surface until the pressure reaches its lowest value and velocity its highest value; after passing those points, the pressure increase as the velocity decreases. On the lower surface, a similar phenomenon is also happening, and the difference is in the values and the location of the occurrence. Due to the camber difference between the upper

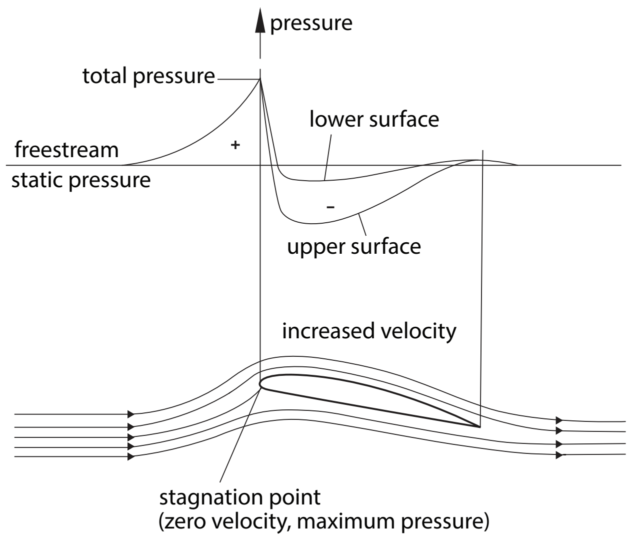


FIGURE 2.4: Pressure distribution over a wing section (Ruijgrok, 2009)

and lower surface, and also the effect of the incident angle of the wing section, the pressure of the wing's lower surface is greater than the upper surface. Thus the wing's shape and the wing's angle of attack resulted in a force.

When fluid particles interact in cohesion, it creates a velocity difference between air laminae. The shear force is induced due to the sliding motion between adjacent layers. The skin friction drag of the body is the resultant effect of all these forces. The boundary layer is the layer where air adheres to the surface where friction is crucial. To understand the characteristic of drag, it is imperative to learn the concept of the boundary layer. Thus the departure point is the concept that viscosity manifests only in a confined area and not in the main flow.

The pattern of a flow around a body can be split into two areas, a thin boundary layer one where friction is significant as well as an area above that where the air functions as a frictionless fluid.

It has been experimentally found that the shear stress τ_0 is determined by the result of both the slope of the surface velocity profile and the dynamic viscosity coefficient μ,

$$\tau_0 = \frac{dV}{dn}_0 \mu. \tag{2.29}$$

The coefficient μ is a physical property roughly proportional to T over the standard spectrum of air temperatures, which refers to the basic type of laminar flow where air layers slide over each other in the form of parallel layers.

The following quantities may determine the aerodynamic forces and moments:

- the shape of the aircraft in general,
- the surface size of the aircraft S,
- condition of the aircraft,
- deflection of the control surface,
- aircraft attitude in relation to the free stream,
- the airspeed V,
- air density ρ,
- dynamic viscosity coefficient μ,
- speed of sound c,
- Mach Number $M_a = V/c$.

The parameter Mach number is essential to indicate the air compressibility on the pressure distribution over the aircraft surface.

The dimensional analysis is a tool used to extract the expression moment M_{cg} and aerodynamic force F_A Stating the aerodynamic force on the aircraft depends on variable μ, ρ, c, S, V as follows

$$F_A = f(S, V, \rho, c, \mu).$$

As we know, F_A is the force dimension; Thus, it must have a force dimension on both the left and right sides of the equation, which the previous was for the left side. Therefore to make sure the dimension is uniform, then the previous equation should be written as

$$F_A = K(S^a V^b \rho^d c^e \mu^f). \tag{2.30}$$

The superscript a, b, d, e, and f are an unknown constant, while the function of the remaining dimensionless variables is K. In terms of time T, length L, and mass M,

$$\frac{M_a L}{T^2} = K(L^2)^a \left(\frac{L}{T}\right)^b \left(\frac{M_a}{L^3}\right)^d \left(\frac{L}{T}\right)^e \left(\frac{M_a}{LT}\right)^f .$$

From the previous equation, the mass exponent on the right-hand side of the equation is $d + f$ while the left-hand side is 1, so

$$1 = d + f, \tag{2.31}$$

and for the length

$$1 = 2a + b - 3d + e - f, \tag{2.32}$$

for the time we have

$$-2 = -b - e - f. \tag{2.33}$$

The three previous equations have five unknowns. If we assume that $\rho, V, \text{and} S$ are priority, then Equation (2.31) until (2.33) using $d, b, \text{and} a$ in terms of f and e, we obtain

$$a = 1 - f/2$$
$$b = 2 - e - f$$
$$d = 1 - f.$$

Replacing the variable in Equation (2.30) with the previous variables yields

$$F_A = K(S)^{1-f/2}(V)^{2-e-f}(\rho)^{1-f}c^e\mu^f.$$

Grouping variables with specific exponents generates

$$F_A = K\rho V^2 S\left(\frac{c}{V}\right)^e\left(\frac{\mu}{\rho V S^{1/2}}\right)^f.$$

As the $S^{1/2}$ dimension corresponds to a length l, we can write

$$F_A = K\rho V^2 S\left(\frac{c}{V}\right)^e\left(\frac{\mu}{\rho V l}\right)^f.$$

The freestream Mach number is the ratio V/c.The relative value of shear and inertia forces within the flow is indicated by the *Reynolds number Re*, also known as the quantity $\rho V l/\mu$. The lower the value of Re, the more viscous forces are relatively significant.

$$F_A = K\rho V^2 S\left(\frac{1}{M_a}\right)^e\left(\frac{\mu}{Re}\right)^f$$
$$K\left(\frac{1}{M_a}\right)^e\left(\frac{1}{Re}\right)^f = \frac{C_R}{2},$$

thus

$$F_A = C_R\frac{1}{2}\rho V^2 S = C_R q S. \tag{2.34}$$

The wing planform area is usually used as the reference surface for an aircraft;

this area is often called the wing area. In the Equation (2.34), there is a representation of lift L, drag D, as well as the aerodynamic force side force components:

$$\begin{aligned} L &= C_L \frac{1}{2} V^2 S \\ D &= C_D \frac{1}{2} V^2 S \\ S &= C_S \frac{1}{2} V^2 S, \end{aligned} \tag{2.35}$$

$C_S, C_D, \text{and} C_L$ are the coefficient of side force, drag, and lift, respectively. Applying the dimensional analysis technique into moment M_{cg}, we obtain

$$M_{cg} = C_M \frac{1}{2} \rho V^2 S \bar{c}, \tag{2.36}$$

Here the coefficient of the nondimensional moment is referred to as C_M. The wing aerodynamic chord is equal to the factor of the length $\bar{c}$.

2.3.2 Airfoil and wing characteristics

An airfoil or also called aerofoil, is a surface that provides a useful aerodynamic force when moved through the air, when air moves over it or a combination of both. An airfoil is shaped in such ways so that it can generate as much lift without excessive drag.

The curvature of an airfoil is determined by the mean chamber line, as shown in Figure 2.5, the mean chamber line is located between the top and bottom surface. At both ends of the mean chamber, line are the trailing edge and the leading edge. If we draw a straight line between the trailing edge and the leading edge, we have a chord line. The distance measured along the chord line, from one end to another, is called chord or chord length c. The angle α is the angle of attack, which is the angle between the chord line and the freestream direction.

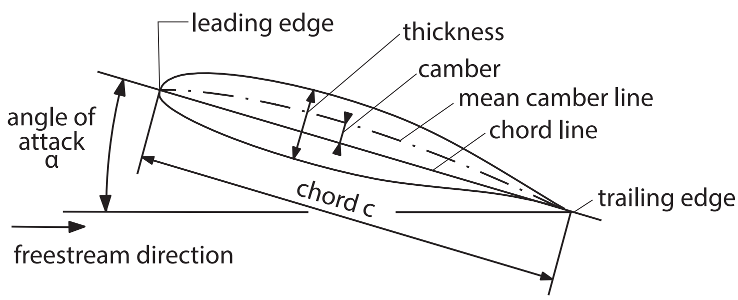


Figure 2.5: Airfoil geometry and nomenclature (Ruijgrok, 2009)

According to Equations (2.35) and (2.36)the two-dimensional lift, drag and moment coefficients are,

$$c_l = \frac{l}{\frac{1}{2}\rho V^2 c}, \tag{2.37}$$

$$c_d = \frac{d}{\frac{1}{2}\rho V^2 c}, \tag{2.38}$$

$$c_m = \frac{m}{\frac{1}{2}\rho V^2 c}, \tag{2.39}$$

l is the Lift, d is the drag, and m is the pitching moment of the aerodynamic force per unit width of the wing. Usually, moment M_o is located in the one-fourth of the chord line, and at low speed, the aerodynamic center is very close to the point M_o; thus the aerodynamic center is at $c/4$.

Several parameters are shown in Figure 2.7. The length between one end of the wing to the other end of the wing is called wing span, denoted as b. The wing area S can be find using $S = \int_{b/2}^{-b/2} c(y)dy$,

The ratio between the tip chord and the root chord,c_t/c_r, is known as the taper ratio.

The geometric chord c_{mg} can be calculated using

$$c_{mg} = \frac{S}{b}.$$

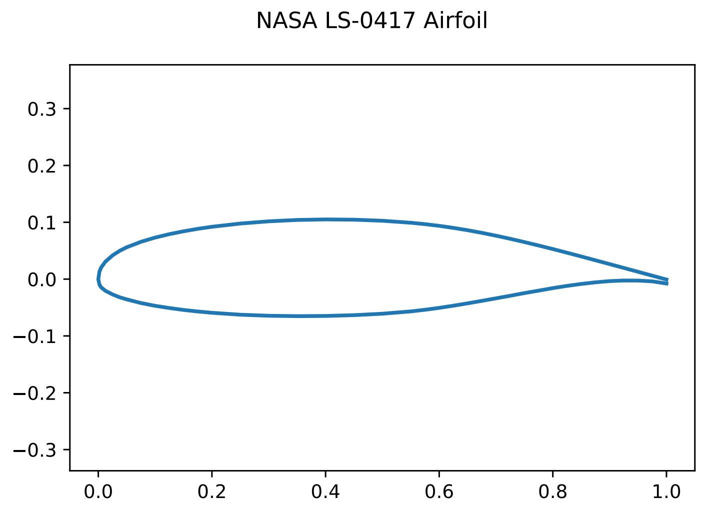


FIGURE 2.6: LS(1)-0417 airfoil

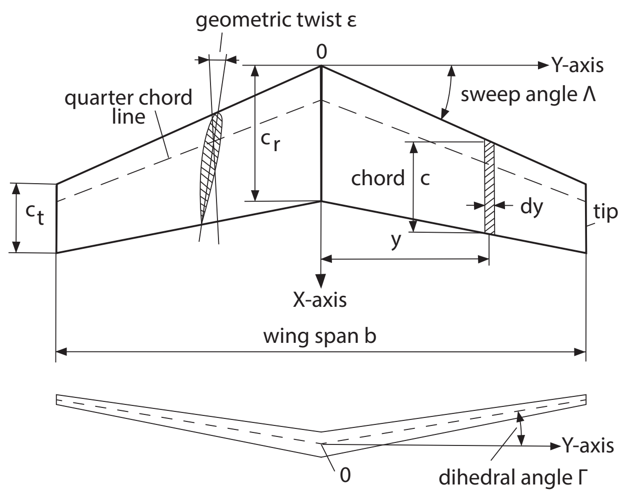


FIGURE 2.7: Wing geometry (Ruijgrok, 2009)

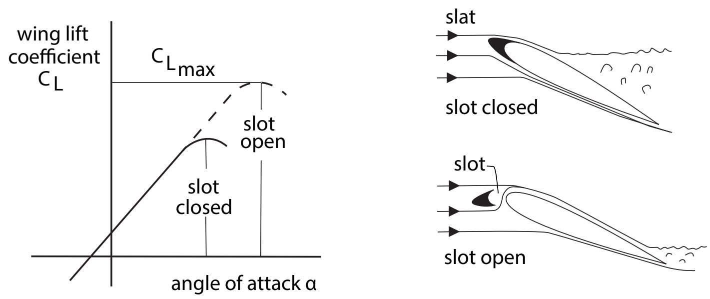


FIGURE 2.8: Effect of a slot on wing lift curve (Ruijgrok, 2009)

The aspect ratio A is the ratio of the wingspan over geometric chord,

$$A = \frac{b}{c_{mg}} = \frac{b^2}{S}. \tag{2.40}$$

The aerodynamic chord $\bar{c}$ is important for assessing the moment coefficient,

$$\bar{c} = \frac{1}{S}\int_{-b/2}^{b/2} c^2(y)dy. \tag{2.41}$$

When the angle of attack is changed, the drag is affected in a sense that

$$C_D = C_{Dp} + C_{Di}, \tag{2.42}$$

where C_{Dp} is the profile drag coefficient, and C_Di is the induced drag coefficient. The maximum lift coefficient C_{Lmax} can be raised by changing the wing configuration; this can be done using the trailing-edge flaps. C_{Lmax} can also be raised using slot, which allows the air to flow from under the wing into the top of the surface as shown in Figure 2.8 . This allows the use of a higher angle of attack and reduces stall speed.

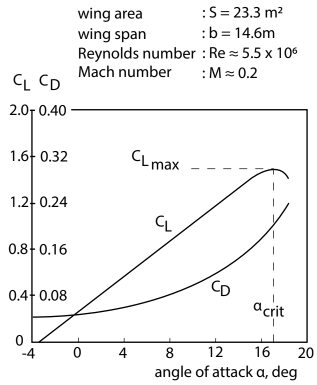


FIGURE 2.9: Aerodynamic characteristics of a low-subsonic airplane (estimated) (Ruijgrok, 2009)

2.3.3 The lift-drag polar

Drag polar is the relationship between an aircraft's lift and its drag, defined in terms of the drag coefficient's dependency on the lift coefficient. An equation may explain it or show it in a diagram called a polar plot.

As shown in Figure 2.9 the drag and lift coefficient on the low-subsonic airplane using clean configuration is an angle of attack function. The wing section essentially has the same form as these curves. When the angle of attack is around 16°, the flow separation begins from the wing, this causes an increase in drag and severe loss of lift. Thus only at below critical angle of attack can an aircraft fly.

the lift-drag polar can be obtained by eliminating angle of attack α from the relations $C_D = f(\alpha)$and$C_L = f(\alpha)$

$$C_D = f(C_L). \tag{2.43}$$

The lift-drag polar from Figure 2.9 is given in Figure 2.10. Furthermore, the aircraft lift-drag polar with its wing's flaps down is plotted. The maximum lift coefficient will increase significantly due to the deflection of the flap. From Equation (2.43) it is known that lift-drag polar is vital if only several parameters are known,

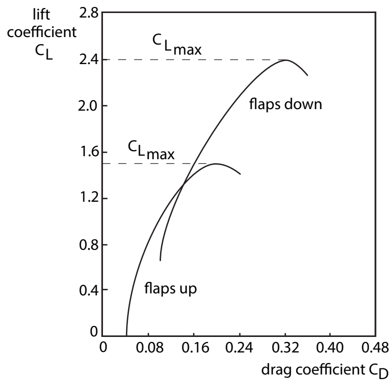


FIGURE 2.10: Lift-drag polar for a low subsonic airplane (estimated) (Ruijgrok, 2009)

which are:

- Aircraft shape,
- Reynolds number,
- Mach number.

The drag increment from the deflection of the control surface can be neglected in most calculations.

Within various aircraft flight phases, typical configurations might be distinguished in concern of the condition of the aircraft. Thus, the aircraft's condition during a particular flight phase will be represented by the lift-drag polars. These phases that will be represented are some of the following:

- takeoff (landing gear down and partly deflected flaps),
- cruise (retracted landing gear and flaps),
- landing (landing gear down and fully deflected flaps).

The lift-drag polars of a retractable landing gear transport aircraft, as seen in Figure 2.11, indicate that the landing gear also provides a major contribution to

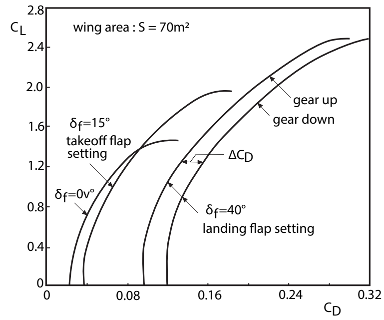


FIGURE 2.11: Typical lift-drag polars for propeller-driven transport airplane (estimated) (Ruijgrok, 2009)

the overall drag of the aircraft in addition to the rise in the drag coefficient due to flap deflection.

In the presence of the boundary layer, the viscous flow manifested. The flow in the boundary layer will become linear when the Reynolds number is low, while turbulent flow is prominent in higher Reynolds numbers. Air particles will oscillate across the boundary layer when the flow is turbulent, which from the boundary layer freestream an energy transfer is happening and among the laminae an exchange of kinetic energy. Shown in Figure 2.12 are both the velocity profiles across the boundary layer for turbulent and laminar flow. If the flow condition is assumed to be the same, the layer with laminar flow is thinner than the one with turbulent flow. Furthermore, The laminar flow layer is much smaller near the surface of the velocity gradient rather than the turbulence layer. Thus, Equation (2.29) shows that skin friction drag will increase due to transition to turbulence.

An aircraft undergo a Reynolds number of 5×10^6 to 10^8 or higher as a characteristic linear measure referenced to the wing chord. Around 90percent of the wing chord portion presents a turbulent flow when at high Reynolds numbers.

Figure 2.13 shows a sketch of the boundary layer development over a wing upper surface. A boundary layer always presents near the leading edge just over the front portion. When flow moves from the wing nose, the layer thickness increases. Passing the minimum local pressure of the air, the turbulence transition occurs; this

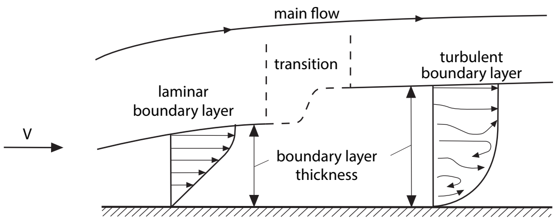


FIGURE 2.12: Typical laminar and turbulent boundary layer velocity profiles (Ruijgrok, 2009)

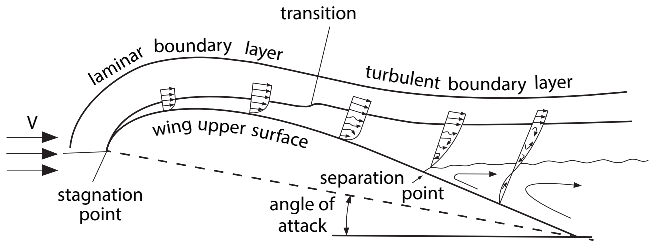


FIGURE 2.13: Development of the boundary layer over the wing (Ruijgrok, 2009)

transition is accompanied by the boundary layer drastic thickening. The increase in downstream distance causes the boundary layer to thickens. The air particles press against the increasing local pressure and the viscous forces when moving over the rear portion of the wing surface. The flow collapsed at a given point and emanated a wake from the separation of the flow.

Typically near the trailing edge at a lower angle of attack, a separation occurs. The thickness of the boundary layer increases if the angle of attack increases, the point of separation moves forward and builds up the coefficient of the drag.

The boundary layer becomes turbulent further upstream as the Reynolds number increases. Separation is postponed at the same time, resulting in a smaller wake. A separation causes a drag measured by the size of the wake. The smaller overall drag coefficient is due to the smaller component of pressure drag, which is caused by the smaller wake. Contrarily, as the Reynolds number grows, the overall lift coefficient that can be achieved increases marginally.

At this stage, it should be remembered in order to detect the approaching stall of an aircraft, an appropriate stall warning, with flaps and landing gear in any normal position, in straight and turning flight, must be present. Either the artificial stall warning device or the aircraft's intrinsic actions can give a warning. The artificial stall warning device usually consists of a pressure vent on the wing's leading edge, positioned such that the stagnation pressure reaches the aperture just before the stall point. The pressure there changes rapidly as a consequence. To create an acoustic signal or warning light, this huge pressure change is used to inform the pilot that the angle of attack is near the angle of the stall. The stall warning starts at a pace that exceeds the speed of the stall and persists until the stall happens.

A tiny vane on the fuselage side near the nose can also assess the stall angle. The vane will rotate freely so that it is aligned with the flight direction, translating the angle of the vane into an electrical signal sent to the cockpit indicator.

With respect to the differences between the maximum lift coefficient and the drag coefficient of the profile, it should be noted that these coefficients vary only at a very small scale within the usual ranges of Reynolds numbers observed during the different flight phases. This observation means that it would normally be necessary to take into account the mean value of the Reynolds number in each flight phase.

2.3.4 Parabolic lift-drag polar

The component drag D_n of the aircraft, and the drag of the wing D_w can be summed into the total drag of an aircraft,

$$D = D_w + D_n.$$

From Equation (2.42) the wing drag can be written as the total of profile drag D_p and induced drag D_i. Thus the previous equation becomes

$$D = D_i + D_p + D_n.$$

Skin friction drag, pressure drag, and wave drag are a part of the profile drag. For subsonic airspeed below the critical Mach number, the wave drag is null. Certain area Sn is referenced as a foundation for a drag coefficient of each component element, C_{Dn}; thus the total drag of the aircraft is given by

$$C_D \frac{1}{2}\rho V^2 S = C_{Di}\frac{1}{2}\rho V^2 S + C_{Dp}\frac{1}{2}\rho V^2 S + (\Sigma C_{Dn} S_n)\frac{1}{2}\rho V^2 S.$$

The aircraft drag coefficient is

$$C_D = C_{Di} + C_{Dp} + \frac{\Sigma C_{Dn} S_n}{S},$$

here the parasite drag coefficient is denoted as $\frac{\Sigma C_{Dn} S_n}{S}$.

In principle, the induced drag coefficient is directly proportional to C_L^2 and inversely proportional to efficiency factor of the wing ϕ and the aspect ratio A,

$$C_{Di} = \frac{C_L^2}{\pi A \phi}.$$

The variable ϕ relies mainly on the wing planform as it shows how similar the distribution of elliptic spanwise lift is received. ϕ should be equal to 1 (minimal coefficient of induced drag) for elliptic lift distribution.

$$C_D = \frac{C_L^2}{\pi A \phi} + C_{Dp} + \frac{\Sigma C_{Dn} S_n}{S}.$$

The previous equation can be written as the following because the parasite and profile drag coefficient are both dependent on the angle of attack,

$$C_D = \frac{C_L^2}{\pi A \phi} + X C_L^2 + \left[C_{Dp} + \frac{\Sigma C_{Dn} S_n}{S}\right]_{C_L=0}.$$

The word XC_L^2 indicates the presumed parabolic change in the lift coefficient with

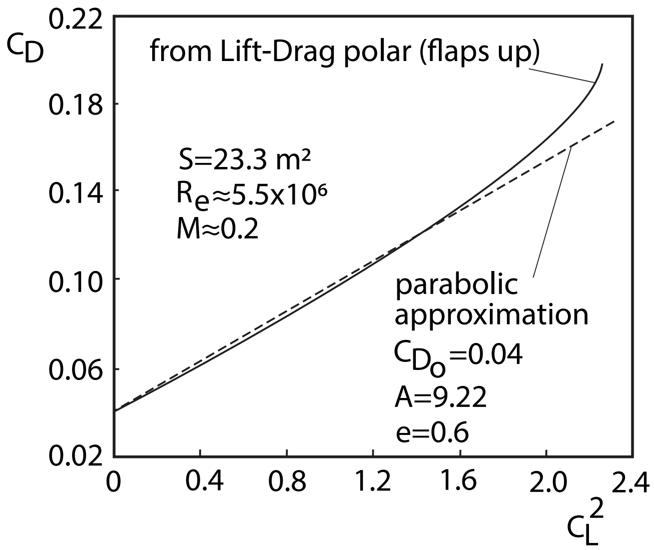


FIGURE 2.14: Parabolic approximation of lift-drag polar of low-subsonic airplane (Ruijgrok, 2009)

the profile and parasite drag coefficients. The quantity in parentheses is called the zero-lift drag coefficient C_{D_0}. Thus the previous equation can be turned into

$$C_D = C_{D_0} + \frac{C_L^2}{\pi A e} \tag{2.44}$$

while e (*Oswald's efficiency factor*) is obtained from $\frac{1}{e} = X\pi A + \frac{1}{\phi}$. This factor evidently accounts for the heterogeneity of the lift coefficient with profile and parasite drag coefficients and the influence of the real span-wise lift distribution on the induced drag coefficient. Thus in several occasion $\frac{1}{e} = X\pi A + \frac{1}{\phi}$ can be written as

$$C_D = C_{D_0} + kC_L^2. \tag{2.45}$$

Induced drag factor k is equal to $1/(\pi A e)$.

Figure 2.14 plotted the drag coefficient of Figure 2.10 against C_L^2. A large portion of the Iift-drag polar is a parabola, but there is some extra drag at lift coefficients above around 1.0.

If the C_{D0} and k values from Equation 2.45 are correctly modified, then the parabolic lift-drag polar can also be used for both supersonic and transonic speed. It should be mentioned that aircraft performance in many respects is calculated by the following aerodynamic ratios: C_L/C_D^2, C_L/C_D, and C_L^3/C_D^2. We differentiate maximum CL/CD with regard to C_L and set the first derivative equal to zero. Since $C_D \neq 0$, then

$$\frac{dC_D}{dC_L} = \frac{C_D}{C_L}.$$

Using Equation 2.44, we can get

$$\frac{2C_L}{\pi Ae} = \frac{C_{D_0} + C_L^2/\pi Ae}{C_L} \quad \text{or} \quad C_L = \sqrt{C_{D_0}\pi Ae}.$$

Substitution of $C_L = \sqrt{C_{D_0}\pi Ae}$ into

$$\left(\frac{C_L}{C_D}\right)_{max} = \frac{\sqrt{C_{D_0}\pi Ae}}{2C_{D_0}} = \frac{1}{2}\sqrt{\frac{\pi Ae}{C_{D_0}}} \tag{2.46}$$

In the same way, differentiation provides the general condition for maximum C_L^3/C_D^2 as follows

$$\frac{dC_D}{dC_L} = \frac{3}{2}\frac{C_D}{C_L}$$

Using parabolic lift-drag polar we obtain

$$\frac{2C_L}{\pi Ae} = \frac{3}{2}\left[\frac{C_{D_0} + C_L^2/(\pi Ae)}{C_L}\right] \quad \text{or} \quad C_L = \sqrt{3C_{D_0}\pi Ae}.$$

Substituting the equation above into 2.44 gives

$$C_D = 4C_{D_0}, \quad \text{and}$$

$$\left(\frac{C_L^3}{C_D^2}\right)_{max} = \frac{3C_{D_0}\pi Ae\sqrt{3C_{D_0}\pi Ae}}{16C_{D_0}^2} = \frac{3\sqrt{3}}{16}\pi Ae\sqrt{\frac{\pi Ae}{C_{D_0}}}. \tag{2.47}$$

For a parabolic lift drag polar, the following can be found

$$C_L = \sqrt{\frac{1}{3}C_{D_0}\pi Ae},$$

$$C_D = \frac{4}{3}C_{D_0}, \quad \text{and}$$

$$\left(\frac{C_L}{C_D^2}\right)_{\max} = \frac{\sqrt{\frac{1}{3}C_{D_0}\pi Ae}}{\frac{16}{9}C_{D_0}^2} = \frac{3\sqrt{3}}{16}\pi Ae\sqrt{\frac{\pi Ae}{C_{D_0}^3}}.$$

Thus it is to be noted that an important aerodynamic quantity of an aircraft is the overall lift-to-drag ratio, $(C_L/C_D)_{\max}$.

2.4 Drag

2.4.1 Types of drag

Estimating the drag of a whole airplane, even for the simplest designs, is a difficult and time-consuming task. A list of definitions for different types of drag reveals part of the basis behind this:

Induced Drag, the drag caused by a trailing vortex system forming downstream of a lifting surface with a limited aspect ratio.

Parasite Drag, the difference between the total drag and the induced drag of an airplane. As a result, it is the drag that is not directly related to the creation of lift. The parasite drag is made up of several drag components, each of which has its own meaning.

Skin Friction Drag, the drag caused by viscous shearing stress on a body's wetted surface. Skin friction drag C_f is expressed by:

$$C_f = \frac{D}{qS_w}, \tag{2.48}$$

where S_w is the wetted surface area that is exposed to the flow.

Form Drag (Pressure Drag), the drag on a body caused by the combined action of static pressure operating normal to its surface, which is resolved in the drag direction.

Interference Drag, the increase in drag that occurs when two bodies are brought close together. The total drag of a wing-fuselage combination, for example, is generally higher than the sum of the wing and fuselage drag when they are considered separately.

Trim Drag, the increase in drag caused by the aerodynamic forces necessary to trim the plane about its center of gravity. This usually manifests itself as increased induced and form drag on the horizontal tail.

Profile Drag, for a two-dimensional airfoil section, it's typically interpreted to mean the sum of skin friction drag and form drag.

Base Drag, a pressure drag attributable to the blunt after-end of a body is a pressure drag due to the blunt after-end of a body.

Wave Drag, this drag is a pressure drag caused by non-canceling static pressure components on each side of a shock wave acting on the surface of the body from which the wave is coming. It is confined to supersonic flow.

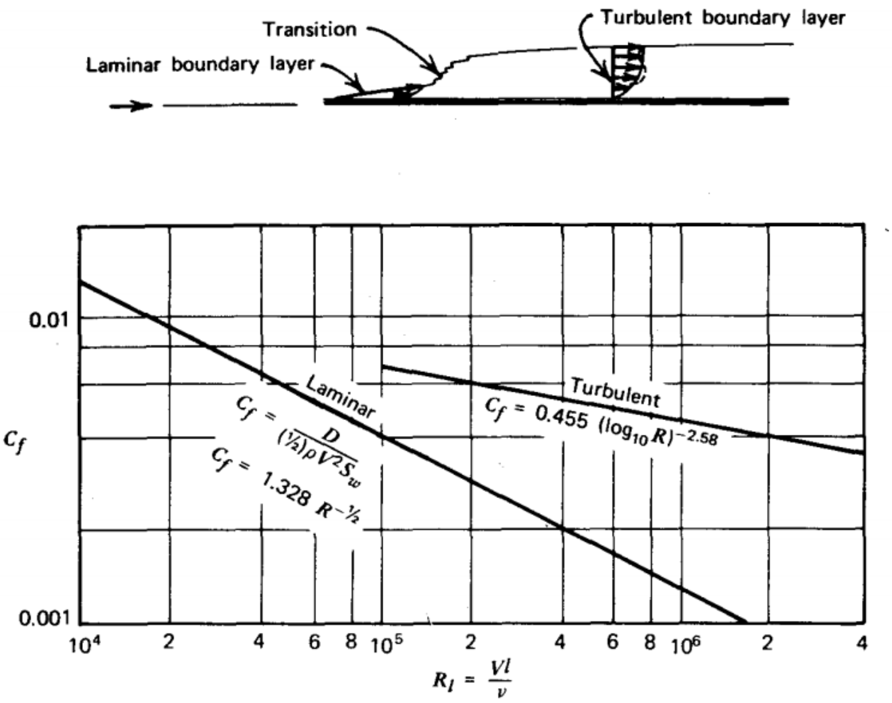


FIGURE 2.15: Drag of a thin flat plate (McCormick, 1994)

2.4.2 Form Drag

A body suffers some type of drag in addition to skin friction drag. Unlike skin friction drag, which is caused by viscous shearing forces tangential to a body's surface, form drag is caused by pressure distribution normal to the body surface. Figure 2.16 depicts the extreme example of a flat plate parallel to the flow. The drag is entirely due to an imbalance in the normal pressure distribution in this scenario. In this instance, there is no skin friction drag.

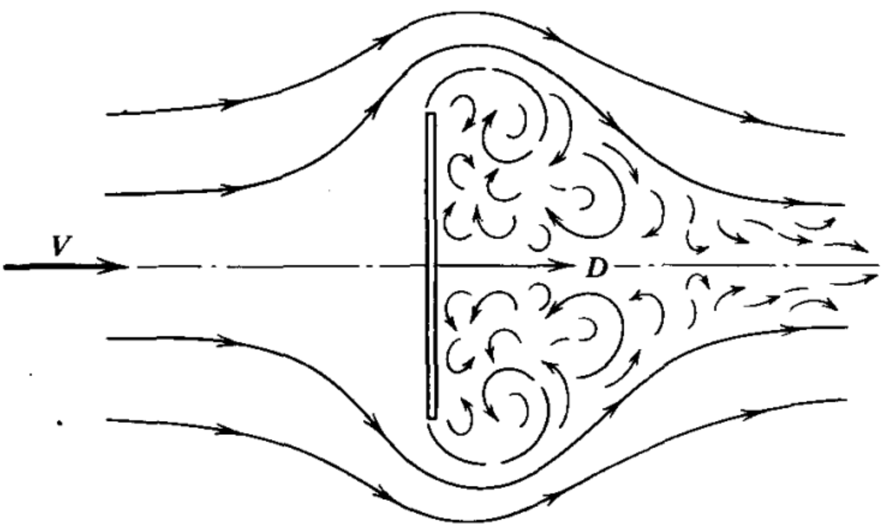


FIGURE 2.16: Flat plate normal to flow (McCormick, 1994)

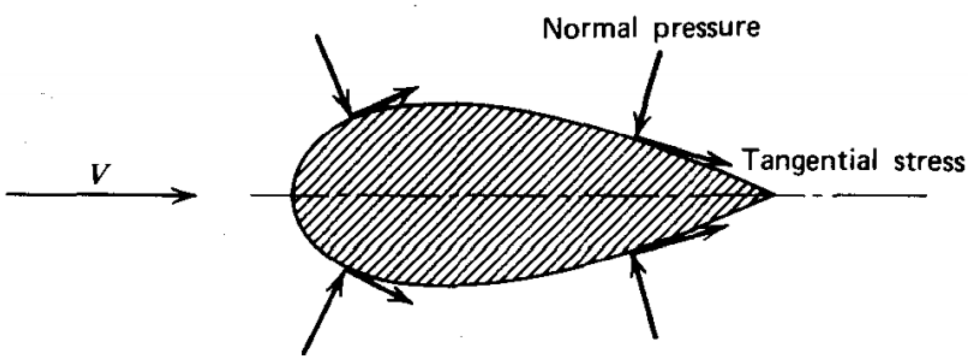


FIGURE 2.17: Body having both skin friction and form drag (McCormick, 1994)

Form drag is notoriously difficult to predict. Except in the simplest instances, skin friction drag is also a problem. As a result, in most instances, such as the one depicted in Figure 2.17, where the total drag is caused by both normal and tangential strains (or pressures), experimental data must be used to estimate the drag.

Form drag, like skin friction drag, is mostly determined by Reynolds number. Consider the flow around the circular cylinder in Figure 2.18 to see why. Figure 2.18a shows a flow with a low Reynolds number. A laminar boundary layer forms here, starting at the stagnation point. The static pressure (normal) on the cylinder's surface is greatest at the stagnation point and lowest at the top and bottom. Moving farther from these locations, the static pressure rises, eventually reaching the stagnation pressure at the very end. The normal pressure distribution would be symmetrical in the absence of viscosity and there would be no drag. D'Alembert's paradox asserts that a body in an inviscid fluid will suffer no drag. As the slower moving fluid in the laminar boundary layer goes past the cylinder's minimum pressure point, its velocity is inadequate to overcome the positive pressure gradient, known as an unfavorable gradient, and the flow splits just past the top and bottom locations. The static pressure in the separated area, which covers the majority of the back section of the cylinder, is constant and equal to the low pressure at the top and bottom. As a result of the high pressure pushing on the front and the low pressure acting on the back, there is a lot of form drag.

Figure 2.18b depicts the high-Reynolds number situation. Before splitting, the laminar boundary layer transitions to a turbulent boundary layer. The ensuing turbulent mixing increases the momentum and energy of the boundary layer, causing it to stick to the back of the cylinder, much past the laminar layer's separation point. As a result, the separation zone is significantly smaller in this case, and the static pressure on the back of the cylinder is much larger than in the laminar case. As a result of the reduced form drag, a cylinder's drag coefficient is lower with higher Reynolds numbers.

Figure 2.19 shows C_d as a function of Reynolds number for both spheres and two-dimensional circular cylinders. The predicted frontal area is used to calculate C_d. Above an R value of around 2×105, notice the fast drop in C_d.This is the so-called critical Reynolds number, where: The transition point and the separation

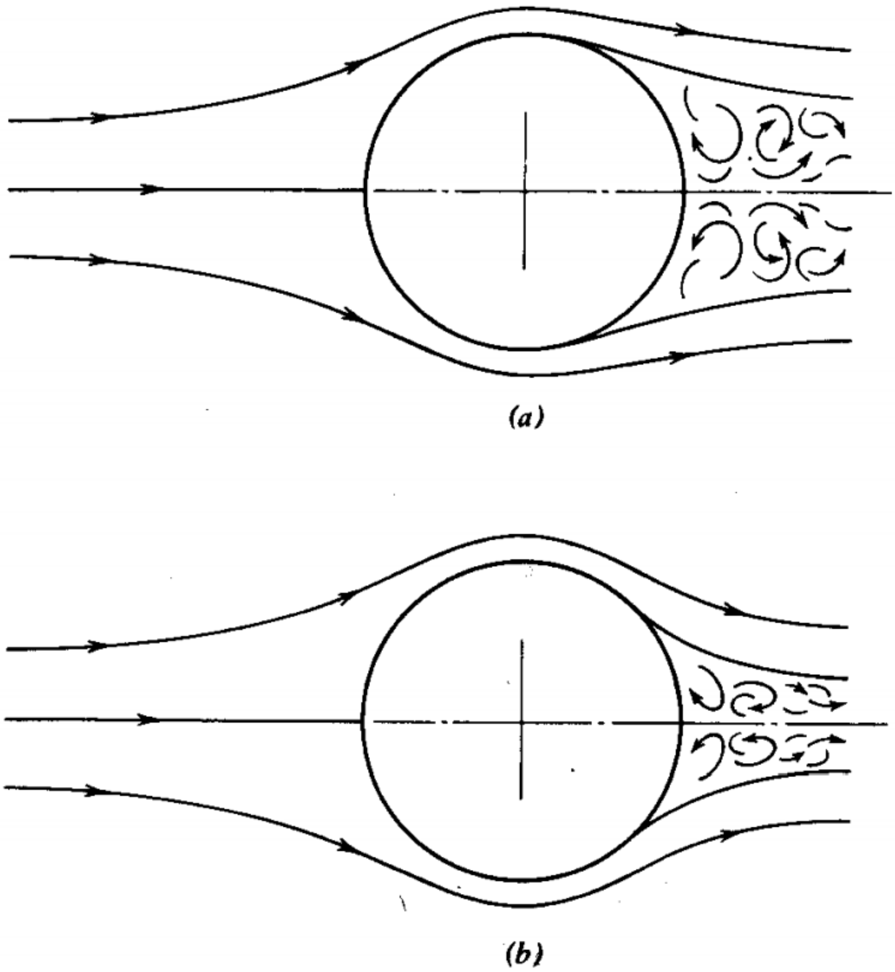


FIGURE 2.18: Flow over a circular cylinder. (a) Low Reynolds number. (b) High Reynolds number. (McCormick, 1994)

point are virtually identical. Flow at Reynolds numbers less than critical is referred to as "subcritical," whereas flow at R values more than critical is referred to as "supercritical." A body form with a clearly-defined separation point, as well as streamlined shapes, will not have a crucial Reynolds number.

Figure 2.19a includes the amount fD/v, commonly known as the Strouhal number, S_t, which describes an intriguing behavior of bluff bodies with rounded trailing edges but is unrelated to drag. As a result, as a body moves through a fluid, the vorticity in the boundary layer is shed symmetrically from the upper and bottom surfaces, forming two opposing rotational vortices. However, the symmetrical placement of the vortex pair is unstable, so that succeeding vortices are then shed alternately from the upper and lower surfaces. The resulting flow patter of periodically spaced vortices downstream of the body is known as a Karman vortex street.

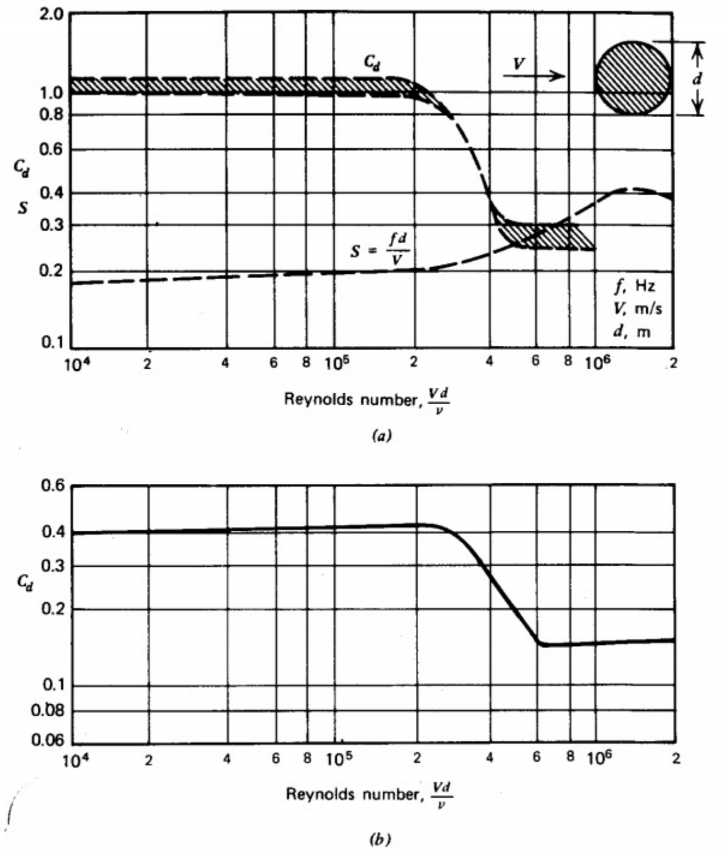


FIGURE 2.19: Drag coefficients of cylinders and spheres vs Reynolds number. (a) Two-dimensional circular cylinder. (b) Spheres. (McCormick, 1994)

The frequency at which the vortices are shed is defined as f in the formulation of the Strouhal number. When a vortex is ejected from one of the cylinder's surfaces, it causes a brief circulation around the cylinder in the opposite direction as the vortex. A force on the cylinder normal to V arises from the Kutta-Joukowski law. The forces switch direction when the next vortex is shed, producing in an alternating force on the cylinder. The "singing" of telephone lines in the wind is caused by this exact phenomena.

Consider the extreme example of form drag shown in Figure 2.16, when the flow separation point is strongly defined and unaffected by Reynolds number. The fact that drag coefficients for such geometries are virtually constant throughout a large range of Reynolds number values is not unexpected. Figure 2.20 depicts a variety of similar forms.

Values for two-dimensional and three-dimensional forms are shown in this diagram. All bodies of revolution are three-dimensional shapes. Take note of the fact that for the same profile form,

$$\frac{C_{d_{2D}}}{C_{d_{3D}}} \approx 1.8 \tag{2.49}$$

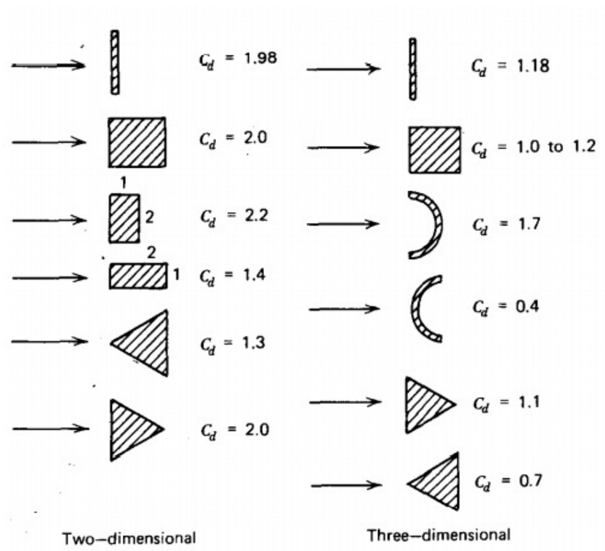


FIGURE 2.20: Examples of shapes having Cd values nearly independent of Reynolds number. (McCormick, 1994)

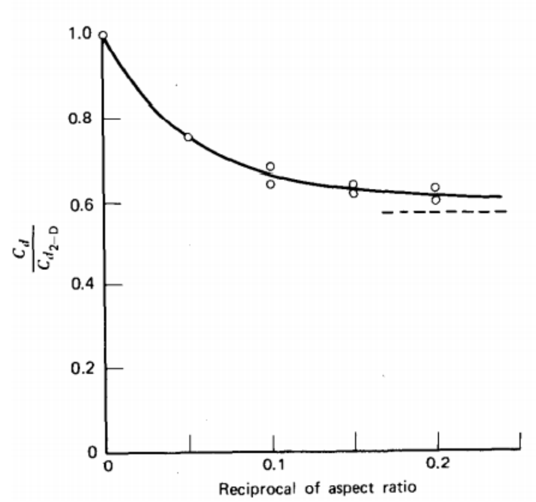


FIGURE 2.21: Transition form three-dimensional to two-dimensional drag for cylinders at supercritical Reynolds number. (McCormick, 1994)

C_d is almost constant and equal to the 3D value if the ratio of the span to the height (or diameter) of a flat plate (or cylinder) normal to the flow is less than 5. For aspect ratios larger than 5, C_d changes roughly in the way seen in Figure 2.21 normalized curve.

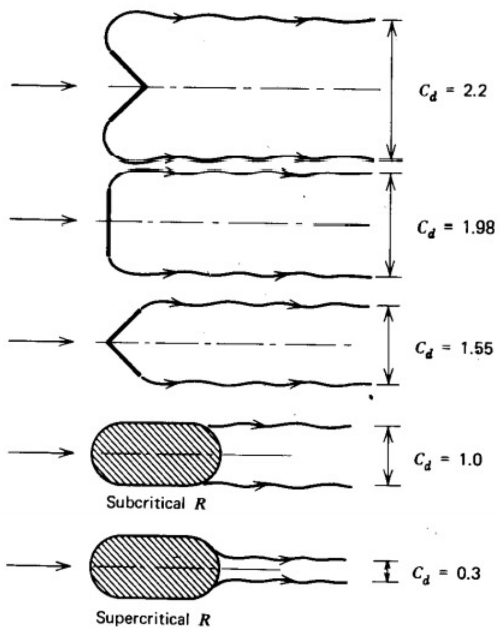


FIGURE 2.22: Qualitative estimate of drag for two-dimensional shapes. (McCormick, 1994)

Some "informed intuition" can be used to make a qualitative assessment of the drag coefficient for a given shape. The drag jected frontal regions, as shown in Figure 2.22, become

$$\frac{C_{d_{2D}}}{C_{d_{3D}}} = \frac{S_w(3D)}{S_w(2D)} \frac{4}{\pi D} \tag{2.50}$$

where D is the maximum three-dimensional body diameter or the maximum thickness of the two-dimensional shape. For an elliptical two-dimensional shape compared to an ellipsoid, this becomes

$$\frac{C_{d_{2D}}}{C_{d_{3D}}} = \frac{\pi}{2} \tag{2.51}$$

For the minimum profile drag coefficients for NACA four- and five-digit airfoils as a function of thickness ratio at a Reynolds number of 6×10^6. C_d is based on the chord length, as is customary for airfoils. Airfoils with varied camber ratios produce several data points for each thickness ratio. It's worth noting that $C_{d_{min}}$ doesn't change much with camber. $C_{d_{min}}$ appears to change almost linearly with t/c and extrapolates to 0.004 for a t/c of zero. A C_f value of 0.002 corresponds to this. According to Figure 2.15, this would necessitate more extensive laminar flow across these portions than one may assume. Transition is most likely postponed until about the 25% chord point, where maximum thickness is found. A $C_{d_{min}}$ value of 0.005 would therefore be expected.

