INTERNATIONAL UNIVERSITY LIAISON INDONESIA

BACHELOR'S THESIS

COMPARISON OF STATISTICAL WEIGHT METHODS APPLIED FOR REGIONAL TURBOPROP AIRCRAFT

By

Raka Pradana Sanferdi
110201601024

Presented to the Faculty of Engineering
In Partial Fulfilment Of the Requirements for the Degree of

SARJANA TEKNIK
In
AVIATION ENGINEERING

FACULTY OF ENGINEERING

Banten 15310
Indonesia
August 2023

APPROVAL PAGE

COMPARISON OF STATISTICAL WEIGHT METHODS APPLIED FOR REGIONAL TURBOPROP AIRCRAFT

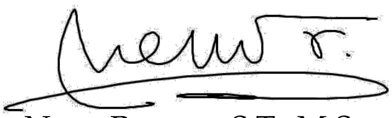
RAKA PRADANA SANFERDI
110201601024
Presented to the Faculty of Engineering
In Partial Fulfillment of the Requirements for the Degree of

SARJANA TEKNIK
In
AVIATION ENGINEERING

FACULTY OF ENGINEERING

Triwanto Simanjuntak, PhD	________
Thesis Advisor	Date
Dr. Eng. Ressa Octavianty	________
Thesis Co-Advisor	Date
Dipl.-Ing. Sentot Wahjoe Georitno, M.Si.	________
Dean of Faculty of Engineering	Date

EXAMINERS APPROVAL PAGE



Neno Ruseno, S.T., M.Sc.
Examiner 1

15/08/2023
Date

Ananta Wijaya, S.T., M. Sc. Tech.
Examiner 2

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.

Raka Pradana Sanferdi
Student

Date

ABSTRACT

Comparison of Statistical Weight Methods Applied For Regional Turboprop Aircraft

by

Raka Pradana Sanferdi

Triwanto Simanjuntak, PhD, Advisor

Dr. Eng. Ressa Octavianty, Co-Advisor

The purpose of this research is to examine and contrast several statistical weight methods for use with regional turboprop aircraft. Estimating the many weight components of an aircraft, such as its empty weight, payload weight, fuel weight, and maximum takeoff weight, requires the use of an essential tool known as the statistical weight techniques. This research examines and compares the Raymer method, the Torenbeek method, Cessna method, and the USAF method, which are all popular choices for calculating statistical weights. A case study is carried out in order to compare various methodologies, and three regional turboprop aircraft are used as the reference aircraft for the study — ATR 42-600, Saab 340, CN-235. This research makes a contribution to the current body of knowledge by assessing and comparing various approaches within the specific context of regional turboprop aircraft. The results from this thesis showed that the Raymer's method tended to overestimate the weight while the Torenbeek's method underestimate it; USAF method gave moderate estimation. Moreover for the three aircraft analyzed, averaging the computations from the three methods gave best estimation — under 10% of error — to the actual total weight components (Empty Maximum Take-off Weight/EMTOW).

Keyword: *aircraft design, regional, turboprop, weights*

ACKNOWLEDGEMENTS

The author would like to begin by expressing his gratitude to god, the almighty Allah SWT, for giving him with the chance to carry out this research and for all of the other favors he has bestowed upon him. The author would also want to express their gratitude to the following wonderful individuals who have supported him in his endeavors and believed in what he does.

1. Triwanto Simanjuntak, PhD, spending time as my thesis advisor, supervisor, and mentor during the whole year. He is an amazing mentor. many thanks for bearing with the author's delays and for providing him with support and confidence.

2. Dr. Eng. Ressa Octavianty, spending time as my second thesis advisor, supervisor, and mentor during my thesis work. She is a kind and patient mentor. Thank you for providing guidance with all the shortcomings that are owned by the author.

3. Pak Neno and Pak Ananta for their valuable inputs as examiners.

4. The author owes a great debt of gratitude to both of his or her parents for the love and support they've provided over the years. They have played a significant role in shaping the author into the person he is today.

5. Some of my closest best friends: Rao, Maheswara, Tauhid, Ifnu, Agha, who are always there to listen to what the author has to say about his brilliant ideas and peculiar ways of thinking. The author would like to express his gratitude and affection to each of them.

6. The author gratitude also goes to Pak Anto, people at IULI, Academic Services, and staff, for helping and allowing the author to conduct the research at IULI facility.

7. Freya, who inspired me to be resolute about working on my thesis.

Contents

List of Figures

List of Tables

List of Abbreviations

AC	**A**lternating **C**urrent
APU	**A**uxiliary **P**ower **Unit**
ATP	**A**dvance **T**urbo **P**rop
AVIC	**AV**iation **I**ndustry **C**orporation of China
CASA	**C**onstrucciones **A**eronáuticas **S**A
EMTOW	**E**mpty **M**aximum **T**ake **O**ff **W**eight
FE	**F**inite **E**lement
GA	**G**eneral **A**ircraft
GDP	**G**ross **D**omestic **P**roduct
HT	**H**orizontal **T**ail
IPTN	**I**ndustri **P**esawat **T**erbang **N**usantara
LSA	**L**ight **S**port **A**ircraft
MTOW	**M**aximum **T**ake **O**ff **W**eight
NATO	**N**orth **A**lantic **T**reaty **O**rganization
RNLAF	**R**oyal **N**etherlands **A**ir **F**orce
STOL	**S**hort **T**ake **O**ff **L**anding
TOGW	**T**ake **O**ff **G**ross **W**eight
TSL	**T**hrust **L**oading
USAF	**U**nited **S**tates **A**ir **F**orce
VT	**V**ertical **T**ail

Dedicated to my lovely parents

CHAPTER 1
INTRODUCTION

1.1 Background

The process of designing a new aircraft may be an extremely difficult and time-consuming endeavor that calls for in-depth knowledge and experience in a wide variety of subjects, including aerodynamics, materials science, propulsion, systems engineering, and many more. It often entails a group of engineers and designers working together to develop a concept for the aircraft that not only satisfies the requirements of the intended purpose of the aircraft but also can be constructed and operated safely and effectively. The procedure might take a lot of time and entails a significant financial commitment to research and development. The design process for an aircraft normally consists of multiple stages, each of which has its own goals and task. The primary phases of design in aircraft design include the Conceptual Design, the Preliminary Design, the Detailed Design, the Prototype and Testing Phase, the Production and Manufacturing Phase, and the Certification and Regulatory Compliance Phase. It can be difficult to accurately calculate aircraft weights during the design phase, and obtaining absolute precision is frequently not achievable due to the presence of a number of different elements. For the purposes of performance analysis, maintaining structural integrity, maintaining stability and control, determining cargo capacity, fuel efficiency, calculating range, and ensuring compliance with regulatory standards, knowing the weights of the separate aircraft components is essential. However, in order to estimate and approximatively determine aircraft weights with the highest possible degree of precision, engineers make use of a variety of analytical approaches, empirical data, computer simulations, and historical data.

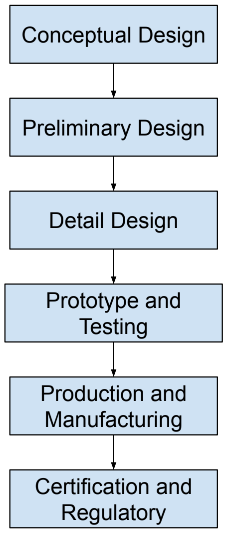


FIGURE 1.1: An elementary outline of the aircraft design process

Many different kinds of procedures may be used to obtain an exact and reliable weight estimations data of an aircraft. Although it is difficult to achieve absolute precision when calculating aircraft weights, this does not mean that it is impossible to do so. The statistical method of weight estimation is the one that is utilized the most frequently. The empirical formulas, regression models, and historical data generated from already-existing aircraft that form the basis of statistical weight estimation methods establish the basis for developing weight estimation relationships. When estimating the weight of components, these connections take into account a variety of criteria, including aircraft size, mission profile, propulsion system, and configuration. The ability to make more precise weight predictions is made possible by using statistical methods, which help detect trends and correlations between weight and design characteristics.

Using statistical methods in order to estimate the weight of aircraft presents both a number of positive and negative aspects. Scalability is enabled via statistical methods, which may be applied to a wide variety of aircraft sizes, configurations, and mission profiles. The relationships that were found through statistical analysis can be used to a wide variety of aircraft designs, which makes the process of estimating weight more effective and adaptable. The process of benchmarking helps find possible areas for making improvements or cutting weight. Statistical tools make it possible to benchmark new aircraft against current ones. Engineers are able to evaluate the feasibility and competitiveness of a design by comparing the estimated weight of a new design to the weight of similar aircraft that have already been constructed and put into operation. However, this is also due to the fact that statistical methods are dependent on historical data as well as correlations between weight and design elements. Despite the fact that they offer helpful estimations, their accuracy is inherently constrained by the quality and relevancy of the data that is readily available. It is possible that the accuracy of the weight predictions will be affected either because the historical data does not fully represent the design that is now being investigated or because there are major design variances.

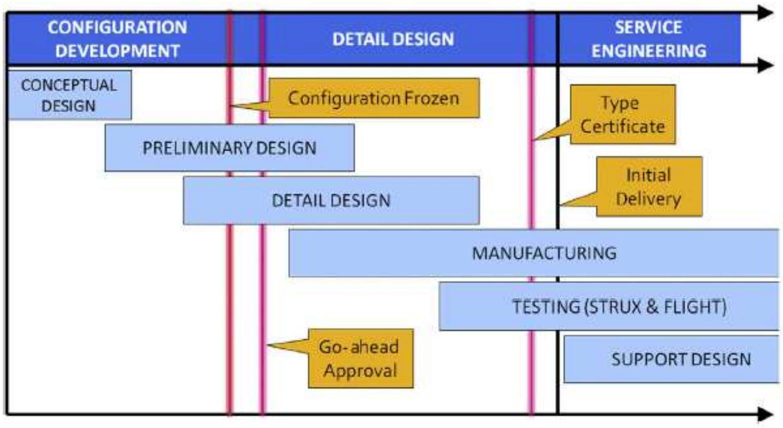


FIGURE 1.2: Aircraft Design Process Proposed by Torenbeek. Reprinted from [1]

Estimating the weight of an airplane can be done using any one of a number of different statistical methods. The equations that were developed as a result can be used to provide an estimate of the weight of new aircraft based on the design specifications of those aircraft. The Raymer method and the Torenbeek method are two of these procedures that are more well-known in the business world and are utilized rather frequently. They offer methodical ways to the estimation of aircraft weight, taking into account a variety of design elements as well as empirical data. On the other hand, the United States Air Force (USAF) technique is largely utilized within the United States Air Force and may be tailored to the requirements of that organization. It is important, considering the current demand trend in regional turboprop aircraft, to investigate how well these various methods of estimating weight perform in the context of calculating the weights of aircraft of this type. Estimating the weights of regional turboprop aircraft is one of the many applications for the Raymer method, which is also commonly employed in the conceptual design of aircraft. It does it by disassembling the airplane into its component parts and calculating the weights of those parts through the application of empirical equations.

FIGURE 1.3: ATR 42-600S

This method offers an organized approach. Estimating the weights of regional turboprop aircraft can also be accomplished with the help of the Torenbeek approach, which is predicated on regression analysis. This method can produce weight estimates that are reliable to a reasonable degree since it involves creating regression relationships between weight and the relevant design factors. In order to create appropriate regression models, it is possible that it will be essential to collect and examine historical data that is unique to regional turboprop aircraft. Even though it is most commonly utilized within the United States Air Force, the USAF weight estimating approach has the potential to also be relevant to regional turboprop aircraft. In order to determine the correlations between different weights, this method uses statistical analysis and regression techniques. However, it can call for some adaptation and calibration based on the data and design considerations that are unique to turboprops.

1.2 Problem Statement

It's possible that the Raymer method, the Torenbeek method, and the USAF method all estimate the weight components of regional turboprop aircraft in different ways. Both the Raymer method and the Torenbeek method place a primary emphasis on estimating the weight of the individual components of an aircraft, such as the wing, the fuselage, the empennage, the landing gear, the systems, and so on. When estimating the weights of these components based on design characteristics and previous data, these methods make use of empirical equations or regression models to come up with estimates. There is a possibility that the Raymer approach and the Torenbeek method will use different equations, data sources, and methodologies altogether. On the other hand, the USAF technique is an approach to weight estimation that was created by the United States Air Force. The specific weight component estimating procedures utilized by the USAF method might not be made available to the general public.

When attempting to calculate the useful payload mass of an aircraft, it is necessary to take into account the weight of the passengers, cargo, and any other objects that are carried throughout the flight. The useful payload mass is estimated using the Raymer method, which takes into account the intended mission profile of the aircraft, the passenger capacity, the cargo capacity, and other pertinent design factors. In order to estimate the payload mass, it frequently makes use of empirical relationships that are generated from historical data. Establishing statistical correlations between weight and design factors is the primary goal of the Torenbeek method, which is predicated on regression analysis. It does so by taking into account variables that are pertinent, such as the length of the fuselage, the volume, or other parameters that are connected to the amount of cargo or passengers that may be carried. In order to determine the payload mass, regression models that were constructed using historical data that was specific to regional turboprop aircraft are utilized. The particular strategy that the USAF system takes in order to estimate the useable payload mass would depend on the method's proprietary protocols, which are not made available to the general public.

1.3 Research Objective

1. The goal is to create a tool (code written in Python) that will be implement in four methods(Raymer, Torenbeek, USAF, Cessna);

2. Utilizing the tools provided, the author will compare the various approaches used to estimate the weights of regional turboprops;

3. To be able to analyze the trends of the methodologies that are used to estimate the useable payload mass of the aircraft.

1.4 Research Scope

1. The main point of this thesis is to compare three different ways to estimate the weight of an an aircraft. The goal of the thesis is to compare how well and accurately the following three methods work; Raymer, Torenbeek, USAF, and with additionaly Cessna method for sanity check.

2. Regional turboprop aircraft are the focus of the thesis because they are unique, have specific operational needs, and are in high demand in the aviation business. The thesis compares things in a way that takes into account and works around the unique challenges and things to think about when estimating weight components of an aircraft.

3. Only three regional turboprop aircraft are used to make comparisons in the thesis. The main point of the thesis is to look at how these three different aircraft weights were estimated using the methods chosen.

4. In this thesis, the author used only data that was readily available to the public, as indicated by the reference.

1.5 Significant of Study

1. This study is possible and can be utilized and implemented for a wider analysis of other turboprop aircraft for the regional turboprop class

2. This study has the potential to serve as a baseline for subsequent research using a variety of aircraft belonging to a variety of classes.

3. This thesis has the potential to encourage additional research and collaboration in the topic of aircraft weight estimate. As a result, this might lead to the improvement of already used methods, the development of new approaches, and the expansion of knowledge in this sector.

CHAPTER 2
LITERATURE REVIEW

2.1 Aircraft Design

The design of modern aircraft places an emphasis on the integration of newly developed technologies and systems with both the conventional and the advanced layouts. This covers the development of brand-new structures, materials, and production methods [2]. The creation of an aircraft that is dependable enough to fly safely for the entirety of the design life of the aircraft while also being strong, lightweight, and economical requires following a certain process that is known as the *Aircraft Design Process*. This approach is used to strike a balance between a number of competing and demanding criteria.

By methodically assessing important parts of the aircraft, the design process makes it possible to find and fix flaws [1]. This is accomplished through the use of mathematical techniques at the conceptual design phase. However, this requires detailed testing of the aerodynamic and structural structure, materials, avionics, control system architecture, and many other things.

2.2 Aircraft Design Objective

There are several reasons why new airplanes are designed. Most are created to perform a specific function or mission that is mandated by potential customers or thought to be necessary for customers. The development of new aircraft is expensive, hence careful planning must be taken when designing them. No matter what kind of aircraft is being built or why, a number of specified tasks need to be finished before it can be constructed and flown.

2.3 Aircraft Design Phases

The major phases of aircraft design consist of three phases: *Conceptual Design Phase*, *Preliminary Design Phase*, and *Detail Design*. However, the requirements phase, often known as the *Initial Phase*, is where the aircraft design process must start before an Aircraft enters the conceptual design stage, during which the *Required Mission*, *Capability*, and *Regulatory* constraints are formulated.

The specifications may be as straightforward as a few lines listing desired features (such as range, cruising speed, and cargo) or as detailed as a document with thousands of pages, addressing factors like environmental effect, operating costs, maintainability, hardware, avionics, and ergonomics, to mention a few [1]. The design lead must demonstrate that the aircraft has a reasonable probability of achieving the requirements during the Conceptual Design Phase, which is the next stage.

2.3.1 Conceptual Design Phase

During the *Conceptual Design* stage of a new aircraft, designers will assess a wide range of various concepts in an effort to find the one that best satisfies the requirements. including aerodynamics, propulsion, performance, structural systems, control systems and many more. Additionally, designers must take into consideration factors like the fuselage shape, the location of the wings, the size of the engines, and more.This calls for them to sketch a concept, examine it, and then rate and contrast how well it works in successive iterations.

2.3.2 Preliminary Design Phase

The following stage is *Preliminary Design*, which comes after conceptual design is finished. The conceptual design is optimized at this point to match the required constraints. During this phase, it is typical to have one or more aircraft components adjusted or redesigned. At this point, testing is done in a wind tunnel, and computational fluid dynamics is used to determine how the flow field surrounding the aircraft should be modeled. At this point, structural and control assessments are also carried out . Before moving on to the third and final stage of the design process, engineers will also check for and fix structural problems and defects. It verifies the idea's validity, highlights potential issues, and provides chances to consider potential remedies.

2.3.3 Detailed Design Phase

The fabrication-related components of the design are completed at the **detailed design stage**. Any design effort that involves the airframe's detailed design and system integration (such as airframe design and engine installation) is referred to by this term. It's important to take into account detailed design from two angles:

(1) When discussing the design of the prototype aircraft's systems and airframe during prototyping.