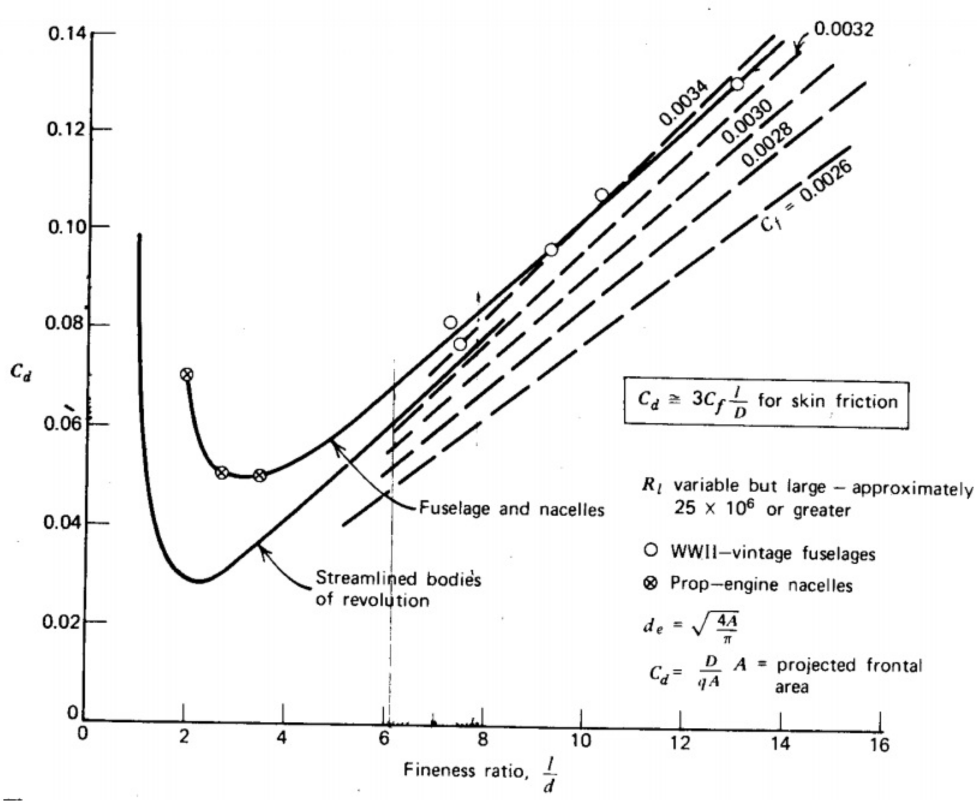


FIGURE 2.23: Drag of fuselages and similar shapes. (McCormick, 1994)

Figure 2.23 shows three-dimensional drag data. Figure 2.23 shows data from real-world fuselage and nacelle construction, as well as C_d findings from torpedo-shaped bodies. Expected C_d values for various values of C_f are also given in the figure, assuming a fair connection between the frontal and wetted regions of such entities. As the fineness ratio increases, the experimental findings should approach one of these lines for a given C_f value.

C_f for a flat plate in completely turbulent flow at a R of 25×10^6 would be 0.0026, but the data appears to be closer to a C of 0.0032 to 0.0034. Surface roughness is likely to blame for the increased skin friction drag on the bodies.

It's fascinating to look at Figure 2.23 data in terms of minimal drag for a particular body volume. This is especially essential for underwater and airship applications. It's also relevant to tip tank design, where the lowest possible drag for a given volume of gasoline is desired. We will define another drag coefficient by denoting the volume by V_m.

$$C_{d_v} = \frac{D}{qV_m^{2/3}} \tag{2.52}$$

C_{d_v} is related to C_d in Figure 2.23 by

$$C_{d_v} = \frac{A}{V_m^{2/3}} C_d \tag{2.53}$$

The ratio of the frontal area, A, to the 2/3 power of the volume, obviously, varies depending on the body shape. We will suppose the body is made up of a hemispherical nose, a cylindrical midbody that extends to the body's midsection, and a tail cone. For this particular shape,

$$\frac{A}{V_m^{2/3}} = \left[\frac{9\pi}{\left(4\frac{l}{d} - 1\right)} \right] \tag{2.54}$$

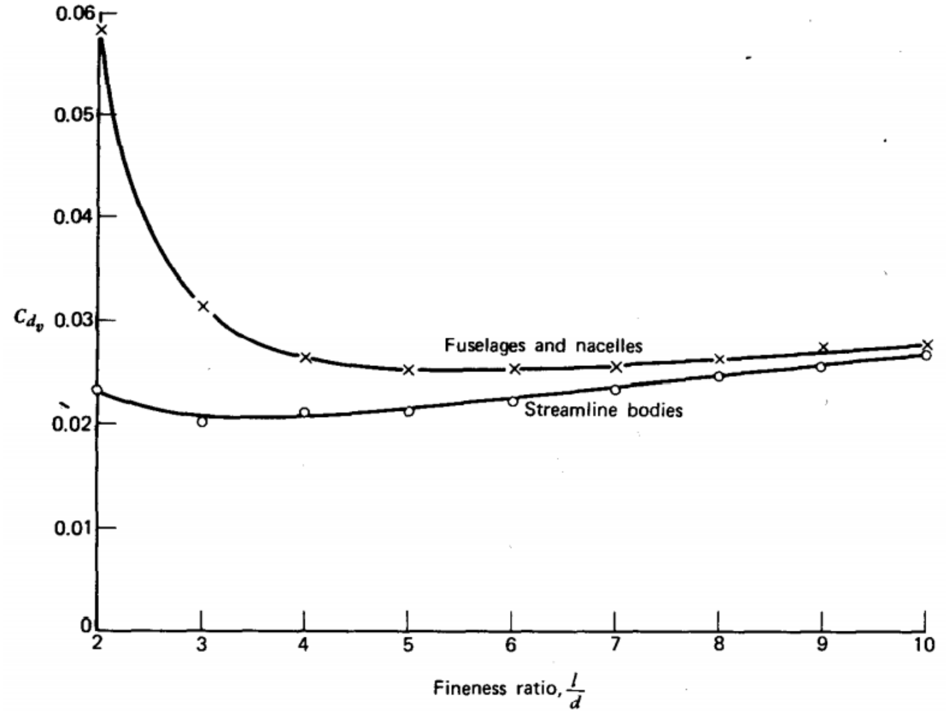


FIGURE 2.24: Drag coefficients based on volume for bodies as a function of fineness ratio. (McCormick, 1994)

The graphs in Figure 2.24 were created using this connection and Figure 2.23. This figure shows that the finesse ratio for enclosing a certain volume with a minimal drag body should be greater than the optimal values in Figure 2.23. Indeed, for l/d values of 4 to 10, the drag for a given volume is virtually constant for fuselages.

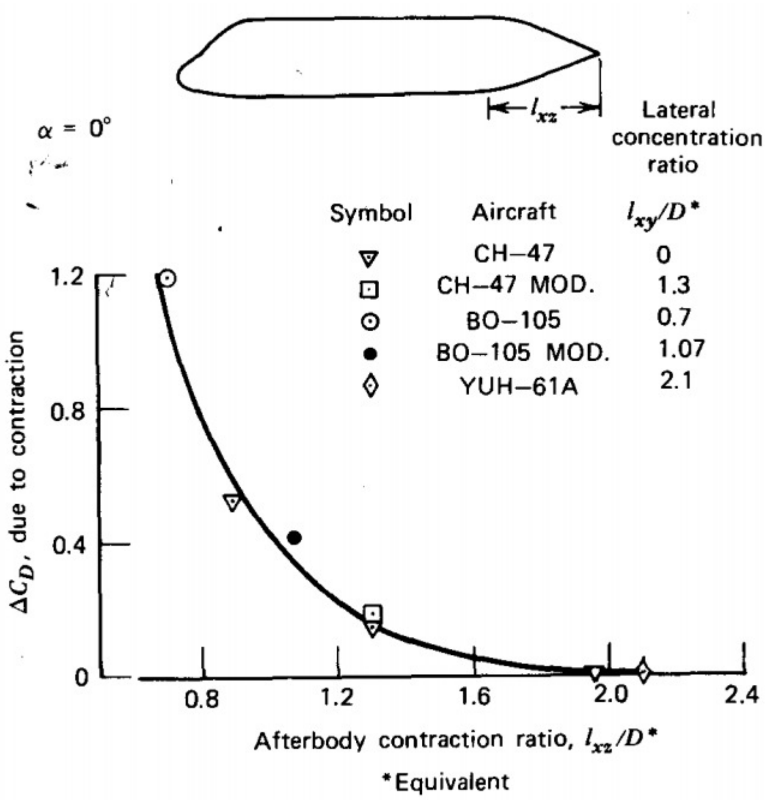


FIGURE 2.25: Effect of afterbody contraction ratio on drag. (McCormick, 1994)

For some purposes, keeping the rear part of a fuselage as broad and bluff as feasible without incurring too much drag is advantageous. Flow separation will occur over the rear if the afterbody is tapered too abruptly, resulting in excessive form drag. Figure 2.25 provides some help in this area. The increase in C_d (based on frontal area) caused by afterbody contraction is plotted as a function of afterbody shape in this figure. The ratio of the afterbody length to the corresponding diameter should be no less than 2.0, according to this diagram.

Figure 2.26, which is drawn to scale, visually illustrates the necessity of streamlining. The ratio of C_d for a circular cylinder to a two-dimensional streamlined form with a fineness ratio of 4 roughly 7.5 (supercritical flow). For the same drag, the height of the streamlied form can be 7.5 times larger than the circular cylinder, as illustrated in Figure 2.26. The contrast gets much more remarkable for subcritical flow, with the ratio climbing to around 25.

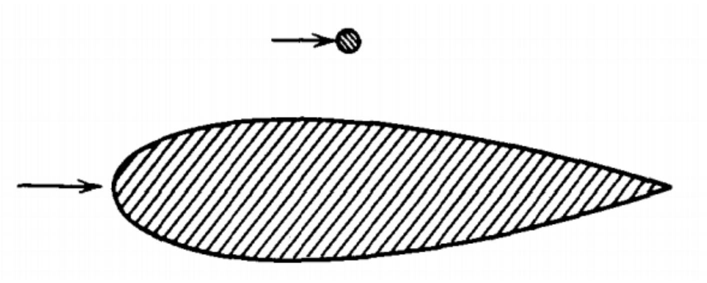


FIGURE 2.26: Two bodies having the same drag (supercritical flow). (McCormick, 1994)

2.4.3 Interference drag

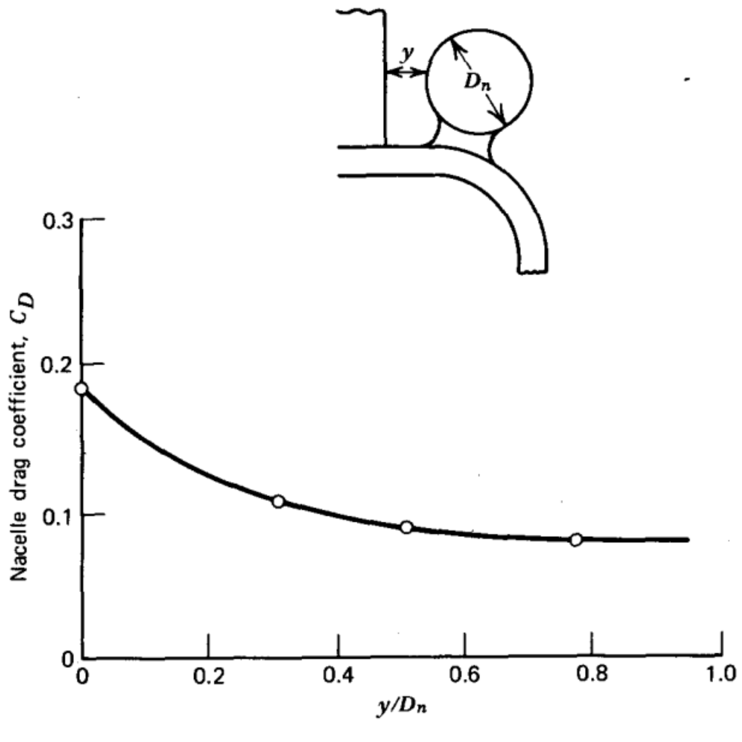


FIGURE 2.27: Effect of nacelle location on interference drag. (McCormick, 1994)

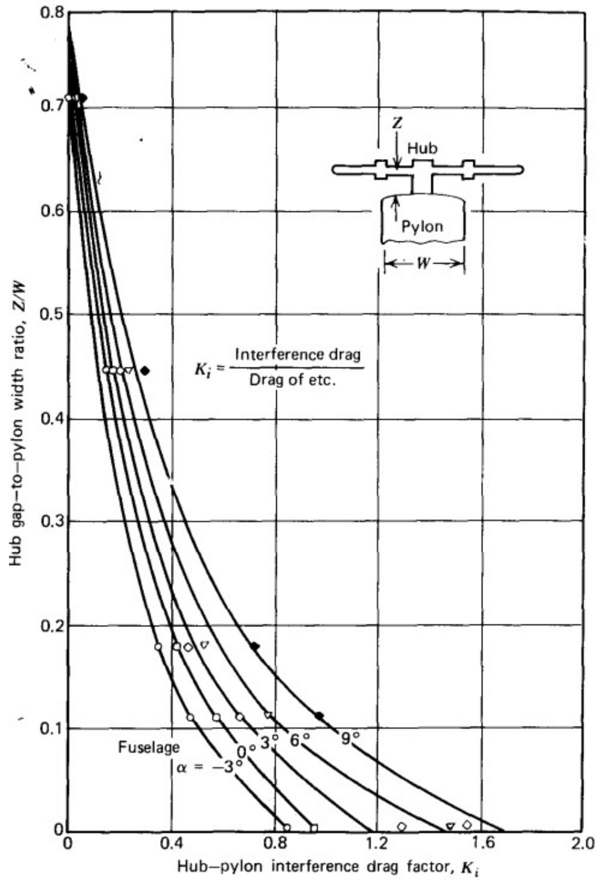


Figure 2.28: Effect of hub pylon gap on interference drag. (McCormick, 1994)

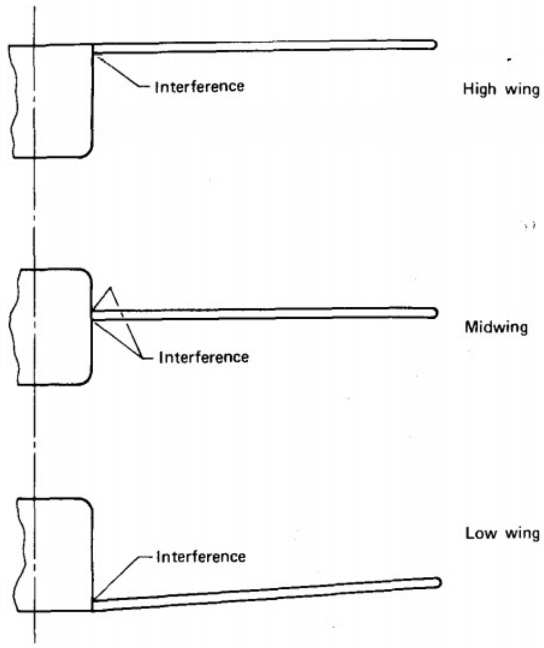


FIGURE 2.29: Wing-fuselage interference drag. (McCormick, 1994)

When the pressure distributions and boundary layers of two forms cross or are positioned close together, they can interact, resulting in a net drag of the combination that is greater than the total of the individual drags. Interference drag is the term for this increase in drag. Interference drag is difficult to predict correctly unless there are particular situations when data is available. Figures 2.27, 2.28, and 2.29 show several examples of interference drag.

On a tandem helicopter, the drag penalty for placing an engine nacelle close to a rear pylon is shown in Figure 2.27. (like a CH-47). Because the nacelle is placed so close to the pylon, the interference drag is approximately equivalent to the drag of the nacelle alone in this case. The interference drag disappears when the distance is more than one-half of a nacelle diameter.

The interference drag between the rotor hub and the pylon for a helicopter is shown in Figure 2.28. The patterns in this graph are similar to those in the preceding graph. In both cases, the additional interference drag is most likely on the pylon rather than the attached component.

A wing abuts the side of a fuselage in Figure 2.29. If the boundary layers from the two airplanes meet at an angle different than 90°, a drag increase occurs at the fuselage-wing juncture. Acute angles between intersecting surfaces, in particular, should be avoided. For example, when the angle drops from 90° to about 60°, the interference drag of a 45 percent thick strut abutting a flat wall doubles. Filleting should be utilized at the junction if sharp angles cannot be avoided.

Interference drag can be advantageous in some situations, such as when one body functions in the wake of another. In the technique of "drafting," race car drivers regularly exploit this to their advantage. Figure 2.30, based on data acquired in Pennsylvania State University's subsonic wind tunnel, shows some evidence of this beneficial interference. The drag on one rectangular cylinder in conjunction with another is plotted as a function of the distance between them. The fineness ratio of the cylinders is 2:1. The long side of the test was oriented both with and against the free-stream velocity. The drag is measured in terms of D_∞, which is the drag on a single cylinder. With respect to the diameter of the cylinder closest to the flow, the gap is rendered dimensionless. When the cylinder on which the drag is measured is downstream of the other, the spacing, x, is positive. The Cylinder's drag is greatly decreased for positive x values, and it even turns negative for tiny positive x values. The drag is somewhat enhanced for tiny negative x values.

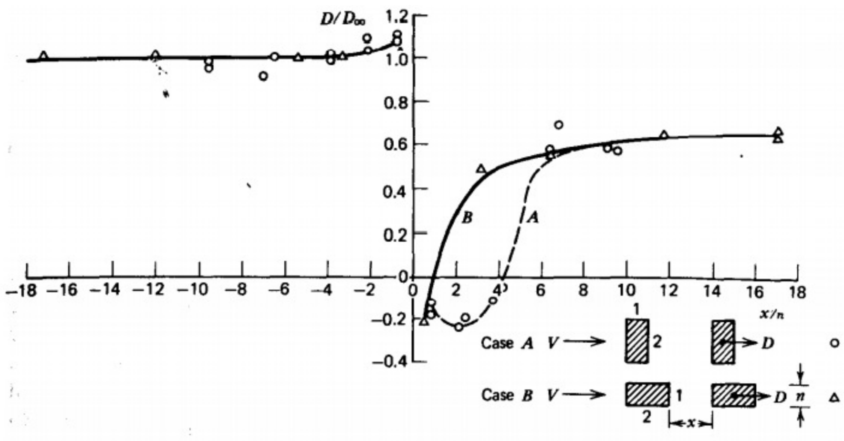


FIGURE 2.30: Interference drag for a two-dimensional rectangular cylinder in tandem with another. (McCormick, 1994)

2.4.4 Drag polar prediction methods

This method is catered for airplanes with straight, tapered wings. Total airplane drag in lbs can be written as:

$$D_{tot} = C_{D_{tot}} \bar{q} S \tag{2.55}$$

where:

- $C_{D_{tot}}$ = total aircraft drag coefficient,
- $\bar{q} = 0.5p(V_1)^2 = 14826M_a^2$, or also known as free stream dynamic pressure.
- δ = pressure ratio

Generally, the total airplane drag can be broken into the following components, (different aircraft can have more or less components):

$$\begin{aligned} C_{D_{tot}} = C_{D_{wing}} + C_{D_{fus}} + C_{D_{emp}} + C_{D_{np}} + C_{D_{flap}} + C_{D_{gear}} + C_{D_{cw}} + \\ C_{D_{store}} + C_{D_{store}} + C_{D_{trim}} + C_{D_{int}} + C_{D_{misc}} \end{aligned} \tag{2.56}$$

where:

- $C_{D_{wing}}$ is wing drag coefficient,
- $C_{D_{fus}}$ is fuselage drag coefficient,
- $C_{D_{emp}}$ is empennage drag coefficient,
- $C_{D_{np}}$ is nacelle/pylon drag coefficient,
- $C_{D_{flap}}$ is leading/trailing edge flap drag coefficient,
- $C_{D_{gear}}$ is landing gear drag coefficient,
- $C_{D_{cw}}$ is canopy/windshield drag coefficient,
- $C_{D_{store}}$ is store(s) drag coefficient,
- $C_{D_{trim}}$ is trim drag coefficient,
- $C_{D_{int}}$ is interference drag coefficient
- $C_{D_{misc}}$ is for miscellaneous drag coefficient such as antennas, struts, speed brake, etc.

2.5 Incompressible Flow over Finite Wings

2.5.1 Downwash and induced drag

Figure 2.31 shows the front and top views of a finite wing. The occurrence of high pressure on the bottom surface and low pressure on the top surface is the physical process for creating lift on the wing. The flow at the wingtips tends to wrap around the tips as a result of the pressure disparity, being driven from the high-pressure zone immediately under the tips to the low-pressure zone on top. The front view of Figure 2.31 shows the flow around the wingtips.

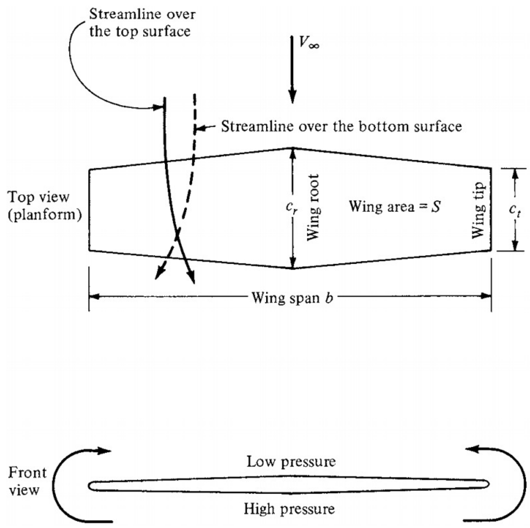


FIGURE 2.31: Finite wing (Anderson, n.d.).

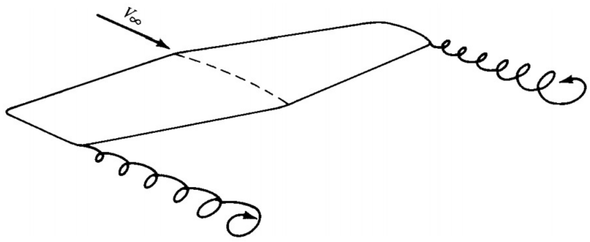


FIGURE 2.32: Schematic of wing-tip vortices (Anderson, n.d.).

Another key influence on the wing's aerodynamics is the tendency for the airflow to "leak" near the wingtips. This flow creates a trailing vortex at each wingtip, which creates a circulatory motion that trails downstream of the wing. Figure 2.32 shows these wing-tip vortices. (These tip vortices can be powerful enough for huge jets like the Boeing 747 to force small planes following too closely to lose control. Such mishaps have happened, which is one of the reasons for significant gaps between planes landing or taking off at airports.) These downstream wing-tip vortices cause a minor downward component of air velocity in the wing's vicinity. Downwash is the term for this downward component, which is represented by the letter w. As a result of the downwash interacting with the freestream velocity V_∞, a regional relative wind is formed that is canted downward in the area of each wing's airfoil section, as shown in Figure 2.33.

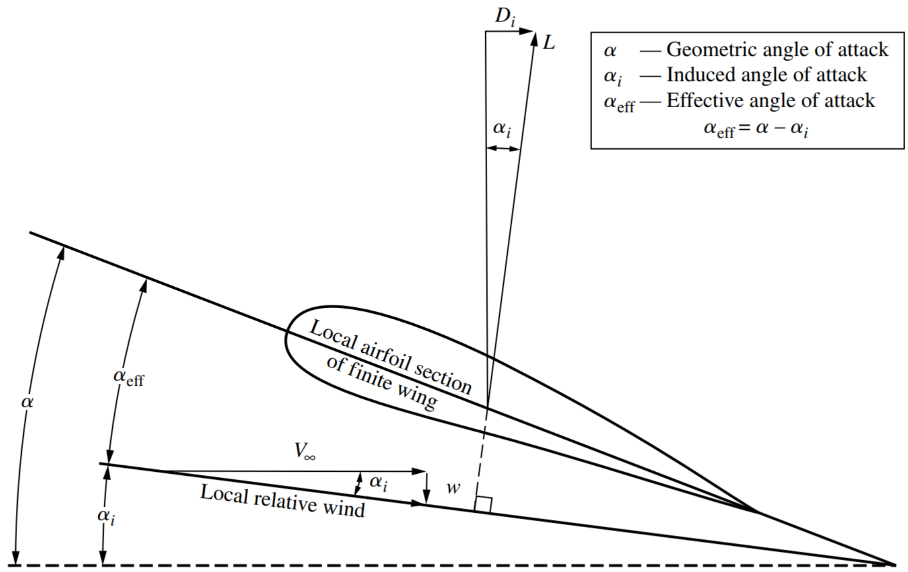


FIGURE 2.33: Effect of downwash on the local flow over a local airfoil section of a finite wing (Anderson, n.d.).

Looking at Figure 2.33, the angle of attack α is the angle formed between the chord line and the direction of V_∞. The local relative wind in Figure 2.33 is angled below the path of V_∞ by the angle α_i which is known as the induced angle of attack. The existence of downwash, and its influence on inclining the local relative airflow in a downward direction, has two key consequences on the local airfoil section, as follows:

1. The local relative wind and the chord line form the angle of attack that are perceived by the local airfoil section. This angle, indicated by the letter α_{eff} in Figure 2.32, is known as the effective angle of attack.

$$\alpha_{eff} = \alpha - \alpha_i \tag{2.57}$$

2. The local lift vector has a component in the direction of V, indicating that there is a drag generated by the presence of downwash. This is shown in Figure 2.33, which is known as Induced drag D_i.

As a result, we can observe that the existence of downwash over a finite wing generates a drag component called induced drag Di.

A finite wing's profile drag coefficient is virtually the same as its airfoil sections at a modest angle of attack. Using

$$c_d = \frac{D_f + D_p}{q_\infty S} \tag{2.58}$$

as the profile drag coefficient and

$$C_{Di} = \frac{D_i}{q_\infty S} \tag{2.59}$$

as the induced drag coefficient, resulted in

$$C_D = c_d + C_{Di} \tag{2.60}$$

as the overall drag coefficient for the finite wing.

2.6 Airplane in Symmetric Flight

2.6.1 Fundamental Equations

The X_a-axis and the Z_a-axis of the air-path axis system and the X_b-axis of the body-axis system are formed at a given point in the trajectory. The X_a-axis is tangent to the flight line, and via the X_a-axis, the Z_a-axis lies perpendicular to the local flight direction in the vertical plane. The flight state is defined by the following kinematic and geometric parameters at one moment along the trajectory:

- V is airspeed, which is the velocity vector of the airplane's center of gravity. The velocity vector coincides with the X_a-axis and lies inside the airplane's plane of symmetry.

- α_T is an angle that determines the thrust vector's inclination to the X_a-axis

- γ is the Flight-path angle, which is the angle on the horizontal plane between the Xa-axis and its projection. The angle L is positive if the airplane climbs relative to the air and negative if the airplane descends. Thus, the flight path angle ranges from $\mu/2$ to -$\mu/2$ from vertical climb to vertical dive.

- α is the angle of attack, which is the angle between the body axis system's X_b-axis and the X_a-axis. The attack angle indicates the aircraft's attitude relative to the incoming air and is positive if the X_b-axis is turned to the X_a-axis in a positive direction.

- θ is the angle of pitch, which is the angle between the horizontal and X_b-axis. According to the sign convention for the flight-path angle, angle V has a positive value if the X_b-axis is above the horizontal plane and negative if it is below that plane.

The angle of pitch, the angle of attack, and the flight path angle of symmetrical flight are correlated by

$$\theta = \alpha + \gamma. \tag{2.61}$$

The three main forces acting on an airplane that dictate its performance are also shown in Figure 2.34. These forces are as follows:

- W is the aircraft's weight, which acts downwards vertically.

- T is the aircraft's thrust, which is assumed to have an angle α_T related to the X_a-axis.

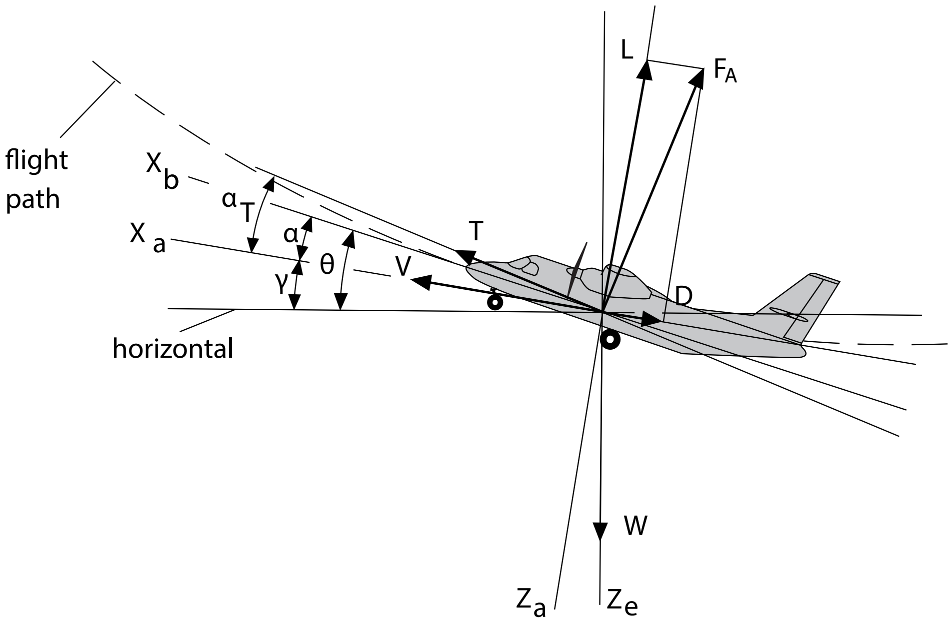


FIGURE 2.34: Airplane In symmetric flight (Ruijgrok, 2009).

- F_A is the aerodynamic force, with its following components, the lift L and drag D. The forces L and D interact, respectively, along the negative Z_a-axis and negative X_a-axis.

The application of Newton's second law of motion along the X_a-axis yields the following equation,

$$\frac{W}{g}\frac{dV}{dt} = T\cos\alpha_T - D - W\sin\gamma. \tag{2.62}$$

dV/dt is the acceleration tangent to the flight path. On the Z_a-axis there is

$$\frac{W}{g}\frac{V^2}{C_R} = T\sin\alpha_T + L - W\cos\gamma, \tag{2.63}$$

C_R is the flight path's curvature radius. With $V = C_R d\gamma/dt$ the equation above can described as

$$\frac{W}{g}\frac{V^2}{C_R} = \frac{W}{g}V\frac{d\gamma}{dt} \tag{2.64}$$

lift L can be written as

$$L = C_L q S = C_L \frac{1}{2} \rho V^2 S \tag{2.65}$$

C_L is denoted as the lift coefficient, q is the dynamic pressure which is $\frac{1}{2}\rho V^2$, and S denote the wing area. For drag D can be written as

$$D = C_D q S = C_D \frac{1}{2} \rho V^2 S \tag{2.66}$$

C_D is denoted as the drag coefficient. Thus the Equation (2.23) and (2.24) become the following

$$\frac{W}{g}\frac{dV}{dt} = T\cos\alpha_T - C_D \frac{1}{2}\rho V^2 S - W\sin\gamma, \tag{2.67}$$

$$\frac{W}{g}\frac{V^2}{C_R} = \frac{W}{g} V \frac{d\gamma}{dt} = T\sin\alpha_T + C_L \frac{1}{2}\rho V^2 S - W\cos\gamma, \tag{2.68}$$

During the flight, the weight of the aircraft reduces continuously due to fuel consumption by the engine(s), thus we have

$$F = -\frac{dW}{dt} \tag{2.69}$$

where F is the fuel weight flow rate

The variance of true altitude per unit time in the absence of wind is the aircraft's rate of climb RC, which is equal to the vertical part of airspeed V,

$$RC = \frac{dh}{dt} = V\sin\gamma \tag{2.70}$$

When the aircraft is ascending, the rate of climb will be positive, and when the aircraft descent, negative.

2.6.2 Point Performance

Point performance is a study that analyses the conditions of performance at a given point in time or a given point in the direction of flight. Parameters such as maximum speed V_max and minimum speed V_min, the minimum radius of turn, rate of climb, and maximum climb angle are part of point performance.

2.6.3 Air Loads

Air loads are a type of force applied to the aircraft due to motion action controlled by the pilot, such as maneuvering loads or by gust loads that cause turbulence.
All forces are in equilibrium when a symmetric flight is sustained at unchanged airspeed and altitude. The flight-path angle γ at Equation (2.67) and (2.68) is zero for this condition, thus

$$0 = T\cos\alpha_T - C_D\frac{1}{2}\rho V^2 S \tag{2.71}$$

$$T\sin\alpha_T + C_L\frac{1}{2}\rho V^2 S - W \tag{2.72}$$

The thrust inclination in most aircraft is very minimal, thus in most cases we can neglect it. Since we can assume $\cos\alpha_T$ is 1 and $\sin\alpha_T$ is 0, then the Equations above can become

$$T = D = C_D\frac{1}{2}\rho V^2 S = C_D\frac{1}{2}\rho_0 V_e^2 S \tag{2.73}$$

$$W = L = C_L\frac{1}{2}\rho V^2 S = C_L\frac{1}{2}\rho_0 V_e^2 S \tag{2.74}$$

- V_e is the (E.A.S.) airspeed
- ρ_0 is the density at sea-level

When the aircraft maneuver from quasi-steady-state and cause the flight path to curve, the lift and drag value will increase and can be conveyed as the following

$$L' = L + \Delta L = (C_L + \Delta C_L)\frac{1}{2}\rho V^2 S \tag{2.75}$$

$$D' = D + \Delta D = (C_D + \Delta C_D)\frac{1}{2}\rho V^2 S \tag{2.76}$$

$load\,factor$ n is used when describing the air load. $load\,factor$ is a parameter that refers to the ratio of the resultant aerodynamic force to the airplane's weight.

$$n = \frac{|A|}{W} \tag{2.77}$$

where A is the vector sum of F_A (Aerodynamic Force) and T (Thrust).
Since the lift increase will be of paramount significance and thus the loads from normal accelerations to the X_a-axis will be of primary concern. Thus in its usual form, the $load\,factor$ can be obtained by the following

$$n = \frac{L'}{W} = 1 + \frac{\Delta C_L}{C_L}. \tag{2.78}$$

For the Z_a-axis, the equation of motion is

$$\frac{W}{g}a_n = L' - W, \tag{2.79}$$

here, a_n is the centripetal acceleration, or the acceleration towards the curvature center.

$$a_n = g(n - 1). \tag{2.80}$$

The variable a_n is zero in quasi-steady level symmetric flight, meaning that the lift is equal to the airplane's weight and that the load factor is equal to one.
If the maximum amount of lift is produced, the greatest maneuvering load factors

will happen. In other words, if the angle of attack is elevated to the critical angle of attack, the maximum lift will be generated. Thus

$$n = \frac{C_{Lmax}\frac{1}{2}\rho V^2 S}{W} \tag{2.81}$$

The C_{Lmax} is actually related to the minimum stalling speed V_{MS} which will be explained later, thus

$$W = C_{Lmax}\frac{1}{2}\rho V_{MS}^2 S. \tag{2.82}$$

Then from Equation (2.81) and (2.82) can obtain n as follows

$$n = \frac{V^2}{V^2{}_{MS}} \tag{2.83}$$

The obtainable $load\, factor\ n$ increases strongly with the increasing airspeed as shown above.

2.6.4 Stalling Speed

The minimum stalling speed is the minimum speed achievable where the aircraft is performing a prescribed stalling maneuver at a certain given configuration.
The load factor first remains approximately constant when reaching the stall, and the lift coefficient steadily increases as shown in the following equation

$$\frac{W}{g}a_n = C_L\frac{1}{2}\rho V^2 S - W \tag{2.84}$$

$$C_L = \frac{nW}{\frac{1}{2}\rho V^2 S} \tag{2.85}$$

When a stall occurs, the lift coefficient goes beyond the maximum lift coefficient; in this state, the load factors drop tremendously. To recover from a stall, the pilot requires maneuvering the aircraft so that the nose is pitching down to increase

airspeed, which will recover lift.
The minimum stalling speed changes with different configurations, such as flap and landing gear configuration, at a certain point in time. It is also possible to subtract the V_{MS} from the following measurement data

$$V_S = \sqrt{\frac{W}{S}\frac{2}{\rho}\frac{1}{C_{Lmax}}} \tag{2.86}$$

With lift equal to drag, L = W, the lift coefficient obtained by

$$C_{Lmax} = \frac{W}{12\rho V^2 S} \tag{2.87}$$

2.7 Performance in Steady Symmetric Flight

2.7.1 Basic relations

From Figure 2.35, the forces parallel to the flight path is as follow

$$T\cos\alpha_T - D - W\sin\gamma = 0 \tag{2.88}$$

and for the forces acting perpendicular to the flight path we have

$$T\sin\alpha_T + L - W\sin\gamma = 0 \tag{2.89}$$

Since most aircraft, when under standard flight conditions, have a very small α_T, the α_T can be neglected, thus simplifying the previous equation into

$$T - D - W\sin\gamma = 0 \tag{2.90}$$

$$L - W\sin\gamma = 0. \tag{2.91}$$

Equation (2.90) can be multiply using airspeed V for convenience as follows

$$TV - DV - WV \sin \gamma = 0. \tag{2.92}$$

Equation 2.92 can be modified further by introducing the rate of climb RC as follows

$$RC = V \sin \gamma \tag{2.93}$$

thus

$$TV = DV + W(RC). \tag{2.94}$$

As we can see from Equation (2.94), the left-hand side of the equation TV indicates the power delivered by the power-plant of the aircraft at certain airspeed V; this variety can also be declared as P_a,

$$P_a = TV. \tag{2.95}$$

For the right-hand side of Equation (2.94) the equation DV for Drag D multiplied by the airspeed V, is the power required P_r which translate to

$$P_r = DV. \tag{2.96}$$

Excess power is the difference between the power available and the power required, thus

$$P_c = P_a - P_r \tag{2.97}$$

In other words, the equation (2.94) can be translated into

$$P_a = P_r + W(RC)$$

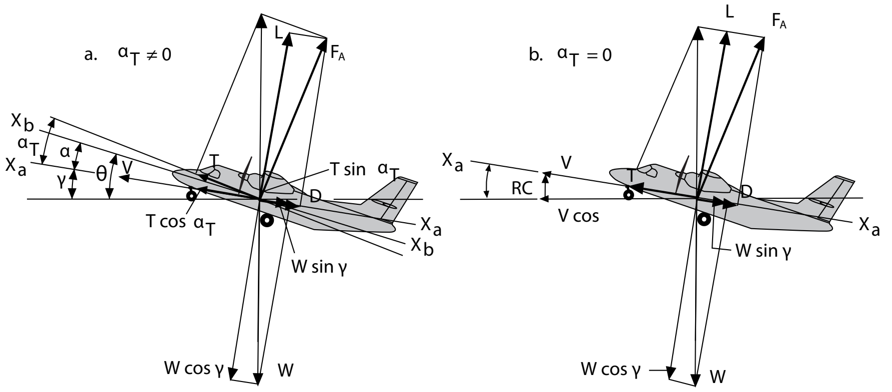


FIGURE 2.35: Airplane In steady symmetric flight (Ruijgrok, 2009).

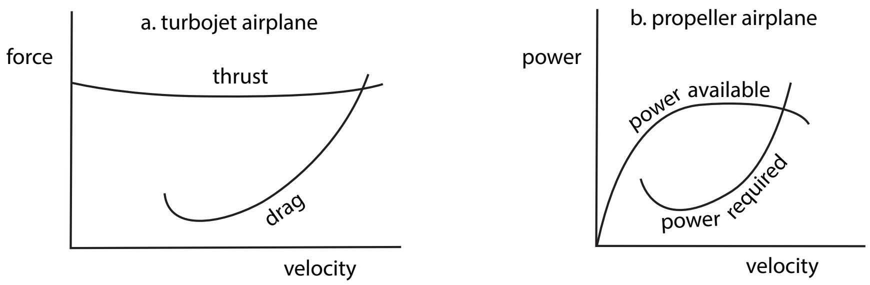


FIGURE 2.36: Performance diagram (Ruijgrok, 2009).

$$P_c = W(RC) \tag{2.98}$$

The typical performance curve types are seen as regards both force and power in Figure 2.36. When evaluating turbojet and turbofan-powered aircraft, adding together the two thrust and drag curves are valuable because their engines are rated in terms of thrust (Figure 2.36a).
Piston- and turboprop engines, on the other hand, are classified in terms of shaft power.

2.7.2 Drag and power required

Using L from Equation (2.74) and implementing it into Equation (2.91) we can obtain airspeed as

$$V = \sqrt{\frac{W}{S}\frac{2}{\rho}\frac{1}{C_L}\cos\gamma} \tag{2.99}$$

and implementing Equation (2.99) into Equation (2.73) will get D,

$$D = \frac{C_D}{C_L}W\cos\gamma \tag{2.100}$$

then implement Equation (2.99) and (2.100) into $P_r = DV$ resulted in

$$P_r = W\sqrt{\frac{W}{S}\frac{2}{\rho}\frac{C_D^2}{C_L^3}\cos^3\gamma} \tag{2.101}$$

The equations above displayed that a low-subsonic aircraft has V, D, and P_r as functions of angle of attack α and flight=path angle γ at a given altitude and weight. Most of the time, γ is neglected in performance analysis for simplification. Thus safe to assume that γ is very small and negligible or $\gamma = 0$. Thus by having $\cos\gamma = 1$, Equation (2.99),(2.100), and (2.101) becomes

$$V = \sqrt{\frac{W}{S}\frac{2}{\rho}\frac{1}{C_L}} \tag{2.102}$$

$$D = \frac{C_D}{C_L}W \tag{2.103}$$

$$P_r = W\sqrt{\frac{W}{S}\frac{2}{\rho}\frac{C_D^2}{C_L^3}} \tag{2.104}$$

Parabolic approximation of lift-drag polar can be obtained by

$$C_D = C_{D0} + \frac{C_L^2}{\pi A e} \tag{2.105}$$

- C_{D0} is denoted as the zero-lift drag coefficient
- A is denoted as the aircraft's wing aspect ratio
- e is the Oswald's efficiency factor

With the establishment of parabolic drag equation, the Equation (2.100) can be rewrite as

$$D = C_{D0}\frac{1}{2}\rho V^2 S + \frac{C_L^2}{\pi A e}\frac{1}{2}\rho V^2 S \tag{2.106}$$

Insert C_L from Equation (2.74) into Equation (2.106) yields

$$D = C_{D0}\frac{1}{2}\rho V^2 S + \frac{W^2}{\pi A e \frac{1}{2}\rho V^2 S} = D_0 + D_i \tag{2.107}$$

- D_0 is the zero-lift drag
- D_i is the induced drag

The condition $dD/dV = 0$ correspond to the minimum drag speed. We can differentiate Equation (2.107) with respect to V and set the derivative to zero to obtain

$$V_{Dmin} = \sqrt{\frac{W}{S}\frac{2}{\rho}\frac{1}{\sqrt{C_{D0}\pi A e}}}. \tag{2.108}$$

Then we insert Equation (2.108) into (2.107) to get the following

$$D_0 = D_i = W\sqrt{\frac{D_{D0}}{\pi A e}} and \tag{2.109}$$

$$D_{min} = 2W\sqrt{\frac{D_{D0}}{\pi A e}}. \tag{2.110}$$

Equation (2.110) can also be obtained by deriving Equation () and (2.103).

$$D_{min} = \frac{W}{(C_L/C_D)_{max}} = 2W\sqrt{\frac{D_{D0}}{\pi Ae}}. \tag{2.111}$$

From Equation (2.96) we have $P_r = DV$, we also have D from Equation (2.73) which result in $P_r = \frac{1}{2}\rho V^3 C_D S$, and from Equation (2.105) we have C_D which will yield $P_r = C_{D0}\frac{1}{2}\rho V^3 S + C_{D0}\frac{C_L^2}{\pi Ae}$. We also have C_L from Equation (2.74) thus we have P_r,

$$P_r = C_D\frac{1}{2}\rho V^3 S = C_{D0}\frac{1}{2}\rho V^3 S + \frac{W^2}{\pi Ae\frac{1}{2}\rho VS}. \tag{2.112}$$

The ratio $dP_r/dV = 0$ for the minimum power required. If we derivate the equation above with respect to V and equating it to zero, we can obtain

$$V_{P_{rmin}} = \sqrt{\frac{W}{S}\frac{2}{\rho}\frac{1}{\sqrt{3C_{D0}\pi Ae}}}. \tag{2.113}$$

Since we now have V for minimum power, we can find the minimum power required P_{rmin} by substituting Equation (2.113) into (2.112). Thus we have

$$P_{rmin} = \frac{4}{3}W\sqrt{\frac{W}{S}\frac{2}{\rho}\sqrt{\frac{3C_{D0}}{(\pi Ae)^3}}}. \tag{2.114}$$

If we combine Equation (2.108) with (2.113) we obtain the following

$$V_{Dmin} = \sqrt[4]{3}V_{P_rmin}. \tag{2.115}$$

Mach number can be obtained by

$$M = \frac{V}{c} = \frac{V}{\sqrt{\gamma RT}}. \tag{2.116}$$

Drag coefficient can also be obtained by

$$C_D = C_{D0} + kC_L^2, \tag{2.117}$$

where k is the drag factor $k = \frac{1}{\pi Ae}$.
Compressibility increase the drag power and power required in transonic and supersonic flight.

2.7.3 Thrust and power available

As turbojets and turbofans have their output characteristics as thrust T, the power curves that are available for these engine types are simply obtained from $P_a = TV$. The calculation process is actually more difficult for propeller-driven aircraft because the power available comes from the product of the engine's shaft brake power and propulsive efficiency.

$$P_a = \eta_j P_{br} \tag{2.118}$$

- η_j is the propulsive efficiency
- P_{br} is the shaft brake power

A shaft brake power in the engine rpm is identified from the standard power diagram for a piston engine aircraft with a propeller. The shaft brake power is known directly from chosen values, the inlet manifold pressure, and engine rpm at a given altitude.
The propulsive efficiency can be obtained by

$$\eta_j = \frac{C_T}{C_P} J \tag{2.119}$$

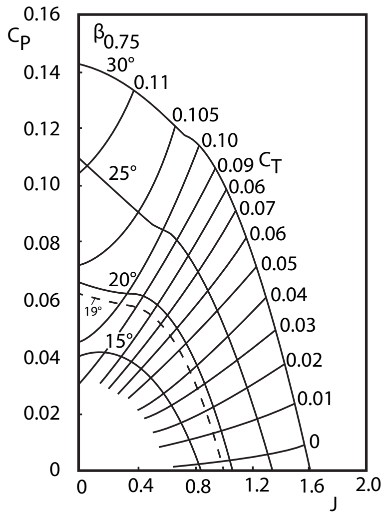


FIGURE 2.37: Propeller chart of 2-bladed propeller (estimated) (Ruijgrok, 2009).

C_P or power coefficient can be obtained by

$$C_P = \frac{P_{br}}{\rho n_p^3 D^5} \tag{2.120}$$

in condition that $P_P = P_{br}$. The advance ratio J can be calculated using

$$J = \frac{V}{n_p D}. \tag{2.121}$$

The value C_T can be determined by referring to Figure 2.37 using C_P and J

Most of the gas generator power in a turboprop is extracted by the turbine from the gas stream through the engine to drive the propeller, while a small portion is used to produce jet thrust by expanding the exhaust gases in the nozzle. The turboprop's power available is then the total power generated from the propeller with the jet thrust power. Thus for turboprop, the function for power available is

$$P_a = \eta_j P_{br} + T_j V, \tag{2.122}$$

where T_j is the jet thrust.

2.7.4 The performance diagram

The maximum forward velocity is defined by the condition that $P_c = 0$ in equation (2.97) at which the unaccelerated flight velocity can be sustained. Thus

$$P_a = P_r. \tag{2.123}$$

The rate of climb of the aircraft can be obtained by combining equation (2.97) and (2.98),

$$RC = \frac{P_a - P_r}{W}. \tag{2.124}$$

A hodograph is one of few ways to display the climbing performance curve, the plot of the climb rate against the horizontal airspeed component, $V_h = V \cos\gamma$. The slope for radius vector from the origin and intersecting the curve can be obtained as from the following ratio

$$\frac{RC}{V_h} = \frac{V \sin\gamma}{V \cos\gamma} = \tan\gamma \tag{2.125}$$

For fixed-pitch propeller aircraft, the rate of climb and the climb angle at each flight velocity would be lower due to the fact that for a reduction in engine speed with decreasing airspeed, the shaft brake power and available power is lower for a fixed-pitch propeller.

The thrust and drag curve intersection in Figure 2.38 sets the maximum airspeed for level flight,

$$T = D \tag{2.126}$$

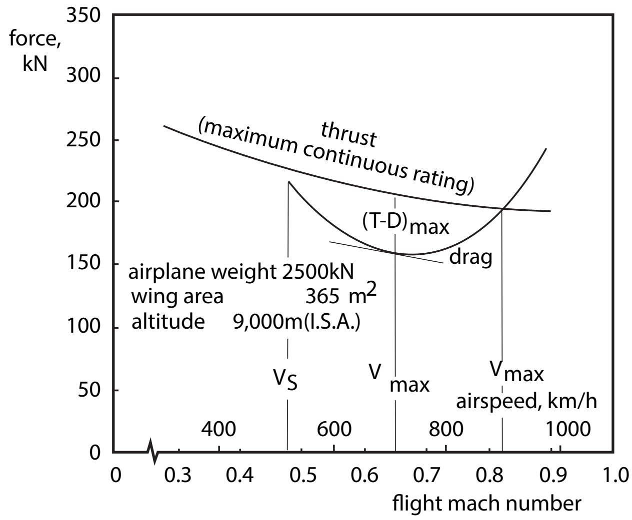


FIGURE 2.38: Performance diagram for turbofan transport airplane (Ruijgrok, 2009).

The excess thrust explicitly defines the angle of climb at velocities below Vmax, as we find from Equation (2.90), the angle of climb is directly determined.

$$\sin\gamma = \frac{T-D}{W} \tag{2.127}$$

The equation above reveals that with a maximum excess thrust, the maximum climb angle is reached.

The relationship between actual climb angle vs climb angle with $\cos\gamma = 1$ is

$$\sin\gamma = \frac{T-D}{W} = \frac{T-D_1}{W} + \frac{\Delta D_i}{W} \tag{2.128}$$

- Subscript 1 is for the case $\cos\gamma = 1$
- ΔD_i is the induced drag surplus. $\Delta D_i = \Delta C_{Di}\frac{1}{2}\rho V^2 S$

with $\Delta D_i = \Delta C_{Di}\frac{1}{2}\rho V^2 S$ and $C_{Di} = kC_L^2$, another form for Equation (2.128) is as follows

$$\sin\gamma = \sin\gamma_1 + k\sin^2\gamma\frac{W}{\frac{1}{2}\rho V^2 S} \tag{2.129}$$

By setting $\gamma = \gamma_1$ on the right-hand side of Equation (2.129) in the second term, the problem can be adequately defined so that

$$\frac{\sin\gamma}{\sin\gamma_1} = \frac{RC}{RC_1} = 1 + k\sin\gamma_1 \frac{W}{\frac{1}{2}\rho V^2 S} \tag{2.130}$$

2.7.5 Performance prediction using analytical expression

In order to achieve a full analytical view of the results curves, we must also incorporate simplistic assumptions in terms of thrust and power available curves. However, these empirical performance computation methods can only provide an evaluation of the actual performance and are particularly useful for gaining insight into the performance effects of the different parameters.
The cold air flowing through the bypass duct and the hot air going through the exhaust nozzle generates the thrust of a turbofan engine. The thrust of the turbofan usually decreases with airspeed, which behavior can be easily defined by:

$$\frac{T}{T_{static}} = 1 - k(V)^{\frac{1}{2}} \tag{2.131}$$

in this equation, k is a constant for a given bypass ratio, control setting, and altitude.

The available power for a constant speed propeller is basically constant across the airplane's speed range. Therefore, it can be concluded that the power available for propeller-driven airplanes is independent of airspeed, given that the setting and altitude of the engine control remain unchanged.

$$\frac{P_a}{P_{a0}} = (\frac{\rho}{\rho_0})^n \tag{2.132}$$

- subscript "0" is a designates sea-level condition
- in the troposphere n is less than 1

At flight velocity, the thrust of a subsonic turbojet engine is relatively constant. Therefore, it seems worthwhile to conclude that a jet-powered airplane's thrust has a constant value in the subsonic speed range.

$$\frac{T}{T_0} = (\frac{\rho}{\rho_0})^n \tag{2.133}$$

going back to the rate of climb equation, the maximum rate of climb can be obtained when the power available is unchanging,

$$RC_{max} = \frac{P_a - P_{rmin}}{W} \tag{2.134}$$

Substituting P_r from Equation (2.104) into the equation above yields

$$RC_{max} = \frac{P_a}{W} - \sqrt{\frac{W}{S}\frac{2}{\rho}\frac{1}{(C_L^3/C_D^2)_{max}}} \tag{2.135}$$

From Equations (2.127) and (2.103), we can write the following for climb angle,

$$\sin\gamma = \frac{T-D}{W} = \frac{T}{W} - \frac{C_D}{C_L}. \tag{2.136}$$

At the maximum lift drag ratio the climb angle become

$$\sin\gamma = \frac{T}{W} - \frac{1}{(C_L/C_D)_{max}}. \tag{2.137}$$

For the rate of climb, integrating Equations (2.102) and (2.136) results in the equation given,

$$RC = V\sin\gamma = \sqrt{\frac{W}{S}\frac{2}{\rho}\frac{1}{C_L}}\left[\frac{T}{W} - \frac{C_D}{C_L}\right]. \tag{2.138}$$

It is possible to achieve the optimal rate of climb by setting the first derivative of the equation above with respect to CL equal to zero,

$$\frac{dRC}{dC_L} = \frac{d}{dC_L}\left[\sqrt{\frac{W}{S}\frac{2}{\rho}\frac{1}{C_L}}\left[\frac{T}{W} - \frac{C_D}{C_L}\right]\right]. \tag{2.139}$$

We can obtain the condition for the maximum rate of climb by following through with this differentiation.

$$\frac{T}{W} = 3\frac{C_D}{C_L} - 2\frac{dC_D}{dC_L}. \tag{2.140}$$

Substituting Equation (2.105) into the equation above yields the following equation for a parabolic variation of C_D and C_L

$$\frac{T}{W} = 3\frac{C_{D0}}{C_L} - 2\frac{dC_D}{dC_L}. \tag{2.141}$$

lift coefficient for maximum rate of climb,

$$C_{LC} = \frac{\pi Ae}{w}\frac{T}{W}\left[-1 + \sqrt{1 + 12\frac{C_{D0}}{\pi Ae}\left(\frac{W}{T}\right)^2}\right] \tag{2.142}$$

The conditions for maximum level steady flight speed

$$W = C_L\frac{1}{2}\rho V^2 S \tag{2.143}$$

$$T = C_D\frac{1}{2}\rho V^2 S \tag{2.144}$$

Equation (2.107) into (2.144) V_max,

$$T = C_D\frac{1}{2}\rho V_{max}^2 S + \frac{W^2}{\pi Ae\frac{1}{2}\rho V_{max}^2 S} \tag{2.145}$$

Equation (2.145) solved into

$$V_{max} = \sqrt{\frac{T}{\rho C_{D0} S}\left[1 \pm \sqrt{4\frac{C_D 0}{\pi A e}\left(\frac{W}{T}\right)^2}\right]} \tag{2.146}$$

2.8 Cruise Performance

2.8.1 Range and endurance

The expression range is used in cruising flights for the horizontal straight line distance in which the aircraft operates. A block distance, stage length, or total range is where an aircraft travels from ascending, cruising, and descending, forming a complete cycle. In general, the total range where the aircraft operates, is limited by the the aircraft's fuel capacity. The function for fuel consumption per unit time can be seen as follow

$$F = \frac{dW_f}{dt}, \tag{2.147}$$

W_f is the total fuel load

since $dW_f = -dW$, the weight of the aircraft is affected by the fuel weight flow

$$F = -\frac{dW}{dt} \tag{2.148}$$

The aircraft's range can be obtained by the following

$$R = \int_{t_1}^{t_2} V dt = \int_{W_1}^{W_2} -\frac{V}{F} dW = \int_{W_2}^{W_1} \frac{V}{F} dW \tag{2.149}$$

- V/F is the range per unit fuel weight, or the specific range
- 1 and 2 is a subscript that refer to the start and end of the cruise, respectively, for initial and final state.

For the amount of time spent on cruising flights, the term endurance is used. The equation is written as

$$E = \int_{t_1}^{t_2} dt = \int_{W_1}^{W_2} -\frac{dW}{F} = \int_{W_2}^{W_1} \frac{dW}{F}. \tag{2.150}$$

For the equilibrium condition of propeller propulsion where power available P_a is equal to power required P_r, there is a certain value of specific fuel consumption c_p and the propulsive efficiency η_j. Thus the engine power can be obtained from the following

$$\begin{aligned} \eta_j &= \frac{P_a}{P_{br}} \\ P_{br} &= \frac{P_a}{\eta_j} \end{aligned} \tag{2.151}$$

For the fuel weight flow rates is obtained from

$$\begin{aligned} c_p &= \frac{F}{P_{br}} \\ F &= c_p P_{br} \end{aligned} \tag{2.152}$$

2.8.2 Range and endurance for propeller aircraft

Specific range and fuel weight flow rate may also be correlated to the parameters of the airplane and propulsion system by utilizing Equations 2.151 and 2.152 to get analytic formulas for range and endurance. The following can be written if we assume a quasi-level and quasi-steady flight,

$$F = c_p P_{br} = c_p \frac{P_a}{\eta_j} = c_p \frac{P_r}{\eta_j} = c_p \frac{DV}{\eta_j}. \tag{2.153}$$

Then by utilizing Equation 2.102 and 2.103 the following can be obtained,

$$\frac{V}{F} = \frac{\eta_j}{c_p}\frac{C_L}{C_D}\frac{1}{W} \tag{2.154}$$

$$F = \frac{c_p}{\eta_j} W \sqrt{\frac{W}{S}\frac{2}{\rho}\frac{C_D^2}{C_L^3}} \tag{2.155}$$

Integrating Equation 2.154 into 2.149, and Equation 2.155 into 2.150 resulted in,

$$R = \int_{W_1}^{W_2} \frac{\eta_j}{c_p}\frac{C_L}{C_D}\frac{dW}{W} \tag{2.156}$$

$$E = \int_{W_1}^{W_2} \frac{\eta_j}{c_p}\frac{dW}{W\sqrt{\frac{W}{S}\frac{2}{\rho}(C_D^2/C_L^3)}}. \tag{2.157}$$

Then,

$$R = \frac{\eta_j}{c_p}\frac{C_L}{C_D}\int_{W_1}^{W_2}\frac{dW}{W} = \int_{W_1}^{W_2}\frac{\eta_j}{c_p}\left|\ln W\right|_{W_2}^{W_1} = \frac{\eta_j}{c_p}\frac{C_L}{C_D}\frac{W_1}{W_2} \tag{2.158}$$

$$E = \frac{\eta_j}{c_p}\sqrt{\frac{C_L^3/C_D^2}{\frac{1}{S}\frac{2}{\rho}}}\int_{W_1}^{W_2}\frac{dW}{W\sqrt{W}} = \frac{\eta_j}{c_p}\sqrt{\frac{C_L^3/C_D^2}{\frac{1}{S}\frac{2}{\rho}}}\left|\frac{-2}{\sqrt{W}}\right|_{W_2}^{W_1}$$

$$E = \frac{\eta_j}{c_p}\sqrt{\frac{C_L^3/C_D^2}{\frac{1}{S}\frac{2}{\rho}}}\left[\frac{2}{\sqrt{W_2}} - \frac{2}{\sqrt{W_1}}\right]. \tag{2.159}$$

With initial airspeed of,

$$V_i = \sqrt{\frac{W_1}{S}\frac{2}{\rho}\frac{1}{C_L}}, \tag{2.160}$$

Equation 2.159 can be modified into

$$E = 2\frac{\eta_j}{c_p}\frac{C_L}{C_D}\frac{1}{V_i}\left[\sqrt{\frac{W_1}{W_2}} - 1\right]. \tag{2.161}$$

2.8.3 Range and endurance for battery powered aircraft

In the case of electric aircraft, instead of fuel, a battery provides the required power to overcome the drag. Usually, a battery pack capacity is labeled in ampere-hours Ah or milliampere-hours mAh. A 1 Ah battery would be able to supply a 1 A current continuously for 1 hour (Usually, battery capacity relies on a 1 h discharge for compact batteries or 20 h for larger installations). Thus, in a sense, it seems that a 1 Ah battery could provide a 2 A current for half an hour, but this is not the case as there is a law called Peukert's effect; thus, a 1 Ah battery if it were to supply a 2 A current, it would last less than half an hour. This indicates that a higher current draw would reduce the effective capacity of the battery. This also means that if the current were less than the battery capacity, the effective capacity would increase. Peukert's equation is as follows:

$$t = \frac{C}{i^n}, \tag{2.162}$$

where C is the capacity of the battery in Ah, i is the discharge current in amperes, t is time in hours. n is a discharge parameter that is dependent on the type and the temperature of the battery. As the battery ages, the parameter n changes such that the capacity usually diminishes. Only if the discharge of the battery is 1 A will the previous equation be valid, which often is not the case as discharge varies as needed. To account the effect of the discharge rate, the equation needs to be modified into the following

$$t = \frac{Rt}{i^n}\left(\frac{C}{Rt}\right)^n, \tag{2.163}$$

where Rt is the time (in hours) of discharge over which the capability has been defined, or simply battery hour rating. The power output of a battery can be estimated by

$$P_B = V_o i, \tag{2.164}$$

where V_o is the battery voltage in volt. We can find i by rearranging Equation (2.163),

$$i^n = \frac{Rt}{t}\left(\frac{C}{Rt}\right)^n,$$

$$i = \sqrt[n]{\frac{Rt}{t}}\frac{C}{Rt},$$

$$i = \left(\frac{Rt}{t}\right)^{1/n}\frac{C}{Rt}. \tag{2.165}$$

Substituting Equation (2.165) into Equation (2.164) will yield the following

$$P_B = V_o\frac{C}{Rt}\left(\frac{Rt}{t}\right)^{1/n}. \tag{2.166}$$

Battery power can also be calculated by dividing the power required with the total efficiency η_{tot}

$$P_B = \frac{P_r}{\eta_{tot}}. \tag{2.167}$$

Losses in the propulsion system, which consists of motor, motor driver, and propeller, would decrease the battery power output. Each element has its own efficiency, and they will be merged into a total efficiency η_{tot} for this calculation and further. Thus integrating Equation (2.112) and (2.167) into (2.166) with $k = 1/(\pi Ae)$ yield

$$\frac{\frac{1}{2}\rho V^3 SC_{D0} + (2W^2k/\rho VS)}{\eta_{tot}} = V_o\frac{C}{Rt}\left(\frac{Rt}{t}\right)^{1/n},$$

$$\frac{C}{Rt}\left(\frac{Rt}{t}\right)^{1/n} = \frac{1}{\eta_{tot}V_o}\left[\frac{1}{2}\rho V^3 SC_{D0} + \frac{2W^2k}{\rho VS}\right]. \tag{2.168}$$

Solving the equation in terms of time t give us

$$\sqrt[n]{\frac{Rt}{t}} = \frac{Rt}{\eta_{tot}V_oC}\left[\frac{1}{2}\rho V^3SC_{D0} + \frac{2W^2k}{\rho VS}\right],$$

$$\left(\frac{Rt}{t}\right) = \left(\frac{Rt}{\eta_{tot}V_oC}\left[\frac{1}{2}\rho V^3SC_{D0} + \frac{2W^2k}{\rho VS}\right]\right)^n,$$

$$t = Rt\left(\frac{\eta V_o \times C}{\frac{1}{2}\rho V^3SC_{D0} + (2W^2k/\rho VS)}\frac{1}{Rt}\right)^n,$$

$$t = Rt\frac{1}{Rt^n}\left[\frac{\eta V_o \times C}{\frac{1}{2}\rho V^3SC_{D0} + (2W^2k/\rho VS)}\right]^n,$$

$$E = t = Rt^{1-n}\left[\frac{\eta V_o \times C}{\frac{1}{2}\rho V^3SC_{D0} + (2W^2k/\rho VS)}\right]^n, \tag{2.169}$$

where E is the endurance of the aircraft in hours. To measure the endurance of a battery-powered electric aircraft we can use Equation (2.169), this equation also accounts the battery discharge rate at any flight velocity. Maximum endurance and range have their own condition, for maximum endurance the condition is

$$C_{D0} = \frac{1}{3}kC_L^2 \tag{2.170}$$

$$C_L^2 = 3C_{D0}/k$$

$$C_L = \sqrt{3C_{D0}/k}$$

and for maximum range the condition is

$$C_{D0} = kC_L^2. \tag{2.171}$$

$$C_L^2 = C_{D0}/k$$

$$C_L = \sqrt{C_{D0}/k}$$

Equation (2.170) and (2.171), combined with Equation (2.74) result in the following for maximum range V_R and maximum endurance V_E;

$$V^2 = \frac{W}{C_L \frac{1}{2}\rho S},$$

$$V = \sqrt{\frac{W}{C_L \frac{1}{2}\rho S}},$$

$$V = \sqrt{\frac{2W}{\rho S}\frac{1}{C_L}},$$

thus

$$V_E = \sqrt{\frac{2W}{\rho S}\sqrt{\frac{k}{3C_{D0}}}}, \tag{2.172}$$

$$V_R = \sqrt{\frac{2W}{\rho S}\sqrt{\frac{k}{C_{D0}}}}. \tag{2.173}$$

V_E and V_R are required to find the aircraft endurance and range, respectively. To estimate the endurance of the aircraft V_E is used in Equation (2.169). For maximum range estimation, V_R is used in Equation (2.169) to find the endurance when the aircraft is flying for maximum range. Then, the maximum range can be estimated using

$$R_{max} = E \times V_R \tag{2.174}$$

where the endurance E is the endurance at maximum achievable range.

Another way to estimate the maximum endurance is by integrating Equation (2.170) and Equation (2.102) into

$$P_r = \frac{1}{2}\rho V^3 S(C_{D0} + kC_L^2) \tag{2.175}$$

which resulted in

$$P_r = 2\rho S C_{D0}\left(\frac{2W}{\rho S}\sqrt{\frac{k}{3C_{D0}}}\right)^{3/2}$$

$$P_r = \frac{2}{\sqrt{\rho S}} C_{D0}^{1/4}\left(2W\sqrt{\frac{k}{3}}\right)^{3/2}. \quad (2.176)$$

Since the denominator in Equation (2.169) is the power required P_r then we can simply replace it with the power required from Equation (2.176) which yields

$$E_{max} = Rt^{1-n}\left(\frac{\eta V_o \times C}{\frac{2}{\sqrt{\rho S}} C_{D0}^{1/4}\left(2W\sqrt{\frac{k}{3}}\right)^{3/2}}\right)^n h. \quad (2.177)$$

Similarly for maximum range R_{max} with $C_L = \sqrt{C_{D0}/k}$ will yield

$$P_r = \frac{1}{2}\rho V^3 S\left(C_{D0} + k\frac{3C_{D0}}{k}\right)$$

$$P_r = 2\rho V^3 S C_{D0},$$

then integrate V from Equation (2.172) we got,

$$P_r = 2\rho S C_{D0}\left(\frac{2W}{\rho S}\sqrt{\frac{k}{3C_{D0}}}\right)^{3/2},$$

$$P_r = 2\rho S C_{D0}\frac{\sqrt{\rho S}}{(\rho S)^2}\frac{4\sqrt{C_{D0}}}{C_{D0}}\left(2W\sqrt{k/3}\right)^{3/2}$$

$$P_r = 2\frac{\sqrt{\rho S}}{\rho S} C_{D0}^{1/4}\left(2W\sqrt{k/3}\right)^{3/2}$$

$$P_r = \frac{2}{\sqrt{\rho S}} C_{D0}^{1/4}\left(2W\sqrt{k/3}\right)^{3/2}. \quad (2.178)$$

Thus, the endurance for maximum range will be

$$E = Rt^{1-n} \left(\frac{\eta V_o \times C}{\frac{2}{\sqrt{\rho S}} C_{D0}^{1/4} \left(2W\sqrt{k/3} \right)^{3/2}} \right)^n h. \tag{2.179}$$

Integrating Equations (2.179) and (2.173) into Equation (2.174) will yield

$$R_{max} = Rt^{1-n} \left(\frac{\eta V_o \times C}{\frac{2}{\sqrt{\rho S}} C_{D0}^{1/4} \left(2W\sqrt{k} \right)^{3/2}} \right)^n \sqrt{\frac{2W}{\rho S}\sqrt{\frac{k}{C_{D0}}}} \cdot 3.6km, \tag{2.180}$$

the 3.6 multiplier is for converting from m/s to km/h.

CHAPTER 3
RESEARCH METHODOLOGY

TABLE 3.1: N-219 Fuel/Battery Configuration for 11 PAX

Fuel/Battery (F/B)	Airbreathing A	Electric A	Electric C
Type	Avtur	Li-Ion	Li-Ion
Mass	1600 kg	1600 kg	4996.8 kg
Volume	1975 L	632.4 L	1975 L
Specific Energy	42.8 MJ/kg (11,889 Wh/kg)	0.954 MJ/kg (265 Wh/kg)	0.954 MJ/kg (265 Wh/kg)
Energy Density	34.668 MJ/L (9,630 Wh/L)	2.412 MJ/kg (670 Wh/L)	2.412 MJ/kg (670 Wh/L)
Density	@ 15 °C ≈ 0,81 kg/L	2.53 kg/L	2.53 kg/L
Total Energy	68,470 MJ ≈ 19 MW	1525 MJ ≈ 423.6 kW	4763.7 MJ ≈ 1.3 MW
Total Weight	7030 kg	7030 kg	10603 kg

TABLE 3.2: N-219 Fuel/Battery Configuration for 21 PAX

Fuel/Battery (F/B)	Airbreathing B	Electric B	Electric D
Type	Avtur	Li-Ion	Li-Ion
Mass	644 kg	644 kg	4996.8 kg
Volume	795 L	254.5 L	1975 L
Specific Energy	42.8 MJ/kg (11,889 Wh/kg)	0.954 MJ/kg (265 Wh/kg)	0.954 MJ/kg (265 Wh/kg)
Energy Density	34.668 MJ/L (9,630 Wh/L)	2.412 MJ/kg (670 Wh/L)	2.412 MJ/kg (670 Wh/L)
Density	@ 15 °C ≈ 0,81 kg/L	2.53 kg/L	2.53 kg/L
Total Energy	68,470 MJ ≈ 19 MW	1525 MJ ≈ 423.6 kW	4763.7 MJ ≈ 1.3 MW
Total Weight	7030 kg	7030 kg	11603 kg

In order to identify whether changing from fuel to battery would pose any issues, a configuration comparison is necessary; as seen in Table 3.1 and 3.2, this comparison also intended to foresee the potential of a battery compared to fossil fuel. According to Figure 1.8, different payload affects the maximum range. That is because, in order to increase the payload, some part of the fuel must be sacrificed, and vice versa. That is why there are two different tables for different payloads, as each payload has its own fuel/battery capacity, with exception of configuration Electric C and D. The Airbreathing configuration and the Electric A and B configuration obey the total weight limit of MTOW (7030 kg), where the Electric C and D configuration is made solely for research purposes that disregards any structural barrier and has no limit in any aspect. That is why the Electric C and D configuration has an alarmingly extra weight up to 3 tonnes more than the other configurations.

3.1 Methodology/Steps of work

Figure 3.1 represents the steps taken throughout the making of this thesis, and as shown, in order to calculate the range and endurance for electric N-219, the author first needs to complete a specific task. From the top left, the author needs to collect the N-219 data and performance. This is one of the most important data to obtain and research aside from the influence of aircraft electrification, as this is the fundamental of range and endurance performance.

The ISA computational model was made in order to smoothen the overall progression of calculating the range and endurance. Thus this makes the reconfirmation of data more accessible. The author does not have to manually calculate the atmospheric properties every time the author decides to recode or calculate the related data.

The research on the influence of aircraft electrification is done because different systems use different measurement methods. For example, a fossil fuel based system takes into account the change in aircraft weight for calculating its range and endurance; meanwhile, with the use of a battery, an electric aircraft retains

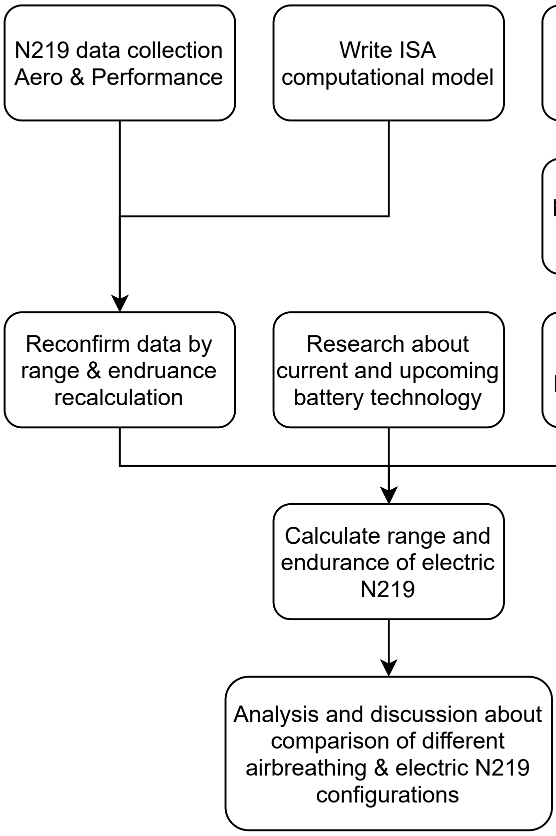


Figure 3.1: Methodology Flow Chart

its weight from takeoff to landing. This alone already changes the equation by a bit; furthermore, there are other effects of battery voltage and current that must be considered for using an electric vehicle.

Then after taking note of current and future battery specifications, the author can calculate the range and endurance of electric N-219 with the combination of data and research the author did previously. A calculation that takes into account battery characteristics using N-219 data and the latest battery technology. Thus with the calculation done, analysis and discussion can be created to compare the performance between the electric N-219 and the airbreathing N-219.

3.2 N-219 data collection

Table 3.3: N-219 General Data (Aerospace, n.d.-b)

Maximum Take Off Weight (MTOW)	7,030 Kg
Max. Landing Weight	6,940 Kg
Max. Fuel Capacity	1,600 Kg
Maximum Range with Maximum Fuel	828 NM
Maximum Payload	2,313 Kg
Take Off Distance	435 m
Landing Distance	509 m
Maximum Cruise Speed	210 Kts
Economical Cruise Speed	170 Kts
Stall Speed	59 Kts
Range with 19 pax	480 NM
Range at Max Fuel	828 NM
Operating altitude	10,000 ft
Ceiling Altitude	24,000 ft

Finding a credible source for the N-219 data was not an easy task, as the aircraft itself is still in production and certification state. This results in very few papers about N-219 and other sources of N-219 data. Furthermore, the Indonesian Aerospace website only provides a fraction of N-219 data, which mostly is a general performance specification as shown in Table 3.3.

Luckily most of the data required for this thesis can be found within some of these papers; most notably is the thesis by Mirna Sari about lift coefficient optimization for single slotted flap and double slotted flap on N-219 aircraft (Sari, 2018). The data obtained here can be seen from Table 3.4. Another important piece of data obtained from this thesis is the N-219 wing profile; finding out the wing profile is crucial as it enables the advisors to assist the author in finding one of the most important variables: the zero-lift drag coefficient of the N-219 aircraft.

TABLE 3.4: N-219 Geometric Characteristic (Sari, 2018)

Wing Area	41.5 m^2
Wing Span	19.5 m
Wing Aspect Ratio	9.16
Wing Taper Ratio	0.52
Wing Sweep Angle	-0.79°
Mean Aerodynamic Chord, MAC	2.13 m
Fuselage Diameter	1.3 m
Horizontal Tail (HT) Area	10.99 m^2
HT Aspect Ratio	5.17
HT Taper Ratio	0.53
HT Sweep Angle (deg)	3.43°
Vertical Tail (VT) Area	8.34 m^2
VT Aspect Ratio	1.6
VT Taper Ratio	0.5
VT Sweep Angle	30.95°
Landing Gear Wheel Base	5.126 m
Landing Gear Wheel Track	3.7 m

The N-219 use an airfoil profile LS(1)-0417, as shown in Fig. 3.2, this airfoils has a characteristic as follow:

- Maximum airfoil thickness $0.17c$;
- $\alpha_{c_l=0} = -4$ deg;
- $c_{l_\alpha} = 6.3814$/rad;
- $c_{m_\alpha} = 0.2455$/rad for $\alpha > 6$ deg and $c_{m_\alpha} = -0.1705$/rad for $\alpha < 4$ deg;

According to Figure 3.3, at moderate α with $Re = 6 \times 10^6$, the c_d varies from 0.01 to 0.016. The N-219 have a maximum cruise speed of 108 m/s, which is less than Mach 0.3, thus the air can be considered as incompressible.

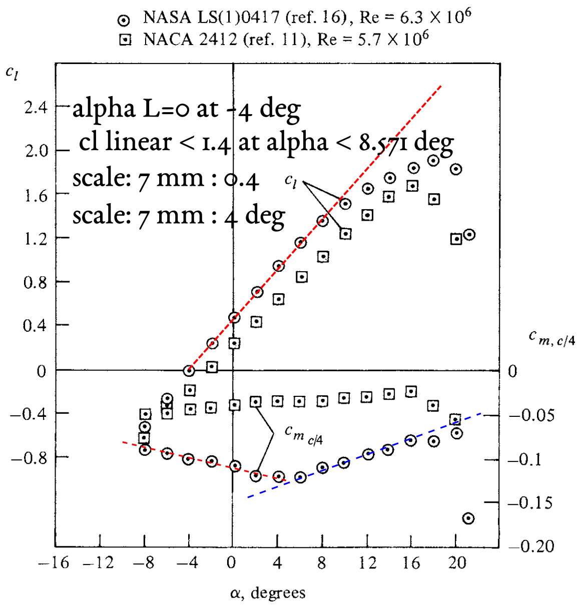
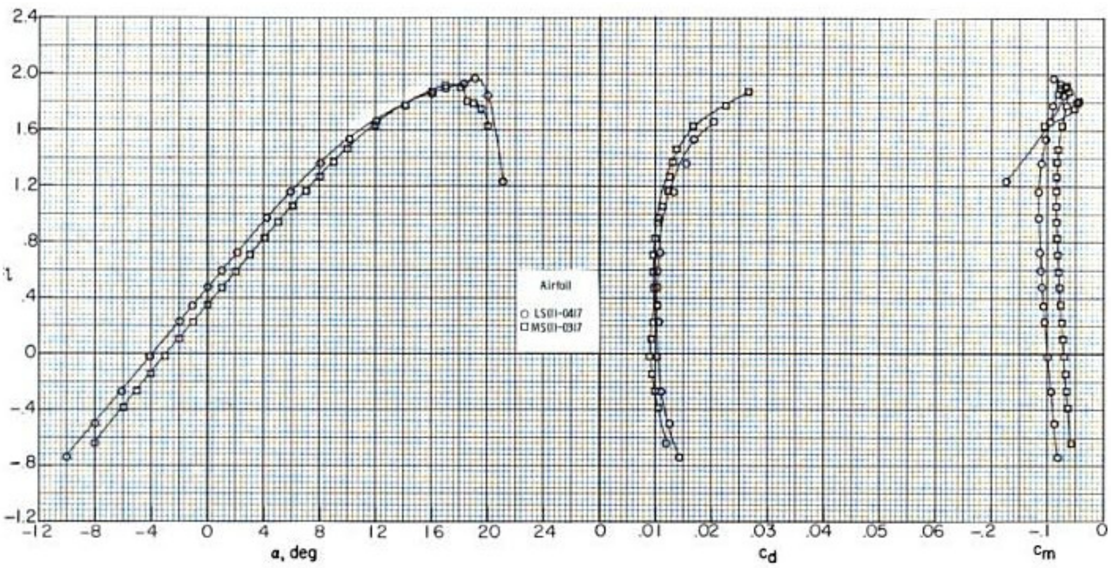


FIGURE 3.2: Airfoil series LS(1) -0417



(c) R = 6.0 × 10⁶.

FIGURE 3.3: c_l, c_d, and c_m at $Re = 6 \times 10^6$

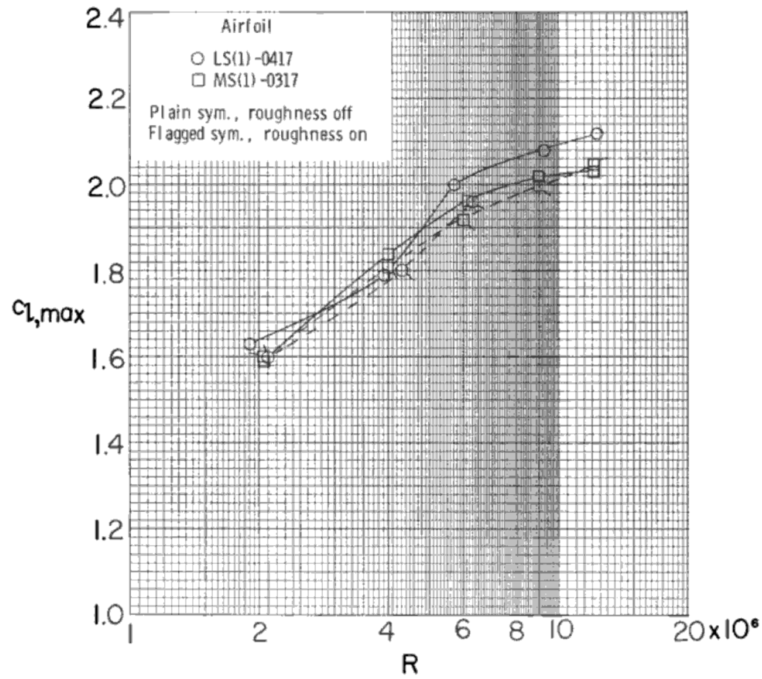


FIGURE 3.4: $c_{l_{max}}$ for various Reynolds numbers.

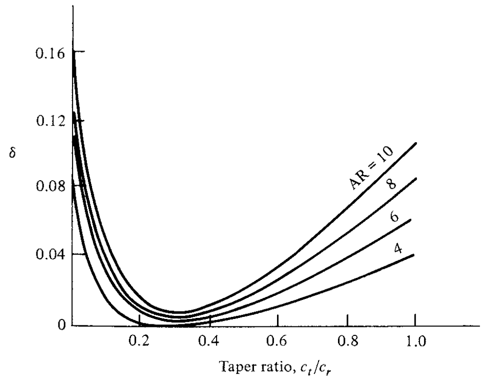


FIGURE 3.5: Induced drag coefficient vs taper ratio.

Wing theory

General lift equations,

$$\Gamma = 2bV_\infty \sum_{1}^{N} A_n \cos n\theta, \tag{3.1}$$

$$\frac{d\Gamma}{d\theta} = 2bV_\infty \sum_{n=0}^{N} nA_n \cos n\theta, \tag{3.2}$$

Downward velocity = w,

$$w = \frac{1}{4\pi} \int_{-b/2}^{b/2} \frac{(d\Gamma/dy)dy}{y - y_0} \tag{3.3}$$

$$w = \frac{1}{4\pi} \int_0^\pi \frac{2bV_\infty \sum_n A_n \cos n\theta d\theta}{-1/2(\cos\theta - \cos\theta_0)}$$

$$w = \frac{V_\infty}{\pi} \int_0^\pi \frac{\sum_n A_n \cos n\theta d\theta}{\cos\theta - \cos\theta_0}$$

$$w = \frac{V_\infty}{\pi} \sum nA_n \int_0^\pi \frac{\cos n\theta d\theta}{\cos\theta - \cos\theta_0}$$

$$w = V_\infty \sum nA_n \frac{\sin n\theta}{sin\theta} \tag{3.4}$$

Lift per unit span,

$$L' = \frac{1}{2}\rho V_\infty^2 c c_l' \tag{3.5}$$

$$\rho V_\infty \Gamma = \frac{1}{2}\rho V_\infty^2 c c_l'$$

$$c_l' = \frac{2\Gamma}{V_{\infty c}} \tag{3.6}$$

where:

$$c_l' = a_0(\alpha_{eff} - \alpha_0),$$

$$c_l' = a_0(\alpha - \alpha_i - \alpha_0),$$

$$c_l' = a_0[(\alpha - \alpha_0) - \alpha_i],$$

and

- $\alpha_0 = \alpha_{L=0}$,

- $\alpha_{eff} = \alpha - \alpha_i$,
- $\alpha_i = \tan^{-1} \frac{w}{V_\infty} \frac{w}{V_\infty}$.

$$\frac{2\Gamma}{a_0 c} = V_\infty \left[(\alpha - \alpha_0) - \frac{w}{V_\infty} \right] = V_\infty(\alpha - \alpha_0) - \frac{V_\infty \Sigma_n A_n \sin n\theta}{\sin\theta}$$

$$\frac{2bV_\infty \sum_{n=1}^{N} A_n \sin n\theta}{a_0 c} = V_\infty(\alpha - \alpha_0) - \frac{V_\infty \Sigma_n A_n \sin n\theta}{\sin\theta}$$

$$\frac{4b}{ca_0} \sum_{n=1}^{N} A_n \sin n\theta = (\alpha - \alpha_0) - \frac{\sum_n A_n \sin n\theta}{\sin\theta}$$

$$\sum A_n \sin n\theta = \frac{ca_0}{4b}(\alpha - \alpha_0) - \frac{ca_0}{4b} \frac{\sum_n A_n \sin n\theta}{\sin\theta}$$

$$\frac{ca_0}{4b}(\alpha - \alpha_0) = \sum A_n \sin n\theta \left(1 + \frac{ca_0}{4b} \frac{n}{\sin\theta}\right)$$

suppose that $K = \frac{ca_0}{4b}$

$$K(\alpha - \alpha_0) = \sum A_n \sin n\theta \left(1 + \frac{Kn}{\sin\theta}\right)$$

$$K \sin\theta(\alpha - \alpha_0) = \sum A_n \sin n\theta \, (\sin\theta + Kn)$$

TABLE 3.5: N-219 wing properties

For N-219	Mid span	Wing tip
c	2.622	1.456
$a_0 = dC_L/d\alpha$	6.3814	6.3814
$\alpha_{L=0}$	$-4°$ (-0.0698 rad)	$-4°$ (-0.0698 rad)
α incident angle	$-2°$ (-0.035 rad)	$-2°$ (-0.035 rad)

$$L = \int_{-b/2}^{b/2} \rho V_\infty \Gamma dy = \int_0^{\pi} \rho V_\infty \Gamma \left(\frac{b}{2} \sin\theta d\theta\right) \tag{3.7}$$

$$L = \frac{1}{2} \int_0^{\pi} \rho V_\infty b \Gamma \sin\theta d\theta = \frac{1}{2} \int_0^{\pi} \rho V_\infty b 2bV_\infty \Sigma A_n \sin n\theta \sin\theta d\theta$$

$$L = b^2 \rho V_\infty^2 \int_0^\pi \Sigma A_n \frac{1}{2} \left[\cos(n-1)\theta + \cos(n+1)\theta \right] d\theta$$

$$L = b^2 \rho V_\infty^2 \frac{1}{2} \int_0^\pi \Sigma A_n \left[\frac{\cos(n-1)\theta}{n-1} + \frac{\cos(n+1)\theta}{n+1} \right] \Bigg|_0^\pi$$

$$L = b^2 \rho V_\infty^2 \frac{1}{2} A_n \left(\frac{\sin(n-1)\theta}{n-1} \right) \Bigg|_0^\pi$$

$$L = b^2 \rho V_\infty^2 \frac{1}{2} A_n \left[\lim_{n \to 1} A_1 \frac{\sin(n-1)\theta}{n-1} \right]_0^\pi$$

$$L = b^2 \rho V_\infty^2 \frac{1}{2} A_1 \pi \tag{3.8}$$

$$\frac{L}{\frac{1}{2}\rho V_\infty^2} = b^2 A_1 \pi$$

$$C_L = A_1 \pi AR \tag{3.9}$$

Induced drag

$$D = \int_{b/2}^{b/2} \rho W \Gamma dy = \int_0^\pi \rho \frac{V_\infty \Sigma_n A_n \sin n\theta}{\sin\theta} 2bV_\infty \Sigma A_n \sin n\theta \frac{b}{2} \sin\theta d\theta \tag{3.10}$$

$$D = \rho V_\infty^2 b^2 \int_0^\pi \Sigma_n A_n \sin n\theta \Sigma A_n \sin n\theta d\theta$$

$$D_i = \rho V_\infty^2 b^2 \left(\frac{\pi}{2} \Sigma_n A_n^2 \right) \tag{3.11}$$

$$A_1 = \frac{C_L}{\pi AR} \tag{3.12}$$

$$c_{di} = \frac{C_L}{A_1} \sigma_n A_n^2 = C_L \sigma_n \frac{A_n^2}{A_1} \frac{A_1}{A_1} = C_L \sigma_n \frac{A_n^2}{A_1^2} A_1 = C_L \frac{C_L}{\pi AR} \sigma_n \frac{A_n^2}{A_1^2}$$

$$c_{di} = \frac{C_L^2}{\pi AR}\left[1 + \left(\frac{3A_3^2}{A_1^2} + \frac{5A_5^2}{A_1^2} + ...\right)\right] \tag{3.13}$$

$$\delta = \frac{3A_3^2}{A_1^2} + \frac{5A_5^2}{A_1^2} + ... \tag{3.14}$$

$$c_{di} = \frac{C_L^2}{\pi AR}(1+\delta) = \frac{C_L^2}{\pi eAR} \tag{3.15}$$

$$e = \frac{1}{1+\delta} \tag{3.16}$$

From numerical calculation using array of A, we obtain

$$e = \frac{1}{1+\delta} = 0.62921$$

C_L wing at level flight = $C_L = \pi AR\, A_1 = 0.320817$ (α is only $\alpha_{incidence}$ $2°$)

Power Law:

$$\frac{\mu}{\mu_0} = (\frac{T}{T_0})^{0.7}. \tag{3.17}$$

$$\mu = (1.789 \times 10^{-5})(\frac{268.67}{288.15})^{0.7} = 1.703 \times 10^{-5}$$

$$Re = \frac{0.90926(30m/s)(2.1772)}{1.703 \times 10^{-5}} = 34.86 \times 10^5 = 3.5 \times 10^6$$

$$\nu = \frac{\mu}{\rho} = 1.87345 \times 10^{-5}$$

For skin friction coefficient C_f,

$$C_f = \frac{1.328}{\sqrt{Re}} = \frac{1.328}{\sqrt{3.5 \times 10^6}} = 0.0007098 \leftarrow \text{laminar}$$

$$\frac{C_f}{0.455} = (log_{10}R)^{-2.58} => C_f = 0.003574 \leftarrow \text{turbulent}$$

From airfoil profile, we have $C_d = 0.011 \leftarrow$ including profile drag and skin friction drag.

Fuselage estimation

known variables:

- Fuselage ν-cabin: 18.1 m^3
- N-219 ν-baggage: 3.87 m^3
- S_{wetted} : 86.74 m^2

We can estimate the profile drag coefficient C_d and skin friction drag coefficient C_f

- $\frac{l}{d} = \frac{11.202}{2} = 5.6 => C_d \approx 0.058$
- $C_f = \frac{C_d}{3} \times \frac{D}{l} = 0.00345$

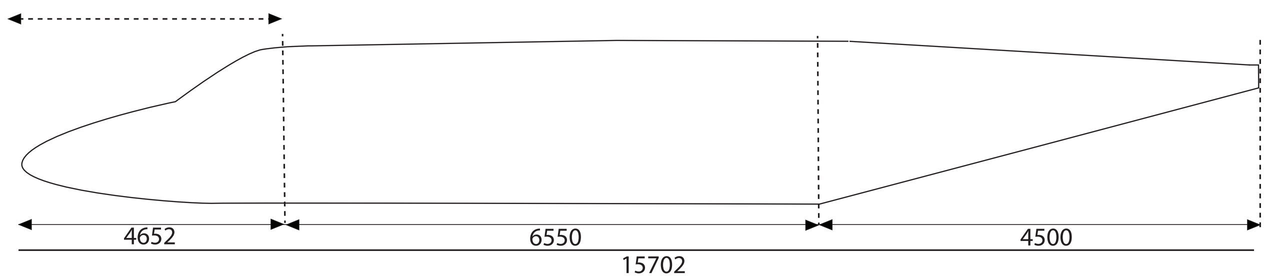


Figure 3.6: N-219 fuselage estimation

Horizontal tail: MS(1)-0313 mod airfoil.