(2) During the development of manufacturing, when it refers to the design of the production aircraft's airframe and systems. Sustaining engineering is a term used to describe some of this type of design work.

2.4 Preliminary Sizing of Aircraft Design

The *Preliminary Sizing* process begins after the rapid sizing process is complete and is typically the most resource-intensive stage of the sizing operations. Preliminary sizing or initial sizing is the estimation of aircraft design take-off gross weight. It is often carried out during the preliminary design phase of an aircraft and is based on a more accurate finite element(FE) model that shows the explicit structural layout of the wing box but omits some specific structural details [3]. The preliminary sizing handles a more complex trade-off on the ideal balance between weight and price.

2.5 Preliminary Sizing Process in The Design of Aircraft

The process of defining an overall aircraft size by estimating essential factors such as takeoff gross weight (TOGW), wing reference area, and thrust is known as aircraft *Initial Sizing*. Thrust loading (TSL/WTO) and wing loading (WTO/S) are used to analyze these critical factors. It is critical to choose the right combination of thrust and wing loadings because different combinations result in distinct geometry aircrafts. Therefore, finding an optimum combination of thrust and wing loadings is required.

The objective of the constraint analysis, which is one of the first steps in the process of sizing, is to identify the ideal ratio of thrust to wing loadings. The production of a constraint diagram is one of the initial jobs in any new aircraft design. The diagram makes it possible to determine the power plant and wing area requirements for the aircraft in order to ensure that all performance standards are met [1]. The constraint diagram is created by mapping constraints onto the unique, two-dimensional design space graph. It is the collection of all potential outcomes given the selected variables. A constraint is a requirement for a certain design that must be met. An isopleth is used to depict it.

Typically, it is expressed as thrust loading (T/W), where T is thrust, W is weight, and S is wing area. This form can be written as T/W=f(W/S), where T is thrust, W is weight, and S is wing area. Because the wing loading (W/S) is plotted along the x-axis and the thrust-to-weight ratio (T/W) is plotted along the y-axis in this way, you should think of W/S as representing x and T/W as representing y. The graph can be evaluated by noting that any W/S and T/W combinations that are above the constraint curves indicate that the design exceeds the necessary values. This can be done by comparing the W/S ratio to the T/W ratio. [1].

The designers of aircraft have made use of constraint analysis in order to choose the most promising design among a number of different combinations of thrust and wing loadings. Using their intuition and prior design experiences, designers have roughly chosen an optimal place where constraint lines cross or a position with a small margin. However, due to uncertainty in the parameters used to determine performance requirements, this chosen point may not adhere to the limits, ultimately leading to an unreliable design.

2.6 Aircraft Weight Major Contributor

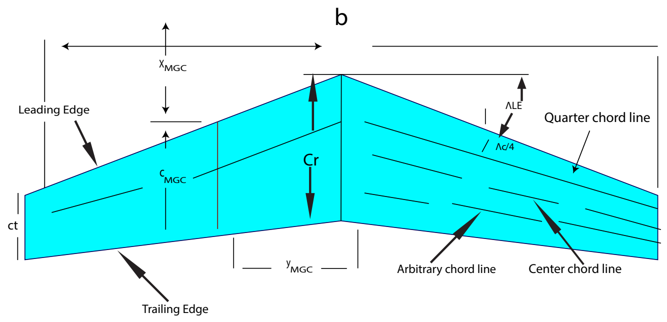
One of the many factors that contribute to the safe and efficient operation of an aircraft is having proper control over the aircraft's weight and balance. The correct loading of the aircraft, the maintenance of the weight and balance records, and the weighing of the aircraft are the three components of the weight and balance system that is widely employed by aircraft. Each of these three aspects of the system is of equal significance. A mistake in any one of these components renders the system useless. There exist five primary components that are responsible for the overall mass of an aircraft. The aircraft comprises five primary components, namely the engine, wing, landing gear, fuselage, and empennage. Aircrafts are composed of numerous components, however, there exist five fundamental constituents that are deemed crucial in their operation.

2.6.1 Engine

Every airplane's engine provides the power that propels it forward. It is the power house of the aircraft. As piston engine, a type of internal combustion engine, is the engine found in the majority of aircraft. By doing so, it implies that it burns fuel inside of a combustion chamber, producing heat and pressure that drive the pistons that propel air at a high speed through the fan. The crankshaft, which turns and spins inside the engine case, is what drives the engines, which are normally seen on the front of airplanes.

2.6.2 Wing

One of the components of an aircraft that is essential for flight is the wing, which can also be referred to as the foils. The wings are positioned on the outside of the craft and are called "wings." The majority of the necessary upward force for flight is generated by the airflow that passes over the wings.. Additionally, the aerodynamic support that wings provide for an aircraft's stability during takeoff and landing includes increased lift, decreased drag, directional stability, and changes in surface area that enable lift to be maintained for longer periods of time than would be possible without wings in specific positions or configurations [4]. Of all its components, an airplane's wings are the longest and thickest.



Fundamental definitions of a trapezoidal wing planform

FIGURE 2.1: Wing Geometry

2.6.3 Landing Gear

One of an airplane's most vital components is the landing gear. It keeps the aircraft in the air and keeps it from colliding with the earth. It landing gear will be lowered so that it wheels can touch down precisely at the end of the runway, protecting the aircraft from harm. A retractable, horizontal surface called the landing gear anchors the aircraft to the ground. Its purpose is to direct the aircraft as it descends, and when it is retracted, it increases lift for takeoff and landing. To ensure that it can support the weight of the aircraft, the landing gear is mostly carbon fiber composite materials and springs.

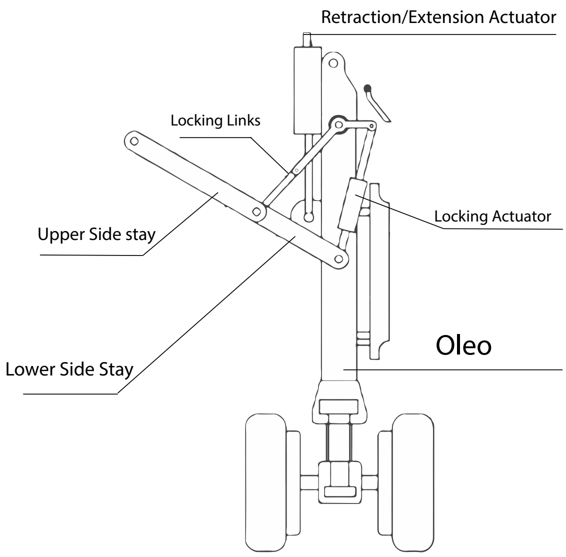


FIGURE 2.2: Landing Gear

2.6.4 Fuselage

An airplane's primary body is called the fuselage. It is typically a long, cylindrical tube that houses the fuel tanks, the engines, the passengers, the cargo, the flying controls, and other interior parts. Near the front of the fuselage is where the cockpit is situated. The fuselage produces a huge empty space around the wings and tail sections and supports their structural integrity. It is essential to the safety of airplanes since it contains all these essential components. Additionally, the fuselage is in charge of transporting the whole cabin of an airplane, which includes all of the passengers' belongings, luggage, and other trip-related necessities.

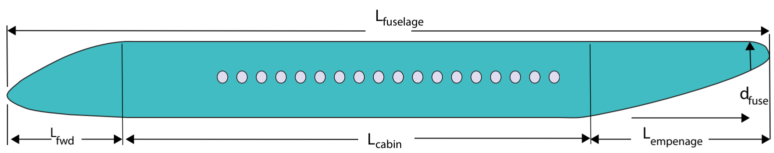


FIGURE 2.3: Fuselage

2.6.5 Empennage

A fixed-wing aircraft's empennage is the back part of the tail assembly. It contains what referred to as flight control surfaces, or horizontal and vertical stabilizers. These control surfaces aid in the plane's lateral and vertical glide as well as maintaining its stability during flight.

This section also includes additional components that are essential to performance and security, like the wingtips and airfoil cowlings. The vertical stabilizer, rudders (little yokes that control the angle of the aircraft's nose), elevators (smaller yokes that control how much your aircraft leans up or down while it travels), and vertical stabilizers are the extensions of these parts of the aircraft.

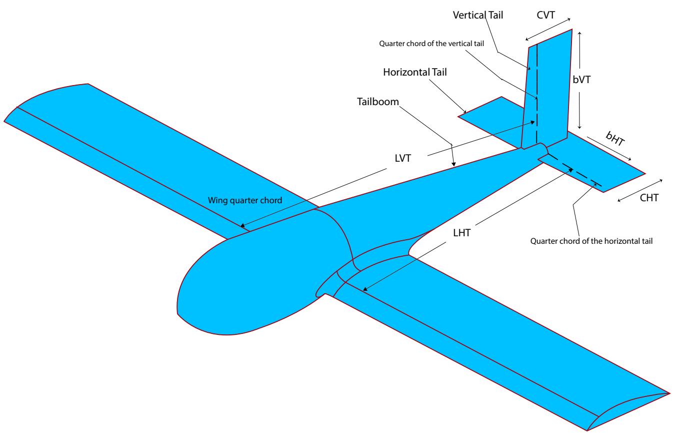


FIGURE 2.4: Empennage

2.7 Aircraft Weight Estimation Analysis Method

The most crucial design variables utilized in aircraft development is weight of the aircraft itself. The weight of the aircraft must be appropriate for it to complete its intended mission without degrading its performance. The cost of an airplane, which is another important factor for customers (airlines), is primarily determined by its weight. As a result, manufacturers constantly make great efforts to make the aircraft as light as it possibly can without having to disrupt the functionality requested or demanded by the customers. Early in the aircraft design phase, estimating weight accurately is a challenging issue. The airplane weight may be precisely determined once the detailed design drawings are finished by assessing each component and adding them all together.

One of the most crucial responsibilities in the entire aircraft design process is weight estimation. Although sophisticated mathematical tools are not required, the task can be rather difficult. Weight estimating approaches are being expanded in tandem with the design processes

2.7.1 Initial Weight Analysis Method

Initial weight analysis method is the first method. This method outlines the steps that need to be taken in order to complete the initial weight estimation of an airplane. The analysis includes the initial weights of the vehicle with fuel, without fuel, and gross. These weights are then refined utilizing secondary weight estimate techniques. For this objective, three approaches are described. The first step is to determine the empty and fuel-weight ratios of previously built aircraft that belong to the same class as the one that is presently being designed. The next step is to make the case that if the new aircraft's mission and certification basis are comparable to those of the reference aircraft, then its empty- and fuel-weight ratios ought to be comparable to levels that have been historically observed. The empty, fuel, and gross weights of the brand-new airplane can all be calculated with the help of an approximation of these ratios. The number of reference aircraft and the degree to which they resemble the aircraft for which these procedures are being developed are two factors that determine how accurate these procedures will be.

1. **Initial Gross Weight Estimation Using Historical Relations**

   If the total weight is unknown, this technique can be used. Take care not to underestimate or exaggerate the situation. Make that the airplanes in the database are all of the same class and have the same characteristics.

   Using ratios of empty to fuel weight, we may express the fuel and empty weights as,

$$\text{Fuel weight:} \quad W_f = \left(\frac{W_f}{W_0}\right) W_0 \tag{2.1}$$

$$\text{Empty weight:} \quad W_e = \left(\frac{W_e}{W_0}\right) W_0 \tag{2.2}$$

$$\text{Design Gross Weight:} \quad W_0 = \left(\frac{W_e}{W_0}\right) W_0 + W_c + \left(\frac{W_f}{W_0}\right) W_0 + W_p \tag{2.3}$$

This can be solved for W_0, giving us a formula we can use to estimate the gross weight based on the weight ratios.

$$W_0 = \frac{W_c + W_p}{1 - \left(\frac{W_e}{W_0}\right) - \left(\frac{W_f}{W_0}\right)} \tag{2.4}$$

2. **Historical Empty Weight Fractions**

When the asymptotical total gross weight is acquired, this method is then used. In this particular situation, we wish to assume that this is the case for many other types of aircraft, such as the light-sport aircraft (LSA), which should not weigh more than 1320 lbf or 1430 lbf if it is amphibious. There are also circumstances in which it is desirable for the aircraft to have a gross weight that is greater than the specified limit.

The following set of equation below let us to estimate a "historical" empty-weight ratio for the newly designed aircraft, provided that the gross weight for the aircraft is known. As a result, it enables to estimate the weight when it is empty, followed by the usable load, and so on.

$$\text{Sailplanes(35):} \quad \frac{W_e}{W_0} = \begin{cases} 0.2950 + 0.0386 \cdot \ln W_0 & \text{if } W_0 \text{ is in lb}_\text{f} \\ 0.3255 + 0.0386 \cdot \ln W_0 & \text{if } W_0 \text{ is in kg} \end{cases} \tag{2.5}$$

$$\text{Powered Sailplanes(13):} \quad \frac{W_e}{W_0} = \begin{cases} 0.3068 + 0.0510 \cdot \ln W_0 & \text{if } W_0 \text{ is in lb} \\ 0.3471 + 0.0510 \cdot \ln W_0 & \text{if } W_0 \text{ is in kg} \end{cases} \tag{2.6}$$

$$\text{LSA(land): } \frac{W_e}{W_0} = \begin{cases} 1.5451 - 0.1402 \cdot \ln W_0 & \text{if } W_0 \text{ is in lb}_\text{f} \\ 1.4343 - 0.1402 \cdot \ln W_0 & \text{if } W_0 \text{ is in kg} \end{cases} \tag{2.7}$$

$$\text{LSA(amphib): } \frac{W_e}{W_0} = \begin{cases} 1.6351 - 0.1402 \cdot \ln W_0 & \text{if } W_0 \text{ is in lb}_\text{f} \\ 1.5243 - 0.1402 \cdot \ln W_0 & \text{if } W_0 \text{ is in kg} \end{cases} \tag{2.8}$$

$$\text{GA Single Engine(86): } \frac{W_e}{W_0} = \begin{cases} 0.8841 - 0.0333 \cdot \ln W_0 & \text{if } W_0 \text{ is in lb}_\text{f} \\ 0.8578 - 0.0333 \cdot \ln W_0 & \text{if } W_0 \text{ is in kg} \end{cases} \tag{2.9}$$

$$\text{GA Twin Piston(12): } \frac{W_e}{W_0} = \begin{cases} 0.4074 + 0.0253 \cdot \ln W_0 & \text{if } W_0 \text{ is in lb}_\text{f} \\ 0.4274 + 0.0253 \cdot \ln W_0 & \text{if } W_0 \text{ is in kg} \end{cases} \tag{2.10}$$

$$\text{GA Twin Turboprop(28)} \frac{W_e}{W_0} = \begin{cases} 0.5319 + 0.0066 \cdot \ln W_0 & \text{if } W_0 \text{ is in lb}_\text{f} \\ 0.5371 + 0.0066 \cdot \ln W_0 & \text{if } W_0 \text{ is in kg} \end{cases} \tag{2.11}$$

$$\text{Agricultural(5): } \frac{W_e}{W_0} = \begin{cases} 1.4029 - 0.0995 \cdot \ln W_0 & \text{if } W_0 \text{ is in } lb_f \\ 1.3242 - 0.0995 \cdot \ln W_0 & \text{if } W_0 \text{ is in kg} \end{cases} \tag{2.12}$$

$$\text{Business Jett(72): } \frac{W_e}{W_0} = \begin{cases} 0.9038 - 0.03163 \cdot \ln W_0 & \text{if } W_0 \text{ is in } lb_f \\ 0.8788 - 0.03163 \cdot \ln W_0 & \text{if } W_0 \text{ is in kg} \end{cases} \tag{2.13}$$

3. **Initial Gross Weight Estimation Using Mission Analysis**

You can use this method when the gross weight is *UNKNOWN* and you are constructing an aircraft to deliver a given payload over a specific range (or endurance) in accordance with a specialized design mission (including, but not limited to, long range or long endurance aircraft). In other words, when the gross weight is unknown, you can use this method. This method computes the gross weight by first analyzing the anticipated mission profile and then integrating the results of that analysis with the empty weight ratios derived from the Historical Empty Weight Fractions Equations.

For this weight estimate, a fully stated design goal is used. The aircraft's flight path is used to figure out its gross weight. Starting at the start-of-position (0), the engine will start at the (design) gross weight (W0) and run until the end of the task, when the engine will be turned off(5). Along each section, fuel weight and flight time are used to guess how much the plane weighs. This is done by putting the chain of weight parts in relation to the overall weight (W0) in the following way:

$$\text{Weight for mission segment 0 to 1: } W_1 = W_0\left(\frac{W_1}{W_0}\right) \tag{2.14}$$

$$\text{Weight for mission segment 1 to 2: } W_2 = W_1\left(\frac{W_2}{W_1}\right) = W_0\left(\frac{W_1}{W_0}\right)\left(\frac{W_2}{W_1}\right) \tag{2.15}$$

$$\text{Weight for mission segment 2 to 3: } W_3 = W_2\left(\frac{W_3}{W_2}\right) = W_0\left(\frac{W_1}{W_0}\right)\left(\frac{W_2}{W_1}\right)\left(\frac{W_3}{W_2}\right) \text{ etc.} \tag{2.16}$$

Using this method, the aircraft's final mission weight can be expressed as follows.

$$\begin{aligned} W_N &= W_0\left(\frac{W_1}{W_0}\right)\left(\frac{W_2}{W_1}\right)\cdots\left(\frac{W_i}{W_{i-1}}\right)\cdots\left(\frac{W_N}{W_{N-1}}\right) \\ &= W_0\prod_{i=1}^{N}\frac{W_i}{W_{i-1}} \end{aligned} \tag{2.17}$$

Thus, the weight fraction at the end-of-mission is:

$$\frac{W_N}{W_0} = \prod_{i=1}^{N} \frac{W_i}{W_{i-1}} \tag{2.18}$$

When accounting for all of the reserve fuel, it is ideal to do it in terms of weight fractions. If we make the assumption that the aircraft uses up all of its fuel by the time the trip is through, then the final weight fraction may be linked to the aircraft's empty weight, crew weight, and cargo in the following way:

$$\frac{W_N}{W_0} = \frac{W_e + W_c + W_p}{W_0} \Leftrightarrow \left(\frac{W_e}{W_0}\right)_{m\text{ miss}} = \frac{W_N}{W_0} - \frac{W_c + W_p}{W_0} \tag{2.19}$$

2.7.2 Secondary Weight Analysis Methods

Secondary weight analysis refers to any and all processes for weight estimates that are applied after the original weight analysis has been completed. Additional knowledge about the new aircraft can be gained by the designer through the use of the secondary weight analysis. Obviously, it takes a lot more time to finish as well, at least while it is being generated in a spreadsheet or by computer code. Because of the component weight that is provided at this stage, target weights of sub-components can be prepared, and a weight budget can be established.

2.7.3 Statistical Weight Estimation Methods

Aircraft weight estimation is crucial in the design process, as it affects various aspects such as performance, fuel efficiency, structural integrity, and overall safety. Traditional weight estimation methods involve using engineering equations and historical data to estimate the weight of individual components and subsystems. Statistical weight estimation methods rely on historical data from existing aircraft. These methods are particularly useful in early stages of aircraft design when detailed information might be lacking or when quick estimations are required. Statistical weight estimation methods are always based on a certain class of aircraft, such as general aviation planes, commercial planes, combat planes, and so forth [1]. Such classes share characteristics that boost the formulation's correctness. Direct weight estimation, also known as component weight estimation based on material volume and density, is typically necessary for determining the weight of components such as wings, fuselage, HT, VT, and control surfaces. The method makes use of a streamlined structural investigation of an idealized aluminum wing as its basis. The technique can readily be adapted to work with a variety of different lifting surfaces.

2.7.4 Weight of Aircraft Components in Statistical Estimation

The data that is utilized in the statistical methods that are used to estimate the weight of aircraft originates from aircraft that are currently in operation. It is important to know the weight of the wing structure for a population of aircraft that fall into a specific class (for example, GA aircraft), in order to build correlations based on geometric parameters such as wing area, aspect ratio, taper ratio, ultimate load factors, and so on. These parameters include the area of the wing, the aspect ratio, the taper ratio, and the ultimate load factors.

The parts of an airplane are made from many different kinds of materials and are put together with rivets, bolts, screws, welding, or adhesive. The parts of an airplane that hold it together are made to carry weight or prevent stress. There may be more than one stress on a single part of the assembly. Most of the time, the structural parts are made to carry loads, not bend. That is, they are made to be under tension or compression, not bending.

1. **Wing Weight Structure**

   When an airplane moves quickly through the air, the wings are made to lift off the ground. The design of any given plane relies on a number of things, like its size, weight, how it will be used, the speed it wants to fly and land at, and how fast it wants to climb. Some wings on aircraft have a "cantilever" shape, which means that they don't need any support from the outside. The skin is part of the structure of the wing and bears some of the forces on the wing. Other aircraft wings use braces, wires, and other types of external bracing to help hold the wing up and carry the aerodynamic and landing loads. Wings can be made out of both aluminum metal and magnesium alloy.

Cessna :

$$W_W = 0.04674 \cdot (n_z W_0)^{0.397} S_W^{0.360} AR_W^{1.712} \text{ (Cantilever)} \tag{2.20}$$

$$W_W = 0.002933 \cdot n_z^{0.611} S_W^{1.018} AR_W^{2.473} \text{(Strut-braced))} \tag{2.21}$$

Raymer :

$$\begin{aligned} W_W =& 0.036 \cdot S_W^{0.758} W_{FW}^{0.0035} \left(\frac{AR_W}{\cos^2 \Lambda_{c/4}}\right)^{0.6} \\ & \cdot q^{0.006} \lambda_W^{0.04} \left(\frac{100 \cdot t/c}{\cos \Lambda_{c/4}}\right)^{-0.3} (n_z W_0)^{0.49} \end{aligned} \tag{2.22}$$

Torenbeek :

$$\begin{aligned} W_W =& 0.00125 \cdot W_0 \left(\frac{b_W}{\cos \Lambda_{c/2}}\right)^{0.75} \\ & \cdot \left(1 + \sqrt{\frac{6.3 \cos \Lambda_{c/2}}{b_W}}\right)^{n_z} 0.55 \\ & \cdot \left(\frac{b_W S_W}{t_{W\max} W_0 \cos \Lambda_{c/2}}\right)^{0.30} \end{aligned} \tag{2.23}$$

USAF :

$$\begin{aligned} W_W =& 96.948 \cdot \left[\left(\frac{n_z W_0}{10^5}\right)^{0.65} \left(\frac{AR_W}{\cos^2 \Lambda_{c/4}}\right)^{0.57} \right. \\ & \left. \cdot \left(\frac{S_W}{100}\right)^{0.61} \left(\frac{1 + \lambda_W}{2(t/c)}\right)^{0.36} \sqrt{1 + \frac{V_H}{500}}\right] 0.993 \end{aligned} \tag{2.24}$$

Where :

$b_W =$ Wingspan in ft

$S_W =$ Trapezoidal wing area in ft^2

$AR_W =$ Aspect Ratio of wing

$\lambda_W =$ Taper ratio of wing

$\Lambda_{c/4} = W$ Wing sweep at 25%MGC

$\Lambda_{c/2} =$ Wing sweep at 50%MGC

$t/c =$ Wing thickness-to-chord ratio (maximum)

t_W max = Max thickness of the wing root chord in ft

$W_W =$ Predicted weight of wing in lb_f

$W_{FW} =$ Weight of fuel in wing in $\text{lb}_{\text{f.}}$ (If $W_{FW} = 0$ then let $W_{FW}^{0.0035} = 1$)

$q =$ Dynamic pressure at cruise $\left(\text{lb}_\text{f}/\text{ft}^2\right)$

$n_Z =$ Ulimate load factor (= $1.5 \times$ limit load factor)

$W_0 =$ Design gross weight in lb_f

$V_H =$ Maximum level airspeed at S – L in KEAS

The Cessna equations should only be used for aircraft of the Cessna type, which are tiny, have very poor performance, and have maximum speeds of less than 200 knots. Cantilever wings and strut braced wings are the sorts of wings that can be modeled using these equations. Both equations take into account the weight of the wing control surfaces and the wing tip fairing, but they do not take into account the influence of the sweep angle on the fuel tanks or the carry through structure of the wing and fuselage spars. Wings in this category have maximum thickness of ratio of around 18 percent or 0.18. The equation used by the USAF is applicable to aircraft of the light and utility types with performance of up to about 300 knots. And Torenbeek is applicable to light transport aircraft that have a maximum take-off weight of less than 12,500 pounds.

Variables	Cessna	Raymer	Torenbeek	USAF
bW	X	X	✓	X
Sw	✓	✓	X	✓
ARw	✓	✓	X	✓
λ_w	X	✓	X	✓
$\Lambda_{c/4}$	X	✓	X	✓
$\Lambda_{c/2}$	X	X	✓	X
t/c	✓	X	X	✓
twmax	X	X	✓	X
Ww	✓	✓	✓	✓
WFW	X	✓	X	X
q	X	X	✓	X
nz	✓	✓	✓	✓
Wo	✓	✓	✓	✓
VH	X	X	X	✓

TABLE 2.1: Wing Weight Variables.

2. **Empennage**

People often refer to the empennage as the ”tail section,.” The empennage is the whole tail group, which comprises solid parts like the vertical fin or stabilizer and the horizontal stabilizer, as well as moving parts like the rudder and rudder trim tabs, the elevator and elevator trim tabs, and so on. The plane's horizontal rotation (called ”yaw”) and vertical rotation (called ”pitch”) are controlled by these surfaces that can move [5]. In some places, the horizontal surface of the empennage can be carried as a single unit from the pilot to change the plane's pitch attitude or trim. These kinds of shapes are usually called stabilizers, flying tails, or slab tails. So, the empennage gives the plane direction and horizontal balance (stability) and provides the pilot with a way to control and move the plane.

The weight of the Horizontal tail (stabilizer and elevator) may be predicted using the expressions that are provided below.

Cessna :

$$W_{HT} = \frac{3.184 W_0^{0.887} S_{HT}^{0.101} AR_{HT}^{0.138}}{174.04 t_{HT\,\max}^{0.223}} \tag{2.25}$$

Raymer :

$$\begin{aligned} W_{HT} =& 0.016 \left(n_z W_0\right)^{0.414} q^{0.168} S_{HT}^{0.896} \left(\frac{100 \cdot t/c}{\cos \Lambda_{c/4}}\right)^{-0.12} \\ & \cdot \left(\frac{AR_W}{\cos^2 \Lambda_{HT}}\right)^{0.043} \lambda_{HT}^{-0.02} \end{aligned} \tag{2.26}$$

Torenbeek :

$$W_{EMP} = 0.04 \left[n_z \left(S_{HT} + S_{VT}\right)^2\right]^{0.75} \tag{2.27}$$

USAF :

$$\begin{aligned} W_{HT} =& 71.927 \left[\left(\frac{n_z W_0}{10^5}\right)^{0.87} \left(\frac{S_{HT}}{100}\right)^{1.2} \left(\frac{l_{HT}}{10}\right)^{0.483} \right. \\ & \left. \cdot \sqrt{\frac{b_{HT}}{t_{HT\,\max}}}\right] 0.458 \end{aligned} \tag{2.28}$$

where:

b_{HT} = HT span in ft

S_{HT} = Trapezoidal HT area in ft^2

AR_{HT} = Aspect Ratio of HT

λ_{HT} = HT taper ratio

Λ_{HT} = HT sweep at 25%MGC

W_{HT} = Predicted weight of HT *in* lb_f

$W_{EMP} = W_{HT} + W_{VT}$ = Combined weight of HT and VT in lb_f

l_{HT} = Horizontal tail arm, from wing c / 4 to HT c / 4 in ft

$t_{HT\,\max}$ = Max root chord thickness of HT *in* ft

The following formula, which applies to both conventional and T-tail layouts, may be used to make predictions about the weight of the VT (fin and rudder).

Cessna :

$$W_{VT} = (1 + 0.2F_{\text{tail}})\,\frac{1.68W_0^{0.567}S_{VT}^{0.1249}AR_{VT}^{0.452}}{639.95t_{VT\,\text{max}}^{0.747}\,(\cos\Lambda_{VT})^{0.882}} \tag{2.29}$$

Raymer :

$$\begin{aligned} W_{VT} =&0.073\,(1 + 0.2F_{\text{tail}})\,(n_z W_0)^{0.376}\,q^{0.122} \\ &\cdot S_{VT}^{0.873}\left(\frac{100\cdot t/c}{\cos\Lambda_{VT}}\right)^{-0.49}\cdot\left(\frac{AR_W}{\cos^2\Lambda_{VT}}\right)^{0.357}\lambda_{VT}^{0.039} \end{aligned} \tag{2.30}$$

Torenbeek : Weight of HT and VT combined in Equation

USAF :

$$W_{VT} = 55.786\,(1 + 0.2F_{\text{tail}})\left[\left(\frac{n_z W_0}{10^5}\right)^{0.87}\left(\frac{S_{VT}}{100}\right)^{1.2}\sqrt{\frac{b_{VT}}{t_{VT\,\text{max}}}}\right]^{0.458} \tag{2.31}$$

where :

b_{VT} = VT span in ft

S_{VT} = Trapezoidal VT area in ft^2

AR_{VT} = Aspect Ratio of VT

λ_{VT} = VT taper ratio

Λ_{VT} = VT sweep at 25%MGC

$t_{VT\,\text{max}}$ = Max root chord thickness of VT in ft

W_{VT} = Predicted weight of VT in lb_f

F_{tail} = 0 for conventional tail, $= 1 for$ T $- tail$

The Cessna equations should only be used for aircraft of the Cessna type-class, which are small, have very poor performance, and have maximum speeds of less than 200 knots. Take note that there is no consideration given to horizontal tail sweep in the equation for vertical tail. The equation used by the USAF is applicable to aircraft of the light and utility types with performance of up to about 300 knots. Take note that the sweep angle is not a consideration in the calculation for the vertical tail. The torenbeek equation may be used to light transport aircraft that have a design dive speed of up to 250 knots and that have a standard layout for their tails. In addition, Raymer's equation provides the most conservative estimate of the overall value.

Variables	Cessna	Raymer	Torenbeek	USAF
bHT	X	X	X	✓
SHT	✓	✓	✓	✓
ARHT	✓	X	X	X
λ_{HT}	X	✓	X	X
Λ_{HT}	X	✓	X	X
WHT	✓	✓	✓	✓
WEMP	X	X	✓	X
LHT	X	X	X	✓
THTMAX	✓	X	X	✓
q	X	✓	X	X
nz	X	✓	✓	✓
Wo	✓	✓	X	✓

TABLE 2.2: Horizontal Weight Variables.

Variables	Cessna	Raymer	Torenbeek	USAF
bVT	X	X	X	✓
SVT	✓	✓	X	✓
ARVT	✓	X	X	X
λ_{VT}	X	✓	X	X
Λ_{VT}	X	✓	X	X
TVTMAX	✓	X	X	✓
WVT	✓	✓	✓	✓
Ftail	✓	✓	X	✓
q	✓	X	X	X
nz	X	✓	✓	✓
Wo	✓	✓	X	✓

TABLE 2.3: Vertical Tail Variables.

3. **Fuselage**

The fuselage of the aircraft serves as the principal structural component of the aircraft. It affords space for passengers, controls, and a variety of accessories in addition to equipment. In single-engine aircraft, it also serves as the location for the engine. It is possible for the engines of a multi-engine aircraft to be housed within the wing structure, attached to the wing structure, or even suspended from the wing structure. They differ mostly in terms of the arrangement and size of the various compartments.

Cessna :

Cessna:

$$W_{\mathrm{FUS}} = 0.04682 W_0^{0.692} R_{\max}^{0.374} 0_{FS}^{590} \text{ (Low-wing)}$$

$$W_{\mathrm{FUS}} = 14.86 W_0^{0.144} \left(\frac{l_{FS}}{R_{\max}}\right)^{0.778} l_{FS}^{0.383} N_{OCC}^{0.455} \tag{2.32}$$

(High-wing)

Raymer :

$$W_{FUS} = 0.052 \cdot S_{FUS}^{1.086} \left(n_z W_0\right)^{0.177} l_{HT}^{-0.051} \left(\frac{l_{FS}}{d_{FS}}\right)^{-0.072} \cdot q^{0.241} + 11.9 \left(V_P \Delta P\right)^{0.271} \tag{2.33}$$

Torenbeek : No expression given for GA aircraft

USAF :

$$W_{\mathrm{FUS}} = 200 \left[\left(\frac{n_z W_0}{10^5}\right)^{0.286} \left(\frac{l_F}{10}\right)^{0.857} \left(\frac{w_F + d_F}{10}\right) \left(\frac{V_H}{100}\right)^{0.338}\right]^{1.1} \tag{2.34}$$

where :

W_{FUS} = Predicted fuselage weight in $\mathrm{lb_f}$

S_{FUS} = Fuselage wetted area in ft^2

w_F = Fuselage max width in ft

d_F = Fuselage max depth in ft

d_{FS} = Depth of fuselage structure in ft

V_P = Volume of pressurized cabin section in ft^3

l_F = Fuselage length in ft

l_{FS} = Length of fuselage structure (forward bulkhead to aft frame) in ft

R_{max} = Fuselage maximum perimeter in ft

N_{OCC} = Number of occupants (crew and passengers)

ΔP = Cabin pressure differential, in psi (typically 8 psi)

The Cessna equations should only be used for aircraft of the Cessna type-class, which are tiny, have very poor performance, and have maximum speeds of less than 200 knots. In the case of aircraft with high wings, pressurized fuselages were not included into the calculation. In the context of this equation, the number of crew members is included in the total number of passengers. And the USAF equation may be used for light and utility type aircraft with performance of up to roughly 300 knots. Raymer's equation provides the most conservative estimate of the overall value

Variables	Cessna	Raymer	Torenbeek	USAF
WFUS	✓	✓	✓	✓
SFUS	X	✓	X	X
wf	X	X	X	✓
df	X	X	X	✓
dfs	X	✓	X	X
vp	X	✓	X	X
lf	X	X	X	✓
lfs	✓	✓	X	X
Rmax	✓	X	X	X
Nocc	✓	X	X	X
Δ_p	X	✓	X	X
nz	X	✓	X	✓
Wo	✓	✓	X	✓
VH	X	X	X	✓

TABLE 2.4: Fuselage Variables.

4. **Landing Gear**

The landing gear is the part of the plane that holds it up when it's landing, stopping, or moving around on the ground. Shock struts in the landing gear take the impact shock and move around. Typically landing gear is attached to the plane's frame by a gear-retraction mechanism, which lets the gear lengthen and retract. Either a nose wheel or a tail wheel is part of the landing gear. Landing gear that has a nose wheel is generally set up to steer with the nose wheel. Nose-wheel planes have a tail skid or bumper at the back of the body to protect it.

The weight of the main landing gear is estimated using the following equations.