- Area = 10.88 m^2
- Span = 7.5 m
- $c_t/c_r = 0.527$
- $c_r = 1.9$ m

- $c_t = 1.0$ m
- $\alpha_{incidence} = 0$

Vertical tail : Joukowsky airfoil.

- Area = 8.344 m^2
- Span = 3.7 m
- $c_r = 3$ m
- $c_t = 1.51$ m
- t = 13%

$$\frac{C_l}{C_d} = 20$$

$$\frac{2\pi\alpha}{C_d} = 20(\text{assume } \alpha < .2)$$

$$C_d \approx 0.0055$$

Nacelle drag coefficient
min 0.08

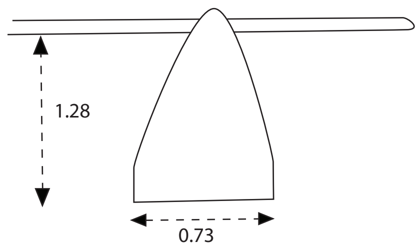


FIGURE 3.7: N-219 nacelle estimation

Wheel + fairings

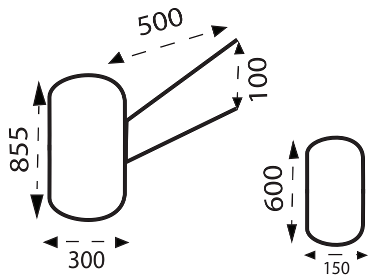


FIGURE 3.8: N-219 wheel + fairings estimation

The C_D for the wheel is 0.7. For 3D shape, a correction is required,

$$\frac{C_{D_{20}}}{C_{D_{30}}} = 1.8$$

thus, $C_{D_{30}} = 0.39$

Wheel strut, from Figure x, the C_D is 0.6 (cylinder). Thus,

$$\frac{D}{q} = C_{D_{wing}} S + C_{D_f}(Sf) + C_{f_f}(Sf) + C_{D_{VT}}(S) + C_{D_{HT}}(S) + C_{D_N}(S) + C_{D_{VT}}(S) + C_{D_W}(S) + C_{D_{WS}}(S)$$

$$\frac{D}{q} = (0.011)(41.5) + (0.058)(2)(2) + (0.00346)(86.74) + (0.0055)(8.344) + (0.009)(10.88) + (0.08)(\pi)\left(\frac{0.73}{2}\right)^2 + (0.39)(0.3)(0.855) \times (2) + (0.39)(0.15)(0.6) + (0.6)(0.5)(0.1) \times (2)$$

$$\frac{D}{q} = 1.5836 => C_{D_0} = \frac{D}{q} \times \frac{1}{S} = 0.0357$$

One particular data were not available for the public and that is the specific data for Pratt & Whitney Canada PT6A-42 (the main engine for N-219 aircraft) and the Hartzell propeller. In order for the author to calculate the range and endurance of the airbreathing N-219, the author needs to know parameters such as Specific Fuel Consumption SFC, Shaft Horsepower SHP, Thrust T, and propeller efficiency η_j. Only the engine manufacturer and the aircraft manufacturer hold these data for this specific type of engine; thus, the author of this thesis proposes a formal letter to Indonesian Aerospace for a data retrieval, Appendix A.

3.3 ISA Computational model

In order to accurately calculate the range and endurance of aircraft with ease, the author of this thesis formulates a python code to compute the atmospheric properties on any given temperature and altitude. This python code is one of the fundamental aspects for the final code later on in this thesis. This code is also present in almost every other code the author creates throughout this thesis. Figure 3.9 shows the workflow of the ISA python code; The code for ISA computational model can be found in Appendix B.

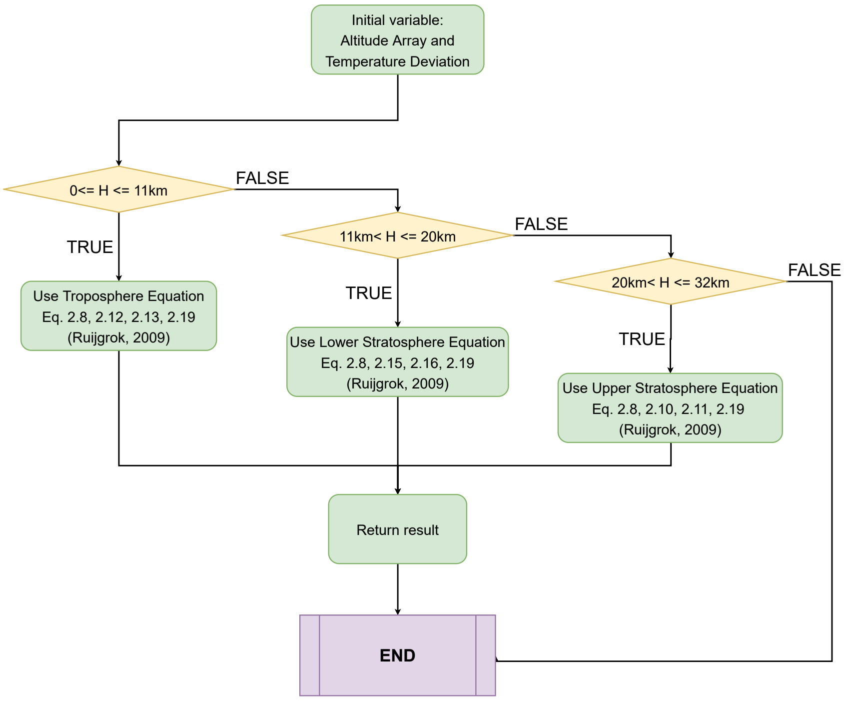


FIGURE 3.9: ISA computational model flow chart (H is geometrical altitude)

3.4 Confirmation of N-219 data using range & endurance

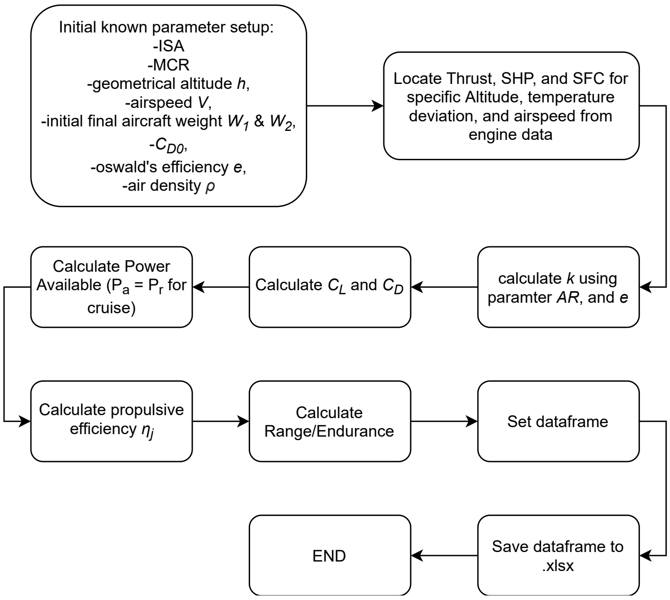


FIGURE 3.10: Range and endurance airbreathing aircraft flow chart table

TABLE 3.6: N-219 aerodynamics data

H (ft)	W (kg)	rho (kg/m^3)	S (m^2)	A	e	C_{D0}	V_R (kts)	V_E (kts)
0	7030	1.225	41.5	9.16	0.62921	0.0357	112.93	82.78
5000	7030	1.0556	41.5	9.16	0.62921	0.0357	121.62	92.41
10000	7030	0.9046	41.5	9.16	0.62921	0.0357	131.38	99.83

It is known from the Indonesian Aerospace webpage about N-219 that the N-219, under optimal operating altitude and velocity, could reach a maximum range of around 828 nmi (1534 km). Thus with the obtained data, the author can calculate the range and endurance of N-219 to confirm the data and calculation with the published result of N-219. For this particular section, the author also devised a computational model to calculate the airbreathing range and endurance of the N-219 aircraft, Appendix B. Figure 3.10 shows the flowchart of the airbreathing range and endurance computational model. So far, it is known that an N-219 with 1600kg of AVTUR and 7030 total weight, flying at 10,000 ft (3048 m) with an airspeed of 170 kts (87.5 m/s), could reach a distance of around 828 nmi (1534 km). But before everything else, the author must confirm that the airspeed given is the same as the V_R for the N-219 aircraft.

Using Equation 2.173, the author can determine the airspeed for N-219 maximum range, which can be seen in Table 3.6. It is to be noted that when using any aircraft formula, the weight should be in newton. As stated previously in Chapter 2.8, to achieve maximum endurance, the aircraft requires a different airspeed which caters to the changing lift drag ratio. Using Equation 2.172 the author finds the maximum endurance airspeed V_E.

Now that the optimal airspeed is known for each flight type, the next thing to do is to identify the Thrust T, Specific Fuel Consumption SFC, and Propeller Efficiency η_j as these parameters are dependent on airspeed, altitude, and temperature deviation. A slight inconvenience here is that the data is only available at a specific interval of airspeed, altitude, and temperature. Luckily for the altitude and temperature deviation has data for 10,000 ft and ISA +0. But the inconvenience comes from the fact that the airspeed data only comes at 20 kts interval, as seen in Appendix A. Thus since the calculated airspeed does not match any airspeed data given, then the airspeed will be rounded to the closest available data; in this case, both are rounded to 100 kts (51.4 m/s) for V_E and 140 kts (72 m/s) for V_R.

TABLE 3.7: N-219 engine data

ISA	Altitude (ft)	Airspeed (kts)	Thrust (kgf)	Fuel Flow (lb/hr)	η_p	$SHP(hp)$	$SFC(lb/hr/lb)$
0	10000	140	434.62	326.32	0.823	507.12	0.34
0	10000	100	523.43	320.7	0.734	489.15	0.28

Now that the author knows the airspeed and the altitude, the author can determine the Thrust produced and Fuel Flow rate consumed during that specific airspeed and altitude, Table 3.7. For Thrust T, Fuel Flow FF, and Propeller

Efficiency η_p were taken directly from Appendix A, which is the data given from Indonesian Aerospace. SHP can be found using

$$SHP = \frac{VT}{\eta_p}, \tag{3.18}$$

it is to be noted that to achieve SHP in hp; the calculation must use imperial units. For SFC calculate using

$$SFC = \frac{FF}{T}, \tag{3.19}$$

and again, for SFC in $lb/hr/lb$ the units must be in imperial. One thing to note that the data from Table 3.7 is only for one engine, and since N-219 uses two Pratt & Whitney Canada PT6A-42, then the data for T, FF, SHP, and SFC must be multiplied by the total engine.

Then with the data from Table 3.7 and 3.6 the range and endurance of N-219 can be estimated using step from Figure 3.10. With $k = \frac{1}{\pi Ae}$, the C_L can be taken from Equation 2.91; since at cruse the lift is equal to aircraft weight, Thus

$$C_L = \frac{2W}{\rho V^2 S}. \tag{3.20}$$

The C_D can be found from Equation 2.45. Then since $P_a = P_r$, using Equation 2.112 the following equation can be obtained,

$$Pa = Pr = \frac{1}{2}\rho V^3 S C_{D0} + \frac{2W^2 k}{\rho V S}. \tag{3.21}$$

Then with $SHP = P_{br}$, the propulsive efficiency can be found from using Equation 2.118 into,

$$\eta_j = \frac{P_a}{SHP}. \tag{3.22}$$

Now that all the prerequisite variable is found, the author can proceed to calculate the range and endurance. Since the variable obtained for the maximum range is mostly in Imperial units, the author calculates the maximum range using the Imperial unit tuned Breguet equation, which includes the final conversion multiplier. This decision were made in order to minimize unit conversion; rather than converting each variable individually, it is much quicker to combine the conversion unit at the end. Thus the maximum range equation becomes,

$$R = 375 \frac{\eta_j}{c_p} \frac{C_L}{C_D} \frac{W_1}{W_2}, \tag{3.23}$$

and the result will be in miles (Saarlas, 2007). In order to get the result in Metric unit then the equation becomes,

$$R = 375 \frac{\eta_j}{c_p} \frac{C_L}{C_D} \frac{W_1}{W_2} 1.61 = 603.75 \frac{\eta_j}{c_p} \frac{C_L}{C_D} \frac{W_1}{W_2}. \tag{3.24}$$

For endurance angkanya masih salah

3.5 Electrification of Aircraft

This particular section of this chapter covers the research on the influence of aircraft electrification on aircraft's performance, research about Peukert's law and its effect on aircraft, and research on modification of Breguet equation for electric aircraft from Figure 3.1. The most notable influence of aircraft electrification on aircraft's performance lies in the capability of the aircraft to achieve a maximum range and endurance under other similar performance aspect as fossil fuel based system. This has been demonstrated by companies such as Eviation and magniX, who had successfully flown a 9 seater all-electric aircraft. While Eviation designed a 9 seater electric aircraft from scratch, magniX retrofitted an existing Cessna B208 Grand Caravan from its fossil fuel based system into an all-electric system. This difference resulted in Eviation able to achieve a farther range of 540-650 nmi (1,000-1,200 km) while magniX only able to achieve around 87 nmi (161 km), around one-tenth of it's fossil fuel counterparts. Albeit, it is still an impressive result for a 30-minute test flight.

Since the Breguet equation is specifically designed for an airbreathing engine, it is crucial to find an equation for aircraft's range and endurance that caters to the electric aircraft specification. A paper by Lance W. Traub has an in-depth calculation and its derivation about calculating electric aircraft range and endurance with a modified Breguet equation that takes into account the electric properties of electric aircraft (Traub, 2011). This, as well as Peukert's effect, which is also explained in Traub's paper, has been explained in Chapter 2.8.3.

3.6 Current and future battery technology

Table 3.8: Battery Specification Short (Raymer, 2006)("TYPES OF LITHIUM-ION BATTERIES", 2021)

Battery type/ manufacturer	Specific Energy (Wh/kg)	Energy Density (Wh/L)	Voltage (V)	Charge (C-rate)	Discharge (C-rate)	Charge Cycle
Lead Acid	45	100	-	-	-	-
Alkaline	100	300	-	-	-	-
Nickle Cadmium	60	160	-	-	-	-
Nickle Metal Hydride	90	300	1.35	1	2-4	-
Lithium ion	100-265	250-700	3.0-4.2	0.7-1	1-2	500-2000
Lithium Sulfur	500	1000	3.82	0.5	2-5	-

Table 3.8 shows a small list of battery types specification. The one showed in Table 3.8 is one of the most commonly used battery types over the years. Although Lithium-Sulfur (LiS) is the best in terms of overall capabilities, it is not yet widely used due to the technology being very recent, and not many products adopt the technology yet. Rather, Lithium-ion (Li-ion) dominates more than 80% of the battery market share.

With that being said, the main calculation for the electric N-219 will be using the Li-ion battery specification and characteristic due to the certainty of the battery availability.

3.7 Calculation of electric N-219 range and endurance

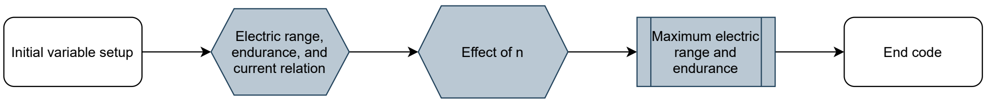


FIGURE 3.11: Electric Range and Endurance flowchart

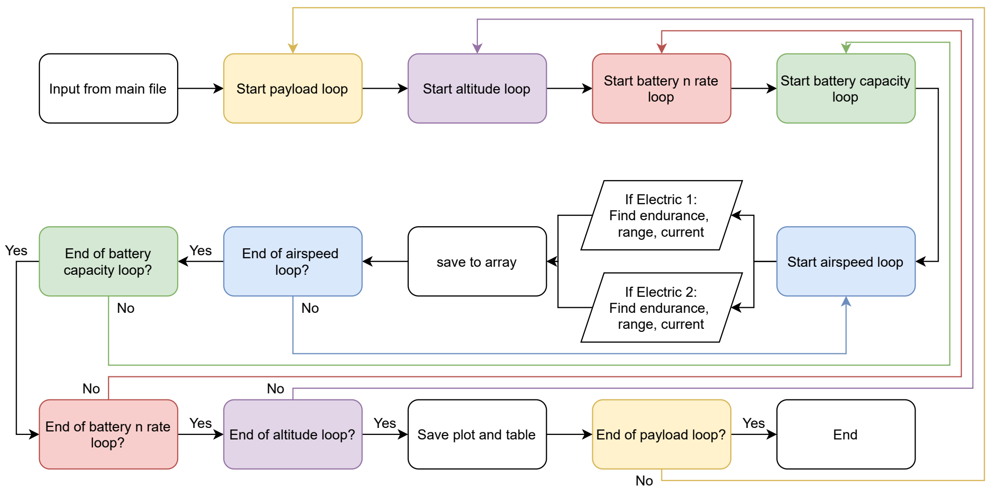


FIGURE 3.12: Electric range, endurance, and current relation flow chart

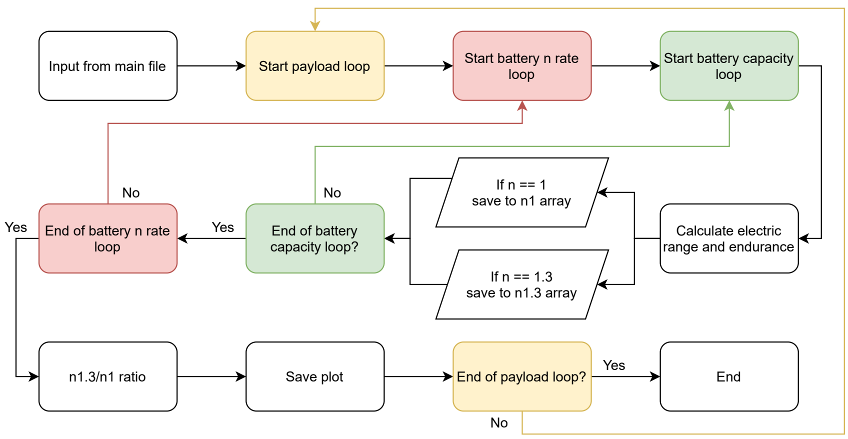


FIGURE 3.13: Effect of n flow chart

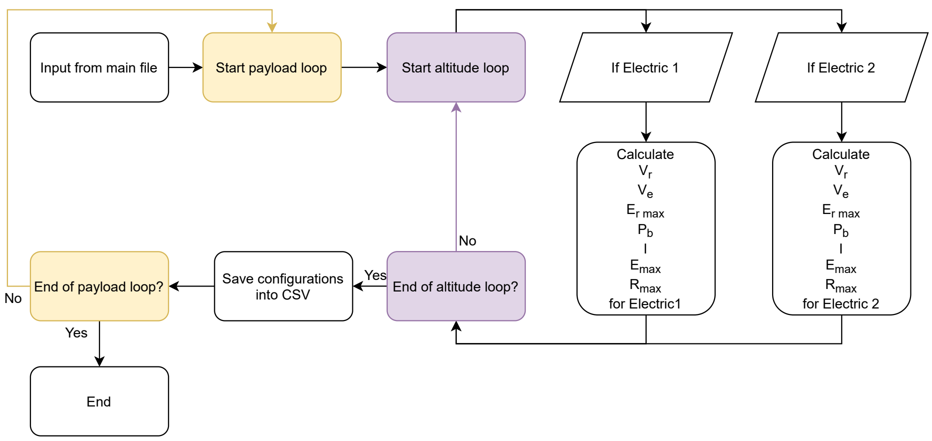


FIGURE 3.14: Maximum electric range and endurance flow chart

Figure 3.11 visualize the flow of work of maximum range and endurance calculation, which is the main topic of this thesis. Generally, this part is divided into three calculation phases, the first one being the calculation of the relation of range, endurance, and current in terms of airspeed, which can be seen in Figure 3.12. Then in Figure 3.13 is the calculation of the effect of n for range and endurance. Finally, in Figure 3.14 there is the calculation for the maximum range and endurance.

There are several things to note before going deep into the inner working of the calculations. For the discharge parameter n, the author will be using $n = 1$ for a general battery type and $n = 1.3$ for lithium-ion battery. There are also several pre-defined variables for this section, which can be seen in Table 3.9.

TABLE 3.9: N-219 Electric Variables

Altitude	10,000 ft (3048 m)
Temperature deviation	ISA+0
Total efficiency η_{tot}	0.81
Airdensity ρ	0.904637
Rt	1 h
Aircraft weight	$7030kg$ ($68941N$)
Aircraft weight w/o fuel/battery	$5430kg$ ($53250N$)
Fuel/battery weight W_f	$15690.64N$
Fuel/battery volume V_f	$1975L$
Wing area S	$41.5m^2$
Vertical Tail (VT) Area	8.34 m^2
Voltage $Volt$	$567V$
Zero-lift drag coefficient C_{D0}	0.0357
Wing aspect ratio A	9.16
Oswald's efficiency e	0.62921
Battery specific energy BSE	$265Wh/kg$
Battery energy density BED	$700Wh/kg$

The calculation of the relation of range, endurance, and current in terms of airspeed. The calculation will be done for $n = 1$ and $n = 1.3$, for every electric configurations, and current i, along a specified airspeed of 0 m/s to 120 m/s. Calculating the endurance is done using Equation 2.169, which, if multiplied by the airspeed at that moment, will result in the achievable range for the aircraft at that specific airspeed. Then to find the electrical current i, the author needs to find the power battery P_B first. Since $E = t$, the author can find P_B using Equation 2.166 , then using Equation 2.164 the author extract the i. The result will be shown in a graph. The purpose of this calculation is to see at what speed does the range and endurance peak and fall off, as well as to see the behavior of the electrical current throughout different airspeeds.

The calculation for effect of n. From here onwards, the calculation for range and endurance will be on a fixed airspeed using eV for maximum endurance E_{max} and rV for the maximum range R_{max}. The calculation will also be for every electric configurations with $n = 1$ and 1.3. The calculation for maximum range and endurance can be calculated directly using Equation 2.180 and 2.177 respectively. Then afterward, the result from $n = 1.3$ is divided by $n = 1$ to see the ratio between them.

The calculation for the maximum range and endurance will solely be using $n = 1.3$. Then for each configuration, the author first calculates the V_R using Equation 2.173 and V_E using Equation 2.172. The author then calculates the endurance for the maximum range E_{Rmax} (Equation 2.179), which is necessary to find the electrical current for the maximum range. From E_{Rmax}, the author can find P_B then i in the same manner as the calculation for effect of n. Then the author calculate the R_{max} directly using Equation 2.180. Moving on to E_{max} calculated using Equation 2.177, and the find the P_B as well as i. This concludes all calculations for the maximum range and endurance, and the result will be shown in a table.

CHAPTER 4
RESULTS AND DISCUSSIONS

In this chapter, we will be discussing the results obtained by using the methods from Chapter 3. In section 4.1 we will be analyzing how different altitudes and atmospheric properties would affect the range and endurance of the N-219 aircraft. Furthermore, we will also see how different aircraft configurations such as different payload and fuel capacity would affect the range and endurance and even the optimal airspeed of the aircraft. Due to the data obtained at a specific interval, the calculation for range and endurance for the airbreathing N-219 would be provided in a summarized table, with a full table available in Appendix A. An official data from Indonesian Aerospace for range and endurance also available and can be seen in Figure 1.8. This also acts as a benchmark to see whether the author's calculation for the range is accurate or not. If the result from Section 4.1 is similar to the Figure 1.8 for the same configuration, then the result can be said as accurate enough.

Section 4.2 will discuss the entirety of the electrification of the N-219 and its effects on the performance. There will be multiple configurations as well for the electric range and endurance. The first is that the configuration will be separated based on the payload or passenger capacity. Then for each payload, there will be two battery capacities. There will be two payloads: for nine passengers and two pilots, and nineteen passengers and two pilots; The passengers and pilot each is estimated as 100 kg, including the luggage. Then for the battery configurations, the first one will be filling up unused available weight with batteries. Available weight is calculated by subtracting the MTOW with the fuselage weight and payload weight.

Thus for the first battery configurations, the total weight will always be at MTOW. The trade-off for this configuration is that the more payload the aircraft carries, the lesser the batteries can be integrated. For the other configurations, the volume of the batteries will be constant, which replace the entirety of the 1600L Fuel Tank with a 1600L worth of batteries. In turn, the latter battery configurations will exceed the MTOW by a significant amount. All the electric N-219 were calculated using lithium-ion batteries with 265 Wh/kg of specific energy and 700 Wh/L energy density unless specified otherwise.

Both results from 4.1 and 4.2 will be compared and analyzed in section 4.3.

4.1 N-219 airbreathing range & endurance

TABLE 4.1: N-219 Airbreathing Results, 11 PAX (1100 kgpayload)

Altitude	$\rho(kg/m^3)$	C_L	C_D	$Pr(hp)$	$V(m/s)$	Endurance (h)	Range (km)
0.0	1.225	0.707	0.063	507.614	61.733	3.548	1045.358
0.0	1.225	0.52	0.051	644.245	72.022	3.522	1079.651
0.0	1.225	0.398	0.044	844.431	82.311	3.5	1099.609
0.0	1.225	0.314	0.041	1113.484	92.6	3.479	1109.291
0.0	1.225	0.255	0.039	1457.774	102.889	3.459	1111.13
0.0	1.225	0.21	0.038	1884.243	113.178	3.428	1104.524
5000.0	1.056	0.821	0.073	503.575	61.733	3.899	1129.807
5000.0	1.056	0.603	0.056	611.851	72.022	3.876	1177.886
5000.0	1.056	0.462	0.047	777.255	82.311	3.856	1207.902
5000.0	1.056	0.365	0.043	1003.576	92.6	3.838	1225.006
5000.0	1.056	0.295	0.041	1295.828	102.889	3.817	1231.819
5000.0	1.056	0.244	0.039	1659.694	113.178	3.787	1229.148
10 000.0	0.905	0.958	0.086	511.188	61.733	4.55	1307.297
10 000.0	0.905	0.704	0.063	592.612	72.022	4.514	1363.528
10 000.0	0.905	0.539	0.052	725.839	82.311	4.48	1397.515
10 000.0	0.905	0.426	0.046	913.169	92.6	4.446	1416.662
10 000.0	0.905	0.345	0.042	1158.331	102.889	4.404	1421.193
10 000.0	0.905	0.285	0.04	1465.834	113.178	4.349	1413.104

Table 4.2: N-219 Airbreathing Results, 21 PAX (2100 kgpayload)

Altitude	$\rho(kg/m^3))$	C_L	C_D	$Pr(hp)$	$V(m/s)$	Endurance (h)	Range (km)
0.0	1.225	0.712	0.064	510.414	61.733	1.428	388.59
0.0	1.225	0.523	0.051	646.645	72.022	1.418	401.337
0.0	1.225	0.4	0.045	846.531	82.311	1.409	408.756
0.0	1.225	0.316	0.041	1115.351	92.6	1.401	412.355
0.0	1.225	0.256	0.039	1459.454	102.889	1.392	413.039
0.0	1.225	0.212	0.038	1885.77	113.178	1.38	410.583
5000.0	1.056	0.826	0.073	506.824	61.733	1.569	419.982
5000.0	1.056	0.607	0.056	614.637	72.022	1.56	437.854
5000.0	1.056	0.465	0.048	779.692	82.311	1.552	449.012
5000.0	1.056	0.367	0.043	1005.742	92.6	1.545	455.37
5000.0	1.056	0.297	0.041	1297.778	102.889	1.536	457.902
5000.0	1.056	0.246	0.039	1661.466	113.178	1.524	456.909
10 000.0	0.905	0.964	0.087	514.98	61.733	1.831	485.96
10 000.0	0.905	0.708	0.063	595.862	72.022	1.817	506.862
10 000.0	0.905	0.542	0.052	728.683	82.311	1.803	519.496
10 000.0	0.905	0.428	0.046	915.697	92.6	1.79	526.614
10 000.0	0.905	0.347	0.042	1160.606	102.889	1.773	528.298
10 000.0	0.905	0.287	0.04	1467.902	113.178	1.75	525.291

Table 4.1 and 4.2 shows the result for achievable range and endurance of the airbreathing N-219 over various altitude and airspeed. As expected, the airbreathing N-219 has a better range and endurance at higher altitudes, and this phenomenon is more pronounced in the 11 PAX (1100 kg) payload. The airspeed for maximum range also increases. Supposedly, the optimum airspeed for the maximum range would increase as the altitude increase since to maintain the same lift, a higher airspeed is needed for lower air density. But due to the limitation of engine data, the shift of the optimum airspeed is not easily observable. Albeit, we can still slightly perceive that at sea level, the optimum airspeed is around 93 to 103 m/s,

and this increase in 5000 ft to around 103 to 113 m/s, but recede to around 103 m/s in 10 000.0feet.

It seems that for maximum endurance, the optimum airspeed might be much lower than the available data. This is relevant since endurance uses the highest lift to drag ratio to obtain the minimum power required. The velocity for minimum power required usually is slightly above the stalling airspeed. But nevertheless, the endurance also increases in time as the altitude increases. Note that the only difference between the 11 PAX (1100 kg payload) and 21 PAX (2100 kg payload) for airbreathing N-219 is just the fuel capacity, with 11 PAX fuel capacity at 1600 kg and 21 PAX fuel capacity at around 644 kg; everything else is the same. That is why for both payloads, the power required Pr is identical, and due to less drag at higher altitude, the power required is greatly reduced.

4.2 N-219 electric range & endurance

Figure 4.1 and 4.2 show the relation between aircraft velocity and its achievable range and endurance along with its required current. Theoretically, it is impossible for aircraft to cruise with airspeed below the stall airspeed, around 50 m/s for N-219 aircraft. But the graph provides a wide range of airspeed so that we can also observe the effect of Peukert's law which is differentiated by the continuous line and dashed lines. The continuous line has an n rate of 1.3, which is a typical n rate for lithium battery types, and the dashed line has an n rate of 1, which is what it would be for a perfect battery. The n rate is affected by many things, and those are the battery types, temperature, and the age of the battery.

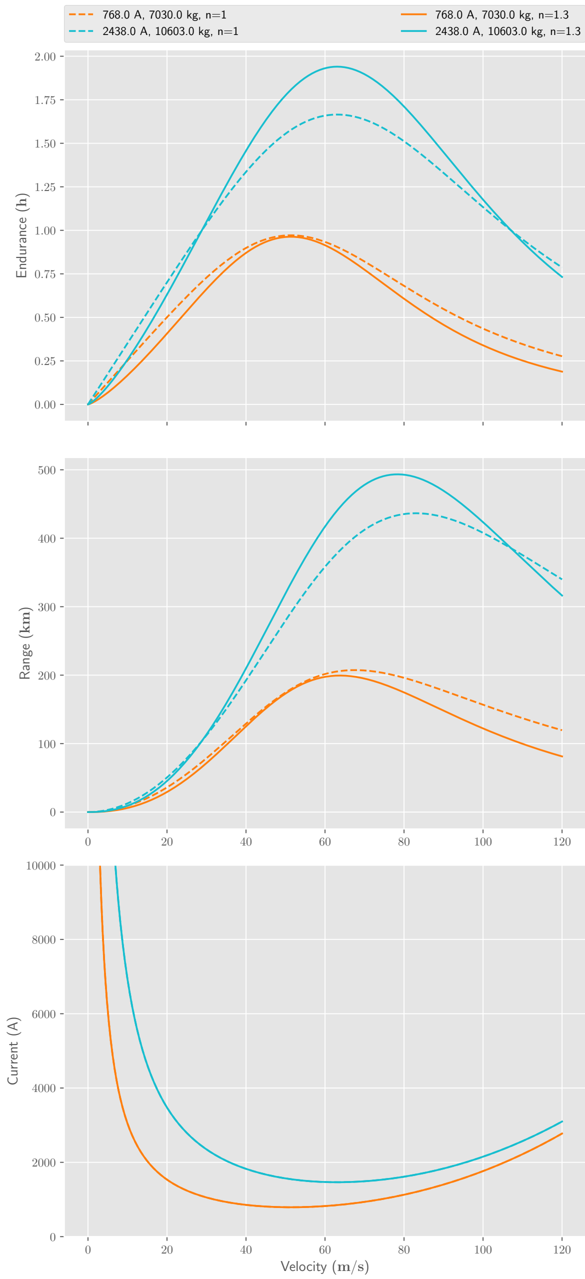


Figure 4.1: Range on Endurance based on airspeed 11 PAX (1100 kg payload).

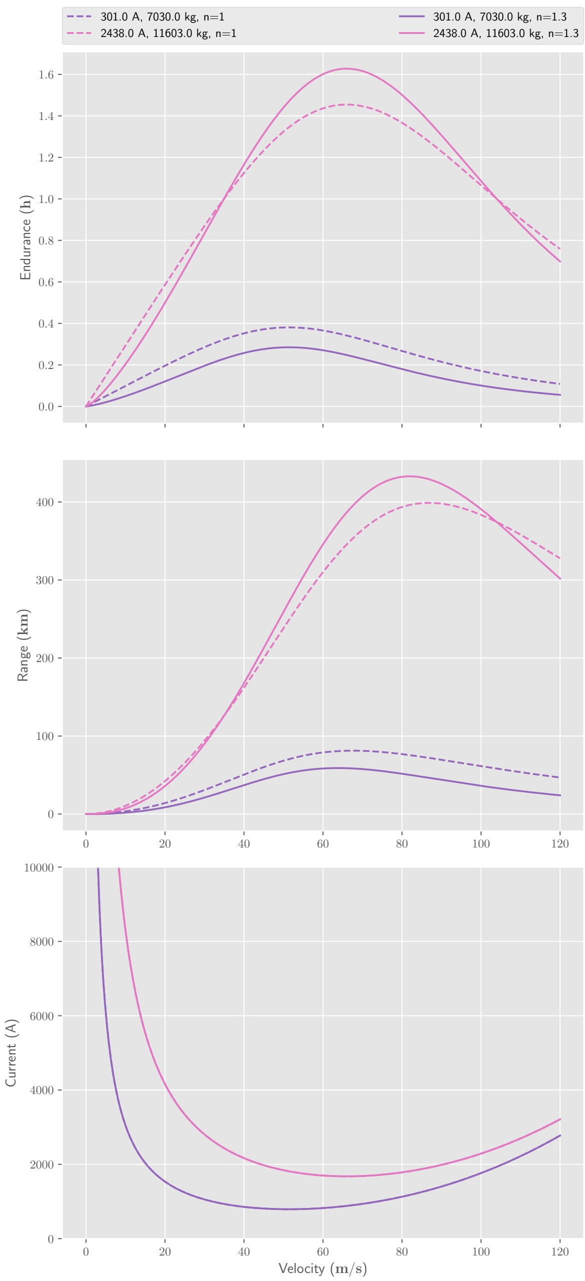


Figure 4.2: Range on Endurance based on airspeed 21 PAX (2100 kg payload).

The apex of each line indicates the maximum endurance it can achieve and its optimum airspeed for each cruise type and configuration. Now here on the apex line for the constant battery configurations (Electric C and D), we can clearly see that the battery with higher n rates benefits more when cruising at the optimum airspeed. This is in accordance with Peukert's effect. When a battery is discharged by a current lower than its capacity, it will last longer, but its capacity would last much shorter if a higher current discharges it. Note that at the apex of the endurance and range for the 2438 A config (Electric C and D), the battery capacity is only being discharged by a current lower than 2438 A, and when the discharge current rises above that, the dashed lines and continuous lines crossed paths.

Now the opposite effect can be seen with the 768 A battery capacities (Electric A). We can see that the continuous line almost touches the dashed line at the lowest current if we observe closely. Again this is in line with what Peukert's effect implies. Since the battery capacity is small (768 A), and the lowest current is still slightly higher than the battery capacity, the battery would be drained a bit faster compared to a battery with no Peukert's effect. One of the few ways to benefit from Peukert's effect is by increasing the battery capacities and reducing the discharge current. Increasing the battery capacity can be done in two ways.

The first is by adding more batteries, and more batteries mean more weight. An aircraft can only take so much weight before it damages the structural integrity of the aircraft, which for the case of N-219, only has 7030 MTOW and 6940 Maximum Landing Weight. Increasing the weight also resulted in an increase of current draw as well, since the current draw of the battery is related to the battery power, which is affected by the power required. The effect of weight on current can also be seen in the current section of Figure 4.1, here the current for 2438 A configuration (Electric C) is significantly higher than the current for 768 A configuration (Electric A). This is because the higher battery capacity configuration has a much heavier total weight at 10603 kg rather than 7030 kg of MTOW for the lower battery capacity configuration.

The other way to increase the battery capacity is by using better batteries that have a higher specific energy. The problem is that this can only be done when there is a new battery technology available to the market, in which a significant improvement only happens once in a few years.

Suppose we focus only on the endurance part. In that case, the gap for the Peukert's effect and the perfect battery is around 10%, which if we have much bigger battery capacities, the difference would be more significant. Realistically speaking, the 768 A configurations (Electric A) are the most reasonable estimation for an electric N-219 without deviating too much from the basic configuration for current technology. With 11 occupants, the 768 A configuration (Electric A) could approximately achieve a one-hour flight. While the 2438 A configuration (Electric C) has more than double the battery capacity of the 768A configuration (Electric A), the added weight from the extra battery capacities contributes to the increase of power required, which burns more power.

We can see that the maximum endurance for battery with n = 1.3 and n = 1 differs depends on the gap between the battery capacities and the discharged current, which when both are the same, the line intersect. The 768 A configuration (Electric A) has an optimum airspeed of around 50 m/s and can achieve a maximum endurance of almost 1 hour. At the same time, the 2438 A configuration (Electric C) has a higher optimum airspeed of approximately 60 m/s and a much higher endurance of around 1.75 hours. But despite that, for each configuration, the optimum airspeed for n = 1.3 and n = 1 is roughly the same. Thus the optimum airspeed seems only to be affected by the total aircraft's weight.

Things are a bit different if we look at the range section, we can roughly see that the optimum airspeed for maximum endurance differs even for different Peukert'seffectst. Peukert's effect is clearly seen to be more beneficial for the configurations that use less power and current. Meanwhile, the batteries with $n = 1$, or the "perfect" battery, do not benefit from a higher current or lower current. From here, we can understand that Peukert's effect can be a double edge sword for electric aircraft. Peukert's effect significantly benefits the aircraft when the aircraft is the most optimum airspeed or at economical cruising. Still, it can suffer a loss when the discharge current is more powerful than the battery capacity. Thus, take-off, climbing, and maneuvering can drain batteries with a higher n rate much faster.

Now the downfall of Peukert's effect really shows in Figure 4.2. Since the 301 A configuration (Electric B) has a much lower battery capacity than the discharge current, the gap between the perfect battery and the battery with $n = 1.3$ is more significant. Peukert's effect makes the aircraft lose much more range and endurance than it should have if using a perfect battery.

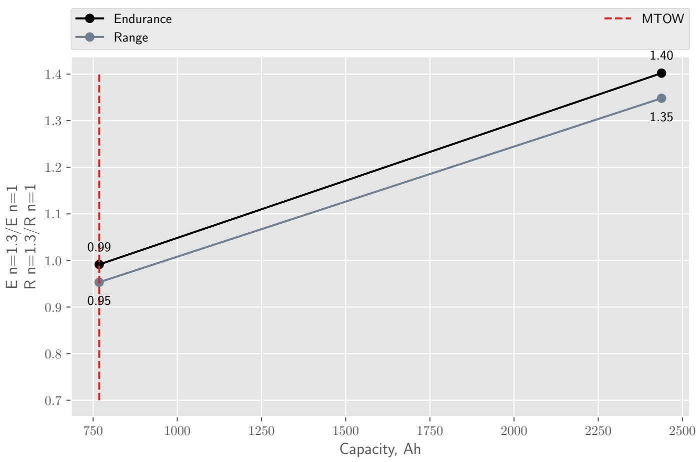


FIGURE 4.3: Effect of battery n-rate (11 PAX).

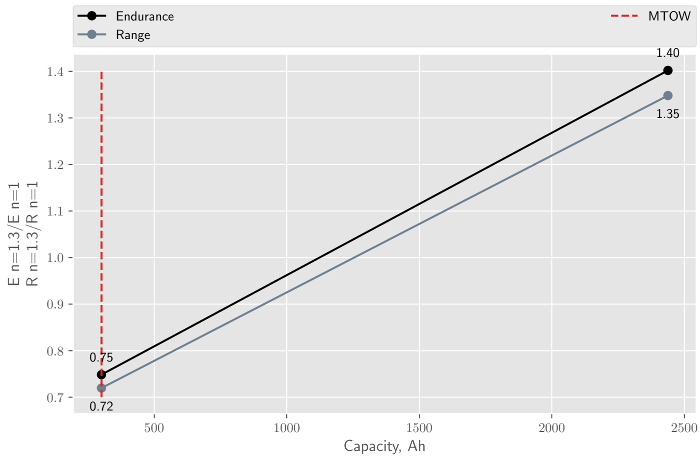


FIGURE 4.4: Effect of battery n-rate (21 PAX).

Figures 4.3 and 4.4 represent the Peukert's effect for different battery capacities. Take the peak of the continuous lines from Figure 4.1 and 4.2, then divide it by the peak of the dashed line, we got Figure 4.3 and 4.4. Here we can observe in a more simplified manner on how Peukert's effect behaves. As mentioned before, Peukert's effect does favour bigger batteries more. Both Figure 4.3 and 4.4 shows that the bigger battery capacities can have up to 30% better performance during maximum range and endurance cruise, whilst the batteries that have lower capacity than its discharge current suffer almost 30% performance loss depending on its payload.

Thus to maximize Peukert's effect, we must have the biggest batteries we can have, and we must achieve the lowest possible n rate. For weight, as previously mentioned, we will need better batteries with higher specific energy; Or modify the aircraft to accommodate higher take-off and landing weight. But if we change the N-219 in such a way, would it still be an N-219 aircraft? or would it be a completely new model? Well, it's a discussion for another topic. For now, we will be focusing on the performance aspect of the electric N-219.

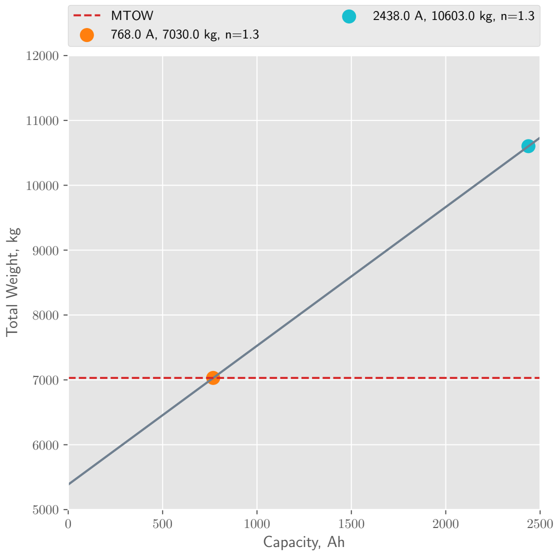


Figure 4.5: Effect of increasing battery capacity on weight (11 PAX).

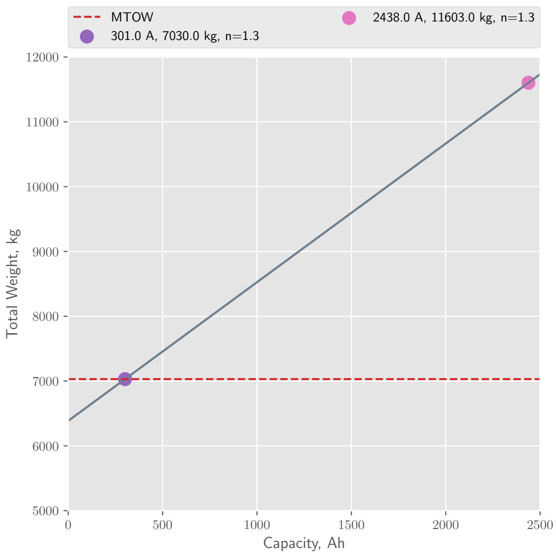


Figure 4.6: Effect of increasing battery capacity on weight (21 PAX).

Speaking of weight, figure 4.5 and 4.6 shows the relation between battery capacities and total weight. The red dashed lines represent the N-219 maximum take-off weight, which if the N-219 were to be operated above that line, it would pose problems and potentially suffer permanent damage. We can clearly see that the configurations with only 11 occupants (Electric A) can store significantly more batteries with more than double the capacity of the 21 occupants (Electric B)of the same total weight. At 11 occupants, the configuration with 2438 A battery capacities (Electric C) exceed the maximum take-off weight by more than three tonnes and even more for the 21 occupants payload (Electric D). That extra weight is almost half of the maximum take-off weight. Unless the N-219 were to be structurally configured to support those extra weight, there is zero chance the 2438 A battery capacity configuration (Electric C and D) would be able to fly. These results will be crucial for later discussion.

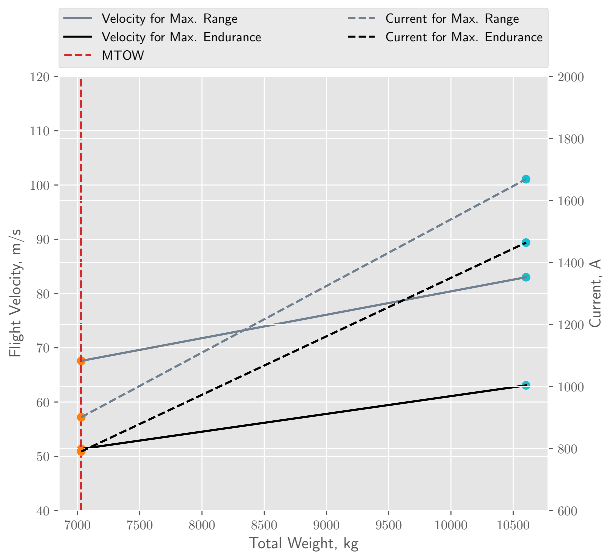


FIGURE 4.7: Effect of increasing total weight on optimum airspeed and currente (11 PAX).

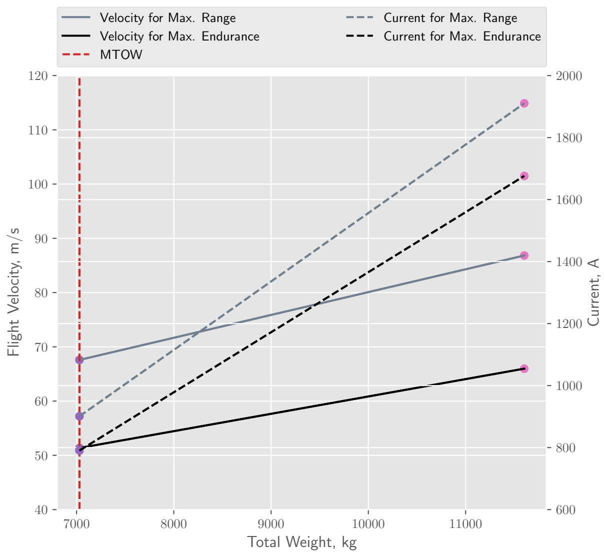


Figure 4.8: Effect of increasing total weight on optimum airspeed and current (21 PAX).