Cessna :

$$
\begin{aligned}
W_{MNLG} =& 6.2 + 0.0143W_0 \\
&+ 0.362W_l^{0.417}n_l^{0.950}L_m^{0.183} \\
&+ 0.007157W_l^{0.749}n_zL_n^{0.788} \\
W_{MNLG} =& 6.2 + 0.0283W_0 \\
&+ 0.362W_l^{0.417}n_l^{0.950}L_m^{0.183} \\
&+ 0.007157W_l^{0.749}n_zL_n^{0.788}
\end{aligned}
\tag{2.35}
$$

Raymer :

$$W_{MLG} = 0.095\,(n_lW_l)^{0.768}\,L_m^{0.409} \tag{2.36}$$

Torenbeek :

$$
\begin{aligned}
W_{LG} &= A + BW_0^{0.75} + CW_0 + DW_0^{1.5} \quad \text{(Low wing)} \\
W_{LG} &= 1.08\left(A + BW_0^{0.75} + CW_0 + DW_0^{1.5}\right) \quad \text{(High wing)}
\end{aligned}
\tag{2.37}
$$

USAF :

$$W_{MNLG} = 0.054\,(n_lW_l)^{0.684}\,L_m^{0.501} \tag{2.38}$$

Where :

W_{MLG} = Predicted weight of the main landing gear inlb$_f$

W_{MNLG} = Predicted weight of the entire landing gear inlb$_f$

W_{LG} = Predicted weight of a specific landing gear (main, nose, or tail) inlb$_f$

n_l = Ultimate landing load factor (typical range 3.5-5.5)

W_l = Design landing weight inlb$_f$

L_m = Length of the main landing gear shock strut inft

The weight of the Nose landing gear is estimated using the following equations.

Cessna :

$$W_{NLG} = 0 \text{ (Included in } W_{MNLG}\text{)} \tag{2.39}$$

Raymer :

$$W_{NLG} = 0.125\,(n_lW_l)^{0.566}\,L_n^{0.845} \tag{2.40}$$

Torenbeek : See the equation 2.37

USAF :

$$W_{NLG} = 0 \text{ (Included in } W_{MNLG}) \tag{2.41}$$

Where :

n_l = Ultimate landing load factor

W_l = Design landing weight in lb_f

W_{NLG} = Predicted weight of the nose landing gear in lb_f

L_n = Length of the nose landing gear strut in ft

Only aircraft with maximum speeds lower than 200 knots should be modeled using the Cessna equations. These equations are designed for use with aircraft of a tiny, rather low performance type. The equation used by the United States Air Force is applicable to light and utility type aircraft with performance up to around 300 knots. The torenbeek's equation was used to compute the weight of each separate landing gear.

Variables	Cessna	Raymer	Torenbeek	USAF
WMLG	X	✓	X	X
WMNLG	✓	X	X	✓
WLG	X	X	✓	X
nl	✓	✓	X	✓
Wl	✓	✓	X	✓
Lm	✓	✓	X	✓
Wo	✓	X	✓	X
nz	X	X	X	✓
Ln	✓	X	X	X

TABLE 2.5: Main Landing Gear Variables.

Variables	Cessna	Raymer	Torenbeek	USAF
nl	✓	X	X	✓
Wl	✓	✓	X	✓
Ln	X	✓	X	X
Wo	✓	X	✓	X
nz	✓	X	X	X

TABLE 2.6: Nose Landing Gear Variables.

5. **Nacelle/Cowling Weight**

Nacelles, also referred to as pods, are aerodynamically designed structures that serve as housings for the engines of multi-engine aircraft. The objects in question exhibit a circular or spherical morphology and are typically situated in a superior, inferior, or anterior position relative to the wing on aircraft with multiple engines. In the event that an aircraft possesses a solitary engine, conventionally, it is situated at the anterior section of the fuselage, whereby the nacelle serves as the aerodynamically refined extension of the fuselage. The term "cowling" generally pertains to the removable casing of specific regions that require frequent accessibility, such as engine compartments, accessory segments, and engine mount or firewall regions.

Cessna :

$$W_{NAC} = 0.37 P_{\max} N_{ENG} \quad \text{(Radial piston engine)} \tag{2.42}$$

$$W_{NAC} = 0.24 P_{\max} N_{ENG} \quad \text{(HOP engine)} \tag{2.43}$$

Raymer: Included in equation 2.51

Torenbeek :

$$W_{NAC} = 2.5 \sqrt{P_{\max}} \text{ (Single-engine tractor propeller)} \tag{2.44}$$

$$W_{NAC} = 0.32 P_{\max} N_{ENG} \quad \text{(Multi-engine HOP)} \tag{2.45}$$

$$W_{NAC} = 0.045 P_{\max}^{1.25} N_{ENG} \quad \text{(Multi-engine radial piston)} \tag{2.46}$$

$$W_{NAC} = 0.14 P_{\max} N_{ENG} \text{ (Multi-engine turboprop)} \tag{2.47}$$

$$W_{NAC} = 0.055 T_{\max} \quad \text{(Podded turbojet or-fan)} \tag{2.48}$$

$$W_{NAC} = 0.065 T_{\max} \quad \text{(HBPR turbofan on a pylon)} \tag{2.49}$$

USAF : Included in equation 2.53

Where :

W_{NAC} = Predicted weight of all engine nacelles in lb_f
N_{ENG} = Number of engines
$P_{\max}$ = Maximum rated power per engine in BHP or ESHP

Variables	Cessna	Raymer	Torenbeek	USAF
WNAC	✓	✓	✓	✓
NENG	✓	✓	✓	✓
Pmax	✓	✓	✓	X

TABLE 2.7: Nacelle/Cowling Weight Variables.

The maximum rated power per engine in USAF equation in nacelle equation weight estimation included in installed engine weight estimation equation. While other calculations include The maximum rated power per engine into the formula

6. **Engine**

The aero engine, commonly known as the aircraft engine, serves as the propulsive element of an aircraft's propulsion mechanism. The majority of aircraft propulsion systems can be classified as either reciprocating piston engines or gas turbines, with a limited number of instances of rocket-powered engines. In contemporary times, electric motors have been utilized in numerous small unmanned aerial vehicles. An aircraft is equipped with a minimum of one and a maximum of eight engines that generate the necessary thrust for flight. Numerous aircraft makes and models exist presently. However, they all share a fundamental purpose of utilizing the air in front of the aircraft, increasing its velocity, and expelling it behind the aircraft.

Cessna :

$$W_{EI} = (1.3P_{\max} + W_{PROP})\,N_{ENG} + W_{NAC} \tag{2.50}$$

Raymer :

$$W_{EI} = 2.575W_{ENG}^{0.922}N_{ENG} \tag{2.51}$$

Torenbeek :

$$W_{EI} = (W_{ENG} + W_{PROP})\,N_{ENG} + 1.03N_{ENG}^{0.3}P_{\max}^{0.7} + W_{NAC} \tag{2.52}$$

USAF :

$$W_{EI} = 2.575W_{ENG}^{0.922}N_{ENG} \tag{2.53}$$

Where :

W_{EI} = Predicted weight of all installed engines in lb_f

W_{ENG} = Weight of each uninstalled engine in lb_f

W_{PROP} = Weight of a single propeller in lb_f

The following expressions can be used to determine dry engine weight.

Piston Engines :

$$W_{ENG} = 50.56 + 1.352P_{\max} \tag{2.54}$$

Turboprop Engines :

$$W_{ENG} = 71.65 + 0.3658 P_{\max} \quad (2.55)$$

Turbofan Engines :

$$W_{ENG} = 295.5 + 0.1683 T_{\max} \quad (2.56)$$

Variables	Cessna	Raymer	Torenbeek	USAF
Wprop	✓	X	X	X
WNAC	✓	X	✓	X
NENG	✓	✓	✓	✓
Pmax	✓	X	✓	X
WENG	X	X	✓	✓

TABLE 2.8: Installed Engine Weight Variables.

Variables	Cessna	Raymer	Torenbeek	USAF
Pmax	✓	✓	✓	✓
Tmax	✓	✓	✓	✓

TABLE 2.9: Uninstalled Engine Weight Variables.

7. **Fuel System**

The fuel system is comprised of the gasoline tanks, fuel lines, fuel pumps, fuel vents, and any other components that are required to transport fuel from the fuel supply to the engine. The fuel system of an aircraft provides the crew with the ability to pump, control, and deliver aviation fuel to the propulsion system and auxiliary power unit (APU) of the aircraft. Fuel systems are highly varied from one another because of the numerous ways in which aircraft can be flown.

Cessna :

$$W_{FS} = 0.40 Q_{\text{tot}} \text{ (Avgas } - \text{ no tip-tanks)} \quad (2.57)$$

$$W_{FS} = 0.4467 Q_{\text{tot}} \text{ (Jet A – no tip-tanks)} \quad (2.58)$$

$$W_{FS} = 0.70 Q_{\text{tot}} \text{ (Avgas – tip-tanks)} \quad (2.59)$$

$$W_{FS} = 0.7817 Q_{\text{tot}} \text{ (Jet A – tip-tanks)} \quad (2.60)$$

Raymer :

$$W_{FS} = 2.49 Q_{\text{tot}}^{0.726} \left(\frac{Q_{\text{tot}}}{Q_{\text{tot}} + Q_{\text{int}}} \right)^{0.363} N_{TANK}^{0.242} N_{ENG}^{0.157} \quad (2.61)$$

Torenbeek :

$$W_{FS} = 2 Q_{\text{tot}}^{0.667} \text{ (Single-engine piston)} \quad (2.62)$$

$$W_{FS} = 4.5 Q_{\text{tot}}^{0.60} \text{ (Multi-engine piston)} \quad (2.63)$$

$$W_{FS} = 1.6 Q_{\text{tot}}^{0.60} \text{ (Multi-engine piston)} \quad (2.64)$$

USAF :

$$W_{FS} = 2.49 \left[Q_{\text{tot}}^{0.6} \left(\frac{Q_{\text{tot}}}{Q_{\text{tot}} + Q_{\text{int}}} \right)^{0.3} N_{TANK}^{0.2} N_{ENG}^{0.13} \right]^{1.21} \quad (2.65)$$

Where :

Q_{tot} = Total fuel quantity in US gallons
Q_{int} = Fuel quantity in integral tanks in US gallons
N_{TANK} = Number of fuel tanks
W_{FS} = Predicted weight of the fuels system in lb_{f}
W_f = Maximum fuel quantity aircraft can carry in lb_{f}

Variables	Cessna	Raymer	Torenbeek	USAF
Qtot	✓	✓	✓	✓
Qint	X	✓	X	✓
NTANK	✓	X	X	✓
WFS	✓	✓	✓	✓

TABLE 2.10: Fuel System Weight Variables.

8. **Flight Control System**

The flight controls (aileron, elevator, rudder, and flaps) are all part of the flight control system. It is made up of wires, pushrods, pulleys, bell-cranks, cockpit controls, and any structural supports that are needed. Primary and secondary flight controls are two categories of flight control systems. The principal flying controls of an aircraft are comprised of the ailerons, elevators (or stabilators, depending on the installation), and rudder, and they are the only means by which an aircraft may be piloted in a safe manner. Secondary flight controls include things like flight spoilers and trim systems, high-lift devices like slats and flaps, and high-lift devices like slats and flaps. Their major purpose is to either increase the performance characteristics of the aircraft or reduce the harsh control loads experienced by the aircraft. .

Cessna :

$$W_{CTRL} = 0.0168 W_0 \quad \text{(Manual control system)} \tag{2.66}$$

The equation above only for W0 less than or same than 8000 lbf

Raymer :

$$W_{CTRL} = 0.053 l_{FS}^{1.536} b_W^{0.371} \left(n_z W_0 \times 10^{-4}\right)^{0.80} \tag{2.67}$$

Torenbeek :

$$W_{CTRL} = 0.23 W_0^{0.667} \quad \text{(Manual single control system)} \tag{2.68}$$

$$W_{\mathrm{CTRL}} = 0.44 W_0^{0.667} \text{ (Manual transport aircraft)} \tag{2.69}$$

$$W_{CTRL} = 0.64W_0^{0.667} \text{ (Powered transport aircraft)} \tag{2.70}$$

USAF :

$$W_{CTRL} = 1.066W_0^{0.626} \text{(Manual control system)} \tag{2.71}$$

$$W_{CTRL} = 1.08W_0^{0.7} \quad \text{(Powered control system)} \tag{2.72}$$

Where :

W_{CTRL} = Predicted weight of the flight control system in lb_f,
b_W = W Wingspan in ft

Variables	Cessna	Raymer	Torenbeek	USAF
bW	X	✓	X	X
nz	X	✓	X	X
lFS	X	✓	X	X
Wo	✓	✓	✓	✓

TABLE 2.11: Flight Control System Variables.

9. **Hydraulic System**

Hydraulics is a way to send power through pipes and control devices using liquid as the working medium. For some tasks, hydraulic systems are better than mechanical or electrical ones because it is easy to apply force, the force can be increased as needed, it is easy to route the pipes, and there is no backlash between the parts. When it comes to smaller aircraft, the hydraulic system is typically just used for the brakes, the retractable landing gear, and occasionally the flaps. Hydraulic boost is also used for flying controls, spoilers, and thrust reversers in bigger aircraft. The weight of the hydraulic systems that are utilized for the flight controls is typically included in the weight of the Flight Control System. This is the case in the majority of instances.

Since the weight of the hydraulic systems used for the flight controls are frequently included in the weight of the flight control system, the following formula will be used for the other components.

All :

$$W_{HYD} = 0.001W_0 \tag{2.73}$$

Where :

W_{HYD} = Predicted weight of the hydraulics system in lb_f.

Variables	Cessna	Raymer	Torenbeek	USAF
WHYD	✓	✓	✓	✓
Wo	✓	✓	✓	✓

TABLE 2.12: Hydraulic System Variables.

10. **Avionics System**

Avionics refers to the many electronic systems that are installed aboard aircraft. Avionic systems include communications, navigation, the display and management of many systems, as well as the hundreds of systems that are added to aircraft to perform distinct activities. These are all considered to have avionic functions.

All :

$$W_{AV} = 2.11W_{UAV}^{0.933} \tag{2.74}$$

Where :

W_{AV} = Predicted weight of the avionics installation in lb_f

W_{LAS} = Weight of the uninstalled avionics in lb_f.

Variables	Cessna	Raymer	Torenbeek	USAF
WUav	✓	✓	✓	✓

TABLE 2.13: Avionics System Variables.

11. **Electrical System**

Every component of a modern aircraft, from the lights and avionics to the auxiliary fuel pump and engine starter motor, must be powered by the electrical system in order for it to operate properly. This system is an absolute necessity for the operation of a modern aircraft. When it comes to powering the electrical systems of an aircraft, there might be various different types of power sources present. These power sources consist of generators that produce Alternating Current (AC), which are driven by an engine; Auxiliary Power Units (APUs); and external power [5]. Flight instruments, life-support systems like de-icing, and passenger services like cabin lighting are all run by the aircraft's electrical power system. This system also provides electricity for the aircraft's onboard entertainment system.

Cessna :

$$W_{EL} = 0.0268W_0 \tag{2.75}$$

Raymer/USAF :

$$W_{EL} = 12.57\left(W_{FS} + W_{AV}\right)^{0.51} \tag{2.76}$$

Torenbeek :

$$W_{EL} = 0.0078\left(W_0 - W_u\right)^{1.2} - W_{HYD} \tag{2.77}$$

Where :

W_{EL} = Predicted weight of the electronics system in l_f,

W_u = Target useful load in l_f.

Variables	Cessna	Raymer	Torenbeek	USAF
Wo	✓	X	✓	X
WFS	X	✓	X	X
WAV	X	✓	X	X
Wu	X	X	✓	X
WHYD	X	X	X	✓

TABLE 2.14: Electrical System Variables.

12. **Air Conditioning, Pressurization, and Anti Icing**

The primary functions of an aircraft's air conditioning system include air supply, heating, cooling, temperature control, and temperature distribution. This system's goal is to keep the flight crew, passengers, and either compartment at the desired temperature at all times. The humidity control section may also be included as a component of the air conditioning system in certain aircraft. In pressurized aircraft, the systems for air conditioning and pressure installation are inextricably intertwined. The controlled discharge of pressured and conditioned air is what maintains the cabin altitude at the chosen setting. As a form of anti-icing protection, anti-icing solutions might take the form of pneumatic inflatable boots or bleed air heated elements.

All :

$$W_{\mathrm{AC}} = 0.265 W_0^{0.52} N_{OCC}^{0.68} W_{AV}^{0.17} M^{0.08} \quad (2.78)$$

Where :

W_{AC} = Predicted weight of the A C and anti icing installation in lb_f

N_{OCC} = Number of occupants (crew and passengers)

M = Mach Number

Variables	Cessna	Raymer	Torenbeek	USAF
Wo	✓	✓	✓	✓
M	✓	✓	✓	✓
WAV	✓	✓	✓	✓
Nocc	✓	✓	✓	✓

TABLE 2.15: Air Conditioning System Variables.

13. **Furnishings**

In a civil aircraft system, there are many different subsystems that serve as furnishings and equipment. Some of these subsystems include the pilot seat, the observer seat, the cabin attendant seat, the galley, the lavatory, the passenger seat, the overhead bin, the cockpit lining, the cabin lining, the cargo lining, the emergency equipment, the passenger service unit, and the plaquecards and markings. Insulation, soundproofing, lighting, galley, emergency equipment, and associated electric systems are also included in this.

Cessna :

$$W_{\text{FURN}} = 0.0412 N_{\text{OCC}}^{1.145} W_0^{0.499} \tag{2.79}$$

Raymer :

$$W_{FURN} = 0.0582 W_0 - 65 \tag{2.80}$$

USAF :

$$W_{FURN} = 34.5 N_{\text{CREW}}\, q_H^{0.25} \tag{2.81}$$

W :

W_{FURN} = Predicted weight of furnishings in lb_f

N_{CREW} = Number of crew,

q_H = Dynamic pressure at max level airspeed, $\text{lb}_\text{f}/\text{ft}^2$

Variables	Cessna	Raymer	Torenbeek	USAF
Wo	✓	✓	X	X
Nocc	✓	X	X	X
Ncrew	X	X	X	✓
qH	X	X	X	✓

TABLE 2.16: Furnishings Variables.