Figure 4.7 and 4.8 represent the change in optimum airspeed according to total aircraft weight. As previously mentioned, the optimum airspeed increases as the weight increases so do the current. This is because at cruise lift is equal to the weight, Equation 2.74; Thus to maintain an efficient level flight, a faster airspeed is required, which leads to more power required and more current being drawn, Equation 2.96, 2.167, and 2.164.

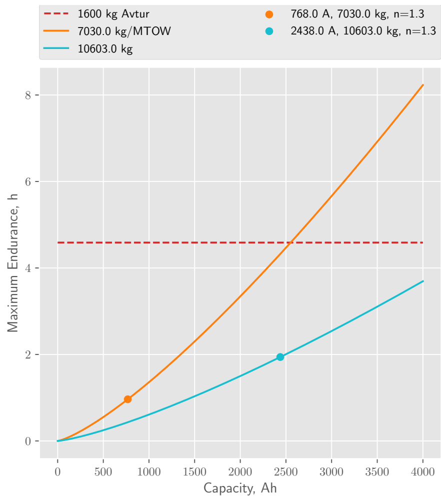


FIGURE 4.9: Effect of increasing battery capacity on maximum endurance (11 PAX).

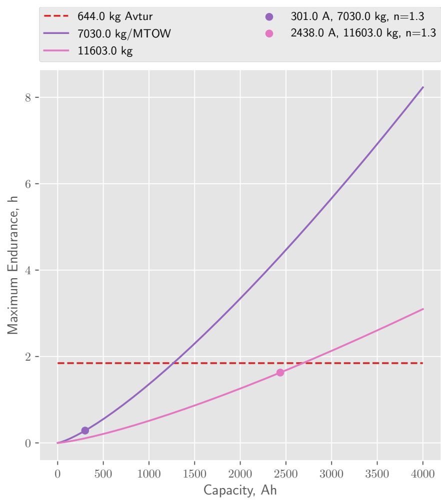


FIGURE 4.10: Effect of increasing battery capacity on maximum endurance (21 PAX).

Figure 4.9 and 4.10 show the scaling of battery capacity and its maximum achievable endurance. Each line represents the total weight for its configurations, and the dot represents where we are at with current battery technology. If we are going to electrify the N-219 aircraft, we want to have the longest endurance and farthest range possible. These plots show what we can expect from future battery technology. Notice that the line is not linear; that is because of Peukert's effect. Since the weight is constant, the current will also be consistent. Thus if the battery capacity were to be increased while maintaining the total weight, Peukert's effect would be more significant. We can see that the maximum achievable endurance is more than double the initial value for the 7030 kg total weight (Electric A and B) if the battery capacity were doubled. But as mentioned before, the increase of a battery specific energy happens gradually at a slow rate.

The only option to increase the battery capacity with current technology is by adding more batteries, which we could see if we observe the 2438 A configurations (Electric C and D) from Figure 4.9 and 4.10. Even though the battery capacity is more than double, the total achievable maximum endurance is not doubled due to the added power required from extra weight. This excess weight is the trade-off we must embrace in order to achieve better range and endurance using current battery technology.

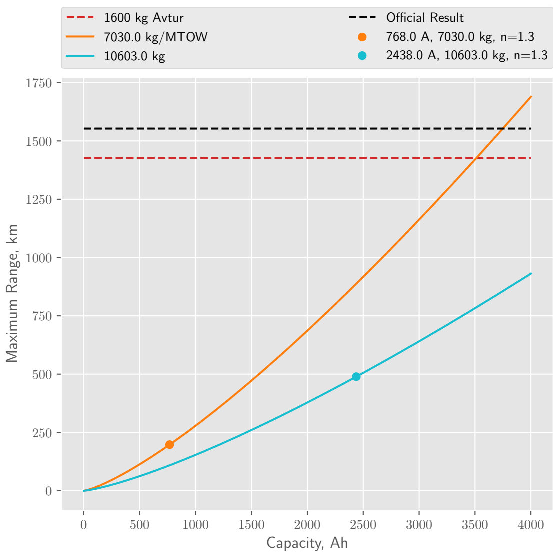


FIGURE 4.11: Effect of increasing battery capacity on maximum range (11 PAX).

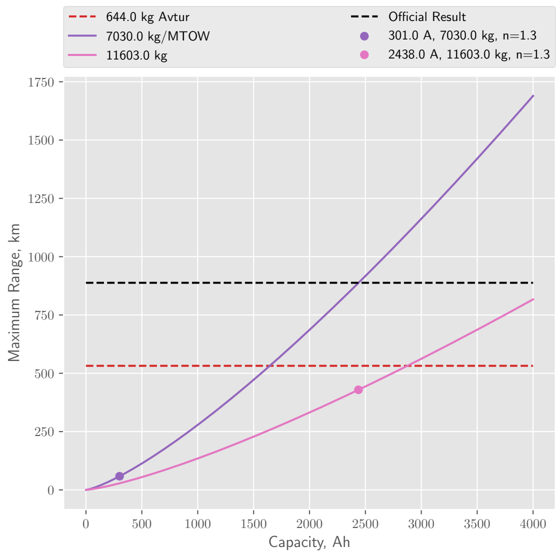


FIGURE 4.12: Effect of increasing battery capacity on maximum range (21 PAX).

Although the near one-hour maximum endurance might not sound that convincing, the result for the maximum range gives a gist of hope for electric N-219. Figure 4.11 shows that at current battery technology with MTOW as consideration (Electric A), the N-219 aircraft could achieve around 200km of range. The result might not mean much compared to the airbreathing counterpart, but it is still quite good for an electrified aircraft.

The following Table 4.3, 4.4, and 4.4 will show about the maximum endurances and their respective optimum airspeed for electric N-219 on various altitude with constant lift-drag ratio. Where Table 4.6, 4.7, and 4.7 is for the maximum range. Note that these tables does not include the overweight configuration (Electric C and D) that maximize the battery capacity, thus all weight are at MTOW of 7030 kg. The tables also include a 7 PAX payload configuration, this is also to see how much more benefit can the electric N-219 obtain from a bigger payload reduction and battery capacity increase. Which is shown that the payload reduction is very beneficial, reducing the payload from 11 PAX to 7 PAX allows for lager battery capacities. The endurance could be increased by around half an hour and the range could be increased by up to 70 km depending on the altitude.

The most notable thing among these tables is that the range and endurance much higher at sealevel compared to higher altitude, contrary to the airbreathing performance where higher altitude resulted in better range and endurance. These result is greatly unexpected considering the airbreathing results. This phenomenon is likely caused by the current which increases exponentially as the airspeed increases. The optimum airspeed, current, and power required increase by up to 50% more depending on the altitude for both maximum range and endurance. Wile the maximum endurance itself is decreased by up to 40% and 20% for the maximum range. The only major difference between the maximum endurance and maximum range table are the lift drag and the optimum airspeed.

TABLE 4.3: N-219 electric results for maximum endurance, 7 PAX

Altitude	ρ	C_L	C_D	Pr	I	V	Endurance
0.0	1.225	1.393	0.143	418.219	679.32	44.132	1.557
5000.0	1.056	1.393	0.143	450.539	731.819	47.543	1.413
10 000.0	0.905	1.393	0.143	486.67	790.506	51.355	1.279
15 000.0	0.771	1.393	0.143	527.225	856.381	55.635	1.152
20 000.0	0.653	1.393	0.143	572.95	930.652	60.46	1.034
25 000.0	0.549	1.393	0.143	624.751	1014.793	65.926	0.924

Table 4.4: N-219 electric results for maximum endurance, 11 PAX

Altitude	ρ	C_L	C_D	Pr	I	V	Endurance
0.0	1.225	1.393	0.143	418.219	679.32	44.132	1.173
5000.0	1.056	1.393	0.143	450.539	731.819	47.543	1.065
10 000.0	0.905	1.393	0.143	486.67	790.506	51.355	0.963
15 000.0	0.771	1.393	0.143	527.225	856.381	55.635	0.868
20 000.0	0.653	1.393	0.143	572.95	930.652	60.46	0.779
25 000.0	0.549	1.393	0.143	624.751	1014.793	65.926	0.696

Table 4.5: N-219 electric results for maximum endurance, 21 PAX

Altitude	ρ	C_L	C_D	Pr	I	V	Endurance
0.0	1.225	1.393	0.143	418.219	679.32	44.132	0.347
5000.0	1.056	1.393	0.143	450.539	731.819	47.543	0.315
10 000.0	0.905	1.393	0.143	486.67	790.506	51.355	0.285
15 000.0	0.771	1.393	0.143	527.225	856.381	55.635	0.257
20 000.0	0.653	1.393	0.143	572.95	930.652	60.46	0.231
25 000.0	0.549	1.393	0.143	624.751	1014.793	65.926	0.206

Table 4.6: N-219 electric results for maximum range, 7 PAX

Altitude	ρ	C_L	C_D	Pr	I	V	Range
0.0	1.225	0.804	0.071	476.666	774.257	58.081	274.656
5000.0	1.056	0.804	0.071	513.504	834.093	62.57	268.591
10 000.0	0.905	0.804	0.071	554.684	900.982	67.587	262.446
15 000.0	0.771	0.804	0.071	600.907	976.063	73.219	256.219
20 000.0	0.653	0.804	0.071	653.022	1060.714	79.57	249.906
25 000.0	0.549	0.804	0.071	712.062	1156.614	86.763	243.5

Table 4.7: N-219 electric results for maximum range, 11 PAX

Altitude	ρ	C_L	C_D	Pr	I	V	Range
0.0	1.225	0.804	0.071	476.666	774.257	58.081	206.897
5000.0	1.056	0.804	0.071	513.504	834.093	62.57	202.328
10 000.0	0.905	0.804	0.071	554.684	900.982	67.587	197.7
15 000.0	0.771	0.804	0.071	600.907	976.063	73.219	193.009
20 000.0	0.653	0.804	0.071	653.022	1060.714	79.57	188.253
25 000.0	0.549	0.804	0.071	712.062	1156.614	86.763	183.427

TABLE 4.8: N-219 electric results for maximum range, 21 PAX

Altitude	ρ	C_L	C_D	Pr	I	V	Range
0.0	1.225	0.804	0.071	476.666	774.257	58.081	61.224
5000.0	1.056	0.804	0.071	513.504	834.093	62.57	59.872
10 000.0	0.905	0.804	0.071	554.684	900.982	67.587	58.502
15 000.0	0.771	0.804	0.071	600.907	976.063	73.219	57.114
20 000.0	0.653	0.804	0.071	653.022	1060.714	79.57	55.707
25 000.0	0.549	0.804	0.071	712.062	1156.614	86.763	54.279

4.3 N-219 airbreathing vs electric range & endurance

The red dashed lines on Figures 4.9, 4.10, 4.11, and 4.12 represent the performance calculated for the airbreathing N-219 using same parameters, while the black dashed lines are the official result from Indonesian Aerospace regarding the N-219 performance. For 11 PAX endurance (Electric A), to catch up with the airbreathing performance, we need at least three times the specific energy of the current state-of-the-art battery, and quadrupling it will even outperform the airbreathing's performance. This is slightly different for the 21 PAX endurance (Electric B), as with current technology with 7030 MTOW, the electric N-219 can only achieve less than 1 hour with 301 A of battery. But in order to catch up with the 2-hour endurance of the airbreathing N-219, the battery technology must at least quadruple its specific energy to achieve around 1200 A worth of battery within the same total weight. If we disregard the MTOW as a limitation for research purposes, adding more battery to increase the battery capacity could potentially compete with the airbreathing endurance for 21 PAX (Airbreathing B). But it is to be noted that the increase in total weight would exceed 3 tonnes.

Surprisingly, the disparity between the electric and airbreathing performance is quite significant disparity for the range performance compared to the endurance performance. Using fossil fuel is still more than six times better than using current battery technology for 11 PAX (Electric A). Using current battery technology with 11 PAX (Electric A), the N-219 can achieve around 200 km of maximum range while using fossil fuel (Airbreathing A) could achieve more than 1400 km of range depending on the conditions. This means that for electric commuter aircraft to compete with its airbreathing counterpart in terms of range performance, it will need to wait several decades of battery development unless some heavy modification were made to the aircraft, such as increasing the MTOW and Landing weight, increasing wing area, or even making a new aircraft from scratch. A range of 200 km can still be utilized in many ways. Things such as commuter flight or other short-range mission that are less than 200 km can significantly benefit from electric N-219.

Table 4.9 and 4.10 were made using the same airspeed as Table 4.1 and 4.2 in order to have a coherent comparison. The electric endurance and range seem to be performing well in 5000 ft compared to other altitudes, but the airbreathing result is still far superior compared to the electric result. And again, the airbreathing N-219 greatly benefit from higher altitude while the electric N-219 have a performance increase on lower altitudes. This might seem to limit the N-219 aircraft to fly at a low altitude if it were to be electrified. But then again, the altitude increase does not affect the electric N-219 range by a ton. Thus, it would not matter much if the N-219 were to be operated at a higher altitude.

Now Figure 4.13 and 4.14 shows both the MTOW (Electric A and B) and overweight (Electric C and D) configuration of electric N-219. Figure 4.14 seems to indicate that the overweight N-219 (Electric D) could almost compete with the airbreathing N-219 (Airbreathing B). But it is to be noted that the overweight N-219 (Electric D) has 4 tonnes more weight compared to the airbreathing N-219 at 7030 kg MTOW. Despite that, it is still good to know that an electric N-219 could achieve such a feat despite the added disadvantages. Furthermore, we can clearly see that for the range, the optimum airspeed differs by quite a lot for the airbreathing N-219 (Airbreathing B) and the electric N-219 (Electric B and D). While the electric N-219 has an optimum airspeed for a maximum range of around 70 m/s, the airbreathing N-219 has an optimum airspeed for a maximum range of around 100 m/s. This happens because of the significant increase of the current past 70 m/s for the electric N-219, while the increase in power required for airbreathing N-219 was likely to be offset by the fuel consumption.

Table 4.9: N-219 electric results at constant airspeed, 11 PAX

Altitude	ρ	C_L	C_D	Pr	I	V	Endurance	Range
0.0	1.225	0.712	0.064	510.414	829.074	61.733	0.905	201.196
0.0	1.225	0.523	0.051	646.645	1050.356	72.022	0.666	172.584
0.0	1.225	0.4	0.045	846.531	1375.034	82.311	0.469	138.971
0.0	1.225	0.316	0.041	1115.351	1811.684	92.6	0.328	109.238
0.0	1.225	0.256	0.039	1459.454	2370.616	102.889	0.231	85.57
0.0	1.225	0.212	0.038	1885.77	3063.088	113.178	0.166	67.457
5000.0	1.056	0.826	0.073	506.824	823.243	61.733	0.914	203.051
5000.0	1.056	0.607	0.056	614.637	998.365	72.022	0.711	184.358
5000.0	1.056	0.465	0.048	779.692	1266.467	82.311	0.522	154.653
5000.0	1.056	0.367	0.043	1005.742	1633.644	92.6	0.375	124.962
5000.0	1.056	0.297	0.041	1297.778	2108.002	102.889	0.269	99.68
5000.0	1.056	0.246	0.039	1661.466	2698.748	113.178	0.195	79.528
10 000.0	0.905	0.964	0.087	514.98	836.491	61.733	0.895	198.88
10 000.0	0.905	0.708	0.063	595.862	967.869	72.022	0.74	191.945
10 000.0	0.905	0.542	0.052	728.683	1183.612	82.311	0.57	168.872
10 000.0	0.905	0.428	0.046	915.697	1487.382	92.6	0.423	141.167
10 000.0	0.905	0.347	0.042	1160.606	1885.193	102.889	0.311	115.26
10 000.0	0.905	0.287	0.04	1467.902	2384.338	113.178	0.229	93.423

Table 4.10: N-219 electric results at constant airspeed, 21 PAX

Altitude	ρ	C_L	C_D	Pr	I	V	Endurance	Range
0.0	1.225	0.712	0.064	510.414	829.074	61.733	0.268	59.537
0.0	1.225	0.523	0.051	646.645	1050.356	72.022	0.197	51.07
0.0	1.225	0.4	0.045	846.531	1375.034	82.311	0.139	41.123
0.0	1.225	0.316	0.041	1115.351	1811.684	92.6	0.097	32.325
0.0	1.225	0.256	0.039	1459.454	2370.616	102.889	0.068	25.321
0.0	1.225	0.212	0.038	1885.77	3063.088	113.178	0.049	19.961
5000.0	1.056	0.826	0.073	506.824	823.243	61.733	0.27	60.086
5000.0	1.056	0.607	0.056	614.637	998.365	72.022	0.21	54.554
5000.0	1.056	0.465	0.048	779.692	1266.467	82.311	0.154	45.764
5000.0	1.056	0.367	0.043	1005.742	1633.644	92.6	0.111	36.978
5000.0	1.056	0.297	0.041	1297.778	2108.002	102.889	0.08	29.497
5000.0	1.056	0.246	0.039	1661.466	2698.748	113.178	0.058	23.534
10 000.0	0.905	0.964	0.087	514.98	836.491	61.733	0.265	58.852
10 000.0	0.905	0.708	0.063	595.862	967.869	72.022	0.219	56.799
10 000.0	0.905	0.542	0.052	728.683	1183.612	82.311	0.169	49.972
10 000.0	0.905	0.428	0.046	915.697	1487.382	92.6	0.125	41.773
10 000.0	0.905	0.347	0.042	1160.606	1885.193	102.889	0.092	34.107
10 000.0	0.905	0.287	0.04	1467.902	2384.338	113.178	0.068	27.645

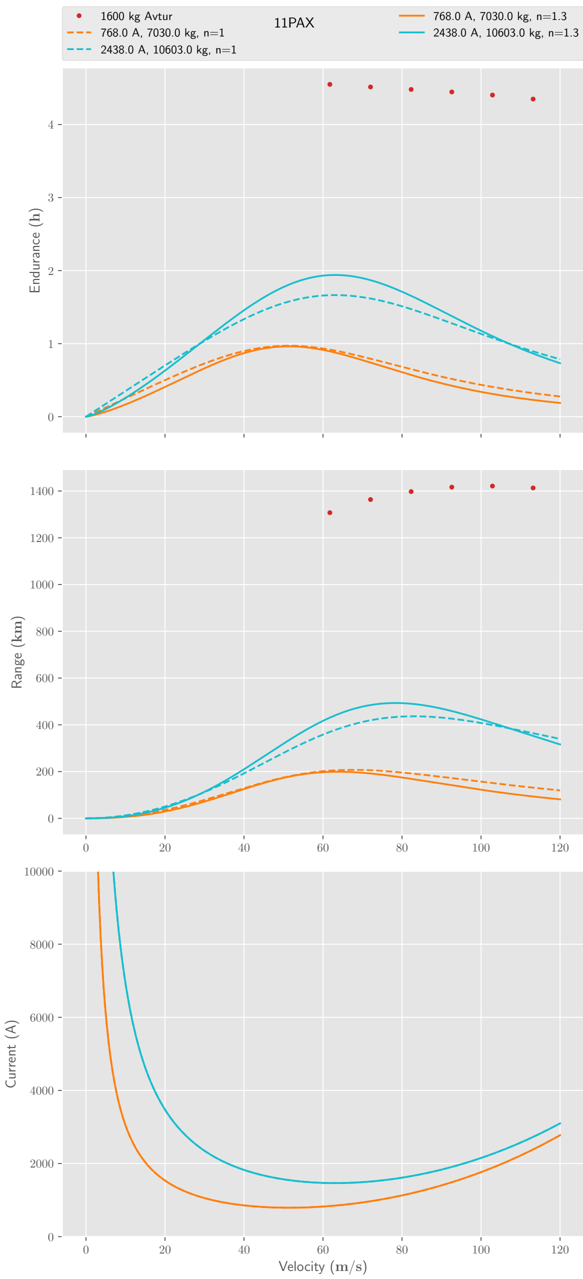


FIGURE 4.13: Electric and airbreathing N-219 comparison (11 PAX).

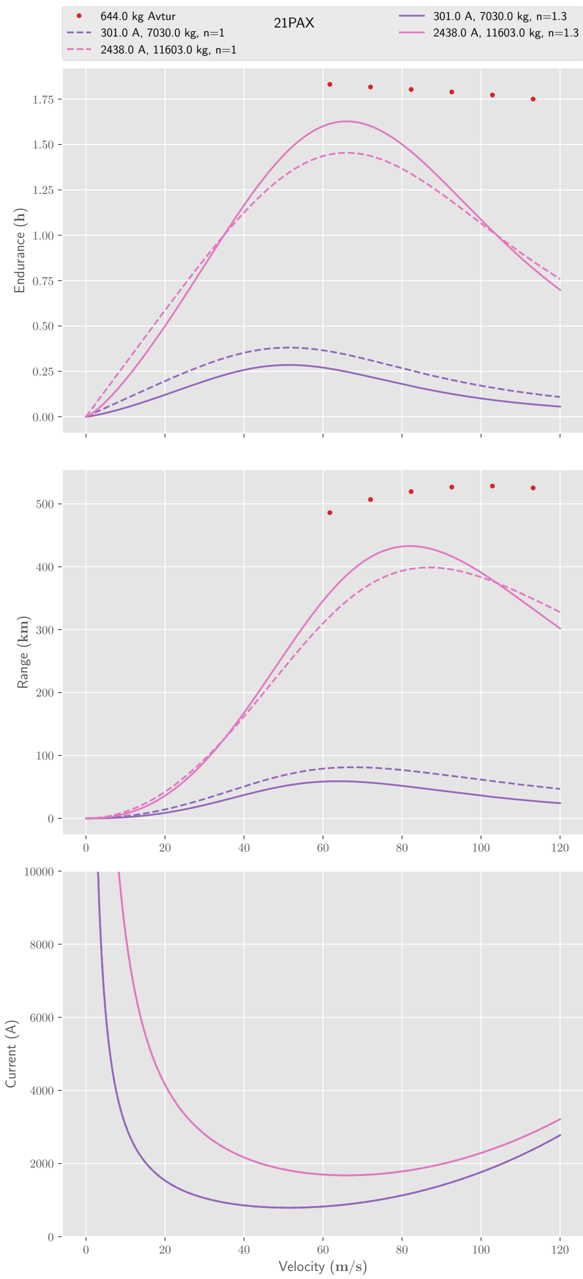


Figure 4.14: Electric and airbreathing N-219 comparison (21 PAX).

CHAPTER 5
SUMMARY, CONCLUSION, RECOMMENDATION

5.1 Summary

The following is the summarization of this thesis that is based on what has been shown and discussed in it:

1. The author managed to make an ISA computational model to expedite other calculations and make it more flexible.
2. The author has successfully obtained the engine data for N-219 and process it to produce range and endurance results in order to reconfirm the accuracy of the data.
3. The aerodynamic properties, especially the zero-lift drag coefficient and Oswald's efficiency, has been meticulously calculated in order to obtain the most precise and accurate data.
4. Various battery types and specifications have been collected and compared in order to find the best in slot for the electric N-219 aircraft.
5. The effect of Peukert's law has been properly researched and integrated into the electric aircraft equation in order to retrieve a more realistic result.

6. Successfully asses the Peukert's effect from the electric N-219 results and compare the effect according to different configurations.

7. The author has successfully evaluate the result from the electric N-219 range and endurance and compare it with the airbreathing N-219 result.

5.2 Conclusion

Many interesting things were unveiled within Chapter 4. How the electric N-219 benefit more from a lower altitude and the airbreathing N-219 benefit more from higher altitude is a very interesting phenomenon. It seems that the airbreathing result was heavily affected by the specific fuel consumption, while the electric aircraft does not possess such things. This also happens because of peukert's effect, since higher altitude requires higher velocity, more current draw is required. Thus, for electric N-219, increasing altitude reduce the maximum range and endurance.

Then there is a difference between optimum airspeed for airbreathing and electric N-219. The difference was around 30-40 m/s, which is significant. The reason why the electric N-219 could not achieve a similar airspeed is due to the substantial increase of current past 70 m/s and the effect of fuel efficiency for the airbreathing engine. Thus, increasing airspeed would increase the current exponentially.

For the performance comparison, at best scenario the electric N219 performs around 1/5 the performance of the airbreathing N219, and that is on a sea level altitude. At worst, the N219 performs around 1/15 the performance of the airbreathing N219, this happens because the airbreathing performance increase significantly at higher altitude, while the electric performance suffers on higher altitude.

Peukert's effect can affect the performance by quite a lot, depending on the situation it can boost or reduce the performance by 30%. Increasing battery capacity would significantly increase the maximum range and endurance of electric N-219.

Increasing the battery weight can result in a significant increase in performance, but the weight increase is also significant which according to configuration Electric D can lead to 4 ton of excess weight. Furthermore, increasing the payload capacity while maintaining MTOW will significantly reduce the achieveable range of the electric N-219.

So, should the N-219 be made an all-electric variant? Well, the answer depends on the situation and use case of the aircraft. Suppose we are talking about a long-range flight of around 1000 km. In that case, the answer is no, not by using current battery technology, at least unless the N-219 were to be modified in such ways that it could handle the heavier weight and have a bigger wing area to obtain greater lift with less power. But if the N-219 would be used for flights around 200 km, then electrifying the N-219 could bring many benefits.

Nevertheless, if there is a plan for an electric N-219 somewhere in the future, there's nothing wrong to start the development process of electric N-219 as early as possible. That is because research and development, legalization and other stuff might take some times. Perhaps, when the extensive research and development were done, by then, a battery technology such as lithium-sulfur, which double the capacity of the current battery, might already be in the mass market. Especially with the added benefits of Peukert's effect, any increment to the battery capacity would greatly benefit the range and endurance of the aircraft.

5.3 Recommendation

1. The first recommendation I would give is that doing market research for flight around 200 km would be a great start for utilizing the electric N-219. If there is a high demand for flight around 200 km, then an electric N-219 is very plausible and beneficial.

2. A research on other method or technology to improve cruising performance while maintaining MTOW is recommended. For example, research on how solar panel on the wing and upper body of the N-219 would affect its performance could be a great option.

3. Since the Peukert's effect does affect the performance by quite a lot (up to 30% depending on the situation, and likely more for a much bigger/lower battery capacity), It is recommended to atleast have a battery capacity as big as the cruising current draw in order to not have any performance reduction.

4. If lithium-sulfur were to be mass-produced it could give more flexibility to the electric N-219 mission profile as it can double the range and endurance of the electric N219.

5. If one would expand on this thesis, an airfield performance and Landing Take-off analysis is necessary for configuration with 5 tonnes of battery. Other things such as climbing performance and maximum airspeed would also give a broader perspective about electric N-219 aircraft.

6. The data for airbreathing N-219 was a bit limited by an airspeed increment of 20 knots. Thus if obtaining more broad data is possible, then it would greatly increase the observe-ability of the airbreathing N-219.

References

Aerospace, I. (n.d.-a). N219 nurtanio. Retrieved from https://www.indonesian-aerospace.com/aircraft/detail/11_n219nurtanio

Aerospace, I. (n.d.-b). *Ptdi internal report* (Tech. Rep.). Author.

Alcock, C. (2020, Jan). *Eviation's electric alice aircraft catches fire during ground tests.* Retrieved from https://www.ainonline.com/aviation-news/business-aviation/2020-01-24/eviations-electric-alice-aircraft-catches-fire-during-ground-tests

Anderson, J. D. (n.d.). *Fundamentals of aerodynamics* (5th ed.). McGraw-Hill.

Calderwood, D. (2020, May). *First flight for all-electric magnix ecaravan.* Retrieved from https://www.flyer.co.uk/first-flight-for-magnix-ecaravan-with-all-electric-power/

Garrett-Glaser, B. (2020, Jun). *First flight of magnix ecaravan showcases maturity of electric aviation.* Aviation Today. Retrieved from https://www.aviationtoday.com/2020/05/29/historic-flight-of-magnixs-ecaravan-showcases-maturity-of-electric-aviation/

Jackson, R. (2021a). The causes of climate change. *Climate Change: Vital Signs of the Planet*. Retrieved from https://climate.nasa.gov/causes/

Jackson, R. (2021b). The effects of climate change. *Climate Change: Vital Signs of the Planet*. Retrieved from https://climate.nasa.gov/effects/

Lavars, N. (2019, Oct). *Magnix's 750-hp electric motor turns its first propellor at full bore.* Retrieved from https://newatlas.com/aircraft/magnixs-750-hp-electric-motor-first-propellor/

Loftin, L. K. (1980). *Subsonic aircraft: Evolution and the matching of size to performance.* National Aeronautics and Space Administration, Scientific and Technical Branch.

McCormick, B. W. (1994). *Aerodynamics, aeronautics, and flight mechanics* (2nd ed.). Wiley.

Meszaros, J. (2019, Mar). *Indonesia's n219 on track for 2019 certification.* Retrieved from https://www.ainonline.com/aviation-news/air-transport/2019-03-29/indonesias-n219-track-2019-certification

N219 solution for frontier gateway. (2018, January). L Pajajaran No. 154 Bandung, 40174, Indonesia: PT. Dirgantara Indonesia (Persero).

Narishkin, A. (2020, Nov). *Inside a 4 million electric plane, the first full-size, all-electric passenger aircraft in the world.* Business Insider. Retrieved from https://www.businessinsider.com/inside-alice-first-full-size-passenger-electric-plane-eviation-2020-10

Pearce, B. (2017, April). *State of the airlineindustry.* Retrieved from https://www.iata.org/en/iata-repository/publications/economic-reports/state-of-the-airline-industry/

Pesawat n219 melengkapi jam terbang sebelum diproduksi massal: Ekonomi. (2019, Jul). Retrieved from https://ekonomi.bisnis.com/read/20190717/257/1125325/pesawat-n219-melengkapi-jam-terbang-sebelum-diproduksi-massal

Pilot's handbook of aeronautical knowledge. (2016, August). Federal Aviation Administration. Retrieved from https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/phak/

Raymer, D. P. (2006). *Aircraft design: A conceptual approach* (6. ed. ed.). Reston, Va.: American Institute of Aeronautics and Astronautics. Retrieved from http://www.loc.gov/catdir/toc/ecip068/2006004706.html

Ruijgrok, G. J. J. (2009). *Elements of airplane performance* (Second ed.). Vssd Pub.

Saarlas, M. (2007). *Aircraft performance*. John Wiley.

Sari, M. (2018). Optimasi coefficient lift (cl) single slotted flap dan double slotted flap pada pesawat n219 menggunakan simulasi computational fluid dynamic. *Repository Universitas Jember*. Retrieved from https://repository.unej.ac.id/bitstream/handle/123456789/85673/MirnaSari.pdfSDH.pdf?sequence=1&isAllowed=y

Schäfer, A. W., Barrett, S. R., Doyme, K., Dray, L. M., Gnadt, A. R., Self, R., ... Torija, A. J. (2019). Technological, economic and environmental prospects of all-electric aircraft. *Nature Energy*, *4*(2), 160–166.

Statistical summary of commercial jet airplane accidents. (n.d.). Boeing. Retrieved from https://www.boeing.com/resources/boeingdotcom/company/about_bca/pdf/statsum.pdf

Traub, L. W. (2011, March). Range and endurance estimatesfor battery-powered aircraft. *JOURNAL OF AIRCRAFT*, *48*, 703-707.

Types of lithium-ion batteries. (2021). *BATTERY UNIVERSITY*. Retrieved from https://batteryuniversity.com/learn/article/types_of_lithium_ion

Viswanathan, V., & Knapp, B. M. (2019). Potential for electric aircraft. *Nature Sustainability*, *2*(2), 88–89.

Working towards ambitious targets. (n.d.). Retrieved from https://www.iata.org/en/programs/environment/climate-change/

Appendices

Appendix A: Tables

Table 1: PT6A-42 Data 70% MCR

RPM	RC	ISA C	ALT ft	V kn	THRT kgf	FF lb/hr	η_p
2000.00	70.00	-10.00	0.00	100.00	725.66	471.37	0.7340
2000.00	70.00	-10.00	0.00	120.00	657.31	475.02	0.7860
2000.00	70.00	-10.00	0.00	140.00	600.29	478.23	0.8205
2000.00	70.00	-10.00	0.00	160.00	552.45	480.90	0.8435
2000.00	70.00	-10.00	0.00	180.00	510.30	483.50	0.8570
2000.00	70.00	-10.00	0.00	200.00	473.23	486.51	0.8655
2000.00	70.00	-10.00	0.00	220.00	444.40	490.86	0.8685
2000.00	70.00	-10.00	5000.00	100.00	698.55	427.81	0.7170
2000.00	70.00	-10.00	5000.00	120.00	636.84	430.61	0.7735
2000.00	70.00	-10.00	5000.00	140.00	581.85	433.03	0.8120
2000.00	70.00	-10.00	5000.00	160.00	534.79	435.18	0.8390
2000.00	70.00	-10.00	5000.00	180.00	494.09	437.23	0.8565
2000.00	70.00	-10.00	5000.00	200.00	458.00	439.92	0.8670
2000.00	70.00	-10.00	5000.00	220.00	428.88	443.45	0.8725
2000.00	70.00	-10.00	10000.00	100.00	629.98	366.39	0.7080
2000.00	70.00	-10.00	10000.00	120.00	575.44	369.34	0.7660
2000.00	70.00	-10.00	10000.00	140.00	529.00	372.30	0.8070
2000.00	70.00	-10.00	10000.00	160.00	488.94	375.34	0.8345

2000.00	70.00	-10.00	10000.00	180.00	453.19	378.49	0.8535
2000.00	70.00	-10.00	10000.00	200.00	423.58	382.72	0.8655
2000.00	70.00	-10.00	10000.00	220.00	397.51	387.69	0.8730
2000.00	70.00	-10.00	15000.00	100.00	566.56	320.47	0.6970
2000.00	70.00	-10.00	15000.00	120.00	519.96	322.43	0.7565
2000.00	70.00	-10.00	15000.00	140.00	478.80	324.59	0.7990
2000.00	70.00	-10.00	15000.00	160.00	442.68	326.79	0.8295
2000.00	70.00	-10.00	15000.00	180.00	410.37	328.98	0.8495
2000.00	70.00	-10.00	15000.00	200.00	384.08	332.53	0.8630
2000.00	70.00	-10.00	15000.00	220.00	360.31	336.45	0.8715
2000.00	70.00	-10.00	20000.00	100.00	506.92	280.55	0.6840
2000.00	70.00	-10.00	20000.00	120.00	466.10	282.09	0.7460
2000.00	70.00	-10.00	20000.00	140.00	430.10	284.10	0.7905
2000.00	70.00	-10.00	20000.00	160.00	398.78	286.06	0.8225
2000.00	70.00	-10.00	20000.00	180.00	369.87	287.94	0.8450
2000.00	70.00	-10.00	20000.00	200.00	347.08	291.48	0.8595
2000.00	70.00	-10.00	20000.00	220.00	326.31	295.02	0.8695
2000.00	70.00	-10.00	25000.00	100.00	442.12	241.42	0.6735
2000.00	70.00	-10.00	25000.00	120.00	407.67	242.65	0.7375
2000.00	70.00	-10.00	25000.00	140.00	377.73	244.48	0.7840
2000.00	70.00	-10.00	25000.00	160.00	350.24	246.31	0.8175
2000.00	70.00	-10.00	25000.00	180.00	325.93	248.45	0.8405
2000.00	70.00	-10.00	25000.00	200.00	306.83	252.20	0.8560
2000.00	70.00	-10.00	25000.00	220.00	288.68	255.85	0.8670
2000.00	70.00	0.00	0.00	100.00	659.44	447.31	0.7410
2000.00	70.00	0.00	0.00	120.00	596.61	450.91	0.7910
2000.00	70.00	0.00	0.00	140.00	543.04	454.25	0.8230
2000.00	70.00	0.00	0.00	160.00	499.34	457.11	0.8435

2000.00	70.00	0.00	0.00	180.00	461.64	459.84	0.8560
2000.00	70.00	0.00	0.00	200.00	427.70	462.56	0.8625
2000.00	70.00	0.00	0.00	220.00	400.85	466.68	0.8650
2000.00	70.00	0.00	5000.00	100.00	646.85	407.50	0.7235
2000.00	70.00	0.00	5000.00	120.00	588.57	410.35	0.7780
2000.00	70.00	0.00	5000.00	140.00	536.24	412.82	0.8160
2000.00	70.00	0.00	5000.00	160.00	492.90	414.90	0.8410
2000.00	70.00	0.00	5000.00	180.00	456.05	416.89	0.8570
2000.00	70.00	0.00	5000.00	200.00	420.51	419.18	0.8665
2000.00	70.00	0.00	5000.00	220.00	394.44	422.51	0.8715
2000.00	70.00	0.00	10000.00	100.00	584.54	348.81	0.7145
2000.00	70.00	0.00	10000.00	120.00	533.90	351.67	0.7715
2000.00	70.00	0.00	10000.00	140.00	489.59	354.43	0.8110
2000.00	70.00	0.00	10000.00	160.00	451.62	357.11	0.8375
2000.00	70.00	0.00	10000.00	180.00	418.48	359.86	0.8555
2000.00	70.00	0.00	10000.00	200.00	390.36	363.32	0.8665
2000.00	70.00	0.00	10000.00	220.00	366.71	367.93	0.8725
2000.00	70.00	0.00	15000.00	100.00	531.90	306.22	0.7020
2000.00	70.00	0.00	15000.00	120.00	486.82	308.18	0.7610
2000.00	70.00	0.00	15000.00	140.00	447.30	310.14	0.8030
2000.00	70.00	0.00	15000.00	160.00	413.65	312.10	0.8325
2000.00	70.00	0.00	15000.00	180.00	382.89	314.06	0.8520
2000.00	70.00	0.00	15000.00	200.00	357.73	316.86	0.8645
2000.00	70.00	0.00	15000.00	220.00	335.44	320.19	0.8725
2000.00	70.00	0.00	20000.00	100.00	479.27	268.63	0.6880
2000.00	70.00	0.00	20000.00	120.00	440.37	270.12	0.7490
2000.00	70.00	0.00	20000.00	140.00	405.80	271.93	0.7930
2000.00	70.00	0.00	20000.00	160.00	375.67	273.77	0.8250

2000.00	70.00	0.00	20000.00	180.00	348.60	275.52	0.8470
2000.00	70.00	0.00	20000.00	200.00	325.75	278.36	0.8605
2000.00	70.00	0.00	20000.00	220.00	306.08	281.46	0.8705
2000.00	70.00	0.00	25000.00	100.00	419.73	231.88	0.6765
2000.00	70.00	0.00	25000.00	120.00	386.71	233.08	0.7400
2000.00	70.00	0.00	25000.00	140.00	357.99	234.77	0.7860
2000.00	70.00	0.00	25000.00	160.00	332.07	236.42	0.8185
2000.00	70.00	0.00	25000.00	180.00	308.52	238.02	0.8420
2000.00	70.00	0.00	25000.00	200.00	290.20	241.13	0.8575
2000.00	70.00	0.00	25000.00	220.00	272.78	244.44	0.8680
2000.00	70.00	10.00	0.00	100.00	591.54	424.28	0.7480
2000.00	70.00	10.00	0.00	120.00	534.76	427.86	0.7940
2000.00	70.00	10.00	0.00	140.00	488.21	431.33	0.8235
2000.00	70.00	10.00	0.00	160.00	448.09	434.48	0.8420
2000.00	70.00	10.00	0.00	180.00	414.47	437.48	0.8525
2000.00	70.00	10.00	0.00	200.00	383.41	440.35	0.8570
2000.00	70.00	10.00	0.00	220.00	360.07	444.21	0.8580
2000.00	70.00	10.00	5000.00	100.00	593.04	387.13	0.7315
2000.00	70.00	10.00	5000.00	120.00	536.78	389.96	0.7840
2000.00	70.00	10.00	5000.00	140.00	489.80	392.49	0.8195
2000.00	70.00	10.00	5000.00	160.00	450.23	394.54	0.8430
2000.00	70.00	10.00	5000.00	180.00	415.41	396.47	0.8575
2000.00	70.00	10.00	5000.00	200.00	383.51	398.46	0.8655
2000.00	70.00	10.00	5000.00	220.00	359.81	401.80	0.8700
2000.00	70.00	10.00	10000.00	100.00	541.30	332.14	0.7215
2000.00	70.00	10.00	10000.00	120.00	493.98	334.97	0.7765
2000.00	70.00	10.00	10000.00	140.00	451.34	337.72	0.8145
2000.00	70.00	10.00	10000.00	160.00	416.24	340.31	0.8400

2000.00	70.00	10.00	10000.00	180.00	386.36	342.95	0.8570
2000.00	70.00	10.00	10000.00	200.00	357.99	345.98	0.8670
2000.00	70.00	10.00	10000.00	220.00	336.99	350.05	0.8720
2000.00	70.00	10.00	15000.00	100.00	496.60	291.98	0.7085
2000.00	70.00	10.00	15000.00	120.00	453.34	293.92	0.7655
2000.00	70.00	10.00	15000.00	140.00	416.26	295.70	0.8070
2000.00	70.00	10.00	15000.00	160.00	384.50	297.38	0.8355
2000.00	70.00	10.00	15000.00	180.00	355.84	299.15	0.8540
2000.00	70.00	10.00	15000.00	200.00	331.48	301.48	0.8660
2000.00	70.00	10.00	15000.00	220.00	310.59	304.50	0.8730
2000.00	70.00	10.00	20000.00	100.00	450.47	256.96	0.6930
2000.00	70.00	10.00	20000.00	120.00	413.58	258.41	0.7535
2000.00	70.00	10.00	20000.00	140.00	380.69	259.93	0.7970
2000.00	70.00	10.00	20000.00	160.00	351.79	261.47	0.8280
2000.00	70.00	10.00	20000.00	180.00	326.40	262.98	0.8490
2000.00	70.00	10.00	20000.00	200.00	304.58	265.18	0.8620
2000.00	70.00	10.00	20000.00	220.00	285.79	267.85	0.8720
2000.00	70.00	10.00	25000.00	100.00	398.40	222.83	0.6800
2000.00	70.00	10.00	25000.00	120.00	366.93	224.11	0.7435
2000.00	70.00	10.00	25000.00	140.00	338.82	225.70	0.7880
2000.00	70.00	10.00	25000.00	160.00	314.37	227.30	0.8205
2000.00	70.00	10.00	25000.00	180.00	292.00	228.83	0.8440
2000.00	70.00	10.00	25000.00	200.00	273.86	231.37	0.8585
2000.00	70.00	10.00	25000.00	220.00	257.76	234.15	0.8690
2000.00	70.00	20.00	0.00	100.00	519.58	402.21	0.7510
2000.00	70.00	20.00	0.00	120.00	468.58	405.74	0.7925
2000.00	70.00	20.00	0.00	140.00	425.35	409.27	0.8185
2000.00	70.00	20.00	0.00	160.00	392.58	412.76	0.8345

2000.00	70.00	20.00	0.00	180.00	365.13	416.18	0.8440
2000.00	70.00	20.00	0.00	200.00	338.71	419.47	0.8480
2000.00	70.00	20.00	0.00	220.00	318.99	423.13	0.8470
2000.00	70.00	20.00	5000.00	100.00	540.06	368.03	0.7390
2000.00	70.00	20.00	5000.00	120.00	488.70	370.83	0.7895
2000.00	70.00	20.00	5000.00	140.00	444.22	373.48	0.8220
2000.00	70.00	20.00	5000.00	160.00	408.39	375.70	0.8435
2000.00	70.00	20.00	5000.00	180.00	376.73	377.80	0.8560
2000.00	70.00	20.00	5000.00	200.00	348.34	379.77	0.8635
2000.00	70.00	20.00	5000.00	220.00	325.46	382.75	0.8655
2000.00	70.00	20.00	10000.00	100.00	498.03	315.54	0.7290
2000.00	70.00	20.00	10000.00	120.00	452.17	318.35	0.7820
2000.00	70.00	20.00	10000.00	140.00	413.44	321.15	0.8180
2000.00	70.00	20.00	10000.00	160.00	381.13	323.64	0.8420
2000.00	70.00	20.00	10000.00	180.00	353.39	326.19	0.8575
2000.00	70.00	20.00	10000.00	200.00	326.96	328.91	0.8665
2000.00	70.00	20.00	10000.00	220.00	308.77	332.89	0.8710
2000.00	70.00	20.00	15000.00	100.00	461.15	278.25	0.7150
2000.00	70.00	20.00	15000.00	120.00	420.78	280.21	0.7715
2000.00	70.00	20.00	15000.00	140.00	385.26	282.01	0.8110
2000.00	70.00	20.00	15000.00	160.00	354.87	283.61	0.8380
2000.00	70.00	20.00	15000.00	180.00	328.49	285.17	0.8560
2000.00	70.00	20.00	15000.00	200.00	305.06	286.99	0.8665
2000.00	70.00	20.00	15000.00	220.00	286.34	289.58	0.8725
2000.00	70.00	20.00	20000.00	100.00	423.34	245.88	0.6985
2000.00	70.00	20.00	20000.00	120.00	387.99	247.33	0.7580
2000.00	70.00	20.00	20000.00	140.00	356.42	248.73	0.8005
2000.00	70.00	20.00	20000.00	160.00	329.46	250.10	0.8310

2000.00	70.00	20.00	20000.00	180.00	304.93	251.43	0.8515
2000.00	70.00	20.00	20000.00	200.00	284.36	253.20	0.8645
2000.00	70.00	20.00	20000.00	220.00	266.17	255.53	0.8725
2000.00	70.00	20.00	25000.00	100.00	375.71	213.26	0.6850
2000.00	70.00	20.00	25000.00	120.00	345.71	214.45	0.7475
2000.00	70.00	20.00	25000.00	140.00	318.64	215.79	0.7915
2000.00	70.00	20.00	25000.00	160.00	295.61	217.22	0.8235
2000.00	70.00	20.00	25000.00	180.00	274.74	218.66	0.8455
2000.00	70.00	20.00	25000.00	200.00	256.43	220.72	0.8605
2000.00	70.00	20.00	25000.00	220.00	241.35	223.19	0.8700
2000.00	70.00	25.00	0.00	100.00	484.24	391.56	0.7500
2000.00	70.00	25.00	0.00	120.00	437.20	395.18	0.7895
2000.00	70.00	25.00	0.00	140.00	395.23	398.80	0.8130
2000.00	70.00	25.00	0.00	160.00	365.27	402.49	0.8280
2000.00	70.00	25.00	0.00	180.00	340.32	406.05	0.8370
2000.00	70.00	25.00	0.00	200.00	315.43	409.49	0.8400
2000.00	70.00	25.00	0.00	220.00	298.16	413.19	0.8395
2000.00	70.00	25.00	5000.00	100.00	513.84	358.78	0.7425
2000.00	70.00	25.00	5000.00	120.00	464.68	361.58	0.7915
2000.00	70.00	25.00	5000.00	140.00	422.33	364.26	0.8230
2000.00	70.00	25.00	5000.00	160.00	387.72	366.59	0.8435
2000.00	70.00	25.00	5000.00	180.00	358.15	368.79	0.8550
2000.00	70.00	25.00	5000.00	200.00	330.90	370.87	0.8610
2000.00	70.00	25.00	5000.00	220.00	308.81	373.73	0.8625
2000.00	70.00	25.00	10000.00	100.00	475.74	307.62	0.7335
2000.00	70.00	25.00	10000.00	120.00	431.41	310.36	0.7855
2000.00	70.00	25.00	10000.00	140.00	394.60	313.16	0.8200
2000.00	70.00	25.00	10000.00	160.00	364.04	315.65	0.8430