2.8 Trend of Global GDP and Traffic From 2000 to 2040 in Aviation Industry

In the early stages of each new administration, when policymakers are still formulating national security strategy and attempting to navigate an uncertain future, Global Trends is intended to offer them an analytical framework. It is not the purpose of this project to provide an accurate forecast of the world in the year 2040; rather, it is intended to assist decision-makers and citizens in gaining a better understanding of what may lie beyond the horizon and in preparing for a variety of potential futures [6]. The aviation industry as a whole has faced a number of obstacles over the course of the previous two years, the Covid-19 pandemic era. When borders began to close and supply lines came to a grinding halt, turboprops rose up to the challenge on a global scale.

As a result of their delivery of vaccines and other necessities to local communities all over the world, they became a vital lifeline for those populations. After a delay of two years, aircraft have finally begun to fly again. As we go back toward the "new normal," the aviation sector is shifting its attention back to the ongoing difficulties it faces. The reduction of carbon emissions in the aviation industry is at the top of the priority list for the industry as a whole, while the turboprop sector is anticipated to be the source of innovation in the near future. In a similar vein, turboprop aircraft will prove to be an important entry point for innovative technologies that will assist the aviation sector move forward.

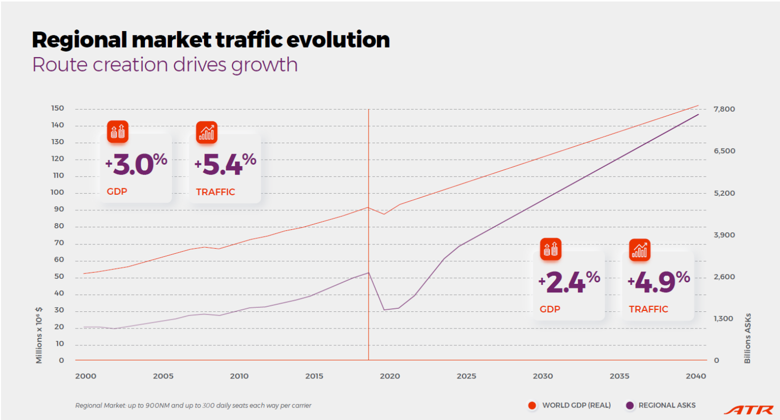


Figure 2.5: Market Traffic evolution

Figure 2.6: Aircraft demand for new route

2.9 Regional Aircraft Market

There are two sizes of regional aircraft: large regional aircraft, which can carry 70–90 passengers, and small regional aircraft, which can take 30–50 passengers. Both sizes are referred to together as regional aircraft. Big companies like Boeing and Airbus have a natural advantage in the market for larger airplanes because of their size and resources. They have achieved this position of dominance in both the narrow-body and wide-body markets thanks to their internal innovation and a string of mergers with other manufacturers. Russia and China are beginning to field new competitors, but it will be quite some time before this makes a difference in the overall market.

2.9.1 Turboprop Aircraft

One or more gas turbine engines that are coupled to a gearbox are utilized by turboprop aircraft in order to provide propulsion for the aircraft while it is both on the ground and in the air. The propeller or propellers are then turned by the gearbox. When air is sucked into the intake of a turboprop engine, it is immediately subjected to compression by the compressor [7]. Combustion of the mixture results from adding fuel to the compressed air in the combustor. The turbine is turned by the hot combustion gasses, which then powers a shaft that rotates the propeller. In a turboprop engine, the propeller receives practically the entire power generated by the turbine. Smaller aircraft that travel at subsonic speeds, such as charter jets and transport aircraft, are the most typical users of turboprop engines. The average cruise speed for one of these ships is approximately 300 knots. However, certain propjet planes can speed over 400 knots. Because they consume less fuel, turboprop aircraft have lower operational costs than jets; however, they are also significantly slower than jets. Businesses that need to fly missions that require a travel distance of 600 to 1,000 miles between general aviation airports, which typically have runways that are too short to handle jets, may find turboprops an interesting choice.

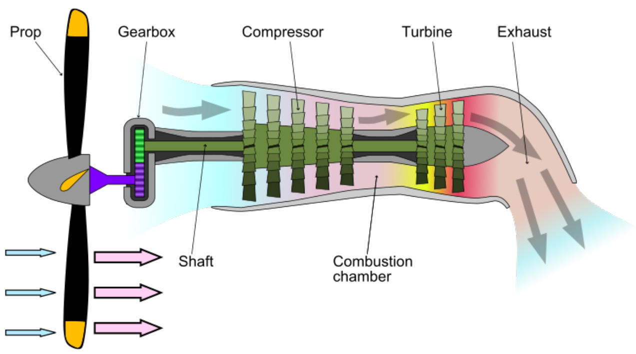


FIGURE 2.7: Chematic diagram of the operation of a turboprop engine

The takeoff performance of turboprop aircraft is superior, and its ability to climb swiftly makes them well suited for use in mountainous regions. The reaction time is far faster than a jet. It can take off and land on runways that are both shorter and rougher than light jets. Many turboprops can take off and land successfully on a runway that is only 3,200 feet long, whereas jets typically require a runway that is at least 5,000 feet long. The use of grass airfields and unmade airstrips is also possible with turboprop aircraft. It has a higher power output relative to its weight than a jet, which gives it the ability to achieve optimal fuel efficiency at low altitudes (preferably below 25,000 feet).

Since turboprops travel at a lesser speed and have a shorter range than jets, they are not the optimal choice for journeys that are very time-sensitive or lengthy. The normal altitude ceiling for turboprop aircraft is between 25,000 and 30,000 feet. At this level, a turboprop won't be able to rise to a higher altitude like a jet can in order to avoid experiencing turbulence or adverse weather conditions. A jet, on the other hand, will be able to do so. It's possible that the journey will be choppy and unpleasant as a result of this. Additionally, very large passenger groups are not the best fit for these vehicles. Even the largest turbo-liners usually only transport between 18 and 30 passengers, while some of them may accommodate as many as 59 people.

FIGURE 2.8: Pilatus PC 12 NG

When it comes to supplying regional connections across short, thin routes, turboprops are the aircraft that should be selected because they are the most efficient option. When compared to a regional jet with a seating capacity that is comparable, the engine technology and customized design point of a turboprop provides a durable competitive advantage. Turboprops are designed specifically for shorter itineraries.

2.9.2 Turboprop Market Demand

We can see that the demand for turboprop aircraft around the globe will reach approximately 2,450 units within the next 20 years. As we emerge from a period that has been difficult for the whole aviation industry, this is an encouraging sign that business will soon resume its normal course. After a delay of two years, the regional aviation industry is getting very close to completing a full recovery. In spite of this, the growth potential that was missed during the pandemic continues to have an impact for the next 20 years. During this time, a number of older aircraft entered retirement, although the average age of the fleet of turboprop aircraft continued to rise. As a consequence of this, aircraft replacement will be a significant driver of demand during the subsequent two decades, accounting for 1,500 brand-new aircraft entering service in 2041. The significance of turboprops as innovative platforms for testing and delivering game-changing technology to the market will become increasingly essential in the coming years.

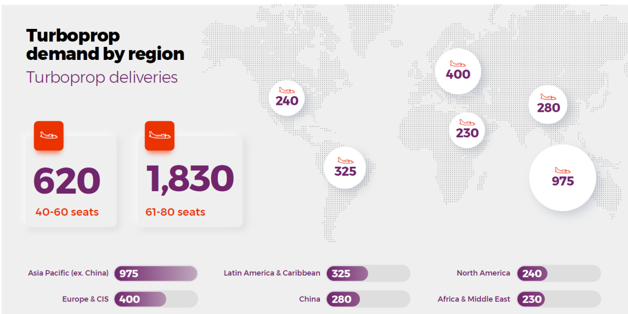


FIGURE 2.9: Turboprop demand by region

The use of turboprop aircraft often spurs on new routes. The use of turboprop aircraft rather than regional jets would result in a significant reduction in CO2 emissions. Even while regional jets produce more carbon dioxide than turboprop aircraft, the environmental impact of short-haul flights is magnified since jet technology is not designed to be used in this market segment. The effects of aviation on the climate are not restricted solely to carbon dioxide emissions.

The presence of contrails has the potential to have the greatest influence on global warming, up to double that of CO2 by itself. This phenomenon is referred to as the ″Non-CO2 effect.″ Because of their lower cruising altitudes, turboprop aircraft have less contribution to developing contrails, lowering the overall impact of aviation's so-called ″Non-CO2 effect.″

2.9.3 Regional Turboprop Aircraft Demand in Asia Pacific

The expansion and development of the business ecosystem as well as the rise in the amount of disposable income that individuals have contributed to the expansion of the general aviation industry in the Asia-Pacific region. In the past ten years, there has been a significant increase in the use of turboprops in South Asia and South-East Asia. They have been the driving force behind the decrease in the cost of air travel as well as the establishment of a new network with the same scale as the one that was in place 20 years ago.

A significant number of the passengers who are currently filling up airplanes had, in the past, endured lengthy and uncomfortable journeys by traveling by boat, train, or road. Demand for regional air travel will continue to rise as an alternative to time-consuming mobility options as economies continue to develop and the middle classes grow in size. In the next 20 years, it is anticipated that the demand for turboprop aircraft will be highest in the Asia-Pacific area.

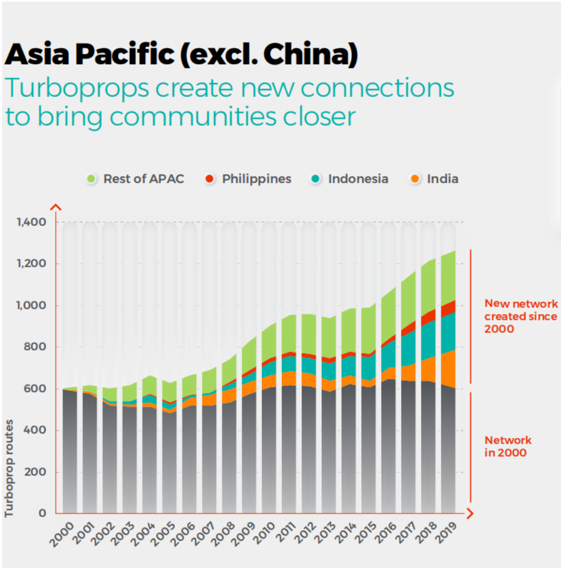


FIGURE 2.10: Demand in Asia Pacific

As the recovery from the Coronavirus epidemic continues, regional aircraft are playing an essential role in linking airports on routes with low demand or in managing problematic airfields. In particular, turboprops are known for having exceptional performance when landing on short runways and traveling across rough terrain. They are the sole aircraft that allow 34 percent of the airports that have scheduled services to maintain their connections to the rest of the globe. The use of turboprop aircraft by airline operators has been a significant factor in the expansion of their networks.

They provide a variety of aircraft that is extraordinarily efficient in terms of both effectiveness and cost, making it possible to open up chances for expansion. Over the course of the previous decade, an annual average of 180 new routes have been introduced into service. Even during the height of the worldwide Covid outbreak, turboprops continued to play a key role in establishing new routes.

Aircraft replacement is the greatest driver for deliveries The establishment of new transportation corridors, which boosts overall regional mobility, is the second most important component. In order to satisfy the growing demand for reduced emission air travel and regional connectivity, the world's top maker of regional aircraft, ATR, forecasts that there will be a requirement for at least

2,450 turboprop aircraft over the next 20 years [6]. According to ATR, the primary driver of such demand would be airlines wanting to replace their existing aircraft fleets. The demand for airplanes that are lighter and more efficient in their use of fuel is growing as shorter flights become more common.

2.10 MTOM of Regional Aircraft

The maximum takeoff mass, or MTOM, of an aircraft is a value that is established by the aircraft manufacturer. This amount is also commonly referred to as the maximum takeoff weight, or MTOW. It is the heaviest load that the airplane might possibly take off with without violating any of its structural or other limitations [8]. The maximum take-off weight is often indicated in either kilograms or pounds. The mass is a constant value that does not change regardless of the temperature, altitude, or amount of runway that is available.

MTOM is a significant parameter that is set by the aircraft manufacturer and is approved by regulatory authorities like the FAA in the United States or the EASA in Europe. The FAA in the United States and the EASA in Europe are two examples of regulatory agencies. During the design and development phases of the aircraft, it is determined by comprehensive testing and analysis, taking into account a variety of aspects such as aerodynamics, structural integrity, engines, systems, and performance requirements.

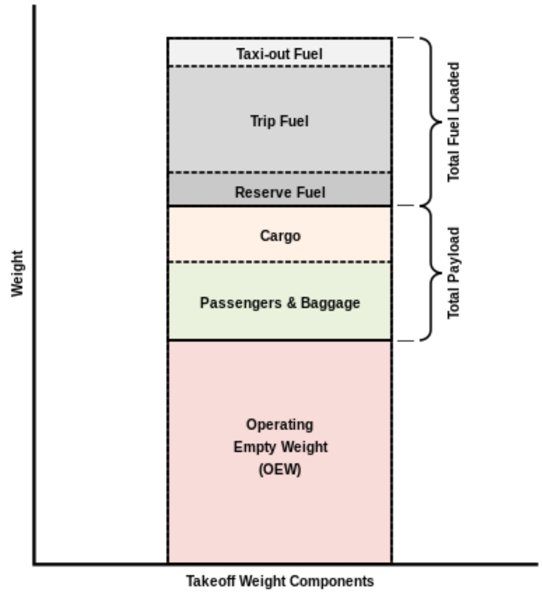


FIGURE 2.11: Takeoff Weight Components

2.10.1 Weight of An Aircraft

When we speak of an aircraft's "empty weight," we are referring to the whole mass of the airframe, as well as the mass of the engines, propellers, rotors, and any other permanent equipment. When determining the weight of the vehicle when it is empty, the weight of the crew and the cargo are subtracted from the total, but the weight of all fixed ballast, unusable fuel supply, undrainable oil, total quantity of engine coolant, and total quantity of hydraulic fluid are included. The term "gross weight" refers to the heaviest load that an airplane can take off with, which includes all of the consumables such as gasoline, oil, and other supplies. The maximum takeoff weight of an aircraft refers to the maximum weight at which it is capable of taking off under normal conditions. The maximum takeoff weight is typically higher than the gross weight, which in turn is higher than the empty weight. The empty weight is typically the lowest of the three.

The MTOM is also an important factor to consider when analyzing the capabilities of an aircraft's performance. The takeoff distance, rate of climb, payload capacity, and range of an aircraft are all impacted by this factor [9]. When the maximum takeoff weight (MTOM) of an aircraft is increased, the takeoff roll, rate of climb, and payload capacity all suffer. On the other hand, having a reduced MTOM enables the aircraft to take off from a shorter distance, have improved climbing capability, and have enhanced payload capacity.

2.10.2 The Regional Turboprop Aircraft Benchmark

The list of regional turboprop aircraft that have substantial build numbers can be found further down. Regional aircraft typically have less than one hundred seats available for passengers, serve as the short-hop component of the hub-and-spoke model of passenger and cargo distribution, and participate in point-to-point transit while traveling a maximum of 810 miles [10].

Aircraft	MTOW (kg)	EMTOW (kg)	Engine Type	Number of Engine	Mengine (kg)	EMTOW-M Engine	ln (MTOW)	ln (EMTOW)	ln (EMTOW-Mengine)\	max V (m/s)	T/W Take Off	Wing Area (m^2)	W/S (N/m^2)
Saab 340	13154	5798	GE CT7-9B	2	244	5310	9.484481174	8.665268309	8.577347114	194.44	3.66	41.81	3086.36
Antonov An-140	19150	12810	PW127A	2	480	11850	9.860057995	9.457981395	9.380083147	150	1.14	51	3683.56
IAMI (HESA) IR.AN-140 FARAZ	19150	11800	Klimov TV3-117VMA-SBM1	2	294	0						51	3683.56
ATR 42	16900	10285	PW120	2	418	9449	9.735068901	9.238441802	9.153664195	147.22	0.9	54.5	3042.00
Saab 2000	22800	13800	RR AE2100P	2	790	12220	10.03451581	9.532423871	9.410829233	185.28	1.12	55.7	4015.58
CASA/IPTN CN-235	15100	9800	GE CT7-9B	2	244	9312	9.622450023	9.190137665	9.13905917	158.45	1.39	59.1	2506.45
De Havilland Canada Dash 8	19505	9424.2	PW123	2	450	8524.2	9.878426122	9.151036128	9.050664456	89.44	1.64	64	2989.75
Fokker 50	20820	13400	PW125B	2	418	12564	9.943669342	9.503009986	9.438590861	96.67	1.57	70	2917.77
Fokker F27 Friendship	19773	11204	RR Dart Mk.5327	2	547	10110	9.89207265	9.324026136	9.221280312	87.5	1.19	70	2771.04
Antonov An-24	21000	13300	Ivchenkp AI-24A	2	600	12100	9.952277717	9.495519314	9.400960732	117.5	0.98	74.98	2747.53
Xian Y-7 / MA60 / MA600	21800	12603	Dongan WJ-5A	2		12603	9.989665249	9.44169016	9.44169016	155.56	1.72	75.26	2841.59
Hawker Siddeley HS 748	21092	12304	RR Rda Dart Mk536-2	2	547	11210	9.956649101	9.417679692	9.324561516	106.11	1.4	77	2687.18
BAe ATP	22930	13959	PW 126	2	480	12999	10.04020138	9.543879741	9.47262771	99.72	1.68	78.3	2872.84
de Havilland Canada Dash 7	19958	12560	PW PT6A-50	4	218	11688	9.901385344	9.43827244	9.366317953	274.44	2.55	80	2447.35

TABLE 2.17: List of Regional Turboprop Aircraft

1. **CASA/IPTN CN-235**

   The CASA/IPTN CN-235 was co-developed by CASA of Spain and IPTN of Indonesia to be a medium-range twin-engine transport aircraft [11]. The military uses it for transport and surveillance missions, in addition to its other uses as a regional airliner. Several Honeywell improvements are available for this medium-range twin-engine aircraft, which is frequently employed by the military for transport and reconnaissance.

FIGURE 2.12: CASA/IPTN CN-235

2. **Antonov An-140**

   Following in the footsteps of its predecessor, the Antonov An-24, the Antonov An-140 is a turboprop-powered, small aircraft that was developed by the Antonov ASTC department in Ukraine. It has an increased cargo capacity and the ability to use unprepared airstrips [12]. Since its first flight on September 17, 1997, the An-140 has been built at the main production line in Kharkiv by KHDABP, in Samara by Aviakor, and in Iran under license by Iran Aircraft Manufacturing Industrial Company (HESA) as the IrAn-140. The IrAn-140 is also known as the "An-140." In addition, the problem of assembly in Kazakhstan has been brought up in trilateral discussions between the governments of Kazakhstan, Ukraine, and Russia.

FIGURE 2.13: Antonov An-140

3. **De Havilland Canada Dash 8**

The De Havilland Canada DHC-8, most frequently referred to as the Dash 8, is a series of turboprop-powered small airliners that were initially introduced by de Havilland Canada (DHC) in the year 1984. Later on, in 1988, Boeing purchased DHC, and then in 1992, Bombardier did the same thing. Finally, in 2019, Longview Aviation Capital purchased DHC, bringing back the De Havilland Canada brand [13]. It was created from the Dash 7 and is powered by two Pratt and Whitney Canada PW100 engines. It has enhanced cruise performance and reduced operational expenses than the Dash 7, but it does not have short takeoff and landing performance.

FIGURE 2.14: De Havilland Canada Dash 8

4. **ATR 42**

Final assembly of the regional airliner known as the ATR 42 takes place in the French city of Toulouse, which is the home base for the Franco-Italian manufacturer ATR. Aérospatiale, which is now known as Airbus, and Aeritalia, which is now known as Leonardo S.p.A [14]. introduced the aircraft on November 4, 1981 under the brand name ATR as part of a joint venture. On August 16, 1984, the ATR 42-300 completed its first flight, and the aircraft received its type certification in September 1985. Launch client Air Littoral carried out its first flight that generated income in the month of December that same year.

FIGURE 2.15: ATR 42

5. **MA600/MA60/Xian Y-7**

   The Xian Aircraft Industry Corporation, which is a subsidiary of the Aviation Industry Corporation of China (AVIC), is responsible for manufacturing the Xian MA600, which is an upgraded version of the Xian MA60. On June 29, 2008, Xi'an Aircraft Industry Corporation successfully completed the rollout of its first MA600 turboprop. On October 10, 2008, the MA600 aircraft completed its first flight [15]. When compared to the MA60, this aircraft features upgraded avionics, a more comfortable passenger cabin, and engines that produce a greater amount of thrust.

FIGURE 2.16: MA600

6. **Saab 340**

   The Saab 340 is a Swedish twin-engine turboprop aircraft that was initially constructed by Saab AB and Fairchild Aircraft [16]. The Saab 340 was designed in Sweden. It is designed to accommodate 30-36 passengers, and as of July 2018, there were 240 aircraft in operation that were operated by 34 distinct operators.

FIGURE 2.17: Saab 340

7. **De Havilland Canada Dash 7**

Turboprop-powered and equipped with short take-off and landing (STOL) capabilities, the de Havilland Canada DHC-7, most commonly referred to as the Dash 7, is a regional aircraft manufactured by de Havilland Canada [17]. It made its first flight in 1975 and continued to be manufactured until 1988, despite the fact that its parent company, de Havilland Canada, was purchased by Boeing in 1986 and later sold to Bombardier. 1975 was the year when the airplane took its maiden voyage. In 2006, Bombardier made the transfer of ownership of the type certificate for the aircraft design to the Victoria-based manufacturer Viking Air. Viking Air is known for producing aircraft.

FIGURE 2.18: De Havilland Canada Dash 7

8. **Fokker 50**

   The Fokker F50 is a turboprop-powered airliner that was developed as an enhanced version of the Fokker F27 Friendship, which was a very successful aircraft for Fokker. The Fokker 50 was later developed into the longer and more capable freighter known as the Fokker 60 [18]. Fokker, a Dutch aircraft manufacturer, was responsible for the production and maintenance of both aircraft. The first flight of a Fokker 50 took place on December 28, 1985, and the aircraft began carrying paying passengers in 1987. Former aircraft of the Royal Netherlands Air Force (RNLAF) are currently used by the Peruvian Naval Aviation and the Air Force of the Republic of China. The Fokker 60 has been used by the Royal Netherlands Air Force (RNLAF).

FIGURE 2.19: Fokker 50

9. **BAe ATP**

British Aerospace is the company that is responsible for designing and manufacturing the airliner known as the Advanced Turbo-Prop (ATP). The Hawker Siddeley HS 748 was a moderately successful feederliner in the 1960s, and this aircraft was a development of that design [19]. Business strategists at British Aerospace made the assumption that there was a market for a short-range, low-noise, fuel-efficient turboprop aircraft as a result of events such as the oil crisis in 1979 and rising public sensitivity surrounding aircraft noise. Because of this belief, the Association of Tennis Professionals (ATP) was founded in the 1980s. On August 6, 1986, it successfully completed its first flight.

FIGURE 2.20: BAe ATP

10. **Antonov An-24**

The Antonov An-24 (Russian/Ukrainian: Antonov An-24) (NATO reporting name: Coke) is a twin turboprop transport/passenger aircraft with 44 seats that was created in 1957 in the Soviet Union by the Antonov Design Bureau. It was constructed by the Kyiv, Irkutsk, and Ulan-Ude Aviation Factories [20]. The Antonov An-24 was given the NATO reporting name of Coke. There are currently 109 An-24s still in use across the globe, the most of which are located in the CIS and Africa. The first flight of the An-24 took place in 1959, and the aircraft went on to be constructed in a total of approximately 1,000 units in a variety of configurations.

FIGURE 2.21: Antonov An-24

11. **Hawker Siddeley HS 748**

    Avro, a British aircraft manufacturer, was the company that came up with the original design for the medium-sized turboprop airliner that would later be constructed by Hawker Siddeley. It was the final aircraft designed by Avro before the company was acquired by Hawker Siddeley and merged into that company [21]. In an effort to realign the company's focus on the growing civil and international markets in the late 1950s, the development of the HS 748 was undertaken. It was primarily built as a contemporary feederliner to serve as a replacement for the outdated Douglas DC-3s that were in widespread use at the time. It was powered by the well-known Rolls-Royce Dart turboprop engine. The HS 748 series 1 made its maiden flight on June 24, 1960, and it entered revenue service in 1961.

FIGURE 2.22: Hawker Siddeley HS 748 Series 2A

12. **Fokker F27 Friendship**

    The Dutch aircraft manufacturer Fokker is responsible for the development and production of the turboprop airliner known as the Fokker F27 Friendship. The F27 was not just one of the most successful European airliners of its day but also the most numerous aircraft produced in the Netherlands after the war [22]. It holds the record for the most post-war aircraft produced in the country. In the early 1950s, the F27 was designed with the goal of developing a viable replacement to the previous piston engine-powered airliners that had become ubiquitous on the market, such as the Douglas DC-3. This was the intention behind the development of the F27.

FIGURE 2.23: Fokker F27 Friendship

13. **Saab 2000**

    Saab, a Swedish aircraft company, is responsible for creating the Saab 2000, which is a twin-engined high-speed turboprop airliner. It can reach a top speed of 665 kilometers per hour (or 413 miles per hour) while carrying between 50 and 58 passengers [23]. The shooting took place at Linkoping, which is located in Sweden. The Saab 2000 made its maiden flight in March of 1992 and was given its certification in the following year. The final aircraft was handed out in April of 1999, bringing the total number of aircraft manufactured to 63. As of October 2022, 27 Saabs 2000 were in service with various airlines and the military.

FIGURE 2.24: Saab 2000

2.11 Turboprop Aircraft Data Analysis

Here some chart data benchmark from 2.17 list of regional turboprop aircraft.

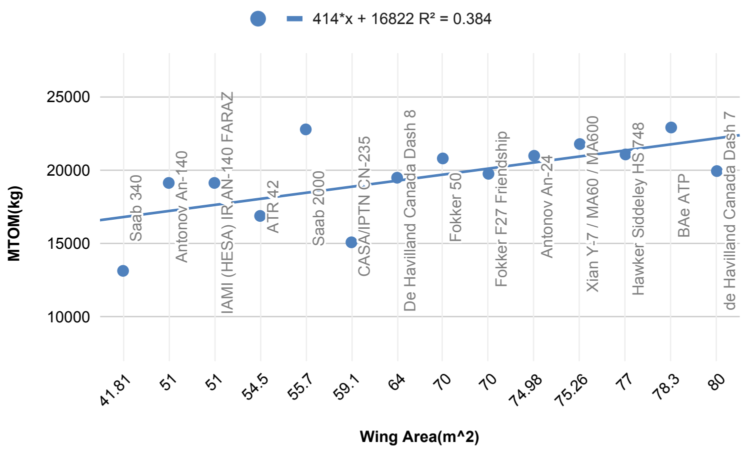


FIGURE 2.25: Wing Area vs MTOM

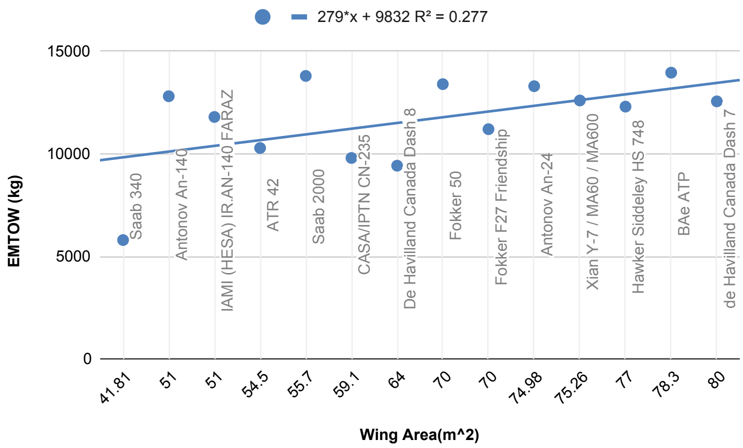


FIGURE 2.26: Wing Area vs EMTOM

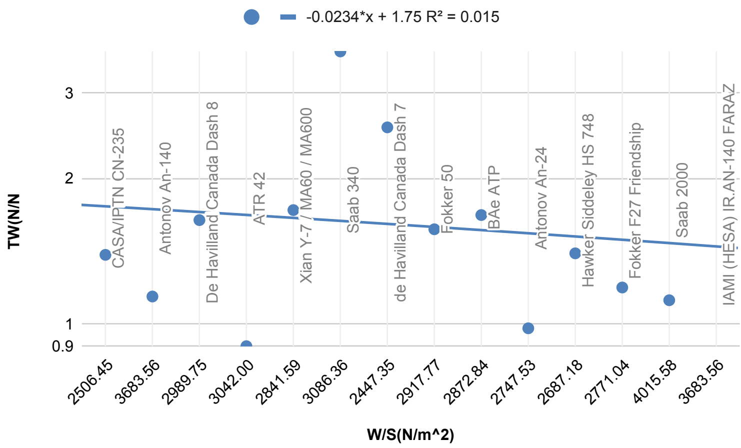


FIGURE 2.27: T/W vs W/S

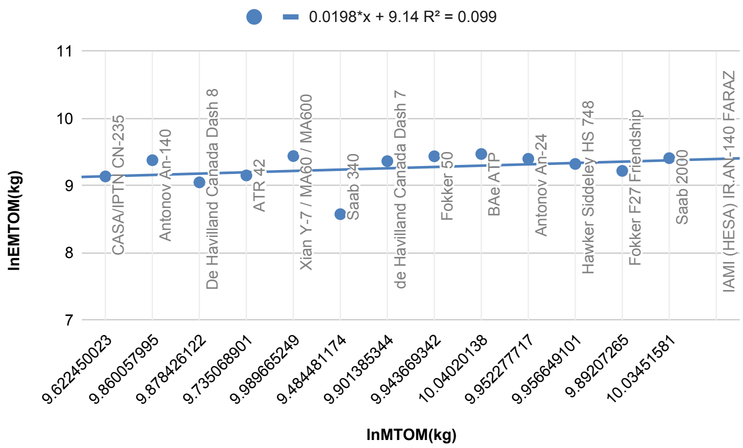


FIGURE 2.29: lnEMTOM vs lnMTOM

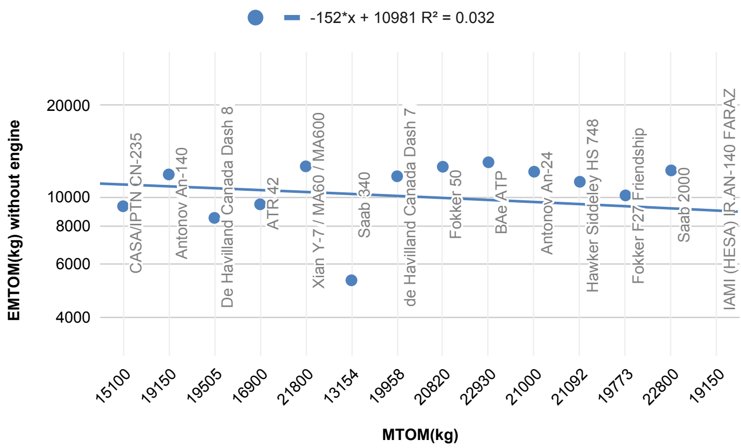


FIGURE 2.28: EMTOM without engine vs MTOM

As can be seen from figure 2.25 it is clear that the MTOM increasing as the value of wing area increases. Looking at the first value of MTOM where it shows the value of MTOM in the y axis. The following values of wing area, where the value increase, shows an increase as well in the MTOM in the y axis. As also proven by the R- squared with the value of 0.384 or almost 40 percent, it can safely be said that MTOM changes, in this case increases, as the value of wing area increases.

It can be seen as well from figure 2.26. The EMTOM increasing as the value of wing area increases as well. it means the increase of area of the wing will also increase the amount of empty weight that can be put on an aircraft. It is seen from the Saab 240 aircraft that it has a relatively small amount of wing area and it can only provide 5,798 Kg amount of EMTOM. If we see the growth of the area, we can see that for each and every aircraft that they have bigger MTOM due to their wing area getting bigger. From the R-squared which show a 0.277 correlation value, thus it is proven that with the increase of wing area, the EMTOM gets bigger as well.

From the 2.27 it can be seen that value of R-squared is relatively small. It means the dependent variable which is Y-axis (T/W), not affected by the X variable. from the 2.28 and 2.29 it can be seen that the value of Y-axis increasing as the value of X-axis. From the R-squared value which show a 0.336 and 0.353 correlation value, that means that the greater the MTOM value, the EMTOM value will also increase simultaneously, and EMTOM is affected by the increasing value of MTOM

CHAPTER 3
RESEARCH METHODOLOGY

3.1 Research Outline

In this part, the author will attempt to define what exactly is meant by the term "research methodology". It is intended to provide an overview of the strategy that the author had utilized in an effort to attempt and find solutions to the issues raised in this thesis.

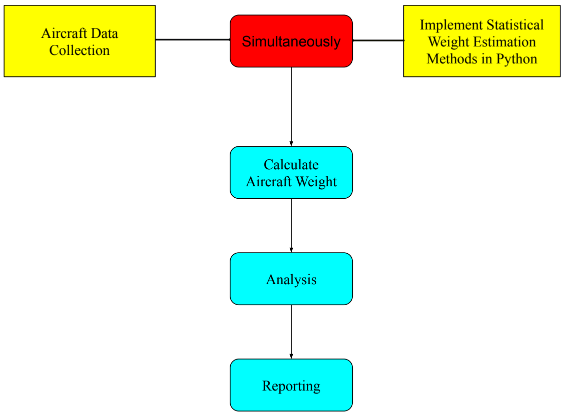


FIGURE 3.1: Thesis Work Flowchart

The above flowchart is a description of the author's thesis work. The author used a five-step process, as depicted in figure 3.1. The first step is compiling information about regional turboprop aircraft. The second is using a computational tools of programming language to realize the statistical estimation weights method. It is possible to perform both the first and second steps simultaneously. Having established the mass of each aircraft, the author moves on to the third stage. The next step is to analyze, compare, and probe the ways in which they differ when the MTOW of an aircraft is changed.

3.2 Stastitical Weight Estimation Methods

Estimating the weight of an airplane can be done using any one of a number of different statistical methods. The estimation of aircraft weight is a more general topic that may be broken down into several more particular ways, such as Cessna method, Raymer's method, Torenbeek's method, and the USAF method. The cessna method, which was developed by cessna aircraft company. The Raymer method, which was invented by Daniel Raymer, is a well-known method for estimating weight that is used in the conceptual design of airplanes. Another approach for estimating weight that is often used is called the Torenbeek method, which was invented by E. Torenbeek. It does this by applying techniques from regression analysis in order to establish correlations between the weight of the aircraft and the various design factors. The United States Air Force method, sometimes known as the USAF method, is a technique for estimating weight that was created by the United States Air Force.

3.2.1 Cessna Method

The Cessna method is a simplified approach to aircraft weight estimation developed by the Cessna Aircraft Company. It is often used for smaller general aviation aircraft. To estimate the weights of various components, the method applies simple mathematical equations and variables based on the aircraft's parameters, such as wing area, span, and engine type. This method is less comprehensive than some others but is suitable for early-stage conceptual design of small aircraft. The Cessna method is often described in various sources related to aircraft design and engineering

3.2.2 Raymer Method

The Raymer method is described in the book *Aircraft Design: A Conceptual Approach* [24]. During the conceptual design phase, the Raymer method was employed to estimate aircraft weights. It breaks down the aircraft into various components and estimates their weights using empirical calculations based on historical data. The technique considers elements such as aircraft size, mission profile, and design characteristics. Raymer's method includes separate equations for fuselage, wing, empennage, landing gear, propulsion systems, and other components.