2000.00	70.00	25.00	10000.00	180.00	337.18	318.20	0.8575
2000.00	70.00	25.00	10000.00	200.00	312.28	320.78	0.8655
2000.00	70.00	25.00	10000.00	220.00	294.53	324.70	0.8700
2000.00	70.00	25.00	15000.00	100.00	443.77	271.64	0.7185
2000.00	70.00	25.00	15000.00	120.00	404.95	273.62	0.7740
2000.00	70.00	25.00	15000.00	140.00	370.05	275.42	0.8130
2000.00	70.00	25.00	15000.00	160.00	340.79	277.00	0.8390
2000.00	70.00	25.00	15000.00	180.00	315.67	278.54	0.8565
2000.00	70.00	25.00	15000.00	200.00	292.22	280.22	0.8670
2000.00	70.00	25.00	15000.00	220.00	274.46	282.75	0.8725
2000.00	70.00	25.00	20000.00	100.00	409.65	240.28	0.7015
2000.00	70.00	25.00	20000.00	120.00	374.79	241.71	0.7605
2000.00	70.00	25.00	20000.00	140.00	344.09	243.01	0.8030
2000.00	70.00	25.00	20000.00	160.00	318.09	244.30	0.8320
2000.00	70.00	25.00	20000.00	180.00	294.26	245.56	0.8525
2000.00	70.00	25.00	20000.00	200.00	274.16	247.14	0.8650
2000.00	70.00	25.00	20000.00	220.00	256.50	249.36	0.8730
2000.00	70.00	25.00	25000.00	100.00	364.51	208.70	0.6875
2000.00	70.00	25.00	25000.00	120.00	335.16	209.89	0.7495
2000.00	70.00	25.00	25000.00	140.00	308.77	211.16	0.7935
2000.00	70.00	25.00	25000.00	160.00	286.16	212.47	0.8255
2000.00	70.00	25.00	25000.00	180.00	266.21	213.75	0.8470
2000.00	70.00	25.00	25000.00	200.00	248.10	215.55	0.8610
2000.00	70.00	25.00	25000.00	220.00	233.48	217.91	0.8705
2000.00	70.00	30.00	0.00	100.00	452.74	381.74	0.7460
2000.00	70.00	30.00	0.00	120.00	407.09	385.34	0.7840
2000.00	70.00	30.00	0.00	140.00	368.03	388.95	0.8055
2000.00	70.00	30.00	0.00	160.00	340.73	392.99	0.8205

2000.00	70.00	30.00	0.00	180.00	317.72	396.96	0.8280
2000.00	70.00	30.00	0.00	200.00	294.89	400.60	0.8300
2000.00	70.00	30.00	0.00	220.00	278.51	404.19	0.8295
2000.00	70.00	30.00	5000.00	100.00	486.79	349.70	0.7460
2000.00	70.00	30.00	5000.00	120.00	439.62	352.50	0.7935
2000.00	70.00	30.00	5000.00	140.00	399.87	355.21	0.8235
2000.00	70.00	30.00	5000.00	160.00	366.76	357.65	0.8430
2000.00	70.00	30.00	5000.00	180.00	339.30	359.96	0.8530
2000.00	70.00	30.00	5000.00	200.00	313.14	362.14	0.8580
2000.00	70.00	30.00	5000.00	220.00	292.71	364.90	0.8585
2000.00	70.00	30.00	10000.00	100.00	453.83	299.89	0.7370
2000.00	70.00	30.00	10000.00	120.00	411.50	302.59	0.7880
2000.00	70.00	30.00	10000.00	140.00	375.68	305.42	0.8215
2000.00	70.00	30.00	10000.00	160.00	346.86	307.98	0.8430
2000.00	70.00	30.00	10000.00	180.00	321.20	310.55	0.8570
2000.00	70.00	30.00	10000.00	200.00	298.20	313.13	0.8650
2000.00	70.00	30.00	10000.00	220.00	280.42	316.79	0.8685
2000.00	70.00	30.00	15000.00	100.00	426.56	265.10	0.7220
2000.00	70.00	30.00	15000.00	120.00	389.02	267.09	0.7770
2000.00	70.00	30.00	15000.00	140.00	354.68	268.91	0.8150
2000.00	70.00	30.00	15000.00	160.00	326.77	270.47	0.8405
2000.00	70.00	30.00	15000.00	180.00	303.09	271.98	0.8570
2000.00	70.00	30.00	15000.00	200.00	279.66	273.54	0.8670
2000.00	70.00	30.00	15000.00	220.00	263.02	276.01	0.8725
2000.00	70.00	30.00	20000.00	100.00	395.78	234.75	0.7050
2000.00	70.00	30.00	20000.00	120.00	361.48	236.19	0.7630
2000.00	70.00	30.00	20000.00	140.00	331.75	237.50	0.8050
2000.00	70.00	30.00	20000.00	160.00	306.60	238.65	0.8340

2000.00	70.00	30.00	20000.00	180.00	283.60	239.77	0.8535
2000.00	70.00	30.00	20000.00	200.00	263.79	241.12	0.8660
2000.00	70.00	30.00	20000.00	220.00	246.93	243.26	0.8735
2000.00	70.00	30.00	25000.00	100.00	353.64	204.27	0.6900
2000.00	70.00	30.00	25000.00	120.00	324.98	205.45	0.7515
2000.00	70.00	30.00	25000.00	140.00	299.26	206.67	0.7955
2000.00	70.00	30.00	25000.00	160.00	277.00	207.92	0.8270
2000.00	70.00	30.00	25000.00	180.00	257.68	209.13	0.8480
2000.00	70.00	30.00	25000.00	200.00	239.90	210.73	0.8620
2000.00	70.00	30.00	25000.00	220.00	225.62	212.85	0.8715

Table 2: N219 Electric Results, 7 PAX

Altitude	ρ	C_L	C_D	Pa	I	V	Endurance	Range
0.0	1.225	1.695	0.194	423.916	688.574	40.0	1.53	220.308
0.0	1.225	1.339	0.135	418.46	679.711	45.0	1.556	252.056
0.0	1.225	1.085	0.101	428.905	696.677	50.0	1.507	271.229
0.0	1.225	0.897	0.08	454.066	737.547	55.0	1.399	277.04
0.0	1.225	0.753	0.067	493.457	801.53	60.0	1.256	271.245
0.0	1.225	0.642	0.058	547.023	888.538	65.0	1.098	257.005
0.0	1.225	0.554	0.053	614.984	998.928	70.0	0.943	237.69
0.0	1.225	0.482	0.049	697.745	1133.36	75.0	0.8	216.118
0.0	1.225	0.424	0.046	795.84	1292.696	80.0	0.675	194.291
0.0	1.225	0.375	0.043	909.888	1477.947	85.0	0.567	173.448
0.0	1.225	0.335	0.042	1040.577	1690.227	90.0	0.476	154.248
0.0	1.225	0.301	0.041	1188.64	1930.728	95.0	0.4	136.96
0.0	1.225	0.271	0.04	1354.846	2200.699	100.0	0.338	121.612
0.0	1.225	0.246	0.039	1539.993	2501.436	105.0	0.286	108.106

0.0	1.225	0.224	0.038	1744.897	2834.266	110.0	0.243	96.278
0.0	1.225	0.205	0.038	1970.393	3200.543	115.0	0.208	85.944
0.0	1.225	0.188	0.038	2217.329	3601.645	120.0	0.178	76.92
5000.0	1.056	1.967	0.249	468.703	761.322	40.0	1.343	193.343
5000.0	1.056	1.554	0.169	452.51	735.019	45.0	1.405	227.683
5000.0	1.056	1.259	0.123	452.316	734.705	50.0	1.406	253.122
5000.0	1.056	1.041	0.095	466.475	757.703	55.0	1.351	267.498
5000.0	1.056	0.874	0.078	494.149	802.654	60.0	1.253	270.752
5000.0	1.056	0.745	0.066	535.001	869.011	65.0	1.131	264.538
5000.0	1.056	0.642	0.058	589.015	956.746	70.0	0.998	251.402
5000.0	1.056	0.56	0.053	656.388	1066.182	75.0	0.867	233.985
5000.0	1.056	0.492	0.049	737.465	1197.877	80.0	0.745	214.517
5000.0	1.056	0.436	0.046	832.695	1352.561	85.0	0.636	194.635
5000.0	1.056	0.389	0.044	942.602	1531.085	90.0	0.541	175.408
5000.0	1.056	0.349	0.042	1067.764	1734.387	95.0	0.46	157.449
5000.0	1.056	0.315	0.041	1208.802	1963.477	100.0	0.392	141.05
5000.0	1.056	0.285	0.04	1366.367	2219.413	105.0	0.334	126.295
5000.0	1.056	0.26	0.039	1541.136	2503.294	110.0	0.286	113.145
5000.0	1.056	0.238	0.039	1733.805	2816.248	115.0	0.245	101.492
5000.0	1.056	0.219	0.038	1945.083	3159.431	120.0	0.211	91.2
10000.0	0.905	2.295	0.327	526.11	854.57	40.0	1.155	166.378
10000.0	0.905	1.814	0.217	498.408	809.572	45.0	1.24	200.811
10000.0	0.905	1.469	0.155	487.183	791.339	50.0	1.277	229.829
10000.0	0.905	1.214	0.117	490.269	796.351	55.0	1.266	250.746
10000.0	0.905	1.02	0.093	506.446	822.628	60.0	1.214	262.237
10000.0	0.905	0.869	0.077	535.077	869.134	65.0	1.13	264.489
10000.0	0.905	0.75	0.067	575.899	935.443	70.0	1.027	258.87
10000.0	0.905	0.653	0.059	628.901	1021.534	75.0	0.916	247.366

10000.0	0.905	0.574	0.054	694.24	1127.665	80.0	0.806	232.04
10000.0	0.905	0.508	0.05	772.196	1254.29	85.0	0.702	214.688
10000.0	0.905	0.453	0.047	863.136	1402.007	90.0	0.607	196.686
10000.0	0.905	0.407	0.045	967.494	1571.517	95.0	0.523	178.985
10000.0	0.905	0.367	0.043	1085.748	1763.6	100.0	0.45	162.176
10000.0	0.905	0.333	0.042	1218.417	1979.095	105.0	0.388	146.585
10000.0	0.905	0.304	0.041	1366.045	2218.891	110.0	0.334	132.35
10000.0	0.905	0.278	0.04	1529.201	2483.907	115.0	0.289	119.489
10000.0	0.905	0.255	0.039	1708.47	2775.097	120.0	0.25	107.951

Table 3: N219 Electric Results, 11 PAX

Altitude	ρ	C_L	C_D	Pa	I	V	Endurance	Range
0.0	1.225	1.695	0.194	423.916	688.574	40.0	1.152	165.957
0.0	1.225	1.339	0.135	418.46	679.711	45.0	1.172	189.873
0.0	1.225	1.085	0.101	428.905	696.677	50.0	1.135	204.315
0.0	1.225	0.897	0.08	454.066	737.547	55.0	1.054	208.693
0.0	1.225	0.753	0.067	493.457	801.53	60.0	0.946	204.328
0.0	1.225	0.642	0.058	547.023	888.538	65.0	0.827	193.601
0.0	1.225	0.554	0.053	614.984	998.928	70.0	0.711	179.05
0.0	1.225	0.482	0.049	697.745	1133.36	75.0	0.603	162.8
0.0	1.225	0.424	0.046	795.84	1292.696	80.0	0.508	146.358
0.0	1.225	0.375	0.043	909.888	1477.947	85.0	0.427	130.658
0.0	1.225	0.335	0.042	1040.577	1690.227	90.0	0.359	116.195
0.0	1.225	0.301	0.041	1188.64	1930.728	95.0	0.302	103.171
0.0	1.225	0.271	0.04	1354.846	2200.699	100.0	0.254	91.61
0.0	1.225	0.246	0.039	1539.993	2501.436	105.0	0.215	81.436
0.0	1.225	0.224	0.038	1744.897	2834.266	110.0	0.183	72.526

0.0	1.225	0.205	0.038	1970.393	3200.543	115.0	0.156	64.741
0.0	1.225	0.188	0.038	2217.329	3601.645	120.0	0.134	57.943
5000.0	1.056	1.967	0.249	468.703	761.322	40.0	1.011	145.644
5000.0	1.056	1.554	0.169	452.51	735.019	45.0	1.059	171.513
5000.0	1.056	1.259	0.123	452.316	734.705	50.0	1.059	190.676
5000.0	1.056	1.041	0.095	466.475	757.703	55.0	1.018	201.505
5000.0	1.056	0.874	0.078	494.149	802.654	60.0	0.944	203.956
5000.0	1.056	0.745	0.066	535.001	869.011	65.0	0.852	199.275
5000.0	1.056	0.642	0.058	589.015	956.746	70.0	0.752	189.38
5000.0	1.056	0.56	0.053	656.388	1066.182	75.0	0.653	176.26
5000.0	1.056	0.492	0.049	737.465	1197.877	80.0	0.561	161.594
5000.0	1.056	0.436	0.046	832.695	1352.561	85.0	0.479	146.618
5000.0	1.056	0.389	0.044	942.602	1531.085	90.0	0.408	132.134
5000.0	1.056	0.349	0.042	1067.764	1734.387	95.0	0.347	118.606
5000.0	1.056	0.315	0.041	1208.802	1963.477	100.0	0.295	106.252
5000.0	1.056	0.285	0.04	1366.367	2219.413	105.0	0.252	95.138
5000.0	1.056	0.26	0.039	1541.136	2503.294	110.0	0.215	85.231
5000.0	1.056	0.238	0.039	1733.805	2816.248	115.0	0.185	76.454
5000.0	1.056	0.219	0.038	1945.083	3159.431	120.0	0.159	68.701
10000.0	0.905	2.295	0.327	526.11	854.57	40.0	0.87	125.332
10000.0	0.905	1.814	0.217	498.408	809.572	45.0	0.934	151.27
10000.0	0.905	1.469	0.155	487.183	791.339	50.0	0.962	173.129
10000.0	0.905	1.214	0.117	490.269	796.351	55.0	0.954	188.885
10000.0	0.905	1.02	0.093	506.446	822.628	60.0	0.915	197.542
10000.0	0.905	0.869	0.077	535.077	869.134	65.0	0.851	199.238
10000.0	0.905	0.75	0.067	575.899	935.443	70.0	0.774	195.006
10000.0	0.905	0.653	0.059	628.901	1021.534	75.0	0.69	186.339
10000.0	0.905	0.574	0.054	694.24	1127.665	80.0	0.607	174.795

10000.0	0.905	0.508	0.05	772.196	1254.29	85.0	0.529	161.724
10000.0	0.905	0.453	0.047	863.136	1402.007	90.0	0.457	148.163
10000.0	0.905	0.407	0.045	967.494	1571.517	95.0	0.394	134.828
10000.0	0.905	0.367	0.043	1085.748	1763.6	100.0	0.339	122.166
10000.0	0.905	0.333	0.042	1218.417	1979.095	105.0	0.292	110.422
10000.0	0.905	0.304	0.041	1366.045	2218.891	110.0	0.252	99.698
10000.0	0.905	0.278	0.04	1529.201	2483.907	115.0	0.217	90.011
10000.0	0.905	0.255	0.039	1708.47	2775.097	120.0	0.188	81.319

Table 4: N219 Electric Results, 21 PAX

Altitude	ρ	C_L	C_D	Pa	I	V	Endurance	Range
0.0	1.225	1.695	0.194	423.916	688.574	40.0	0.341	49.109
0.0	1.225	1.339	0.135	418.46	679.711	45.0	0.347	56.186
0.0	1.225	1.085	0.101	428.905	696.677	50.0	0.336	60.46
0.0	1.225	0.897	0.08	454.066	737.547	55.0	0.312	61.755
0.0	1.225	0.753	0.067	493.457	801.53	60.0	0.28	60.464
0.0	1.225	0.642	0.058	547.023	888.538	65.0	0.245	57.289
0.0	1.225	0.554	0.053	614.984	998.928	70.0	0.21	52.984
0.0	1.225	0.482	0.049	697.745	1133.36	75.0	0.178	48.175
0.0	1.225	0.424	0.046	795.84	1292.696	80.0	0.15	43.31
0.0	1.225	0.375	0.043	909.888	1477.947	85.0	0.126	38.664
0.0	1.225	0.335	0.042	1040.577	1690.227	90.0	0.106	34.384
0.0	1.225	0.301	0.041	1188.64	1930.728	95.0	0.089	30.53
0.0	1.225	0.271	0.04	1354.846	2200.699	100.0	0.075	27.109
0.0	1.225	0.246	0.039	1539.993	2501.436	105.0	0.064	24.098
0.0	1.225	0.224	0.038	1744.897	2834.266	110.0	0.054	21.461
0.0	1.225	0.205	0.038	1970.393	3200.543	115.0	0.046	19.158

0.0	1.225	0.188	0.038	2217.329	3601.645	120.0	0.04	17.146
5000.0	1.056	1.967	0.249	468.703	761.322	40.0	0.299	43.098
5000.0	1.056	1.554	0.169	452.51	735.019	45.0	0.313	50.753
5000.0	1.056	1.259	0.123	452.316	734.705	50.0	0.313	56.424
5000.0	1.056	1.041	0.095	466.475	757.703	55.0	0.301	59.628
5000.0	1.056	0.874	0.078	494.149	802.654	60.0	0.279	60.354
5000.0	1.056	0.745	0.066	535.001	869.011	65.0	0.252	58.968
5000.0	1.056	0.642	0.058	589.015	956.746	70.0	0.222	56.04
5000.0	1.056	0.56	0.053	656.388	1066.182	75.0	0.193	52.158
5000.0	1.056	0.492	0.049	737.465	1197.877	80.0	0.166	47.818
5000.0	1.056	0.436	0.046	832.695	1352.561	85.0	0.142	43.386
5000.0	1.056	0.389	0.044	942.602	1531.085	90.0	0.121	39.101
5000.0	1.056	0.349	0.042	1067.764	1734.387	95.0	0.103	35.097
5000.0	1.056	0.315	0.041	1208.802	1963.477	100.0	0.087	31.442
5000.0	1.056	0.285	0.04	1366.367	2219.413	105.0	0.074	28.153
5000.0	1.056	0.26	0.039	1541.136	2503.294	110.0	0.064	25.221
5000.0	1.056	0.238	0.039	1733.805	2816.248	115.0	0.055	22.624
5000.0	1.056	0.219	0.038	1945.083	3159.431	120.0	0.047	20.33
10000.0	0.905	2.295	0.327	526.11	854.57	40.0	0.258	37.087
10000.0	0.905	1.814	0.217	498.408	809.572	45.0	0.276	44.763
10000.0	0.905	1.469	0.155	487.183	791.339	50.0	0.285	51.232
10000.0	0.905	1.214	0.117	490.269	796.351	55.0	0.282	55.894
10000.0	0.905	1.02	0.093	506.446	822.628	60.0	0.271	58.456
10000.0	0.905	0.869	0.077	535.077	869.134	65.0	0.252	58.958
10000.0	0.905	0.75	0.067	575.899	935.443	70.0	0.229	57.705
10000.0	0.905	0.653	0.059	628.901	1021.534	75.0	0.204	55.141
10000.0	0.905	0.574	0.054	694.24	1127.665	80.0	0.18	51.724
10000.0	0.905	0.508	0.05	772.196	1254.29	85.0	0.156	47.856

10000.0	0.905	0.453	0.047	863.136	1402.007	90.0	0.135	43.844
10000.0	0.905	0.407	0.045	967.494	1571.517	95.0	0.117	39.898
10000.0	0.905	0.367	0.043	1085.748	1763.6	100.0	0.1	36.151
10000.0	0.905	0.333	0.042	1218.417	1979.095	105.0	0.086	32.675
10000.0	0.905	0.304	0.041	1366.045	2218.891	110.0	0.075	29.502
10000.0	0.905	0.278	0.04	1529.201	2483.907	115.0	0.064	26.635
10000.0	0.905	0.255	0.039	1708.47	2775.097	120.0	0.056	24.063

Appendix B: Python Codes

.1 ISA main code

```
import numpy as np
from module import ISAfunc as calc
# import RANGEfunc as RNG
import matplotlib.pyplot as plt
plt.rcParams['axes.grid'] = True
# from scipy import stats
# import xlsxwriter
plt.rcParams['text.usetex'] = True

cols = 'Altitude (km)'
rows = ['{}'.format(row) for row in ['Temperature (k)', 'Pressure (kPa)',
                                      "Density (kg/"'$\displaystyle''{m^3}$'")",
                                      'Viscosity (Pa.S)']]

# workbook = xlsxwriter.Workbook('ISA32.csv')
# worksheet = workbook.add_worksheet()
row = 0
col = 0

del_t = np.arange(-15,16,5)
# del_t = np.arange(0,5,5)


n_alt = 80001

```

```
alt_dat = np.arange(0,n_alt)

t = np.array([])
p = np.array([])
rho = np.array([])
miu = np.array([])

for n in range(7):
    for i in range(n_alt):
        if (alt_dat[i] <= 11000):
            ISA = calc.TROPOSPHERE(del_t[n],alt_dat[i])
        if (11000 < alt_dat[i] <= 20000):
            ISA = calc.TROPOPAUSE(del_t[n],alt_dat[i])
        if (20000 < alt_dat[i] <= 32000):
            ISA = calc.LOWERSTRAT(del_t[n],alt_dat[i])
        if (32000 < alt_dat[i] <= 47000):
            ISA = calc.UPPERSTRAT(del_t[n],alt_dat[i])
        if (47000 < alt_dat[i] <= 51000):
            ISA = calc.STRATPAUSE(del_t[n],alt_dat[i])
        if (51000 < alt_dat[i] <= 71000):
            ISA = calc.LOWERMESO(del_t[n],alt_dat[i])
        if (71000 < alt_dat[i] <= 80000):
            ISA = calc.UPPERMESO(del_t[n],alt_dat[i])

        ISAtotal = np.array([ISA])

        t = np.append(t,ISAtotal[0,0])
        p = np.append(p,ISAtotal[0,1])
        rho = np.append(rho,ISAtotal[0,2])
        miu = np.append(miu,ISAtotal[0,3])


    """Plot"""
```

```python


    #ISA
    fig, axs = plt.subplots(3,sharex=True,figsize=(5,10))
    fig.suptitle('ISA '+ str(del_t[n]))

    axs[0].plot(alt_dat/1000,t)
    axs[1].plot(alt_dat/1000,p/1000)
    axs[2].plot(alt_dat/1000,rho)

    plt.xlabel(cols)

    for ax, row in zip(axs, rows):
        ax.set_ylabel(row, rotation=90, size='large')

    fig.tight_layout(rect=[0, 0.03, 1, 0.95])
    fig.savefig("data_result/ISA/ISA_Plot"+ str(del_t[n]) +".pdf",dpi=100)


    #Temp
    fig, axs = plt.subplots(1,sharex=True,figsize=(5,5))
    fig.suptitle('Temperature Over Altitude, ISA ' + str(del_t[n]))
    axs.plot(alt_dat/1000,t)
    # plt.xlim(0, 80)
    # plt.ylim(180, 320)
    plt.xlabel(cols)
    plt.ylabel(rows[0])
    fig.tight_layout(rect=[0, 0.03, 1, 0.95])
    fig.savefig("data_result/ISA/ISA_Plot_Temp"+ str(del_t[n]) +".pdf",dpi=600)


    #Press
    fig, axs = plt.subplots(1,sharex=True,figsize=(5,5))
```

```
    fig.suptitle('Pressure Over Altitude, ISA ' + str(del_t[n]))
    axs.plot(alt_dat/1000,p/1000)
    # plt.xlim(0, 80)
    # plt.ylim(0, 140)
    plt.xlabel(cols)
    plt.ylabel(rows[1])
    fig.tight_layout(rect=[0, 0.03, 1, 0.95])
    fig.savefig("data_result/ISA/ISA_Plot_Press"+ str(del_t[n]) +".pdf",dpi=600)

    #Dens
    fig, axs = plt.subplots(1,sharex=True,figsize=(5,5))
    fig.suptitle('Density Over Altitude, ISA ' + str(del_t[n]))
    axs.plot(alt_dat/1000,rho)
    # plt.xlim(0, 80)
    # plt.ylim(0, 1.4)
    plt.xlabel(cols)
    plt.ylabel(rows[2])
    fig.tight_layout(rect=[0, 0.03, 1, 0.95])
    fig.savefig("data_result/ISA/ISA_Plot_Dens"+ str(del_t[n]) +".pdf",dpi=600)

    #AIO

    def make_patch_spines_invisible(ax):
        ax.set_frame_on(True)
        ax.patch.set_visible(False)
        for sp in ax.spines.values():
            sp.set_visible(False)


    fig, host = plt.subplots(figsize=(6,5))
    fig.subplots_adjust(right=0.8)

    fig.suptitle('ISA '+ str(del_t[n]))
```

```

    par1 = host.twinx()
    par2 = host.twinx()

    par2.spines["right"].set_position(("axes", 1.15))
    make_patch_spines_invisible(par2)
    par2.spines["right"].set_visible(True)

    p1, = host.plot(alt_dat/1000, t, "tab:blue", label="Temperature (K)")
    p2, = par1.plot(alt_dat/1000, p/1000, "tab:orange", label="Pressure (kPa)")
    p3, = par2.plot(alt_dat/1000, rho, "tab:green",
                    label="Density (kg/"'$\displaystyle''{m^3}$'")")

    host.set_xlim(0, 80)
    host.set_ylim(180, 320)
    par1.set_ylim(0, 140)
    par2.set_ylim(0, 1.4)

    host.set_xlabel("Altitude (km)")
    host.set_ylabel("T,Kelvin")
    par1.set_ylabel("p,kPa")
    par2.set_ylabel(r'$\displaystyle\rho$'",kg/"'$\displaystyle''{m^3}$')

    host.yaxis.label.set_color(p1.get_color())
    par1.yaxis.label.set_color(p2.get_color())
    par2.yaxis.label.set_color(p3.get_color())

    tkw = dict(size=4, width=1.5)
    host.tick_params(axis='y', colors=p1.get_color(), **tkw)
    par1.tick_params(axis='y', colors=p2.get_color(), **tkw)
    par2.tick_params(axis='y', colors=p3.get_color(), **tkw)
    host.tick_params(axis='x', **tkw)

```

```
    lines = [p1, p2, p3]

    host.legend(lines, [l.get_label() for l in lines])
    fig.tight_layout(rect=[0, 0.03, 1, 0.95])
    fig.savefig("data_result/ISA/ISA_Plot_AIO"+ str(del_t[n]) +".pdf",dpi=600)

    #clear
    t = np.array([])
    p = np.array([])
    rho = np.array([])
    miu = np.array([])

    # ISAcompile = np.array([])


```

.2 ISA function code

```
""" ISA Formula from Troposphere to Upper Stratosphere """
""" Using Ruijgrok Equation"""
import numpy as np
import math as math

g0 = 9.80665
Ra = 8314.32
R = 287.053
gamma = 1.4
BETA = 1.458 * 10**-6
S = 110.4

def SPEEDSOUND (t):
    """

```

```

    Parameters
    ----------
    t : TYPE
        temperature (K).

    Returns
    -------
    c : TYPE
        speed of sound (m/s).

    """

    c = np.sqrt(gamma * R * t)

    return c

def MACHNUMBER (V,c):
    """



    Parameters
    ----------
    V : TYPE
        airspeed.
    c : TYPE
        speed of sound.

    Returns
    -------
    M : TYPE
        Mach number.

```

```
    """

    M = V/c

    return M

def airspeed (W,s,rho,Cl):
    """


    Parameters
    ----------
    W : TYPE
        aircraft weight (N).
    s : TYPE
        wing span (m).
    rho : TYPE
        air density (kg/m^3).
    Cl : TYPE
        lift coefficient.

    Returns
    -------
    V : TYPE
        airspeed (m/s).

    """
    #s = wing span
    V = np.sqrt((2*W)/(s*rho*Cl))

    return V

def LiftCoefficient (Cd,Cd0,k):
```

```
    """


    Parameters
    ----------
    Cd : TYPE
        drag coefficient.
    Cd0 : TYPE
        zero lift drag coefficient.
    k : TYPE
        k.

    Returns
    -------
    Cl : TYPE
        lift coefficient.

    """

    Cl = np.sqrt((Cd-Cd0)/k)

    return Cl

def k (A,e):
    """


    Parameters
    ----------
    A : TYPE
        wing aspec ratio.
    e : TYPE
        oswald efficiency.
```

```

    Returns
    -------
    k : TYPE
        k.

    """

    k = 1/(np.pi*A*e)

    return k

def TROPOSPHERE (del_t,alt):
    """


    Parameters
    ----------
    del_t : TYPE
        temperature deviation (k).
    alt : TYPE
        altitude (m).

    Returns
    -------
    t : TYPE
        temperature (k).
    p : TYPE
        pressure (Pa).
    rho : TYPE
        density (kg/m^3).
    miu : TYPE
        viscosity (Pa.s).
```

```

    """

    lmd = -0.0065
    rho0 = 1.225
    t0 = 288.150
    p0 = 101325

    #temperature
    #eq 2.8
    t = t0 + (del_t) + (lmd * float(alt))

    #pressure
    #eq 2.12
    p = p0 * ((t0+(lmd * float(alt))) / t0) ** (-g0 / (lmd * R))

    #density
    #eq 2.13
    rho = rho0 * (t / t0) ** -((g0 / (lmd * R)) + 1)

    #viscosity
    #eq 2.19
    miu = (BETA * t**(3/2))/(t+S)

    return t, p, rho, miu


def TROPOPAUSE (del_t,alt):
    """



    Parameters
    ----------
```

```
    del_t : TYPE
        temperature deviation (k).
    alt : TYPE
        altitude (m).

    Returns
    -------
    t : TYPE
        temperature (k).
    p : TYPE
        pressure (Pa).
    rho : TYPE
        density (kg/m^3).
    miu : TYPE
        viscosity (Pa.s).

    """

    alt1 = 11000
    ISA = TROPOSPHERE (del_t,alt1)
    t1 = ISA[0]
    p1 = ISA[1]
    rho1 = ISA[2]
    lmd = 0

    #temperature
    #eq 2.8
    t = t1 + lmd*(alt-alt1)

    #pressure
    #eq 2.15
    p = float(p1) * math.exp((-g0/((t1-del_t)*R))*(alt-alt1))

```

```
    #density
    #eq 2.16
    rho = rho1 * math.exp((-g0/(t*R))*(alt-alt1))

    #viscosity
    #eq 2.19
    miu = (BETA * t**(3/2))/(t+S)

    return t, p, rho, miu


def LOWERSTRAT (del_t,alt):
    """



    Parameters
    ----------
    del_t : TYPE
        temperature deviation (k).
    alt : TYPE
        altitude (m).

    Returns
    -------
    t : TYPE
        temperature (k).
    p : TYPE
        pressure (Pa).
    rho : TYPE
        density (kg/m^3).
    miu : TYPE
        viscosity (Pa.s).

```

```
    """

    alt1 = 20000
    ISA = TROPOPAUSE (del_t,alt1)
    t1 = ISA[0]
    p1 = ISA[1]
    rho1 = ISA[2]
    lmd = 0.001

    #temperature
    #eq 2.8
    t = t1 + lmd*(alt-alt1)

    #pressure
    #eq 2.10
    p = p1 * (t / t1) ** (-g0 / (lmd * R))
    #density
    #eq 2.11
    rho = rho1 * (t / t1) ** -((g0 / (lmd * R)) + 1)

    #viscosity
    #eq 2.19
    miu = (BETA * t**(3/2))/(t+S)

    return t, p, rho, miu


def UPPERSTRAT (del_t,alt):
    """


    Parameters
    ----------
```

```
    del_t : TYPE
        temperature deviation (k).
    alt : TYPE
        altitude (m).

    Returns
    -------
    t : TYPE
        temperature (k).
    p : TYPE
        pressure (Pa).
    rho : TYPE
        density (kg/m^3).
    miu : TYPE
        viscosity (Pa.s).

    """

    alt1 = 32000
    ISA = LOWERSTRAT (del_t,alt1)
    t1 = ISA[0]
    p1 = ISA[1]
    rho1 = ISA[2]
    lmd = 0.0028

    #temperature
    #eq 2.8
    t = t1 + lmd*(alt-alt1)

    #pressure
    #eq 2.10
    p = p1 * (t / t1) ** (-g0 / (lmd * R))

```

```
    #density
    #eq 2.11
    rho = rho1 * (t / t1) ** -((g0 / (lmd * R)) + 1)

    #viscosity
    #eq 2.19
    miu = (BETA * t**(3/2))/(t+S)

    return t, p, rho, miu


def STRATPAUSE (del_t,alt):
    """


    Parameters
    ----------
    del_t : TYPE
        temperature deviation (k).
    alt : TYPE
        altitude (m).

    Returns
    -------
    t : TYPE
        temperature (k).
    p : TYPE
        pressure (Pa).
    rho : TYPE
        density (kg/m^3).
    miu : TYPE
        viscosity (Pa.s).

```

```
    """

    alt1 = 47000
    ISA = UPPERSTRAT (del_t,alt1)
    t1 = ISA[0]
    p1 = ISA[1]
    rho1 = ISA[2]
    lmd = 0.000

    #temperature
    #eq 2.8
    t = t1 + lmd*(alt-alt1)

    #pressure
    #eq 2.15
    p = float(p1) * math.exp((-g0/((t1-del_t)*R))*(alt-alt1))

    #density
    #eq 2.16
    rho = rho1 * math.exp((-g0/(t*R))*(alt-alt1))

    #viscosity
    #eq 2.19
    miu = (BETA * t**(3/2))/(t+S)

    return t, p, rho, miu


def LOWERMESO (del_t,alt):
    """


    Parameters
```

```
    ----------
    del_t : TYPE
        temperature deviation (k).
    alt : TYPE
        altitude (m).

    Returns
    -------
    t : TYPE
        temperature (k).
    p : TYPE
        pressure (Pa).
    rho : TYPE
        density (kg/m^3).
    miu : TYPE
        viscosity (Pa.s).

    """

    alt1 = 51000
    ISA = STRATPAUSE (del_t,alt1)
    t1 = ISA[0]
    p1 = ISA[1]
    rho1 = ISA[2]
    lmd = -0.0028

    #temperature
    #eq 2.8
    t = t1 + lmd*(alt-alt1)

    #pressure
    #eq 2.10
    p = p1 * (t / t1) ** (-g0 / (lmd * R))
```

```
    #density
    #eq 2.11
    rho = rho1 * (t / t1) ** -((g0 / (lmd * R)) + 1)

    #viscosity
    #eq 2.19
    miu = (BETA * t**(3/2))/(t+S)

    return t, p, rho, miu


def UPPERMESO (del_t,alt):
    """


    Parameters
    ----------
    del_t : TYPE
        temperature deviation (k).
    alt : TYPE
        altitude (m).

    Returns
    -------
    t : TYPE
        temperature (k).
    p : TYPE
        pressure (Pa).
    rho : TYPE
        density (kg/m^3).
    miu : TYPE
        viscosity (Pa.s).
```

```
    """

    alt1 = 71000
    ISA = LOWERMESO (del_t,alt1)
    t1 = ISA[0]
    p1 = ISA[1]
    rho1 = ISA[2]
    lmd = -0.002

    #temperature
    #eq 2.8
    t = t1 + lmd*(alt-alt1)

    #pressure
    #eq 2.10
    p = p1 * (t / t1) ** (-g0 / (lmd * R))

    #density
    #eq 2.11
    rho = rho1 * (t / t1) ** -((g0 / (lmd * R)) + 1)

    #viscosity
    #eq 2.19
    miu = (BETA * t**(3/2))/(t+S)

    return t, p, rho, miu
```

.3 Aircraft Performance function code

```
# -*- coding: utf-8 -*-
"""
Created on Thu Feb 18 19:33:53 2021
```

```

@author: mahes
"""

import numpy as np

def k (AR, e):
    """
    from Elements of airplane performance ruijgrok

    Parameters
    ----------
    AR : TYPE
        wing aspect ratio.
    e : TYPE
        oswald's efficiency.

    Returns
    -------
    k : TYPE
        k.

    """
    k  =  1 / (np.pi * AR * e)

    return k

def PowerRequired(rho,V,S,cd0,W,k):
    """



    Parameters
    ----------
```

```
    rho : TYPE
        air density.
    V : TYPE
        airspeed.
    S : TYPE
        wing area.
    cd0 : TYPE
        zero lift drag coefficient.
    W : TYPE
        aircraft weight N.
    k : TYPE
        k.

    Returns
    -------
    Pr : TYPE
        power required.

    """
    Pr = 0.5 * rho * V ** 3 * S * cd0 + (2 * W**2 * k)/(rho * V * S)

    return Pr

def PowerRequired2(rho,S,cd0,W,k):

    Pr = (2/np.sqrt(rho*S)) * cd0 ** 0.25 * (2 * W * np.sqrt(k/3))**(3/2)

    return Pr

def Vdmin (W, S, rho, k, cd0):
    """
    airspeed at drag minimum/for max range
    From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
```

```
    Journal of Aircraft.
    Equation no. 12.

    Parameters
    ----------
    W : TYPE
        aircraft weight (N).
    S : TYPE
        wing area (m^2).
    rho : TYPE
        air density (kg/m^2).
    k : TYPE
        k.

    Returns
    -------
    V : TYPE
        airspeed (m/s).

    """
    V = np.sqrt( (2*W) / (S * rho * np.sqrt(cd0/k) ) )

    return V

def Vemax (W,S,rho,k,cd0):
    """
    airspeed for max endurance
    From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
    Journal of Aircraft.
    Equation no. 11.

    Parameters
    ----------
```

```
    W : TYPE
        aircraft weight (N).
    S : TYPE
        wing area (m^2).
    rho : TYPE
        air density (kg/m^2).
    k : TYPE
        k.
    cd0 : TYPE
        zero lift drag coefficient.

    Returns
    -------
    V : TYPE
        DESCRIPTION.

    """
    V = np.sqrt( ((2*W)/(rho*S)) * np.sqrt(k/(3*cd0)) )

    return V

def Vrmax (W,S,rho,k,cd0):
    """
    airspeed for max range
    From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
    Journal of Aircraft.
    Equation no. 12.

    Parameters
    ----------
    W : TYPE
        aircraft weight (N).
    S : TYPE
```

```
        wing area (m^2).
    rho : TYPE
        air density (kg/m^2).
    k : TYPE
        k.
    cd0 : TYPE
        zero lift drag coefficient.

    Returns
    -------
    V : TYPE
        DESCRIPTION.

    """
    V = np.sqrt( ((2*W)/(rho*S)) * np.sqrt(k/cd0) )

    return V

def endurancemaxElectric (Rt, n, ntot, Volt, C, rho, S, k, W, cd0):
    """
    maximum endurance of aircraft
    From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
    Journal of Aircraft.
    Equation no. 16.



    Parameters
    ----------
    Rt : TYPE
        battery hour rating (h).
    n : TYPE
        battery discharge parameter.
    ntot : TYPE
```

```
            total efficiency.
        Volt : TYPE
            voltage (V).
        C : TYPE
            battery capacity (Ah).
        rho : TYPE
            air density (kg/m^3).
        S : TYPE
            wing area (m^2).
        k : TYPE
            DESCRIPTION.
        W : TYPE
            aircraft weight (N).
        cd0 : TYPE
            zero lift drag coefficient.

        Returns
        -------
        R : TYPE
            range (km).

        """
        E = (Rt ** (1-n)) * ((ntot*Volt*C) /
                                  ((2/np.sqrt(rho*S)) * (cd0**0.25) *
                                   (2*W*np.sqrt(k/3))**(3/2)   ) ) ** n

        return E

def rangemaxElectric (Rt, n, ntot, Volt, C, rho, S, k, W, cd0):
        """
        maximum range of aircraft
        From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
        Journal of Aircraft.
```

```
    Equation no. 18.

    Parameters
    ----------
    Rt : TYPE
        battery hour rating (h).
    n : TYPE
        battery discharge parameter.
    ntot : TYPE
        total efficiency.
    Volt : TYPE
        voltage (V).
    C : TYPE
        battery capacity (Ah).
    rho : TYPE
        air density (kg/m^3).
    S : TYPE
        wing area (m^2).
    k : TYPE
        DESCRIPTION.
    W : TYPE
        aircraft weight (N).
    cd0 : TYPE
        zero lift drag coefficient.

    Returns
    -------
    R : TYPE
        range (km).

    """
    R = (Rt ** (1-n)) * ((ntot*Volt*C) /
                         ( (1/np.sqrt(rho*S)) * (cd0**0.25) *
```

```
                            (2*W*np.sqrt(k))**(3/2) ))** n * np.sqrt(
                                ((2*W)/(rho*S)) * np.sqrt(k/cd0) ) * 3.6

    return R

def enduranceRmaxElectric (Rt, n, ntot, Volt, C, rho, S, k, W, cd0):
    """
    endurance at maximum range
    From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
    Journal of Aircraft.
    derived from Equation no. 18.


    Parameters
    ----------
    Rt : TYPE
        battery hour rating (h).
    n : TYPE
        battery discharge parameter.
    ntot : TYPE
        total efficiency.
    Volt : TYPE
        voltage (V).
    C : TYPE
        battery capacity (Ah).
    rho : TYPE
        air density (kg/m^3).
    S : TYPE
        wing area (m^2).
    k : TYPE
        DESCRIPTION.
    W : TYPE
        aircraft weight (N).
```

```
    cd0 : TYPE
        zero lift drag coefficient.

    Returns
    -------
    E : TYPE
        endurance (h).

    """
    E = (Rt ** (1-n)) * ((ntot*Volt*C) / ( (1/np.sqrt(rho*S)) * (cd0**0.25) *
                                             (2*W*np.sqrt(k))**(3/2) ))** n

    return E

def enduranceElectric (Rt,n,ntot,Volt,C,rho,V,cd0,W,k,S):
    """
    From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
    Journal of Aircraft.
    Equation no. 9.

    Parameters
    ----------
    Rt : TYPE
        battery hour rating (h).
    n : TYPE
        battery discharge parameter.
    ntot : TYPE
        total efficiency.
    Volt : TYPE
        voltage (V).
    C : TYPE
        battery capacity (Ah).
    rho : TYPE
```

```
            air density (kg/m^3).
        V : TYPE
            airspeed (m/s).
        cd0 : TYPE
            zero lift drag coefficient.
        W : TYPE
            aircraft weight (N).
        k : TYPE
            DESCRIPTION.
        S : TYPE
            wing area (m^2).

        Returns
        -------
        E : TYPE
            endurance (h).

        """
        E = Rt ** (1-n) * ((ntot*Volt*C) / ((0.5 * rho * V**3 * S * cd0) +
                                            ((2 * W**2 * k)/(rho * V * S)))) ** n

        return E

    def rangeElectric (E,V):
        """
        From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
        Journal of Aircraft.
        Equation no. 13.

        Parameters
        ----------
        E : TYPE
            aircraft endurance (h).
```

```
    V : TYPE
        airspeed (m/s).

    Returns
    -------
    R : TYPE
        range (km).

    """
    R = E * V * 3.6

    return R

def powerbatt (Volt,C,Rt,n,E):
    """
    From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
    Journal of Aircraft.
    Equation no. 7.

    Parameters
    ----------
    Volt : TYPE
        Voltage (V).
    C : TYPE
        battery capacity (Ah).
    Rt : TYPE
        battery hour rating (h).
    n : TYPE
        battery discharge parameter.
    E : TYPE
        aircraft endurance (h).

    Returns
```

```
    -------
    Pb : TYPE
        power battery.

    """
    Pb = Volt * (C/Rt) * (Rt/E) ** (1/n)

    return Pb

def current (Volt,Pb):
    """
    From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
    Journal of Aircraft.
    derived from Equation no. 6.

    Parameters
    ----------
    Volt : TYPE
        Voltage (V).
    Pb : TYPE
        Power Battery (W).

    Returns
    -------
    I : TYPE
        current (A).

    """
    I = Pb/Volt

    return I

def Ctot (j,Cbatt):
```

```
    """
    From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
    Journal of Aircraft.
    Equation no. 19.

    Parameters
    ----------
    j : TYPE
        a counter expressing the number of batteries.
    Cbatt : TYPE
        capacity of each battery (Ah).

    Returns
    -------
    Ctot : TYPE
         total capacity of the battery (Ah).

    """
    Ctot = j * Cbatt

    return Ctot

def Wtot (j,Wbatt,BR):
    """
    From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
    Journal of Aircraft.
    Equation no. 20.

    Parameters
    ----------
    j : TYPE
        a counter expressing the number of batteries.
    Wbatt : TYPE
```

```
        weight of each individual battery (N).
    BR : TYPE
        battery weight as a fraction of the total weight.

    Returns
    -------
    Wtot : TYPE
        aircraft's total weight (N).

    """
    Wtot = (j * Wbatt) / BR

    return Wtot

def Wtot2 (W,BR):
    """
    Parameters
    ----------
    W : TYPE
        aircraft weight.
    BR : TYPE
        DESCRIPTION.

    Returns
    -------
    Wtot : TYPE
        DESCRIPTION.

    """
    Wtot = W / (1-BR)

    return Wtot

```

```python
def j1 (Ctot,Cbatt):
    """
    From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
    Journal of Aircraft.
    derived from Equation no. 19.

    Parameters
    ----------
    Ctot : TYPE
        total capacity of the battery (Ah).
    Cbatt : TYPE
        capacity of each battery (Ah).

    Returns
    -------
    j : TYPE
        a counter expressing the number of batteries.

    """
    j = Ctot/Cbatt

    return j

def current0 (Pr,V0,ntot):
    """
    required battery supply current (I) at the rated full-capacity battery
    supply voltage (V)
    From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
    Journal of Aircraft.
    Equation no. 21.

    Parameters
    ----------
```

```
    Pr : TYPE
        Power Required (W).
    V0 : TYPE
        initial airspeed (m/s).
    ntot : TYPE
        aircraft total efficiency.

    Returns
    -------
    I : TYPE
        current (A).

    """
    I0 = Pr/(V0*ntot)

    return I0

def C0 (I0,Rt,C,n):
    """
    effective initial battery capacity
    From "Range and Endurance Estimates for Battery-Powered Aircraft" by Traub,
    Journal of Aircraft.
    Equation no. 22.

    Parameters
    ----------
    I0 : TYPE
        initial current (A).
    Rt : TYPE
        battery hour rating (h).
    C : TYPE
         rated total battery capacity (Ah).
    n : TYPE
```

```
        battery discharge parameter.

    Returns
    -------
    C0 : TYPE
        effective initial battery capacity (Ah).

    """
    C0 = (I0 ** (1-n)) * (Rt ** (1-n)) * (C ** n)

    return C0

def Vj (V0, k, C0, Cjt):
    """



    Parameters
    ----------
    V0 : TYPE
        initial velocity(airspeed).
    k1 : TYPE
        .
    C0 : TYPE
        initial battery capacity.
    Cj : TYPE
        battery capacity at counter j (time t).

    Returns
    -------
    Vj : TYPE
        DESCRIPTION.

    """
```

```
    Vj = V0 - k * (C0 - Cjt)

    return Vj

def Ij (Pr, Vj, ntot):
    """


    Parameters
    ----------
    Pr : TYPE
        power required.
    Vj : TYPE
        airspeed at time t.
    ntot : TYPE
        total efficiency.

    Returns
    -------
    Ij : TYPE
        current at time t.

    """
    Ij = Pr / (Vj*ntot)

    return Ij

def Cj (Ij, Rt, C, n, SIGMA):


    Cj = ((Ij ** (1-n)) * (Rt ** (1-n)) * (C ** n)) - SIGMA

    return Cj
```

```python


#aerodynamic properties

def CL (W, rho, V, S):
    """



    Parameters
    ----------
    W : TYPE
        Aircraft weight in Newton.
    rho : TYPE
        air density (kg/m^3).
    V : TYPE
        airspeed (m/s).
    S : TYPE
        Wing area (m^2).

    Returns
    -------
    CL : TYPE
        Lift Coefficient.

    """

    CL = W / (0.5 * rho * (V**2) * S)

    return CL

def CD (cd0, CL, AR, e):
    """

```

```

    Parameters
    ----------
    cd0 : TYPE
        zero lift drag coefficient.
    CL : TYPE
        Lift Coefficient.
    AR : TYPE
        wing aspect ratio.
    e : TYPE
        Oswald's Efficiency.

    Returns
    -------
    CD : TYPE
        Drag Coefficient.

    """

    CD = cd0 + ((CL**2) / (np.pi * AR * e))

    return CD

def CLCD (CL, CD):
    """


    Parameters
    ----------
    CL : TYPE
        Lift Coefficient.
    CD : TYPE
        Drag Coefficient.
```

```

    Returns
    -------
    CLCD : TYPE
        Lift Drag Ratio.

    """

    CLCD = CL/CD

    return CLCD

def Pa (CD, rho, V, S):
    """


    Parameters
    ----------
    CD : TYPE
        Drag Coefficient.
    rho : TYPE
        air density (kg/m^3).
    V : TYPE
        airspeed (m/s).
    S : TYPE
        wing area (m^2).

    Returns
    -------
    Pa : TYPE
        Power available (HP).

    """
```

```

    Pa = (CD * 0.5 * rho * (V**3) * S) * 0.0013404825737265416


    return Pa

def Pa1 (cd0, rho, V, S, k, W):
    """


    Parameters
    ----------
    cd0 : TYPE
        zero lift drag coefficient.
    rho : TYPE
        air density (kg/m^3).
    V : TYPE
        airspeed (m/s).
    S : TYPE
        wing area (m^2).
    k : TYPE
        k.
    W : TYPE
        aircraft weight (N).

    Returns
    -------
    Pa : TYPE
        power available (hp).

    """

    Pa = ((0.5 * rho * (V**3) * S * cd0) + ((2 * (W**2) * k) / (rho * V * S))) * 0.0013404825737372
```

```

    return Pa

def nj (Pa, SHP):
    """



    Parameters
    ----------
    Pa : TYPE
        Power available (HP).
    SHP : TYPE
        Shaft Horse Power.

    Returns
    -------
    nj : TYPE
        propulsive efficiency.

    """
    nj = Pa/SHP

    return nj

def rangeAirbreathing (nj, SFC, CLCD, W1, W2):

    R = 375.0 * (nj/(SFC)) * (CLCD) * np.log(W1/W2)

    R = R * 1.60934

    return R

def enduranceAirbreathing (nj,SFC,W1,W2,S,rho,CD,CL):
```

```
    E = (nj/SFC) * (1/(np.sqrt((1/S)*(2/rho)*(CD**2/CL**3)))) * np.log(W1/(W2))

    return E

def enduranceAirbreathing2 (nj,SFC,W1,W2,S,rho,CD,CL):

    W1 = W1 * 0.22481

    W2 = W2 * 0.22481

    E = (nj/SFC) * (1/(np.sqrt((W2/S)*(2/rho)*(CD**2/CL**3)))) * np.log(W1/W2)

    return E
```

.4 Electric N219 code

```
# -*- coding: utf-8 -*-
"""
Created on Thu Feb 18 19:59:18 2021

@author: mahes
"""


import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
from module import AircraftPerformance as AP
from module import ISAfunc as ISAfunc


```

```
MCR = 70 #Maxium Continuous Rating percentage

alt = 10000 #ft
altM = alt * 0.3048 #meter

dt = 0
dts = [dt]

#endurance airspeed m/s
eV = 51

#range airspeed m/s
rV = 67

#total efficiency
ntot = 0.81


#air density
for del_t in dts:
    if (altM <= 11000):
        ISA = ISAfunc.TROPOSPHERE(del_t,altM)
    if (11000 < altM <= 20000):
        ISA = ISAfunc.TROPOPAUSE(del_t,altM)
    if (20000 < altM <= 32000):
        ISA = ISAfunc.LOWERSTRAT(del_t,altM)
    if (32000 < altM <= 47000):
        ISA = ISAfunc.UPPERSTRAT(del_t,altM)
    if (47000 < altM <= 51000):
        ISA = ISAfunc.STRATPAUSE(del_t,altM)
    if (51000 < altM <= 71000):
        ISA = ISAfunc.LOWERMESO(del_t,altM)
    if (71000 < altM <= 80000):
```

```
        ISA = ISAfunc.UPPERMESO(del_t,altM)

ISAtotal = np.array([ISA])

rho = ISAtotal[0,2]

#Rt
Rt = 1 #hour

#max aircraft weight/MTOW
W = 68941  #Newton

#aircraft no fuel/battery weight
W0 = 53250.1095 #newton

#fuel/battery weight
Wf = 15690.64 #newton

#fuel/battery volume
Vf = 1600 #litre

#wing area
S = 41.5 #m^2

#Voltage
Volt = 567 #V

#cd0
cd0 = 0.0357

# wing aspec ratio
AR = 9.16

```

```
# oswald's efficiency
e = 0.62921

#k
k = AP.k(AR,e)

#n
ns = [1,1.3]

#battery weight as a fraction of the total weight
BRs = [0.48]

#battery specific energy Wh/kg
# BSE = 250 #tesla 18350
BSE = 265 # Raymer

#battery energy density Wh/l
# BED = 721 #tesla 18350
BED = 700 # Raymer

# #battery capacity
# C1 = round( ((Wf/9.80665) * BSE)/ Volt )
# C2 = round( (Vf * BED) / Volt )

# Cs = [C1,C2]

# #aircraft weights
# W1 = W0 + Wf #Config 1
# W2 = W0 + (((Vf * BED)/BSE)*9.80665) #Config 2

# Wsi = [W1,W2]

#passengger weight
```

```
passengger = 85 #kg
luggage = 15 #kg

PAX = (passengger + luggage) *9.80665

#######################################
#configurations

#number of pilots and passengger, pilots always 2
p1 = 7
p2 = 11
p3 = 21
PAXs = [p1,p2,p3]


MTOW = 7030 *9.80665 #N, Maximum Take-Off Weight
Wb = 42031 #N, Fuselage Weight

#Config X1
Wp = p1 * PAX
Wf = MTOW - (Wb+Wp)
C = round( ((Wf/9.80665) * BSE)/ Volt )
X1 = np.array([Wb,Wf,Wp,C])

#Config X2
Wp = p1 * PAX
Wf = ((Vf * BED)/BSE)*9.80665
C = round( ((Wf/9.80665) * BSE)/ Volt )
X2 = np.array([Wb,Wf,Wp,C])

#Config Y1
Wp = p2 * PAX
Wf = MTOW - (Wb+Wp)
```

```
C = round( ((Wf/9.80665) * BSE)/ Volt )
Y1 = np.array([Wb,Wf,Wp,C])

#Config Y2
Wp = p2 * PAX
Wf = ((Vf * BED)/BSE)*9.80665
C = round( ((Wf/9.80665) * BSE)/ Volt )
Y2 = np.array([Wb,Wf,Wp,C])

#Config Z1
Wp = p3 * PAX
Wf = MTOW - (Wb+Wp)
C = round( ((Wf/9.80665) * BSE)/ Volt )
Z1 = np.array([Wb,Wf,Wp,C])

#Config Z2
Wp = p3 * PAX
Wf = ((Vf * BED)/BSE)*9.80665
C = round( ((Wf/9.80665) * BSE)/ Volt )
Z2 = np.array([Wb,Wf,Wp,C])

Configs = [X1,X2,Y1,Y2,Z1,Z2]
Config = [str(X1),str(X2),str(Y1),str(Y2),str(Z1),str(Z2)]
# Config  = np.array([X1,X2,Y1,Y2,Z1,Z2])

#Battery capacities array
Csi = np.array([])
for x in Configs:
    C = x[3]
    Csi = np.append(Csi,C)

#Total Weight Array
Wsi = np.array([])
```

```
for x in Configs:
    Wb = x[0]
    Wf = x[1]
    Wp = x[2]
    Wtot = Wb + Wf + Wp
    Wsi = np.append(Wsi,Wtot)

########################################
#for refrence
for p in PAXs:
    Wp = p * PAX
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])
    Ci = C

    Wp = p * PAX
    Wf = ((Vf * BED)/BSE)*9.80665
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con2 = np.array([Wb,Wf,Wp,C])

    Configs = [Con1, Con2]

########################################
plt.style.use("ggplot")

matplotlib.rcParams["text.usetex"] = True


for p in PAXs:


    FIG = plt.figure(figsize=(7, 15))
```

```

    # Endurance
    ax1 = FIG.add_subplot(311)
    ax1.grid(True)
    ax1.set_ylabel("Endurance ($\mathbf{h}$)")
    # ax1.set_ylim(0,4)
    plt.setp(ax1.get_xticklabels(), visible=False)

    # Range
    ax2 = FIG.add_subplot(312, sharex=ax1)
    ax2.set_ylabel("Range ($\mathbf{km}$)")
    # ax2.set_xlabel("Velocity ($\mathbf{m/s}$)")
    # ax2.set_xticks(np.arange(0,21,5))
    # ax2.set_xlim(0,25)
    # plt.setp(ax2.get_xticklabels(), visible=True)

    # Current
    ax3 = FIG.add_subplot(313, sharex=ax1)
    ax3.set_ylabel("Current (A)")
    ax3.set_xlabel("Velocity ($\mathbf{m/s}$)")
    # ax3.set_xticks(np.arange(0,21,5))
    # ax3.set_xlim(5,120)
    ax3.set_ylim(0,10000)
    ax3.grid(True)
    # plt.setp(ax2.get_xticklabels(), visible=True)

    #Fig 1
    #electric

    Wp = p * PAX
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])
```

```

    Wp = p * PAX
    Wf = ((Vf * BED)/BSE)*9.80665
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con2 = np.array([Wb,Wf,Wp,C])

    Configs = [Con1, Con2]
    for n in ns:
        for x in Configs:
            Vs = np.linspace(0,120,num=1000)
            Es = np.array([])
            Rs = np.array([])
            Is = np.array([])
            Wb = x[0]
            Wf = x[1]
            Wp = x[2]
            C  = x[3]
            W1 = Wb + Wf + Wp

            for V in Vs:
                E = AP.enduranceElectric(Rt,n,ntot,Volt,C,rho,V,cd0,W1,k,S)
                Es = np.append(Es, E)
                R = AP.rangeElectric(E,V)
                Rs = np.append(Rs, R)
                Pb = AP.powerbatt(Volt,C,Rt,n,E)
                I = AP.current(Volt,Pb)
                Is = np.append(Is, I)

            if str(x) == Config[0]:
                # label = "Config X1"
                label = str(C) + " A, " + str(round(W1/9.80665)) + " kg, n=" + str(n)
                color = "tab:green"
            if str(x) == Config[1]:
```

```
                # label = "Config X2"
                label = str(C) +  " A, " + str(round(W1/9.80665)) + " kg, n=" + str(n)
                color = "tab:blue"
            if str(x) == Config[2]:
                # label = "Config Y1"
                label = str(C) + " A, " + str(round(W1/9.80665)) + " kg, n=" + str(n)
                color = "tab:orange"
            if str(x) == Config[3]:
                # label = "Config Y2"
                label = str(C) + " A, " + str(round(W1/9.80665)) + " kg, n=" + str(n)
                color = "tab:cyan"
            if str(x) == Config[4]:
                # label = "Config Z1"
                label = str(C) + " A, " + str(round(W1/9.80665)) + " kg, n=" + str(n)
                color = "tab:purple"
            if str(x) == Config[5]:
                # label = "Config Z2"
                label = str(C) + " A, " + str(round(W1/9.80665)) + " kg, n=" + str(n)
                color = "tab:pink"


            # label = str(C) + " Ah, " + str(n) + " n"
            if n == 1:
                line = "--"
            else:
                line = "-"

            ax1.plot(Vs, Es, line, label=label,color=color)
            ax2.plot(Vs, Rs, line, label=label,color=color)
            ax3.plot(Vs, Is, line, label=label,color=color)

    plt.suptitle(str(p) + "PAX")

```

```
    ax1.legend(
        bbox_to_anchor=(0.0, 1.02, 1.0, 0.2),
        loc="lower left",
        ncol=2,
        mode="expand",
        borderaxespad=0.0,
    )

    plt.tight_layout()
    plt.savefig("data_result/Electric N219 V4/EelctricN219EnduranceRange"+ str(p) +"PAX.pdf",
                dpi=600)
    plt.show()




#effect of n
for p in PAXs:
    Wp = p * PAX
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])
    Ci = C

    Wp = p * PAX
    Wf = ((Vf * BED)/BSE)*9.80665
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con2 = np.array([Wb,Wf,Wp,C])

    Configs = [Con1, Con2]


    E1 = np.array([])
```

```
    E13 = np.array([])
    R1 = np.array([])
    R13 = np.array([])
    Cs = np.array([])

    for n in ns:
        for x in Configs:
            Wb = x[0]
            Wf = x[1]
            Wp = x[2]
            C  = x[3]
            W1 = Wb + Wf + Wp

            E = AP.endurancemaxElectric(Rt,n,ntot,Volt,C,rho,S,k,W,cd0)
            R = AP.rangemaxElectric(Rt,n,ntot,Volt,C,rho,S,k,W,cd0)
            if n == 1:
                E1 = np.append(E1, E)
                R1 = np.append(R1, R)
                # Cs = np.append(Cs,C)
            if n == 1.3:
                E13 = np.append(E13, E)
                R13 = np.append(R13, R)
                Cs = np.append(Cs,C)


    plt.figure(figsize=(7.5,5))
    plt.suptitle(str(p) + "PAX")
    plt.plot(Cs, E13/E1, "-o", label="Endurance",color="black")
    plt.plot(Cs, R13/R1, "-o", label="Range",color="slategrey")
    plt.plot([Ci,Ci], [0.7,1.4], label="MTOW",linestyle="--",color="tab:red")
    plt.xlabel("Capacity, Ah")
    plt.ylabel("E n=1.3/E n=1 \n R n=1.3/R n=1")
    plt.legend(
```

```
        bbox_to_anchor=(0.0, 1.02, 1.0, 0.2),
        loc="lower left",
        ncol=2,
        mode="expand",
        borderaxespad=0.0,
    )
    # plt.xlim(0,5)
    # plt.ylim(0.8,1.4)
    for x,y in zip(Cs,E13/E1):

        label = "{:.2f}".format(y)

        plt.annotate(label, # this is the text
                     (x,y), # this is the point to label
                     textcoords="offset points", # how to position the text
                     xytext=(0,10), # distance from text to points (x,y)
                     ha='center') # horizontal alignment can be left, right or center
    for x,y in zip(Cs,R13/R1):

        label = "{:.2f}".format(y)

        plt.annotate(label, # this is the text
                     (x,y), # this is the point to label
                     textcoords="offset points", # how to position the text
                     xytext=(0,-17), # distance from text to points (x,y)
                     ha='center') # horizontal alignment can be left, right or center

    plt.tight_layout()
    plt.savefig("data_result/Electric N219 V4/EelctricN219effectofn"+ str(p) +
                "PAX.pdf", dpi=600)
    plt.show()


```

```python

#Fig 3.1

for p in PAXs:
    plt.figure(figsize=(6,6))

    Wp = p * PAX
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])

    Wp = p * PAX
    Wf = ((Vf * BED)/BSE)*9.80665
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con2 = np.array([Wb,Wf,Wp,C])

    Configs = [Con1, Con2]

    Csx = np.arange(1,4001)
    Wsx = np.array([])
    labels = np.array([])

    for x in Configs:
        Ws = np.array([])
        for C in Csx:
            Wfx = (C*Volt/(BSE)) *9.80665
            Wb = x[0]
            Wf = x[1]
            Wp = x[2]
            Wx = Wb + Wfx + Wp
            Ws = np.append(Ws, Wx)

        plt.plot(Csx,Ws/9.80665,linestyle="-",color="slategrey")
```

```


        C  = x[3]
        Wx = Wb + Wf + Wp


        if str(x) == Config[0]:
            # label = "Config X1"
            label = str(C) + " A, " + str(round(Wx/9.80665)) + " kg, n=" + str(n)
            color = "tab:green"
        if str(x) == Config[1]:
            # label = "Config X2"
            label = str(C) + " A, " + str(round(Wx/9.80665)) + " kg, n=" + str(n)
            color = "tab:blue"
        if str(x) == Config[2]:
            # label = "Config Y1"
            label = str(C) + " A, " + str(round(Wx/9.80665)) + " kg, n=" + str(n)
            color = "tab:orange"
        if str(x) == Config[3]:
            # label = "Config Y2"
            label = str(C) + " A, " + str(round(Wx/9.80665)) + " kg, n=" + str(n)
            color = "tab:cyan"
        if str(x) == Config[4]:
            # label = "Config Z1"
            label = str(C) + " A, " + str(round(Wx/9.80665)) + " kg, n=" + str(n)
            color = "tab:purple"
        if str(x) == Config[5]:
            # label = "Config Z2"
            label = str(C) + " A, " + str(round(Wx/9.80665)) + " kg, n=" + str(n)
            color = "tab:pink"

        plt.scatter(C,Wx/9.80665,label=label,color=color)
    plt.plot([0,2000],[7030,7030],linestyle="--",color="tab:red",label="MTOW")
```

```

    plt.xlim(0,2000)
    plt.ylim(5000,11000)

    plt.suptitle(str(p) + "PAX")
    plt.xlabel("Capacity, Ah")
    plt.ylabel("Total Weight, kg")

    plt.legend(
        bbox_to_anchor=(0.0, 1.02, 1.0, 0.2),
        loc="lower left",
        ncol=2,
        mode="expand",
        borderaxespad=0.0,
    )


    plt.tight_layout()
    plt.savefig(
        "data_result/Electric N219 V4/"+
        "EelctricN219 Effect of increasing battery capacity on weight n = 1.3, "+
        str(p) + "PAX.pdf", dpi=600)
    plt.show




#Fig 3.2
for p in PAXs:
    Wp = p * PAX
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])
    Ci = C
```

```

    Wp = p * PAX
    Wf = ((Vf * BED)/BSE)*9.80665
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con2 = np.array([Wb,Wf,Wp,C])

    Configs = [Con1, Con2]

    # plt.figure(figsize=(5.3,5.3))

    n = 1.3
    Ws = np.array([]) #aircraft weights
    Vrs = np.array([]) #V for max range
    Ves = np.array([]) #V for max endurance
    irs = np.array([]) #current for max range
    ies = np.array([]) #current for max endurance

    Www = np.array([])


    ###for plots####################################
    def make_patch_spines_invisible(ax):
        ax.set_frame_on(True)
        ax.patch.set_visible(False)
        for sp in ax.spines.values():
            sp.set_visible(False)

    fig, host = plt.subplots(figsize=(6.5,6))
    fig.subplots_adjust(right=0.8)

    fig.suptitle(str(p) + "PAX")

    par1 = host.twinx()
```

```
    par1.spines["right"].set_position(("axes", 1))
    make_patch_spines_invisible(par1)
    par1.spines["right"].set_visible(True)
    ################################################


    for x in Configs:
        Wb = x[0]
        Wf = x[1]
        Wp = x[2]
        C  = x[3]
        Wx = Wb + Wf + Wp
        Ws = np.append(Ws, Wx)
        Vr = AP.Vrmax(Wx,S,rho,k,cd0)
        Ve = AP.Vemax(Wx,S,rho,k,cd0)

        Www= np.append(Www, Wx)

        Vrs = np.append(Vrs, Vr)
        Ves = np.append(Ves, Ve)

        Er = AP.enduranceRmaxElectric(Rt,n,ntot,Volt,C,rho,S,k,Wx,cd0)
        Pb = AP.powerbatt(Volt, C, Rt, n, Er)
        ir = AP.current(Volt,Pb)
        irs = np.append(irs, ir)

        E = AP.endurancemaxElectric(Rt,n,ntot,Volt,C,rho,S,k,Wx,cd0)
        Pb = AP.powerbatt(Volt, C, Rt, n, E)
        ie = AP.current(Volt,Pb)
        ies = np.append(ies, ie)
```

```
        if str(x) == Config[0]:
            VrX1 = Vr
            VeX1 = Ve
            irX1 = ir
            ieX1 = ie
            WX1 = Wx
        if str(x) == Config[1]:
            VrX2 = Vr
            VeX2 = Ve
            irX2 = ir
            ieX2 = ie
            WX2 = Wx
        if str(x) == Config[2]:
            VrY1 = Vr
            VeY1 = Ve
            irY1 = ir
            ieY1 = ie
            WY1 = Wx
        if str(x) == Config[3]:
            VrY2 = Vr
            VeY2 = Ve
            irY2 = ir
            ieY2 = ie
            WY2 = Wx
        if str(x) == Config[4]:
            VrZ1 = Vr
            VeZ1 = Ve
            irZ1 = ir
            ieZ1 = ie
            WZ1 = Wx
        if str(x) == Config[5]:
            VrZ2 = Vr
            VeZ2 = Ve
```

```
            irZ2 = ir
            ieZ2 = ie
            WZ2 = Wx


        for i in Config:
            if str(x) == Config[0]:
                # label = "Config X1"
                label = str(C) + " A, " + str(round(Wx/9.80665)) + " kg, n=" + str(n)
                color = "tab:green"
            if str(x) == Config[1]:
                # label = "Config X2"
                label = str(C) +  " A, " + str(round(Wx/9.80665)) + " kg, n=" + str(n)
                color = "tab:blue"
            if str(x) == Config[2]:
                # label = "Config Y1"
                label = str(C) + " A, " + str(round(Wx/9.80665)) + " kg, n=" + str(n)
                color = "tab:orange"
            if str(x) == Config[3]:
                # label = "Config Y2"
                label = str(C) + " A, " + str(round(Wx/9.80665)) + " kg, n=" + str(n)
                color = "tab:cyan"
            if str(x) == Config[4]:
                # label = "Config Z1"
                label = str(C) + " A, " + str(round(Wx/9.80665)) + " kg, n=" + str(n)
                color = "tab:purple"
            if str(x) == Config[5]:
                # label = "Config Z2"
                label = str(C) + " A, " + str(round(Wx/9.80665)) + " kg, n=" + str(n)
                color = "tab:pink"


        if str(x) == Config[0]:
```

```
            host.scatter([WX1/9.80665,WX1/9.80665],
                         [VrX1,VeX1],label=label,color=color)
            par1.scatter([WX1/9.80665,WX1/9.80665],
                         [irX1,ieX1],color=color)
        if str(x) == Config[1]:
            host.scatter([WX2/9.80665,WX2/9.80665],
                         [VrX2,VeX2],label=label,color=color)
            par1.scatter([WX2/9.80665,WX2/9.80665],
                         [irX2,ieX2],color=color)
        if str(x) == Config[2]:
            host.scatter([WY1/9.80665,WY1/9.80665],
                         [VrY1,VeY1],label=label,color=color)
            par1.scatter([WY1/9.80665,WY1/9.80665],
                         [irY1,ieY1],color=color)
        if str(x) == Config[3]:
            host.scatter([WY2/9.80665,WY2/9.80665],
                         [VrY2,VeY2],label=label,color=color)
            par1.scatter([WY2/9.80665,WY2/9.80665],
                         [irY2,ieY2],color=color)
        if str(x) == Config[4]:
            host.scatter([WZ1/9.80665,WZ1/9.80665],
                         [VrZ1,VeZ1],label=label,color=color)
            par1.scatter([WZ1/9.80665,WZ1/9.80665],
                         [irZ1,ieZ1],color=color)
        if str(x) == Config[5]:
            host.scatter([WZ2/9.80665,WZ2/9.80665],
                         [VrZ2,VeZ2],label=label,color=color)
            par1.scatter([WZ2/9.80665,WZ2/9.80665],
                         [irZ2,ieZ2],color=color)


    p1, = host.plot(Ws/9.80665,Vrs, label="Velocity for Max. Range",
                    color="slategrey")
```

```
    p1, = host.plot(Ws/9.80665,Ves, label="Velocity for Max. Endurance",
                    color="black")
    p1, = host.plot([7030,7030],[40,100],linestyle="--",color="tab:red",
                    label="MTOW")
    p2, = par1.plot(Ws/9.80665,irs, label="Current for Max. Range",
                    color="slategrey",linestyle="--")
    p2, = par1.plot(Ws/9.80665,ies, label="Current for Max. Endurance",
                    color="black",linestyle="--")

    # host.set_xlim(0, 80)
    host.set_ylim(40, 100)
    par1.set_ylim(600, 1800)

    host.set_xlabel("Total Weight, kg")
    host.set_ylabel("Flight Velocity, m/s")
    par1.set_ylabel("Current, A")

    l1, = host.plot(Ws/9.80665,Vrs, label="Velocity for Max. Range",
                    color="slategrey")
    l2, = host.plot(Ws/9.80665,Ves, label="Velocity for Max. Endurance",
                    color="black")
    l3, = host.plot([7030,7030],[40,100],linestyle="--",color="tab:red",
                    label="MTOW")
    l4, = par1.plot(Ws/9.80665,irs, label="Current for Max. Range",
                    color="slategrey",linestyle="--")
    l5, = par1.plot(Ws/9.80665,ies, label="Current for Max. Endurance",
                    color="black",linestyle="--")


    lines = [l1,l2,l3,l4,l5]

    # host.legend(lines, [l.get_label() for l in lines])
    # fig.tight_layout(rect=[0, 0.03, 1, 0.95])
```

```

    host.legend(lines,[l.get_label() for l in lines],
        bbox_to_anchor=(0.0, 1.02, 1.0, 0.2),
        loc="lower left",
        ncol=2,
        mode="expand",
        borderaxespad=0.0,
    )



    fig.tight_layout()
    fig.savefig(
        "data_result/Electric N219 V4/"+
        "EelctricN219 Effect of increasing total weight on optimum airspeed " +
        "and current, n = 1.3, "+ str(p) + "PAX.pdf", dpi=600)
    plt.show




#Fig 3.3

for p in PAXs:
    plt.figure(figsize=(6,6))

    Wp = p * PAX
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])
    Ci = C

    Wp = p * PAX
```

```
    Wf = ((Vf * BED)/BSE)*9.80665
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con2 = np.array([Wb,Wf,Wp,C])

    Configs = [Con1, Con2]


    for x in Configs:
        n = 1.3
        Ws = np.array([])
        Es = np.array([])
        for C in Csx:
            Wb = x[0]
            Wf = x[1]
            Wp = x[2]
            Cx  = x[3]
            Wtot = Wb + Wf + Wp
            Ws = np.append(Ws, Wtot)

            E = AP.endurancemaxElectric(Rt,n,ntot,Volt,C,rho,S,k,Wtot,cd0)
            Es = np.append(Es, E)

        if str(x) == Config[0]:
            W = Wtot
            EX1 = AP.endurancemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,W,cd0)
        if str(x) == Config[1]:
            W = Wtot
            EX2 = AP.endurancemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,W,cd0)
        if str(x) == Config[2]:
            W = Wtot
            EY1 = AP.endurancemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,W,cd0)
        if str(x) == Config[3]:
            W = Wtot
```

```
            EY2 = AP.endurancemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,W,cd0)
        if str(x) == Config[4]:
            W = Wtot
            EZ1 = AP.endurancemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,W,cd0)
        if str(x) == Config[5]:
            W = Wtot
            EZ2 = AP.endurancemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,W,cd0)

        for i in Config:
            if str(x) == Config[0]:
                # label = "Config X1"
                label = str(Csi[0]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:green"
            if str(x) == Config[1]:
                # label = "Config X2"
                label = str(Csi[1]) +  " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:blue"
            if str(x) == Config[2]:
                # label = "Config Y1"
                label = str(Csi[2]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:orange"
            if str(x) == Config[3]:
                # label = "Config Y2"
                label = str(Csi[3]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:cyan"
            if str(x) == Config[4]:
                # label = "Config Z1"
                label = str(Csi[4]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:purple"
            if str(x) == Config[5]:
                # label = "Config Z2"
                label = str(Csi[5]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:pink"
```

```

        if str(x) == Config[0]:
            plt.scatter(Csi[0],EX1,label=label,color=color)
            # plt.plot([Csi[0],Csi[0]],[0,3.5],label="MTOW",color='tab:red',linestyle="--")
            plt.plot(Csx,Es,label=str(round(W/9.80665))+" kg/MTOW",color=color)
        if str(x) == Config[1]:
            plt.scatter(Csi[1],EX2,label=label,color=color)
            plt.plot(Csx,Es,label=str(round(W/9.80665))+" kg",color=color)
        if str(x) == Config[2]:
            plt.scatter(Csi[2],EY1,label=label,color=color)
            # plt.plot([Csi[2],Csi[2]],[0,3.5],label="MTOW",color='tab:red',linestyle="--")
            plt.plot(Csx,Es,label=str(round(W/9.80665))+" kg/MTOW",color=color)
        if str(x) == Config[3]:
            plt.scatter(Csi[3],EY2,label=label,color=color)
            plt.plot(Csx,Es,label=str(round(W/9.80665))+" kg",color=color)
        if str(x) == Config[4]:
            plt.scatter(Csi[4],EZ1,label=label,color=color)
            # plt.plot([Csi[4],Csi[4]],[0,3.5],label="MTOW",color='tab:red',linestyle="--")
            plt.plot(Csx,Es,label=str(round(W/9.80665))+" kg/MTOW",color=color)
        if str(x) == Config[5]:
            plt.scatter(Csi[5],EZ2,label=label,color=color)
            plt.plot(Csx,Es,label=str(round(W/9.80665))+" kg",color=color)



        # plt.plot(Csx,Es,label=str(round(Wx/9.80665))+" kg")

    plt.suptitle(str(p) + "PAX")
    plt.xlabel("Capacity, Ah")
    plt.ylabel("Maximum Endurance, h")
    # plt.ylim(0,20)

    plt.legend(
        bbox_to_anchor=(0.0, 1.02, 1.0, 0.2),
```

```
            loc="lower left",
            ncol=2,
            mode="expand",
            borderaxespad=0.0,
        )

    # plt.tight_layout()
    plt.savefig(
        "data_result/Electric N219 V4/"+
        "EelctricN219 Effect of increasing battery capacity on maximum endurance"+
        " n = 1.3, "+ str(p) + "PAX.pdf", dpi=600)
    plt.show





#Fig 3.4
for p in PAXs:
    plt.figure(figsize=(6,6))

    Wp = p * PAX
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])
    Ci = C

    Wp = p * PAX
    Wf = ((Vf * BED)/BSE)*9.80665
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con2 = np.array([Wb,Wf,Wp,C])

    Configs = [Con1, Con2]
```

```

    R1s = np.array([])
    for x in Configs:
        n = 1.3
        Ws = np.array([])
        Rs = np.array([])

        for C in Csx:
            Wb = x[0]
            Wf = x[1]
            Wp = x[2]
            Cx  = x[3]
            Wtot = Wb + Wf + Wp
            Ws = np.append(Ws, Wtot)



            R = AP.rangemaxElectric(Rt,n,ntot,Volt,C,rho,S,k,Wtot,cd0)
            Rs = np.append(Rs, R)

        if str(x) == Config[0]:
            W = Wtot
            RX1 = AP.rangemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,Wsi[0],cd0)
        if str(x) == Config[1]:
            W = Wtot
            RX2 = AP.rangemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,Wsi[1],cd0)
        if str(x) == Config[2]:
            W = Wtot
            RY1 = AP.rangemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,Wsi[2],cd0)
        if str(x) == Config[3]:
            W = Wtot
            RY2 = AP.rangemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,Wsi[3],cd0)
        if str(x) == Config[4]:
            W = Wtot
```

```
            RZ1 = AP.rangemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,Wsi[4],cd0)
        if str(x) == Config[5]:
            W = Wtot
            RZ2 = AP.rangemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,Wsi[5],cd0)


        for i in Config:
            if str(x) == Config[0]:
                # label = "Config X1"
                label = str(Csi[0]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:green"
            if str(x) == Config[1]:
                # label = "Config X2"
                label = str(Csi[1]) +  " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:blue"
            if str(x) == Config[2]:
                # label = "Config Y1"
                label = str(Csi[2]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:orange"
            if str(x) == Config[3]:
                # label = "Config Y2"
                label = str(Csi[3]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:cyan"
            if str(x) == Config[4]:
                # label = "Config Z1"
                label = str(Csi[4]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:purple"
            if str(x) == Config[5]:
                # label = "Config Z2"
                label = str(Csi[5]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:pink"

        if str(x) == Config[0]:
```

```
            plt.scatter(Csi[0],RX1,label=label,color=color)
            # plt.plot([Csi[0],Csi[0]],[0,3.5],label="MTOW",color='tab:red',linestyle="--")
            plt.plot(Csx,Rs,label=str(round(W/9.80665))+" kg/MTOW",color=color)
        if str(x) == Config[1]:
            plt.scatter(Csi[1],RX2,label=label,color=color)
            plt.plot(Csx,Rs,label=str(round(W/9.80665))+" kg",color=color)
        if str(x) == Config[2]:
            plt.scatter(Csi[2],RY1,label=label,color=color)
            # plt.plot([Csi[2],Csi[2]],[0,3.5],label="MTOW",color='tab:red',linestyle="--")
            plt.plot(Csx,Rs,label=str(round(W/9.80665))+" kg/MTOW",color=color)
        if str(x) == Config[3]:
            plt.scatter(Csi[3],RY2,label=label,color=color)
            plt.plot(Csx,Rs,label=str(round(W/9.80665))+" kg",color=color)
        if str(x) == Config[4]:
            plt.scatter(Csi[4],RZ1,label=label,color=color)
            # plt.plot([Csi[4],Csi[4]],[0,3.5],label="MTOW",color='tab:red',linestyle="--")
            plt.plot(Csx,Rs,label=str(round(W/9.80665))+" kg/MTOW",color=color)
        if str(x) == Config[5]:
            plt.scatter(Csi[5],RZ2,label=label,color=color)
            plt.plot(Csx,Rs,label=str(round(W/9.80665))+" kg",color=color)

    if str(p) == str(PAXs[1]):
        plt.plot([0,4000],[1553,1553],linestyle="--",color="tab:red",label="Airbreathing's Perfor

    if str(p) == str(PAXs[2]):
        plt.plot([0,4000],[888,888],linestyle="--",color="tab:red",label="Airbreathing's Performa

    plt.suptitle(str(p) + "PAX")
    plt.xlabel("Capacity, Ah")
    plt.ylabel("Maximum Range, km")
    # plt.ylim(0,350,50)

    plt.legend(
```

```
            bbox_to_anchor=(0.0, 1.02, 1.0, 0.2),
            loc="lower left",
            ncol=2,
            mode="expand",
            borderaxespad=0.0,
        )

        plt.tight_layout()
        plt.savefig(
            "data_result/Electric N219 V4/"+
            "EelctricN219 Effect of increasing battery capacity on maximum range" +
            " n = 1.3, "+ str(p) + "PAX.pdf", dpi=600)
        plt.show




# #Fig 3 AIO

# FIG = plt.figure(figsize=(10, 10))

# # top left
# ax1 = FIG.add_subplot(221)
# ax1.grid(True)
# ax1.set_ylabel("Total Weight, $\mathbf{kg}$")
# ax1.set_xlabel("Capacity, $\mathbf{Ah}$")
# plt.setp(ax1.get_xticklabels(), visible=True)

# # top right
# ax2 = FIG.add_subplot(222)
# ax2.set_ylabel("Flight Velocity, $\mathbf{m/s}$ \n Current, $\mathbf{A}$")
# ax2.set_xlabel("Total Weight, $\mathbf{kg}$")
# ax2.grid(True)
```

```
# plt.setp(ax2.get_xticklabels(), visible=True)

# # bot left
# ax3 = FIG.add_subplot(223)
# ax3.set_ylabel("Maximum Endurance, $\mathbf{h}$")
# ax3.set_xlabel("Capacity, $\mathbf{Ah}$")
# ax3.grid(True)
# plt.setp(ax3.get_xticklabels(), visible=True)

# # bot right
# ax4 = FIG.add_subplot(224)
# ax4.set_ylabel("Maximum Range, $\mathbf{km}$")
# ax4.set_xlabel("Capacity, $\mathbf{Ah}$")
# ax4.grid(True)
# plt.setp(ax4.get_xticklabels(), visible=True)



# n = 1.3
# # Wbatt = 0.4
# # Cbatt = 3/135
# Csx = np.arange(1,C2+1)
# Rs = np.array([]) #aircraft ranges
# Ws = np.array([]) #aircraft weights
# Wx = np.array([]) #same as Ws but for BR 0.3
# Es = np.array([]) #aircraft endurances
# Vrs = np.array([]) #V for max range
# Ves = np.array([]) #V for max endurance
# irs = np.array([]) #current for max range
# ies = np.array([]) #current for max endurance
# for C in Csx:
#     W = (C*Volt/(BSE)) *9.80665
#     Wtot = (W0 + W)
```

```
#       Ws = np.append(Ws, Wtot)
#       R = AP.rangemaxElectric(Rt,n,ntot,Volt,C,rho,S,k,Wtot,cd0)
#       Rs = np.append(Rs, R)
#       E = AP.endurancemaxElectric(Rt,n,ntot,Volt,C,rho,S,k,Wtot,cd0)
#       Es = np.append(Es, E)

#       Vr = AP.Vrmax(Wtot,S,rho,k,cd0)
#       Vrs = np.append(Vrs, Vr)

#       Ve = AP.Vemax(Wtot,S,rho,k,cd0)
#       Ves = np.append(Ves, Ve)

#       Er = AP.enduranceRmaxElectric(Rt,n,ntot,Volt,C,rho,S,k,Wtot,cd0)
#       Pb = AP.powerbatt(Volt, C, Rt, n, Er)
#       ir = AP.current(Volt,Pb)
#       irs = np.append(irs, ir)

#       E = AP.endurancemaxElectric(Rt,n,ntot,Volt,C,rho,S,k,Wtot,cd0)
#       Pb = AP.powerbatt(Volt, C, Rt, n, E)
#       ie = AP.current(Volt,Pb)
#       ies = np.append(ies, ie)

#       if C == C1:
#           Vr1 = Vr
#           Ve1 = Ve
#           ir1 = ir
#           ie1 = ie
#           R1 = R
#           E1 = E

#       if C == C2:
#           Vr2 = Vr
#           Ve2 = Ve
```

```
#           ir2 = ir
#           ie2 = ie
#           R2 = R
#           E2 = E






# ax2.plot(Ws/9.80665,Vrs, label="Velocity for Max. Range")
# ax2.plot(Ws/9.80665,Ves, label="Velocity for Max. Endurance")
# ax2.plot(Ws/9.80665,irs, label="Current for Max. Range")
# ax2.plot(Ws/9.80665,ies, label="Current for Max. Endurance")

# ax2.scatter([W1/9.80665,W1/9.80665,W1/9.80665,W1/9.80665],
#               [Vr1,Ve1,ir1,ie1],color="tab:blue",label="Configuration 1")
# ax2.scatter([W2/9.80665,W2/9.80665,W2/9.80665,W2/9.80665],
#               [Vr2,Ve2,ir2,ie2],color="tab:orange",label="Configuration 2")


# ax1.plot(Csx,Ws/9.80665,color="slategrey",linestyle="-")
# ax1.scatter(C1,W1/9.80665,color="tab:blue",label="Configuration 1")
# ax1.scatter(C2,W2/9.80665,color="tab:orange",label="Configuration 2")

# ax3.plot(Csx,Es,color="slategrey")
# ax3.scatter(C1,E1,color="tab:blue",label="Configuration 1")
# ax3.scatter(C2,E2,color="tab:orange",label="Configuration 2")

# ax4.plot(Csx,Rs,color="slategrey")
# ax4.scatter(C1,R1,color="tab:blue",label="Configuration 1")
# ax4.scatter(C2,R2,color="tab:orange",label="Configuration 2")

```

```
# ax1.legend(
#     bbox_to_anchor=(0.0, 1.02, 1.0, 0.2),
#     loc="lower left",
#     ncol=2,
#     mode="expand",
#     borderaxespad=0.0,
# )
# ax2.legend(
#     bbox_to_anchor=(0.0, 1.02, 1.0, 0.2),
#     loc="lower left",
#     ncol=2,
#     mode="expand",
#     borderaxespad=0.0,
# )
# ax3.legend(
#     bbox_to_anchor=(0.0, 1.02, 1.0, 0.2),
#     loc="lower left",
#     ncol=2,
#     mode="expand",
#     borderaxespad=0.0,
# )
# ax4.legend(
#     bbox_to_anchor=(0.0, 1.02, 1.0, 0.2),
#     loc="lower left",
#     ncol=2,
#     mode="expand",
#     borderaxespad=0.0,
# )

# plt.tight_layout()
# plt.savefig(
#     "data_result/Electric N219 V2/"+
#     "EelctricN219 Effect of increasing battery capacity on weight maximum range"+
```

```
#     " endurance flight velocity and current draw n = 1.3.pdf", dpi=600)
# plt.show()
```

.5 N219 comparison code

```
# -*- coding: utf-8 -*-
"""
Created on Thu Feb 18 19:59:18 2021

@author: mahes
"""


import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
from module import AircraftPerformance as AP
from module import ISAfunc as ISAfunc


MCR = 70 #Maxium Continuous Rating percentage

alt = 10000 #ft
altM = alt * 0.3048 #meter

dt = 0
dts = [0]

#endurance airspeed kts
ekts = 100

#endurance airspeed m/s
```

```
eV = 72

#range airspeed kts
rkts = 120

#range airspeed m/s
rV = 87.5

#endurance SHP and SFC
df = pd.read_csv("data_result\PT6A\pandas\PT6-42 full for pandas - "+str(MCR)+
                 "%mcr.csv")
Rating = df.loc[(df['ALT'] == alt) & (df['ISA'] == dt) & (df['V'] == ekts)]
Te = Rating.iloc[0,5]
SHPe = Rating.iloc[0,8]
SFCe = Rating.iloc[0,9]
FFe = Rating.iloc[0,6]

#range SHP and SFC
df = pd.read_csv("data_result\PT6A\pandas\PT6-42 full for pandas - "+str(MCR)+
                 "%mcr.csv")
Rating = df.loc[(df['ALT'] == alt) & (df['ISA'] == dt) & (df['V'] == rkts)]
Tr = Rating.iloc[0,5]
SHPr = Rating.iloc[0,8]
SFCr = Rating.iloc[0,9]
FFr = Rating.iloc[0,6]



#total efficiency
ntot = 0.81


#air density
```

```
for del_t in dts:
    if (altM <= 11000):
        ISA = ISAfunc.TROPOSPHERE(del_t,altM)
    if (11000 < altM <= 20000):
        ISA = ISAfunc.TROPOPAUSE(del_t,altM)
    if (20000 < altM <= 32000):
        ISA = ISAfunc.LOWERSTRAT(del_t,altM)
    if (32000 < altM <= 47000):
        ISA = ISAfunc.UPPERSTRAT(del_t,altM)
    if (47000 < altM <= 51000):
        ISA = ISAfunc.STRATPAUSE(del_t,altM)
    if (51000 < altM <= 71000):
        ISA = ISAfunc.LOWERMESO(del_t,altM)
    if (71000 < altM <= 80000):
        ISA = ISAfunc.UPPERMESO(del_t,altM)

ISAtotal = np.array([ISA])

rho = ISAtotal[0,2]

#Rt
Rt = 1 #hour

#max aircraft weight/MTOW
W = 68941  #Newton

#aircraft no fuel/battery weight
W0 = 53250.1095 #newton

#fuel/battery weight
Wf = 15690.64 #newton

#fuel/battery volume
```

```
Vf = 1600 #litre

#wing area
S = 41.5 #m^2

#Voltage
Volt = 567 #V

#cd0
cd0 = 0.0357

# wing aspec ratio
AR = 9.16

# oswald's efficiency
e = 0.62921

#k
k = AP.k(AR,e)

#n
ns = [1,1.3]

#battery weight as a fraction of the total weight
BRs = [0.48]

#battery specific energy Wh/kg
# BSE = 250 #tesla 18350
BSE = 265 # Raymer

#battery energy density Wh/l
# BED = 721 #tesla 18350
BED = 700 # Raymer
```

```


#passengger weight
passengger = 85 #kg
luggage = 15 #kg

PAX = (passengger + luggage) *9.80665

######################################
#configurations

#number of pilots and passengger, pilots always 2
p1 = 7
p2 = 11
p3 = 21
PAXs = [p1,p2,p3]


MTOW = 7030 *9.80665 #N, Maximum Take-Off Weight
Wb = 42031 #N, Fuselage Weight

#Config X1
Wp = p1 * PAX
Wf = MTOW - (Wb+Wp)
C = round( ((Wf/9.80665) * BSE)/ Volt )
X1 = np.array([Wb,Wf,Wp,C])

#Config X2
Wp = p1 * PAX
Wf = ((Vf * BED)/BSE)*9.80665
C = round( ((Wf/9.80665) * BSE)/ Volt )
X2 = np.array([Wb,Wf,Wp,C])