3.2.3 Torenbeek Method

The Torenbeek method presents an approach to aircraft weight estimation based on historical data and statistical relationships. The method breaks down the aircraft into major components and uses statistical relationships to estimate their weights. The method includes equations for the weights of the wing, tail, fuselage, landing gear, and other components. [25]

3.2.4 USAF Method

The United States Air Force (USAF) has developed its own method for estimating aircraft weights. This technique establishes correlations between weight and particular design characteristics by making use of historical data collected from already-existing aircraft. It considers the aircraft's mission, design parameters, and size to estimate the weights of different components. The USAF method includes equations for estimating the weights of the wing, empennage, fuselage, systems, and more. [26]

3.3 Data Collection

In order for the author to carry out this research, the author required some data. The data used are from Wikipedia and from Jane's All the World's Aircraft book [27] . As a result of this, the author came to the conclusion that some data on regional turboprop aircraft should be collected. The gathered information is an essential component in getting this thesis off the ground. As shown in the previous chapter, several aircraft were introduced, but due to the lack of data of most of the aircraft, the author made the decision to use three distinct types of regional turboprop aircraft, which are the ATR 42-600, the Saab 340 , and the CN-235. The following sections will discuss each aircraft to give a rough overview of what aircrafts were used by the author.

3.3.1 ATR 42-600

ATR, a French aircraft manufacturer, is responsible for the construction of the short-haul, twin-turboprop regional passenger aircraft known as the ATR 42. The number of seats that fall between 40 and 50 is where the name ”42” originates from. After further development, this aircraft became known as the ATR 72. Production of this aircraft began in 1981, and its maiden voyage took place on August 16, 1984. At a press conference that took place in Washington, District of Columbia, on Thursday, October 2, 2007, ATR CEO Stéphane Mayer introduced the new aircraft of the -600 series.

FIGURE 3.2: TransNusa ATR 42-600

The new ATR 42-600 will be outfitted with the most up-to-date technology, which will be constructed using the invaluable expertise that was gathered from earlier aircraft. These new aircraft will have improved efficacy, improved reliability, lower fuel consumption, and lower operating costs. The aircraft will be powered by a PW127M engine, which is the industry standard. This new engine offers improved performance on shorter runways, as well as in hotter and higher altitude environments, as well as a 5 percent boost in thermodynamic power during takeoff. Featuring a ”boost function” that can increase power but is only activated during takeoff, where this function will ever be used.

3.3.2 Saab 340

The Saab Aircraft includes two powerful General Electric engines, a spacious cabin with accommodation for up to 34 passengers, sufficient seats to ensure the comfort of those traveling, a fully stocked restroom, and all of the necessary equipment for providing service on board. The Saab 340 is a kind of aircraft that has two turboprop engines. Saab Aircraft is a Swedish aircraft company that was founded in 1927.

FIGURE 3.3: Saab 340

In the 1970s, Saab began the process of developing what would later be known as the Saab 340. The production of an airplane that could accommodate between 30 and 40 passengers on short-distance journeys was the primary target. Saab made the calculated decision to enter into a cooperation with the aircraft manufacturer Swearingen, which is located in the state of Texas and is a subsidiary of Fairchild Industries.The active aircraft fleet of the airline now consists of two Saab 340As. According to the data provided by ch-aviation, the airline's two Saab 340As have an average age of 37.3 years and are configured with a layout of 30 seats in a one-class configuration.

3.3.3 CN-235

IPTN and CASA, which is now known as Airbus Defense and Space, founded a new joint-venture business known as Aircraft Technology (Airtech) on October 17, 1979, with the intention of designing the CN235. This innovative multi-purpose aircraft has the ability to perform a Short Take-Off and Landing (STOL) at difficult airstrips that are 800 meters long, has a ramp door that allows for easy outgoing and incoming goods transport, and has low maintenance costs. The CASA/IPTN CN-235 is a twin-engined transport aircraft with a medium-range capability.

FIGURE 3.4: CN-235

Its principal military functions are in the areas of air transport, maritime surveillance, and patrolling the oceans. CASA and IPTN, an Indonesian manufacturing company, collaborated on the project and founded the company Airtech to administer the program. The project was a joint venture. The partnership was only applicable to the Series 10 and Series 100/110 models; subsequent versions were created on an individual basis. Over 230 different variants of the CN-235 are now in operation. CASA began research and development on the C-295, a stretched version of the CN-235, in 1995.

3.4 Computational Tools

An application or item of software that serves as a means of carrying out a procedure or accomplishing a goal is known as a computational tool. Throughout the course of this research, the author has considered and been instructed to make use of computers to carry out all of the scientific computing. The author was able to effortlessly collect data and process it in the manner in which it was required thanks to the assistance of a computer. It was speculated that the author made use of various programs in the hopes of making the study somewhat less laborious. Because all of the programs were available at no cost and could be modified by anyone, the author had access to a large number of resources that detailed how to get the most out of each application. The majority of the applications were focused on making use of programming languages and the packages or modules that come with them.

3.4.1 Python

During the course of this study, the Python programming language was the one that was utilized for its various coding purposes. Python is a high-level programming language that can be interpreted and is object-oriented. Python also has dynamic semantics. Python is widely utilized in the process of producing websites and applications, as well as for automating tasks, analyzing data, and visualizing data. Python was selected by the author because, in comparison to other programming languages, it was simpler to understand, and the author has limited experience in the programming field, particularly in scientific computing. These factors contributed to the author's decision to adopt Python as the main programming language for the thesis.

3.4.2 NumPy

NumPy is an essential library that Python users need in order to perform scientific computing. NumPy is a Python library that allows users to perform mathematical and logical operations on arrays [28]. The author required the numerical computing capability of this package, which included massive numerical array objects and procedures to handle them, in order to complete this research, which is why it was very helpful throughout the process.

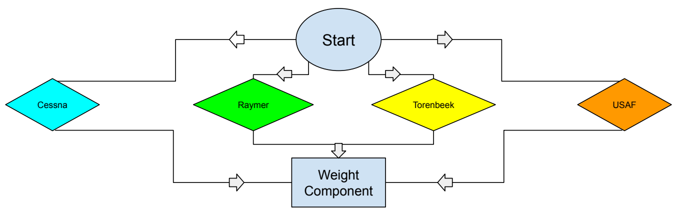


FIGURE 3.5: Aircraft Coding Data Processing

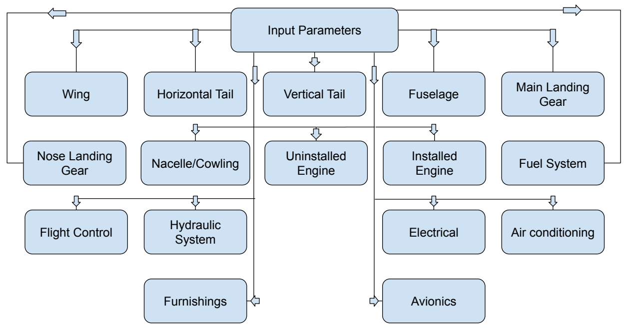


FIGURE 3.6: Weight Components Coding Flow Chart

3.5 Calculate the aircraft's weight

When determining the weight of an aircraft, it is necessary to take into consideration a number of different aspects and components. The weights of an airplane are often broken down into various categories, such as the weight of the aircraft when it is empty, the weight of the payload, and the weight of the fuel. In order for the author to carry out this research, the author use the statistical estimation methods. When it comes to estimating numerous factors and features that are associated with aircraft performance, safety, maintenance, and operational elements, statistical estimation methods are utilized. In order to arrive at accurate estimations and forecasts, these techniques require doing analyses on the data gathered from various aircraft systems, instruments, flight tests, and other sources.

The following hypothetical method is use of a statistical method for estimating weight that the author could take:

1. **Collecting Data**

   The author acquired the data from a variety of sources, such as historical records, reports on maintenance, measurements, and specifications provided by the aircraft manufacturer. Throughout the entirety of the procedure, it was used in both the input and the reference value capacities. The author collected information from members of the general public, which is then made available on a variety of websites that offer data on the characteristics of a variety of different aircraft.

2. **Data Preparation**

   By cleaning and arranging the gathered data, the author assures consistency and eliminates any outliers or errors that may have an effect on the estimating process. In addition, this step prevents any errors from having an effect.

3. **Feature Selection**

   By determining the relevant aspects that can influence the weight of the aircraft, such as the MTOW, EMTOW, wing surface, wing area, and any other relevant components. In the model for estimating weight, these considerations will each take the role of an independent variable.

4. **Model Development**

   Creating a statistical model, such as a trend line chart model or a multiple linear regression model, in which the independent variables are utilized to make predictions about the dependent variable, which in this case is the weight of the aircraft. Statistical software or computer languages could be utilized in the construction of the model.

3.6 Comparison and Analysis

The purpose of statistical estimation is to produce the most accurate estimate feasible of an unknown variable or characteristic, in addition to an indication of the degree to which this estimate is uncertain. Because the author had access to the necessary mathematical tools and they were ready for use, the author was able to obtain the data that was required for comparison. The statistical approaches used for weight estimation are taken from historical data collected from already-existing aircraft. It is possible to generate relationships based on geometrical parameters such as wing area, aspect ratio, taper ratio, ultimate load factors, and so forth. For example, if we know the weight of the wing structure for a population of aircraft that fall into a certain class, we will be able to determine which relationships may be derived from the data. Even if they were made by different companies, the wing weights of two different airplanes in the same class that are certified to the same set of standards and have similar gross weights should be similar.

Variables	SI	Imperial
Vcruise (km/h, ft/s)	535	488
cruise density(kg/m^3, lb/ft^3)	0.6597	0.0411
chord(m, ft)	2.04	6.6929
b(wingspan)(m, ft)	24.57	80.6102
S(wing surface)(ft2̂)	54.5	586.6331
AR	11.07	11.07
λ_w	0.54	0.54
Wing Sweep at 25%	2°	2°
Wing Sweep at 50%	0°	0°
t/c	0.15	0.15
twmax (m, ft)	0.306	1.0039
WFW(kg, lb)	4500	9920.8017
q(kg/m^2, lb/ft^2)	94058.706	4895.2029
nz	3.5	-
EMTOW(lb)	11750	25904.316
MTOW(lb)	18600	41005.981
Lf(fuselage length)(m, ft)	22.7	74.4750
df(fuselage width)(m, ft)	2.6	8.5301
Sh(Horizontal tail surface)(m^2, ft^2)	11.5	123.784
Cvroot(Vertical tail root chord)(m, ft)	3.4	11.1548
bv(vertical tail span(bv)(m, ft)	4.5	14.7637
Sv(vertical tail surface)(m^2, ft^2)	12.7	136.7016
Nocc(number of occupants)	48	48

TABLE 3.1: ATR 42-600 Geometrical Data. Collected from [27]

Variables	SI	Imperial
Vcruise (km/h, ft/s)	455	415
cruise density(kg/m^3, lb/ft^3)	0.54895	0.0342
chord(m, ft)	2.5	8.2020
b(m, ft)	25.81	84.6784
S(m^2, ft^2)	59.1	636.1471
AR	11.27	11.27
λ_w	0.36	0.36
Wing Sweep at 25%	2°	2°
Wing Sweep at 50%	0°	0°
t/c	0.18	0.18
twmax(m, ft)	0.45	1.4763
WFW(kg, lb)	5220	11508.130
q(kg/m^2, lb/ft^2)	56823.1868	2946.2702
nz	3.5	3.5
EMTOW(kg, lb)	9800	21605.302
MTOW(kg, lb)	16100	35494.424
Lf(fuselage length)(m, ft)	21.4	70.2099
df(fuselage width)(m, ft)	2.9	9.5144
Sh(Horizontal tail surface)(m^2, ft^2)	21.2	228.1949
Cvroot(Vertical tail root chord)(m, ft)	2.39	7.8412
bv(vertical tail span(bv)(m, ft)	4.63	15.1902
Sv(vertical tail surface)(m^2, ft^2)	11.11	119.5870
Nocc(number of occupants)	51	51

TABLE 3.2: CN-235 Geometrical Data. Collected from [27]

Variables	SI	Imperial
Vcruise (km/h, ft/s)	524	478.5
cruise density (kg/m^3, lb/ft^3)	0.5489	0.0342
chord(m, ft)	1.95	6.3976
b(wingspan)(m, ft)	21.44	70.3412
S(wing surface) (m^2, ft^2)	41.81	450.0390
AR	11.0	11.0
λ_w	0.4	0.4
Wing Sweep at 25%	2°	2°
Wing Sweep at 50%	0°	0°
t/c	0.16	0.16
twmax(m, ft)	0.312	0.312
WFW (kg, lb)	2580	5687.9
q (kg/m^2, lb/ft^2)	75364.2476	3922.5466
nz	3.5	3.5
EMTOW (kg, lb)	8618	18999.438
MTOW (kg lb)	13154	28999.606
Lf(fuselage length) (m, ft)	19.73	64.7309
df(fuselage width) (m, ft)	2.31	7.578
Sh(Horizontal tail surface) (m^2, ft^2)	14.57	156.8301
Cvroot(Vertical tail root chord)(m, ft)	2.7	8.858
bv(vertical tail span(bv)(m, ft)	3.89	12.7624
Sv(vertical tail surface)(m^2, ft^2)	10.53	113.3439
Nocc(number of occupants)	34	34

TABLE 3.3: Saab 340 Geometrical Data. Collected from [27]

This assumption is based on the notion that the gross weight of the aircraft will be comparable. Because of this, the statistical relationship that was formed by the entirety of the class of aircraft may be used to estimate the wing weight of any aircraft in the same class, provided that the aircraft in question falls somewhere in the middle of the range of aircraft that belong to that class. These types of estimating methods typically require certain dimensions to have been specified before to their application.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Data Results

This section contains all the results of calculations of the weight components for each of methods. But before getting into the results, here a few definitions and their formulations are defined.

1. The expression below is used to calculate the component weight of AVG RTU.

$$\text{AVG RTU} = \frac{\text{Weight}_{\text{Raymer}} + \text{Weight}_{\text{Torenbeek}} + \text{Weight}_{\text{USAF}}}{3} \tag{4.1}$$

2. For each of the methods, the EMTOW (Empty Maximum Take-Off Weight) is the total sum of component weights for the corresponding method.

$$\text{EMTOW} = \text{MTOW} - W_{\text{fuel}} - W_{\text{pax+cabin}} - W_{\text{cargo}} = \text{Total Weight Components} \tag{4.2}$$

3. And here the EMTOW percentage expression is defined as,

$$\text{EMTOW Percentage} = \frac{\text{EMTOW}}{\text{MTOW}} \times 100\% \tag{4.3}$$

4.1.1 Aircraft Total Weight Data Results

This section contains all of the extracted data from all of aircraft component weight summary

Components	Cessna	Raymer	Torenbeek	USAF	AVG RTU
Wing weight(lb)	1799	2921	3022	2972	2972
Horizontal Tail Weight(lb)	405	474	0	548	341
Vertical Tail Weight(lb)	2	675	0	142	272
Empenage Weight(lb)	407	1150	388	691	743
Fuselage Weight(lb)	4372	9814	0	3931	4582
Main Landing Gear(lb)	1413	2226	1326	561	1371
Nose Landing Gear(lb)	0	535	315	0	283
Nacele/Cowling Weight(lb)	1082	0	445	0	148
Uinstalled(dry) Engine Weight(lb)	653	653	653	653	653
Installed Engine Weight(lb)	4983	1896	1963	1896	1918
Fuel System Weight(lb)	642	500	632	500	544
Flight Control System(lb)	596	1389	693	1654	1245
Hydraulic System Weight(lb)	35	351	35	35	140
Avionics System Weight(lb)	1451	1451	1451	1451	
Electrical System(lb)	951	832	694	832	786
Air Conditioning System(lb)	2885	2885	2885	2885	2885
Furnishings(lb)	623	2000	0	508	836
Total Weight(lb)	22299	29752	14502	19259	21171
MTOW(lb)			35494		
Percentage of EMTOW	0.63%	0.84%	0.41%	0.54%	0.6%

TABLE 4.1: CN-235 Weight Summary (for average calculation result only from Raymer to USAF)

Components	Cessna	Raymer	Torenbeek	USAF	AVG RTU
Wing weight(lb)	1585	3094	3837	2913	3281
Horizontal Tail Weight(lb)	496	963	0	1195	719
Vertical Tail Weight(lb)	1	937	0	148	362
Empenage Weight(lb)	498	1900	713	1343	1319
Fuselage Weight(lb)	4651	10225	0	3943	4723
Main Landing Gear(lb)	1601	2360	1517	581	1486
Nose Landing Gear(lb)	0	490	350	0	280
Nacele/Cowling Weight(lb)	1316	0	654	0	218
Uinstalled(dry) Engine Weight(lb)	926	926	926	926	926
Installed Engine Weight(lb)	6029	2226	2426	2226	2293
Fuel System Weight(lb)	554	449	568	450	489
Flight Control System(lb)	688	1739	763	1829	1444
Hydraulic System Weight(lb)	41	415	41	41	166
Avionics System Weight(lb)	1451	1451	1451	1451	
Electrical System(lb)	1098	591	766	591	649
Air Conditioning System(lb)	3010	3010	3010	3010	3010
Furnishings(lb)	624	2321	0	577	966
Total Weight(lb)	24569	33097	17022	21224	23781
MTOW(lb)			41006		
Percentage of EMTOW	0.60%	0.81%	0.42%	0.52%	0.58%

TABLE 4.2: ATR 42-600 Weight Summary (for average calculation result only from Raymer to USAF)

Components	Cessna	Raymer	Torenbeek	USAF	AVG RTU
Wing weight(lb)	1191	2097	2322	2033	2144
Horizontal Tail Weight(lb)	368	573	0	601	391
Vertical Tail Weight(lb)	1	650	0	119	256
Empenage Weight(lb)	370	1223	454	721	799
Fuselage Weight(lb)	2325	10357	0	3050	4469
Main Landing Gear(lb)	765	1608	1020	397	1008
Nose Landing Gear(lb)	0	471	252	0	241
Nacele/Cowling Weight(lb)	1051	0	458	0	153
Uinstalled(dry) Engine Weight(lb)	670	670	670	670	670
Installed Engine Weight(lb)	4848	1851	1947	1851	1883
Fuel System Weight(lb)	317	300	428	300	343
Flight Control System(lb)	487	992	606	1435	1011
Hydraulic System Weight(lb)	28	315	28	28	124
Avionics System Weight(lb)	1451	1451	1451	1451	
Electrical System(lb)	777	566	463	566	532
Air Conditioning System(lb)	1988	1988	1988	1988	1988
Furnishings(lb)	355	1622	0	546	723
Total Weight(lb)	16992	26714	12087	15756	18186
MTOW(lb)			29000		
Percentage of EMTOW	0.59%	0.92%	0.42%	0.54%	0.63%

TABLE 4.3: Saab 340 Weight Summary (for average calculation result only from Raymer to USAF)

4.2 CN-235 Data Analysis

As we can see from the table 4.1, we can see all the components along with the results of the predicted results from each method. For the wing weight, it can be seen that the cessna has the lowest value of the other methods. And so with the horizontal tail weight, vertical tail weight, and several other components. And for average calculation is only from Raymer to USAF. It's because the cessna method is not well accurate for calculations on aircraft classes such as CN-235, ATR 42-600, and Saab 340.