```

```
#Config Y1
Wp = p2 * PAX
Wf = MTOW - (Wb+Wp)
C = round( ((Wf/9.80665) * BSE)/ Volt )
Y1 = np.array([Wb,Wf,Wp,C])

#Config Y2
Wp = p2 * PAX
Wf = ((Vf * BED)/BSE)*9.80665
C = round( ((Wf/9.80665) * BSE)/ Volt )
Y2 = np.array([Wb,Wf,Wp,C])

#Config Z1
Wp = p3 * PAX
Wf = MTOW - (Wb+Wp)
C = round( ((Wf/9.80665) * BSE)/ Volt )
Z1 = np.array([Wb,Wf,Wp,C])

#Config Z2
Wp = p3 * PAX
Wf = ((Vf * BED)/BSE)*9.80665
C = round( ((Wf/9.80665) * BSE)/ Volt )
Z2 = np.array([Wb,Wf,Wp,C])

Configs = [X1,X2,Y1,Y2,Z1,Z2]
Config = [str(X1),str(X2),str(Y1),str(Y2),str(Z1),str(Z2)]
# Config  = np.array([X1,X2,Y1,Y2,Z1,Z2])

#Battery capacities array
Csi = np.array([])
for x in Configs:
    C = x[3]
    Csi = np.append(Csi,C)
```

```

#Total Weight Array
Wsi = np.array([])
for x in Configs:
    Wb = x[0]
    Wf = x[1]
    Wp = x[2]
    Wtot = Wb + Wf + Wp
    Wsi = np.append(Wsi,Wtot)

#######################################
#for refrence
for p in PAXs:
    Wp = p * PAX
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])
    Ci = C

    Wp = p * PAX
    Wf = ((Vf * BED)/BSE)*9.80665
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con2 = np.array([Wb,Wf,Wp,C])

    Configs = [Con1, Con2]

#######################################
plt.style.use("ggplot")

matplotlib.rcParams["text.usetex"] = True


for p in PAXs:
```

```python


FIG = plt.figure(figsize=(7, 15))

# Endurance
ax1 = FIG.add_subplot(311)
ax1.grid(True)
ax1.set_ylabel("Endurance ($\mathbf{h}$)")
# ax1.set_ylim(0,4)
plt.setp(ax1.get_xticklabels(), visible=False)

# Range
ax2 = FIG.add_subplot(312)
ax2.set_ylabel("Range ($\mathbf{km}$)")
# ax2.set_xlabel("Velocity ($\mathbf{m/s}$)")
# ax2.set_xticks(np.arange(0,21,5))
# ax2.set_xlim(0,25)
# plt.setp(ax2.get_xticklabels(), visible=True)

# Current
ax3 = FIG.add_subplot(313, sharex=ax1)
ax3.set_ylabel("Current (A)")
ax3.set_xlabel("Velocity ($\mathbf{m/s}$)")
# ax3.set_xticks(np.arange(0,21,5))
# ax3.set_xlim(5,120)
ax3.set_ylim(0,10000)
ax3.grid(True)
# plt.setp(ax2.get_xticklabels(), visible=True)

#Fig 1
#electric

Wp = p * PAX
```

```
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])

    Wp = p * PAX
    Wf = ((Vf * BED)/BSE)*9.80665
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con2 = np.array([Wb,Wf,Wp,C])

    Configs = [Con1, Con2]




    ##########################################################################
    kts = np.arange(120,221,20)

    Vs = np.array([])
    Es = np.array([])
    Rs = np.array([])

    for V in kts:
        Rating = df.loc[(df['ALT'] == alt) & (df['ISA'] == dt) & (df['V'] == V)]
        T = Rating.iloc[0,5]
        SHP = Rating.iloc[0,8]
        SFC = Rating.iloc[0,9]
        FF = Rating.iloc[0,6]
        # nj = Rating.iloc[0,7]

        Vx = V*0.514444

        # W1 = MTOW

```

```
        Win = MTOW
        Wfin = Wb + Wp

        FuelCap = (Win-Wfin)/9.80665
        MaxFuelCap = 1600



        if FuelCap < MaxFuelCap:

            Win = MTOW
            Wfin = Wb + Wp

            CL = AP.CL(Win,rho,Vx,S)
            CD = AP.CD(cd0,CL,AR,e)
            Pa = AP.Pa1(cd0,rho,Vx,S,k,Win)
            nj = AP.nj(Pa,SHP*2)
            CLCD = AP.CLCD(CL,CD)

            E = FuelCap/FF
            R = AP.rangeAirbreathing(nj,SFC*2,CLCD,Win,Wfin)
            #for label
            airbreathing = str(round(FuelCap)) + " L Avtur"

        if FuelCap > MaxFuelCap:

            FuelCap = 1600
            Win = Wb + Wp + (MaxFuelCap * 9.80665)
            Wfin = Wb + Wp

            CL = AP.CL(Win,rho,Vx,S)
            CD = AP.CD(cd0,CL,AR,e)
            Pa = AP.Pa1(cd0,rho,Vx,S,k,Win)
```

```
            nj = AP.nj(Pa,SHP*2)
            CLCD = AP.CLCD(CL,CD)

            E = FuelCap/FF
            R = AP.rangeAirbreathing(nj,SFC*2,CLCD,Win,Wfin)
            #for label
            airbreathing = str(round(FuelCap)) + " L Avtur"



        Es = np.append(Es, E)
        Rs = np.append(Rs, R)
        Vs = np.append(Vs, V)

    line = "."
    color = "tab:red"
    label = airbreathing

    ax1.plot(Vs*0.514444, Es, line, color=color, label=label)
    ax2.plot(Vs*0.514444, Rs, line, color=color, label=label)

    ##########################################################################





    for n in ns:
        for x in Configs:
            Vs = np.linspace(0,120,num=1000)
            Es = np.array([])
            Rs = np.array([])
            Is = np.array([])
            Wb = x[0]
            Wf = x[1]
```

```
            Wp = x[2]
            C  = x[3]
            W1 = Wb + Wf + Wp

            for V in Vs:
                E = AP.enduranceElectric(Rt,n,ntot,Volt,C,rho,V,cd0,W1,k,S)
                Es = np.append(Es, E)
                R = AP.rangeElectric(E,V)
                Rs = np.append(Rs, R)
                Pb = AP.powerbatt(Volt,C,Rt,n,E)
                I = AP.current(Volt,Pb)
                Is = np.append(Is, I)

            if str(x) == Config[0]:
                # label = "Config X1"
                label = str(C) + " A, " + str(round(W1/9.80665)) + " kg, n=" + str(n)
                color = "tab:green"
            if str(x) == Config[1]:
                # label = "Config X2"
                label = str(C) +  " A, " + str(round(W1/9.80665)) + " kg, n=" + str(n)
                color = "tab:blue"
            if str(x) == Config[2]:
                # label = "Config Y1"
                label = str(C) + " A, " + str(round(W1/9.80665)) + " kg, n=" + str(n)
                color = "tab:orange"
            if str(x) == Config[3]:
                # label = "Config Y2"
                label = str(C) + " A, " + str(round(W1/9.80665)) + " kg, n=" + str(n)
                color = "tab:cyan"
            if str(x) == Config[4]:
                # label = "Config Z1"
                label = str(C) + " A, " + str(round(W1/9.80665)) + " kg, n=" + str(n)
                color = "tab:purple"
```

```
            if str(x) == Config[5]:
                # label = "Config Z2"
                label = str(C) + " A, " + str(round(W1/9.80665)) + " kg, n=" + str(n)
                color = "tab:pink"


            # label = str(C) + " Ah, " + str(n) + " n"
            if n == 1:
                line = "--"
            else:
                line = "-"

            ax1.plot(Vs, Es, line, label=label,color=color)
            ax2.plot(Vs, Rs, line, label=label,color=color)
            ax3.plot(Vs, Is, line, label=label,color=color)

    plt.suptitle(str(p) + "PAX")

    ax1.legend(
        bbox_to_anchor=(0.0, 1.02, 1.0, 0.2),
        loc="lower left",
        ncol=2,
        mode="expand",
        borderaxespad=0.0,
    )

    plt.tight_layout()
    plt.savefig("data_result/N219 Comparison/EelctricN219EnduranceRange"+ str(p) +"PAX.pdf",
                dpi=600)
    plt.show()
```

```
#Fig 3.3

Csx = np.arange(1,4001)

for p in PAXs:
    plt.figure(figsize=(6,6))

    Wp = p * PAX
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])
    Ci = C

    Wp = p * PAX
    Wf = ((Vf * BED)/BSE)*9.80665
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con2 = np.array([Wb,Wf,Wp,C])

    Configs = [Con1, Con2]

    #########################################################################
    Rating = df.loc[(df['ALT'] == alt) & (df['ISA'] == dt) & (df['V'] == ekts)]
    FF = Rating.iloc[0,6]

    CL = AP.CL(W1,rho,eV,S)
    CD = AP.CD(cd0,CL,AR,e)
    Pa = AP.Pa1(cd0,rho,eV,S,k,W1)
    nj = AP.nj(Pa,SHPe*2)
```

```
    CLCD = AP.CLCD(CL,CD)
    Win = MTOW
    Wfin = Wb + Wp

    FuelCap = (Win-Wfin)/9.80665
    MaxFuelCap = 1600


    if FuelCap < MaxFuelCap:
        E0 = FuelCap/FF
        #for label
        airbreathing = str(round(FuelCap)) + " L Avtur"

    if FuelCap > MaxFuelCap:
        FuelCap = 1600
        # E0 =  AP.enduranceAirbreathing(nj,SFCe*2,Win,Win-(1600*9.80665),S,rho,CD,CL)
        E0 = FuelCap/FF
        #for label
        airbreathing = str(round(FuelCap)) + " L Avtur"

    plt.plot([0,4000],[E0,E0],linestyle="--",color='tab:red',label=airbreathing)
    ###########################################################################


    for x in Configs:
        n = 1.3
        Ws = np.array([])
        Es = np.array([])
        for C in Csx:
            Wb = x[0]
            Wf = x[1]
            Wp = x[2]
            Cx  = x[3]
```

```
            Wtot = Wb + Wf + Wp
            Ws = np.append(Ws, Wtot)

            E = AP.endurancemaxElectric(Rt,n,ntot,Volt,C,rho,S,k,Wtot,cd0)
            Es = np.append(Es, E)

        if str(x) == Config[0]:
            W = Wtot
            EX1 = AP.endurancemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,W,cd0)
        if str(x) == Config[1]:
            W = Wtot
            EX2 = AP.endurancemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,W,cd0)
        if str(x) == Config[2]:
            W = Wtot
            EY1 = AP.endurancemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,W,cd0)
        if str(x) == Config[3]:
            W = Wtot
            EY2 = AP.endurancemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,W,cd0)
        if str(x) == Config[4]:
            W = Wtot
            EZ1 = AP.endurancemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,W,cd0)
        if str(x) == Config[5]:
            W = Wtot
            EZ2 = AP.endurancemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,W,cd0)

        for i in Config:
            if str(x) == Config[0]:
                # label = "Config X1"
                label = str(Csi[0]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:green"
            if str(x) == Config[1]:
                # label = "Config X2"
                label = str(Csi[1]) +  " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
```

```
                color = "tab:blue"
            if str(x) == Config[2]:
                # label = "Config Y1"
                label = str(Csi[2]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:orange"
            if str(x) == Config[3]:
                # label = "Config Y2"
                label = str(Csi[3]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:cyan"
            if str(x) == Config[4]:
                # label = "Config Z1"
                label = str(Csi[4]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:purple"
            if str(x) == Config[5]:
                # label = "Config Z2"
                label = str(Csi[5]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:pink"

        if str(x) == Config[0]:
            plt.scatter(Csi[0],EX1,label=label,color=color)
            # plt.plot([Csi[0],Csi[0]],[0,3.5],label="MTOW",color='tab:red',linestyle="--")
            plt.plot(Csx,Es,label=str(round(W/9.80665))+" kg/MTOW",color=color)
        if str(x) == Config[1]:
            plt.scatter(Csi[1],EX2,label=label,color=color)
            plt.plot(Csx,Es,label=str(round(W/9.80665))+" kg",color=color)
        if str(x) == Config[2]:
            plt.scatter(Csi[2],EY1,label=label,color=color)
            # plt.plot([Csi[2],Csi[2]],[0,3.5],label="MTOW",color='tab:red',linestyle="--")
            plt.plot(Csx,Es,label=str(round(W/9.80665))+" kg/MTOW",color=color)
        if str(x) == Config[3]:
            plt.scatter(Csi[3],EY2,label=label,color=color)
            plt.plot(Csx,Es,label=str(round(W/9.80665))+" kg",color=color)
        if str(x) == Config[4]:
```

```
            plt.scatter(Csi[4],EZ1,label=label,color=color)
            # plt.plot([Csi[4],Csi[4]],[0,3.5],label="MTOW",color='tab:red',linestyle="--")
            plt.plot(Csx,Es,label=str(round(W/9.80665))+" kg/MTOW",color=color)
        if str(x) == Config[5]:
            plt.scatter(Csi[5],EZ2,label=label,color=color)
            plt.plot(Csx,Es,label=str(round(W/9.80665))+" kg",color=color)


        # plt.plot(Csx,Es,label=str(round(Wx/9.80665))+" kg")




    plt.suptitle(str(p) + "PAX")
    plt.xlabel("Capacity, Ah")
    plt.ylabel("Maximum Endurance, h")
    # plt.ylim(0,20)

    plt.legend(
        bbox_to_anchor=(0.0, 1.02, 1.0, 0.2),
        loc="lower left",
        ncol=2,
        mode="expand",
        borderaxespad=0.0,
    )

    # plt.tight_layout()
    plt.savefig(
        "data_result/N219 Comparison/"+
        "EelctricN219 Effect of increasing battery capacity on maximum endurance"+
        " n = 1.3, "+ str(p) + "PAX.pdf", dpi=600)
    plt.show
```

```
#Fig 3.4
for p in PAXs:

    plt.figure(figsize=(6,6))

    Wp = p * PAX
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])
    Ci = C

    Wp = p * PAX
    Wf = ((Vf * BED)/BSE)*9.80665
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con2 = np.array([Wb,Wf,Wp,C])

    Configs = [Con1, Con2]


    ##########################################################################
    Rating = df.loc[(df['ALT'] == alt) & (df['ISA'] == dt) & (df['V'] == rkts)]
    FF = Rating.iloc[0,6]

    Win = MTOW
    Wfin = Wb + Wp

    FuelCap = (Win-Wfin)/9.80665
    MaxFuelCap = 1600


```

```

    if FuelCap < MaxFuelCap:



        Win = MTOW
        Wfin = Wb + Wp

        V = AP.Vdmin(Win, S, rho, k, cd0)

        CL = AP.CL(Win,rho,V,S)
        CD = AP.CD(cd0,CL,AR,e)
        Pa = AP.Pa1(cd0,rho,V,S,k,Win)
        nj = AP.nj(Pa,SHPr*2)
        CLCD = AP.CLCD(CL,CD)

        R0 = AP.rangeAirbreathing(nj,SFCr*2,CLCD,Win,Wfin)
        # R0 = (FuelCap/FF) * V *3.6
        #for label
        airbreathing = str(round(FuelCap)) + " L Avtur"

    if FuelCap > MaxFuelCap:

        FuelCap = 1600
        Win = Wb + Wp + (MaxFuelCap * 9.80665)
        Wfin = Wb + Wp

        V = AP.Vdmin(Win, S, rho, k, cd0)

        CL = AP.CL(Win,rho,V,S)
        CD = AP.CD(cd0,CL,AR,e)
        Pa = AP.Pa1(cd0,rho,V,S,k,Win)
        nj = AP.nj(Pa,SHPr*2)
```

```
        CLCD = AP.CLCD(CL,CD)

        R0 = AP.rangeAirbreathing(nj,SFCr*2,CLCD,Win,Wfin)
        # R0 = (FuelCap/FF) * V *3.6
        #for label
        airbreathing = str(round(FuelCap)) + " L Avtur"


    # print(R0)
    plt.plot([0,4000],[R0,R0],linestyle="--",color='tab:red',label=airbreathing)
    ##########################################################################


    R1s = np.array([])
    for x in Configs:
        n = 1.3
        Ws = np.array([])
        Rs = np.array([])

        for C in Csx:
            Wb = x[0]
            Wf = x[1]
            Wp = x[2]
            Cx  = x[3]
            Wtot = Wb + Wf + Wp
            Ws = np.append(Ws, Wtot)


            R = AP.rangemaxElectric(Rt,n,ntot,Volt,C,rho,S,k,Wtot,cd0)
            Rs = np.append(Rs, R)

        if str(x) == Config[0]:
            W = Wtot
```

```
            RX1 = AP.rangemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,Wsi[0],cd0)
            print(RX1)
        if str(x) == Config[1]:
            W = Wtot
            RX2 = AP.rangemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,Wsi[1],cd0)
            print(RX2)
        if str(x) == Config[2]:
            W = Wtot
            RY1 = AP.rangemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,Wsi[2],cd0)
            print(RY1)
        if str(x) == Config[3]:
            W = Wtot
            RY2 = AP.rangemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,Wsi[3],cd0)
            print(RY2)
        if str(x) == Config[4]:
            W = Wtot
            RZ1 = AP.rangemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,Wsi[4],cd0)
            print(RZ1)
        if str(x) == Config[5]:
            W = Wtot
            RZ2 = AP.rangemaxElectric(Rt,n,ntot,Volt,Cx,rho,S,k,Wsi[5],cd0)
            print(RZ2)


        for i in Config:
            if str(x) == Config[0]:
                # label = "Config X1"
                label = str(Csi[0]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:green"
            if str(x) == Config[1]:
                # label = "Config X2"
                label = str(Csi[1]) +  " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:blue"
```

```
            if str(x) == Config[2]:
                # label = "Config Y1"
                label = str(Csi[2]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:orange"
            if str(x) == Config[3]:
                # label = "Config Y2"
                label = str(Csi[3]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:cyan"
            if str(x) == Config[4]:
                # label = "Config Z1"
                label = str(Csi[4]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:purple"
            if str(x) == Config[5]:
                # label = "Config Z2"
                label = str(Csi[5]) + " A, " + str(round(W/9.80665)) + " kg, n=" + str(n)
                color = "tab:pink"

        if str(x) == Config[0]:
            plt.scatter(Csi[0],RX1,label=label,color=color)
            # plt.plot([Csi[0],Csi[0]],[0,3.5],label="MTOW",color='tab:red',linestyle="--")
            plt.plot(Csx,Rs,label=str(round(W/9.80665))+" kg/MTOW",color=color)
        if str(x) == Config[1]:
            plt.scatter(Csi[1],RX2,label=label,color=color)
            plt.plot(Csx,Rs,label=str(round(W/9.80665))+" kg",color=color)
        if str(x) == Config[2]:
            plt.scatter(Csi[2],RY1,label=label,color=color)
            # plt.plot([Csi[2],Csi[2]],[0,3.5],label="MTOW",color='tab:red',linestyle="--")
            plt.plot(Csx,Rs,label=str(round(W/9.80665))+" kg/MTOW",color=color)
        if str(x) == Config[3]:
            plt.scatter(Csi[3],RY2,label=label,color=color)
            plt.plot(Csx,Rs,label=str(round(W/9.80665))+" kg",color=color)
        if str(x) == Config[4]:
            plt.scatter(Csi[4],RZ1,label=label,color=color)
```

```
            # plt.plot([Csi[4],Csi[4]],[0,3.5],label="MTOW",color='tab:red',linestyle="--")
            plt.plot(Csx,Rs,label=str(round(W/9.80665))+" kg/MTOW",color=color)
        if str(x) == Config[5]:
            plt.scatter(Csi[5],RZ2,label=label,color=color)
            plt.plot(Csx,Rs,label=str(round(W/9.80665))+" kg",color=color)

    if str(p) == str(PAXs[1]):
        plt.plot([0,4000],[1553,1553],linestyle="--",color="black",label="Official Result")

    if str(p) == str(PAXs[2]):
        plt.plot([0,4000],[888,888],linestyle="--",color="black",label="Official Result")

    plt.suptitle(str(p) + "PAX")
    plt.xlabel("Capacity, Ah")
    plt.ylabel("Maximum Range, km")
    # plt.ylim(0,350,50)

    plt.legend(
        bbox_to_anchor=(0.0, 1.02, 1.0, 0.2),
        loc="lower left",
        ncol=2,
        mode="expand",
        borderaxespad=0.0,
    )

    plt.tight_layout()
    plt.savefig(
        "data_result/N219 Comparison/"+
        "EelctricN219 Effect of increasing battery capacity on maximum range" +
        " n = 1.3, "+ str(p) + "PAX.pdf", dpi=600)
    plt.show
```

.6 N219 tables code

```
# -*- coding: utf-8 -*-
"""
Created on Sat Jun 19 14:50:27 2021

@author: mahes
"""


import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
from module import AircraftPerformance as AP
from module import ISAfunc as ISAfunc


MCR = 70 #Maxium Continuous Rating percentage

#altitudes
alts = [0,5000,10000]
altsM = [alts[0]*0.3048,alts[1]*0.3048,alts[2]*0.3048] #for metrics

#off ISA/temperature deviation
dt = 0
dts = [0]

#total efficiency
ntot = 0.81

#Rt
Rt = 1 #hour

```

```
#max aircraft weight/MTOW
W = 68941  #Newton

#aircraft no fuel/battery weight
W0 = 53250.1095 #newton

#fuel/battery weight
Wf = 15690.64 #newton

#fuel/battery volume
Vf = 1600 #litre

#wing area
S = 41.5 #m^2

#Voltage
Volt = 567 #V

#cd0
cd0 = 0.0357

# wing aspec ratio
AR = 9.16

# oswald's efficiency
e = 0.62921

#k
k = AP.k(AR,e)


#battery specific energy Wh/kg
# BSE = 250 #tesla 18350
```

```
BSE = 265 # Raymer

#battery energy density Wh/l
# BED = 721 #tesla 18350
BED = 700 # Raymer


#passengger weight
passengger = 85 #kg
luggage = 15 #kg

PAX = (passengger + luggage) *9.80665

#######################################
#configurations

#number of pilots and passengger, pilots always 2
p1 = 7
p2 = 11
p3 = 21
PAXs = [p1,p2,p3]


MTOW = 7030 *9.80665 #N, Maximum Take-Off Weight
Wb = 42031 #N, Fuselage Weight
########################################

plt.style.use("ggplot")

matplotlib.rcParams["text.usetex"] = True
```

```
n = 1.3

df = pd.read_csv("data_result\PT6A\pandas\PT6-42 full for pandas - "+str(MCR)+
                  "%mcr.csv")

for p in PAXs:

    kts = np.arange(120,221,20)

    ResA = np.array([]) #results airbreathing

    Hs = np.array([]) #altitude
    rhos = np.array([]) #air density rho
    CLs = np.array([]) #lift coefficients
    CDs = np.array([]) #drag coefficients
    Pas = np.array([]) #power available
    Varr = np.array([]) #airspeed
    Earr = np.array([]) #endurances
    Rarr = np.array([]) #ranges

    Wp = p * PAX
    Wf = MTOW - (Wb+Wp)

    for alt in alts:

        altM = alt * 0.3048

        for del_t in dts:
            if (altM <= 11000):
                ISA = ISAfunc.TROPOSPHERE(del_t,altM)
            if (11000 < altM <= 20000):
                ISA = ISAfunc.TROPOPAUSE(del_t,altM)
            if (20000 < altM <= 32000):
```

```
                ISA = ISAfunc.LOWERSTRAT(del_t,altM)
            if (32000 < altM <= 47000):
                ISA = ISAfunc.UPPERSTRAT(del_t,altM)
            if (47000 < altM <= 51000):
                ISA = ISAfunc.STRATPAUSE(del_t,altM)
            if (51000 < altM <= 71000):
                ISA = ISAfunc.LOWERMESO(del_t,altM)
            if (71000 < altM <= 80000):
                ISA = ISAfunc.UPPERMESO(del_t,altM)

        ISAtotal = np.array([ISA])

        rho = ISAtotal[0,2]

        for V in kts:
            Rating = df.loc[(df['ALT'] == alt) & (df['ISA'] == dt) & (df['V'] == V)]
            T = Rating.iloc[0,5]
            SHP = Rating.iloc[0,8]
            SFC = Rating.iloc[0,9]
            FF = Rating.iloc[0,6]

            Vx = V*0.514444

            Win = MTOW
            Wfin = Wb + Wp

            FuelCap = (Win-Wfin)/9.80665
            MaxFuelCap = 1600

            if FuelCap < MaxFuelCap:

                Win = MTOW
                Wfin = Wb + Wp
```

```

                CL = AP.CL(Win,rho,Vx,S)
                CD = AP.CD(cd0,CL,AR,e)
                Pa = AP.Pa1(cd0,rho,Vx,S,k,Win)
                nj = AP.nj(Pa,SHP*2)
                CLCD = AP.CLCD(CL,CD)

                E = FuelCap/FF
                R = AP.rangeAirbreathing(nj,SFC*2,CLCD,Win,Wfin)

            if FuelCap > MaxFuelCap:

                FuelCap = 1600
                Win = Wb + Wp + (MaxFuelCap * 9.80665)
                Wfin = Wb + Wp

                CL = AP.CL(Win,rho,Vx,S)
                CD = AP.CD(cd0,CL,AR,e)
                Pa = AP.Pa1(cd0,rho,Vx,S,k,Win)
                nj = AP.nj(Pa,SHP*2)
                CLCD = AP.CLCD(CL,CD)

                E = FuelCap/FF
                R = AP.rangeAirbreathing(nj,SFC*2,CLCD,Win,Wfin)

            Hs = np.append(Hs, alt) #altitude
            rhos = np.append(rhos, round(rho,3)) #air density rho
            CLs = np.append(CLs, round(CL,3))
            CDs = np.append(CDs, round(CD,3))
            Pas = np.append(Pas, round(Pa,3))
            Varr = np.append(Varr, round(Vx,3))
            Earr = np.append(Earr, round(E,3))
            Rarr = np.append(Rarr, round(R,3))
```

```


    ResA = np.vstack((Hs,rhos,CLs,CDs,Pas,Varr,Earr,Rarr))


    ResA_transpose = ResA.transpose()

    dfA = pd.DataFrame(data=ResA_transpose,columns=['Altitude','rho','C_L','C_D','Pa','V','E','R'
    dfA.to_csv("data_result/N219 Airbreathing Table/AirbreathingResults, "+ str(p) +" PAX.csv")
    # print(dfA)



for p in PAXs:
    #Electric table kts

    ResA = np.array([]) #results airbreathing

    Hs = np.array([]) #altitude
    rhos = np.array([]) #air density rho
    CLs = np.array([]) #lift coefficients
    CDs = np.array([]) #drag coefficients
    Pas = np.array([]) #power avai lable
    Is = np.array([]) #currents
    Varr = np.array([]) #airspeed
    Earr = np.array([]) #endurances
    Rarr = np.array([]) #ranges

    Wp = p * PAX
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])

```

```
    Configs = [Con1]

    for alt in alts:

        altM = alt * 0.3048

        for del_t in dts:
            if (altM <= 11000):
                ISA = ISAfunc.TROPOSPHERE(del_t,altM)
            if (11000 < altM <= 20000):
                ISA = ISAfunc.TROPOPAUSE(del_t,altM)
            if (20000 < altM <= 32000):
                ISA = ISAfunc.LOWERSTRAT(del_t,altM)
            if (32000 < altM <= 47000):
                ISA = ISAfunc.UPPERSTRAT(del_t,altM)
            if (47000 < altM <= 51000):
                ISA = ISAfunc.STRATPAUSE(del_t,altM)
            if (51000 < altM <= 71000):
                ISA = ISAfunc.LOWERMESO(del_t,altM)
            if (71000 < altM <= 80000):
                ISA = ISAfunc.UPPERMESO(del_t,altM)

        ISAtotal = np.array([ISA])

        rho = ISAtotal[0,2]

        for x in Configs:
            Vs = np.arange(40,121,5)
            Wb = x[0]
            Wf = x[1]
            Wp = x[2]
            C  = x[3]
            W1 = Wb + Wf + Wp
```

```python

            for V in kts:

                V = V*0.514444

                CL = AP.CL(Win,rho,V,S)
                CD = AP.CD(cd0,CL,AR,e)
                Pa = AP.Pa1(cd0,rho,V,S,k,Win)
                CLCD = AP.CLCD(CL,CD)

                E = AP.enduranceElectric(Rt,n,ntot,Volt,C,rho,V,cd0,W1,k,S)
                R = AP.rangeElectric(E,V)
                Pb = AP.powerbatt(Volt,C,Rt,n,E)
                I = AP.current(Volt,Pb)

                Hs = np.append(Hs, alt) #altitude
                rhos = np.append(rhos, round(rho,3)) #air density rho
                CLs = np.append(CLs, round(CL,3))
                CDs = np.append(CDs, round(CD,3))
                Pas = np.append(Pas, round(Pa,3))
                Is = np.append(Is, round(I,3))
                Varr = np.append(Varr, round(V,3))
                Earr = np.append(Earr, round(E,3))
                Rarr = np.append(Rarr, round(R,3))

    ResE = np.vstack((Hs,rhos,CLs,CDs,Pas,Is,Varr,Earr,Rarr))
    ResE_transpose = ResE.transpose()

    dfE = pd.DataFrame(data=ResE_transpose,columns=['Altitude','rho','C_L','C_D','Pa','I','V','E'
    dfE.to_csv("data_result/N219 Electric Table/ElectricResultsKTS, "+ str(p) +" PAX.csv")

    print(dfE)

```

```


for p in PAXs:
    #Electric table m/s

    ResA = np.array([]) #results airbreathing

    Hs = np.array([]) #altitude
    rhos = np.array([]) #air density rho
    CLs = np.array([]) #lift coefficients
    CDs = np.array([]) #drag coefficients
    Pas = np.array([]) #power avai lable
    Is = np.array([]) #currents
    Varr = np.array([]) #airspeed
    Earr = np.array([]) #endurances
    Rarr = np.array([]) #ranges

    Wp = p * PAX
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])

    Configs = [Con1]

    for alt in alts:

        altM = alt * 0.3048

        for del_t in dts:
            if (altM <= 11000):
                ISA = ISAfunc.TROPOSPHERE(del_t,altM)
            if (11000 < altM <= 20000):
                ISA = ISAfunc.TROPOPAUSE(del_t,altM)
```

```
            if (20000 < altM <= 32000):
                ISA = ISAfunc.LOWERSTRAT(del_t,altM)
            if (32000 < altM <= 47000):
                ISA = ISAfunc.UPPERSTRAT(del_t,altM)
            if (47000 < altM <= 51000):
                ISA = ISAfunc.STRATPAUSE(del_t,altM)
            if (51000 < altM <= 71000):
                ISA = ISAfunc.LOWERMESO(del_t,altM)
            if (71000 < altM <= 80000):
                ISA = ISAfunc.UPPERMESO(del_t,altM)

        ISAtotal = np.array([ISA])

        rho = ISAtotal[0,2]

        for x in Configs:
            Vs = np.arange(40,121,5)
            Wb = x[0]
            Wf = x[1]
            Wp = x[2]
            C  = x[3]
            W1 = Wb + Wf + Wp

            for V in Vs:

                CL = AP.CL(Win,rho,V,S)
                CD = AP.CD(cd0,CL,AR,e)
                Pa = AP.Pa1(cd0,rho,V,S,k,Win)
                CLCD = AP.CLCD(CL,CD)

                E = AP.enduranceElectric(Rt,n,ntot,Volt,C,rho,V,cd0,W1,k,S)
                R = AP.rangeElectric(E,V)
                Pb = AP.powerbatt(Volt,C,Rt,n,E)
```

```
                I = AP.current(Volt,Pb)

                Hs = np.append(Hs, alt) #altitude
                rhos = np.append(rhos, round(rho,3)) #air density rho
                CLs = np.append(CLs, round(CL,3))
                CDs = np.append(CDs, round(CD,3))
                Pas = np.append(Pas, round(Pa,3))
                Is = np.append(Is, round(I,3))
                Varr = np.append(Varr, round(V,3))
                Earr = np.append(Earr, round(E,3))
                Rarr = np.append(Rarr, round(R,3))

    ResE = np.vstack((Hs,rhos,CLs,CDs,Pas,Is,Varr,Earr,Rarr))
    ResE_transpose = ResE.transpose()

    dfE = pd.DataFrame(data=ResE_transpose,columns=['Altitude','rho','C_L','C_D','Pa','I','V','E'
    dfE.to_csv("data_result/N219 Electric Table/ElectricResults, "+ str(p) +" PAX.csv")

    # print(dfE)


#Fig 3.3

for p in PAXs:


    ResE = np.array([]) #results electric

    Hs = np.array([]) #altitude
    rhos = np.array([]) #air density rho
    CLs = np.array([]) #lift coefficients
    CDs = np.array([]) #drag coefficients
    Pas = np.array([]) #power avai lable
```

```
    Is = np.array([]) #currents
    Varr = np.array([]) #airspeed
    Earr = np.array([]) #endurances
    Rarr = np.array([]) #ranges

    Wp = p * PAX
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])


    Configs = [Con1]

    alts = np.arange(0,25001,5000)

    for alt in alts:

        altM = alt * 0.3048

        for del_t in dts:
            if (altM <= 11000):
                ISA = ISAfunc.TROPOSPHERE(del_t,altM)
            if (11000 < altM <= 20000):
                ISA = ISAfunc.TROPOPAUSE(del_t,altM)
            if (20000 < altM <= 32000):
                ISA = ISAfunc.LOWERSTRAT(del_t,altM)
            if (32000 < altM <= 47000):
                ISA = ISAfunc.UPPERSTRAT(del_t,altM)
            if (47000 < altM <= 51000):
                ISA = ISAfunc.STRATPAUSE(del_t,altM)
            if (51000 < altM <= 71000):
                ISA = ISAfunc.LOWERMESO(del_t,altM)
            if (71000 < altM <= 80000):
```

```
                ISA = ISAfunc.UPPERMESO(del_t,altM)

        ISAtotal = np.array([ISA])

        rho = ISAtotal[0,2]



        for x in Configs:
            n = 1.3

            Wb = x[0]
            Wf = x[1]
            Wp = x[2]
            Cx  = x[3]
            Wtot = Wb + Wf + Wp

            V = AP.Vemax(Wtot,S,rho,k,cd0)
            CL = AP.CL(Wtot,rho,V,S)
            CD = AP.CD(cd0,CL,AR,e)
            Pa = AP.Pa1(cd0,rho,V,S,k,Wtot)
            Pr = AP.Pr(cd0,rho,V,S,k,Wtot)
            CLCD = AP.CLCD(CL,CD)

            E = AP.endurancemaxElectric(Rt,n,ntot,Volt,C,rho,S,k,Wtot,cd0)
            Pb = AP.powerbatt(Volt,C,Rt,n,E)
            I = AP.current(Volt,Pb)

            Hs = np.append(Hs, alt) #altitude
            rhos = np.append(rhos, round(rho,3)) #air density rho
            CLs = np.append(CLs, round(CL,3))
            CDs = np.append(CDs, round(CD,3))
            Pas = np.append(Pas, round(Pa,3))
            Is = np.append(Is, round(I,3))
```

```
            Varr = np.append(Varr, round(V,3))
            Earr = np.append(Earr, round(E,3))

    ResE = np.vstack((Hs,rhos,CLs,CDs,Pas,Is,Varr,Earr))
    ResE_transpose = ResE.transpose()

    dfE = pd.DataFrame(data=ResE_transpose,columns=['Altitude','rho','C_L','C_D','Pa','I','V','Em
    dfE.to_csv("data_result/N219 Electric Table/ElectricResultsEmax, "+ str(p) +" PAX.csv")

    # print(dfE)



# Fig 3.4

for p in PAXs:


    #Fig 1

    ResE = np.array([]) #results electric

    Hs = np.array([]) #altitude
    rhos = np.array([]) #air density rho
    CLs = np.array([]) #lift coefficients
    CDs = np.array([]) #drag coefficients
    Pas = np.array([]) #power avai lable
    Is = np.array([]) #currents
    Varr = np.array([]) #airspeed
    Earr = np.array([]) #endurances
    Rarr = np.array([]) #ranges

    Wp = p * PAX
```

```
    Wf = MTOW - (Wb+Wp)
    C = round( ((Wf/9.80665) * BSE)/ Volt )
    Con1 = np.array([Wb,Wf,Wp,C])

    Configs = [Con1]

    alts = np.arange(0,25001,5000)

    for alt in alts:

        altM = alt * 0.3048

        for del_t in dts:
            if (altM <= 11000):
                ISA = ISAfunc.TROPOSPHERE(del_t,altM)
            if (11000 < altM <= 20000):
                ISA = ISAfunc.TROPOPAUSE(del_t,altM)
            if (20000 < altM <= 32000):
                ISA = ISAfunc.LOWERSTRAT(del_t,altM)
            if (32000 < altM <= 47000):
                ISA = ISAfunc.UPPERSTRAT(del_t,altM)
            if (47000 < altM <= 51000):
                ISA = ISAfunc.STRATPAUSE(del_t,altM)
            if (51000 < altM <= 71000):
                ISA = ISAfunc.LOWERMESO(del_t,altM)
            if (71000 < altM <= 80000):
                ISA = ISAfunc.UPPERMESO(del_t,altM)

        ISAtotal = np.array([ISA])

        rho = ISAtotal[0,2]
        for x in Configs:
            n = 1.3
```

```

            Wb = x[0]
            Wf = x[1]
            Wp = x[2]
            Cx  = x[3]
            Wtot = Wb + Wf + Wp

            V = AP.Vemax(Wtot,S,rho,k,cd0)
            CL = AP.CL(Wtot,rho,V,S)
            CD = AP.CD(cd0,CL,AR,e)
            Pa = AP.Pa1(cd0,rho,V,S,k,Wtot)
            Pr = AP.Pr(cd0,rho,V,S,k,Wtot)
            CLCD = AP.CLCD(CL,CD)

            R = AP.rangemaxElectric(Rt,n,ntot,Volt,C,rho,S,k,Wtot,cd0)
            E = R/V

            Pb = AP.powerbatt(Volt,C,Rt,n,E)
            I = AP.current(Volt,Pb)

            Hs = np.append(Hs, alt) #altitude
            rhos = np.append(rhos, round(rho,3)) #air density rho
            CLs = np.append(CLs, round(CL,3))
            CDs = np.append(CDs, round(CD,3))
            Pas = np.append(Pas, round(Pa,3))
            Is = np.append(Is, round(I,3))
            Varr = np.append(Varr, round(V,3))
            Rarr = np.append(Rarr, round(R,3))

    ResE = np.vstack((Hs,rhos,CLs,CDs,Pas,Is,Varr,Rarr))
    ResE_transpose = ResE.transpose()

    dfE = pd.DataFrame(data=ResE_transpose,columns=['Altitude','rho','C_L','C_D','Pa','I','V','R
```

```
    dfE.to_csv("data_result/N219 Electric Table/ElectricResultsRmax, "+ str(p) +" PAX.csv")

    print(dfE)
```

Turnitin Report

ANALYSIS OF RANGE AND ENDURANCE OF N-219 UNDER POTENTIAL ELECTRIC PROPULSION

by . . .

Submission date: 25-Jun-2021 07:37AM (UTC-0400)
Submission ID: 1611997652
File name: MaheswaraFile-1.pdf (7.99M)
Word count: 30518
Character count: 140398

ANALYSIS OF RANGE AND ENDURANCE OF N-219 UNDER POTENTIAL ELECTRIC PROPULSION

ORIGINALITY REPORT

9%	5%	7%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.nature.com Internet Source	1%
2	avionicsengineering.files.wordpress.com Internet Source	1%
3	Andreas W. Schäfer, Steven R. H. Barrett, Khan Doyme, Lynnette M. Dray et al. "Technological, economic and environmental prospects of all-electric aircraft", Nature Energy, 2018 Publication	1%
4	Venkatasubramanian Viswanathan, B. Matthew Knapp. "Potential for electric aircraft", Nature Sustainability, 2019 Publication	1%
5	Lance W. Traub. "Range and Endurance Estimates for Battery-Powered Aircraft", Journal of Aircraft, 2011 Publication	1%
6	www.indonesian-aerospace.com Internet Source	<1%

7	Bandu N. Pamadi. "Performance, Stability, Dynamics, and Control of Airplanes, Second Edition", American Institute of Aeronautics and Astronautics (AIAA), 2004 Publication	<1 %
8	S. K. Ojha. "Flight Performance of Aircraft", American Institute of Aeronautics and Astronautics (AIAA), 1995 Publication	<1 %
9	archive.org Internet Source	<1 %
10	"Fundamentals of Airplane Flight Mechanics", Springer Science and Business Media LLC, 2007 Publication	<1 %
11	E. Torenbeek, H. Wittenberg. "Flight Physics", Springer Science and Business Media LLC, 2009 Publication	<1 %
12	www.yumpu.com Internet Source	<1 %
13	Ali Paziresh, Amir Hossein Nikseresht, Hashem Moradi. "Wing-Body and Vertical Tail Interference Effects on Downwash Rate of the Horizontal Tail in Subsonic Flow", Journal of Aerospace Engineering, 2017 Publication	<1 %

14	Submitted to Cranfield University Student Paper	<1 %
15	Steven A. Brandt, John J. Bertin, Randall J. Stiles, Ray Whitford. "Introduction to Aeronautics", American Institute of Aeronautics and Astronautics (AIAA), 2004 Publication	<1 %
16	Torenbeek. "Aeroplane Performance", Flight Physics, 2009 Publication	<1 %
17	es.scribd.com Internet Source	<1 %
18	Submitted to Whitefriars Catholic College Student Paper	<1 %
19	Sadraey, . "Design of Control Surfaces", Aircraft Design A Systems Engineering Approach, 2012. Publication	<1 %
20	hdl.handle.net Internet Source	<1 %
21	dspace.lib.cranfield.ac.uk Internet Source	<1 %
22	Basic Flight Mechanics, 2016. Publication	<1 %
23	Submitted to UC, Boulder Student Paper	

<1%

24 Mario Asselin. "An Introduction to Aircraft Performance", American Institute of Aeronautics and Astronautics (AIAA), 1997
Publication
<1%

25 Submitted to Military Technological College
Student Paper
<1%

26 Submitted to Technische Universiteit Delft
Student Paper
<1%

27 Submitted to Universidad Carlos III de Madrid
Student Paper
<1%

28 pt.scribd.com
Internet Source
<1%

29 Submitted to Southern New Hampshire University - Continuing Education
Student Paper
<1%

30 Submitted to University of Central Lancashire
Student Paper
<1%

31 Lothar Birk. "Fundamentals of Ship Hydrodynamics", Wiley, 2019
Publication
<1%

32 "Proceedings of the International Conference on Aerospace System Science and
<1%

Engineering 2020", Springer Science and Business Media LLC, 2021
Publication

33	Hsu Myat NAING, Nhu Van NGUYEN, Jae-Woo LEE, Sangho KIM. "Aircraft Design Optimization Process using a Decision Model by Database", TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES, 2015 Publication	<1 %
34	Reza N. Jazar. "Vehicle Dynamics", Springer Science and Business Media LLC, 2017 Publication	<1 %
35	mafiadoc.com Internet Source	<1 %
36	silo.pub Internet Source	<1 %
37	Submitted to Coventry University Student Paper	<1 %
38	digitalcommons.wpi.edu Internet Source	<1 %
39	ntrs.nasa.gov Internet Source	<1 %
40	researchbank.rmit.edu.au Internet Source	<1 %

41	Torenbeek. "High-Speed Flight", Flight Physics, 2009 Publication	<1 %
42	Submitted to Tyler Junior College Student Paper	<1 %
43	Denis Howe. "Aircraft Conceptual Design Synthesis", Wiley, 2000 Publication	<1 %
44	J.J. Wang, Y.C. Li, K.-S. Choi. "Gurney flap—Lift enhancement, mechanisms and applications", Progress in Aerospace Sciences, 2008 Publication	<1 %
45	Submitted to Brooklands College Student Paper	<1 %
46	Frederick O. Smetana. "Introductory Aerodynamics and Hydrodynamics of Wings and Bodies: A Software-Based Approach", American Institute of Aeronautics and Astronautics (AIAA), 1997 Publication	<1 %
47	John T. Lowry. "Performance of Light Aircraft", American Institute of Aeronautics and Astronautics (AIAA), 1999 Publication	<1 %
48	Submitted to Sheffield Hallam University Student Paper	<1 %

49	Submitted to University of Surrey Student Paper	<1 %
50	academicworks.cuny.edu Internet Source	<1 %
51	Submitted to Emirates Aviation College, Aerospace & Academic Studies Student Paper	<1 %
52	Baarspul, M.. "A review of flight simulation techniques", Progress in Aerospace Sciences, 1990 Publication	<1 %
53	Begüm Şenses, Anil V. Rao. "Optimal Finite-Thrust Small Spacecraft Aeroassisted Orbital Transfer", Journal of Guidance, Control, and Dynamics, 2013 Publication	<1 %
54	basicmechanicalengineering.com Internet Source	<1 %
55	bibdigital.epn.edu.ec Internet Source	<1 %
56	en.wikipedia.org Internet Source	<1 %
57	orca.cf.ac.uk Internet Source	<1 %

58 E. Torenbeek, H. Wittenberg. "Chapter 6 Aeroplane Performance", Springer Science and Business Media LLC, 2009 — <1%
Publication

59 Guang-Hong Yang, Kai-Yew Lum. "An optimization approach to integrated aircraft-controller design", Proceedings of the 2003 American Control Conference, 2003., 2003 — <1%
Publication

60 John Hwang, Satadru Roy, Jason Kao, Joaquim Martins, William A. Crossley. "Simultaneous aircraft allocation and mission optimization using a modular adjoint approach", 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2015 — <1%
Publication

61 pdfs.semanticscholar.org — <1%
Internet Source

62 vtechworks.lib.vt.edu — <1%
Internet Source

63 www.coursehero.com — <1%
Internet Source

64 "Advances in Natural Computation", Springer Science and Business Media LLC, 2006 — <1%
Publication

65 Gritt Ahrens. "Transport Systems", Springer Handbook of Mechanical Engineering, 2009
Publication
<1%

66 Hans Jürgen Ohlbach. "Chapter 6 Theory Unification in Abstract Clause Graphs", Springer Science and Business Media LLC, 1986
Publication
<1%

67 Hong Zhang. "Rapid object recognition from high resolution SAR image supported by geo-database", IEEE International IEEE International IEEE International Geoscience and Remote Sensing Symposium 2004 IGARSS 04 Proceedings 2004, 2004
Publication
<1%

68 I.W Eames, S Aphornratana, H Haider. "A theoretical and experimental study of a small-scale steam jet refrigerator", International Journal of Refrigeration, 1995
Publication
<1%

69 Sadraey, . "Tail Design", Aircraft Design A Systems Engineering Approach, 2012.
Publication
<1%

70 aviationdose.com
Internet Source
<1%

71 data.torre-solar.es
Internet Source
<1%

72	de.scribd.com Internet Source	<1 %
73	docplayer.net Internet Source	<1 %
74	www.easa.europa.eu Internet Source	<1 %
75	www.efita.org Internet Source	<1 %
76	www.gpo.gov Internet Source	<1 %
77	"103 Trigonometry Problems", Springer Science and Business Media LLC, 2005 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

Curriculum Vitae

	Basic Information
Name	Maheswara Sinatriyo
Place of Birth	South Jakarta, Indonesia
Date of Birth	26, 01, 1999
Address	Bali View, F7/21, Cirendeu, South Tangerang, Banten, Indonesia 15419
Email	maheswara.sinatriyo@gmail.com
Year	**Education**
2016 - present	International University Liaison Indonesia
2013 - 2016	Kharisma Bangsa Senior High School
Year	**Work Experiences**
2019-2020	TransNusa air (Intern)
2017	Indonesian Aerospace (Intern)
Year	**Organizational Experiences**
2017-2018	Chairman of IULI's student executive board
2017	Head of property division for IULI's orientation week