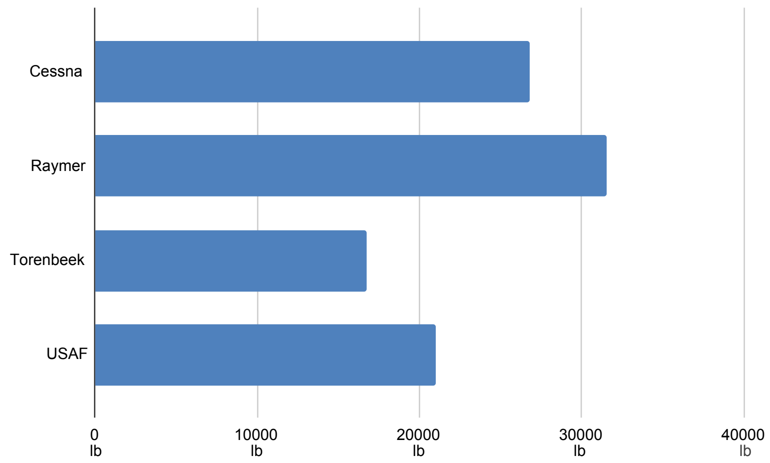


FIGURE 4.1: Comparison of total component weight from each method

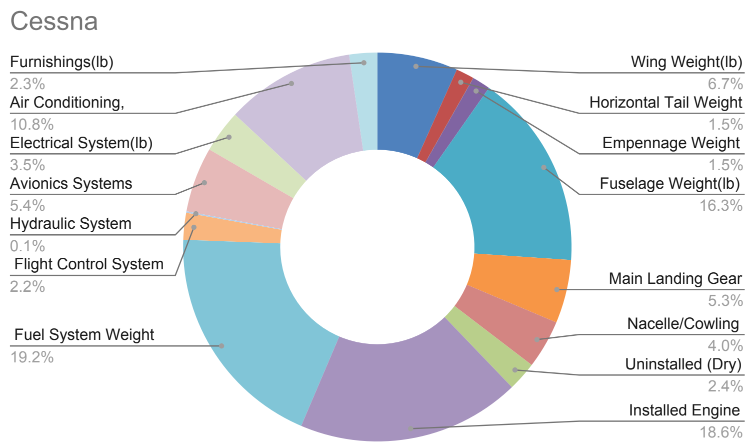


FIGURE 4.2: Cessna chart

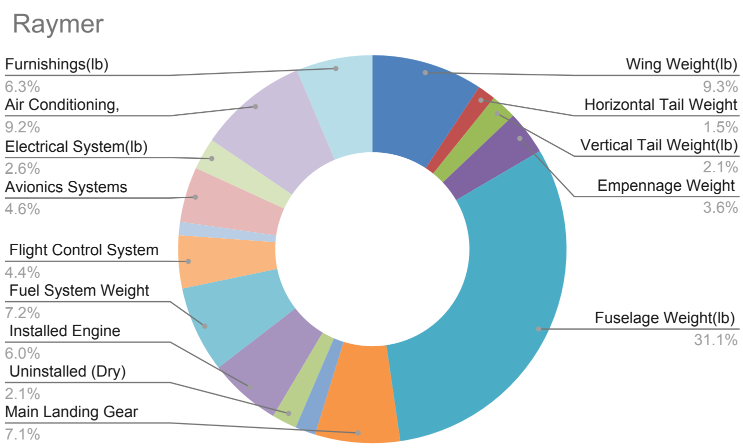


FIGURE 4.3: Raymer chart

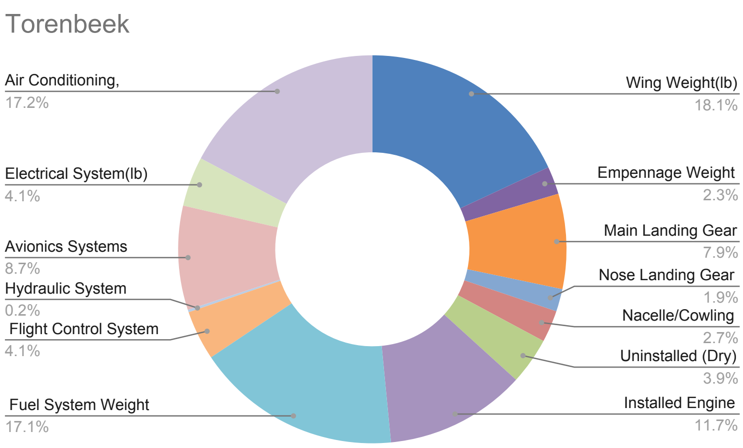


FIGURE 4.4: Torenbeek chart

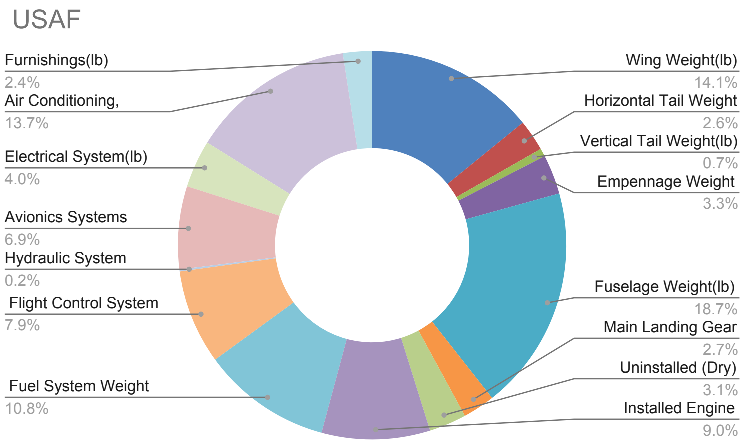


FIGURE 4.5: USAF chart

4.2.1 Effectiveness in Estimating Total Mass (%)

Based on the data from table 4.1 , The Raymer Method estimates a mass percentage that is the highest compared to the other methods, and it offers an estimated empty mass that is 29,752 lb. This shows that the Raymer Method may not account for a major amount of the overall mass and may have poorer accuracy compared to the other ways. Moreover, this suggests that the other methods may be more accurate. The Cessna Method has a mass percentage that is higher after the Raymer Method, and it has a mass that is empty of 22,299 lb.

The Torenbeek Method has the lowest mass percentage among the methods and provides an empty mass of 14,502 lb. This indicates that the Torenbeek Method attributes a larger proportion of the aircraft's total weight of the total mass to the components considered in the method. The USAF Method has an empty mass of 19,259 lb. This suggests that the USAF Method attributes a considerable portion of the total mass to the components considered.

4.2.2 Impact on the Total Mass (%)

To determine which method has a greater impact on the total mass of the aircraft, it can be compare to their mass percentages. Based on the information provided from figure 4.6, the Torenbeek Method yields the greatest mass percentage, which comes in at 0.41%. This leads one to believe that the Torenbeek Method credits the largest amount of the total mass to the components that are evaluated in the method, which indicates that there may be a potentially considerable impact on the overall weight of the aircraft.

The USAF Method comes in second place with a mass percentage of 46 %, falling behind the Cessna Method 0.54%. In comparison to the Torenbeek Method, these approaches contribute a less percentage to the overall mass, but they still have a discernible effect on the final product. The Raymer Method has the highest mass percentage of any of the other ways, coming in at 0.84%. This indicates that a larger proportion of the aircraft's weight is already dedicated to its structure and systems, leaving less available weight for variable items. This can impact performance and operational flexibility.

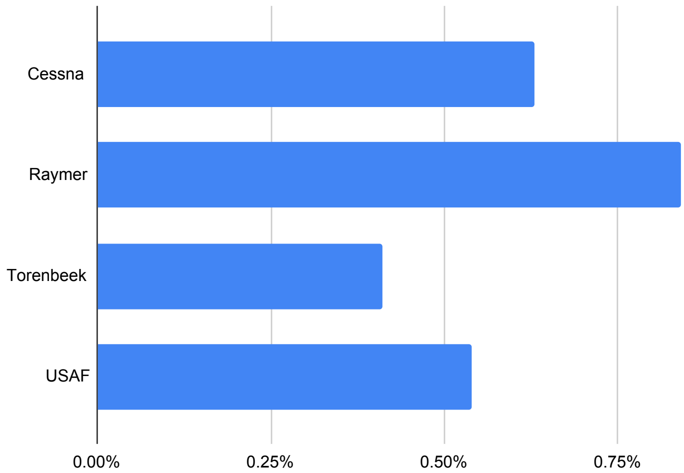


FIGURE 4.6: Comparison of mass percentage from each methods on CN-235

4.3 ATR 42-600 Data Analysis

From figure 4.7 we can see that Raymer's method has the greatest value. then followed by the cessna method in position 2 and the USAF and Torenbeek methods in positions 3 and 4. This is due to the fact that the Raymer method gives a simplified means of calculating the weight of various components of an airplane, in addition to being founded on the statistical analysis of historical data collected from already-existing aircraft. It is based on the idea of weight fractions, which reflect the proportion of the aircraft's maximum takeoff weight (MTOW) that can be attributed to each main component, such as the wing, the fuselage, the empennage, the landing gear, and the propulsion system. This allows the maximum takeoff weight to be calculated more accurately. We can see from figure 4.9, the overall result of each predicted weight of each component is held by Raymer as the highest percentage result on average than the other chart

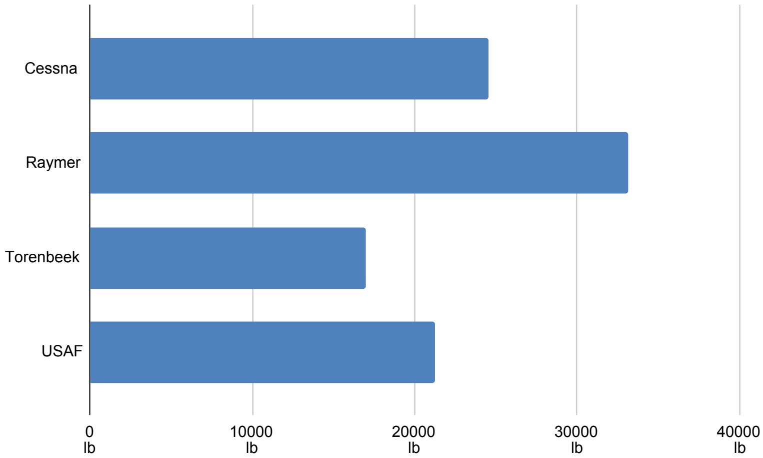


FIGURE 4.7: Comparison of total component weight from each method

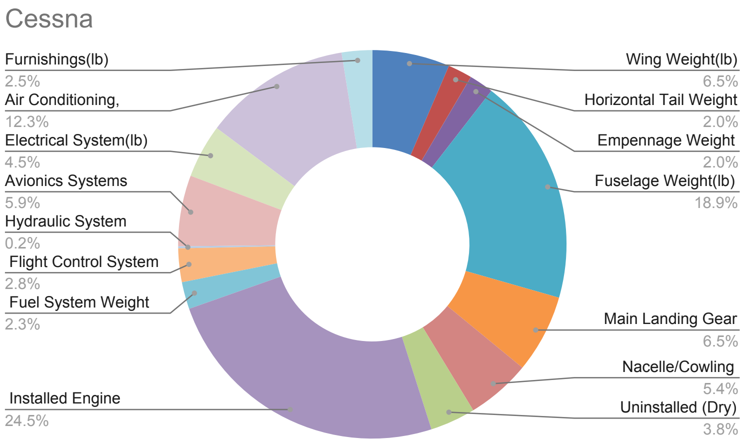


FIGURE 4.8: Cessna chart

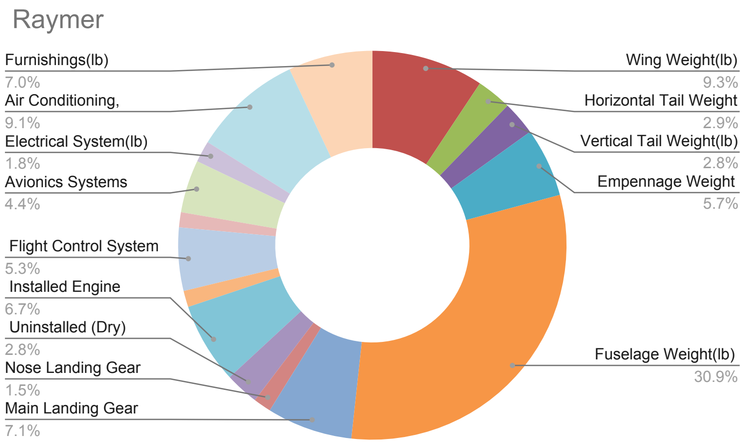


FIGURE 4.9: Raymer chart

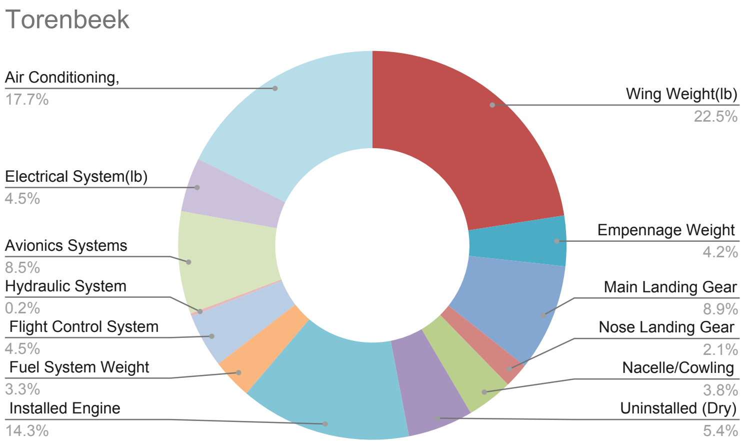


FIGURE 4.10: Torenbeek chart

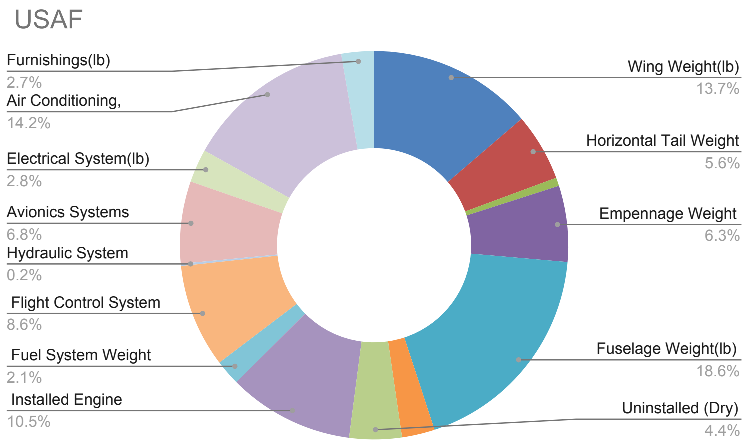


FIGURE 4.11: USAF chart

4.3.1 Effectiveness in Estimating Total Mass (%)

Based on the data from table 4.2, The Torenbeek Method has the greatest mass percentage, which indicates that it attributes a significantly bigger amount of the total mass to the components that are considered in the method. The Torenbeek Method has a mass percentage of 0.42%. This points to the possibility that the Torenbeek Method can produce a more accurate calculation of weight. Following the USAF Method with 0.52% and then 0.60% mass percentage is the Cessna Method, and then the Raymer Method with 0.81% mass percentage. The contributions made by these approaches to the overall mass are not as good as those made by the Torenbeek Method.

4.3.2 Impact on the Total Mass (%)

From the given data, the Torenbeek Method has the better mass percentage of 0.42%. This indicates that the Torenbeek Method attributes the largest portion of the total mass to the components considered in the method, suggesting a potentially significant impact on the overall weight of the aircraft.

The USAF Method has a mass percentage of 0.52%, followed by the cessna Method with 0.60% and the Raymer Method with 0.81%. Therefore, based on the provided data, the Torenbeek Method is likely to have the greatest impact on the total mass of the aircraft among the methods mentioned.

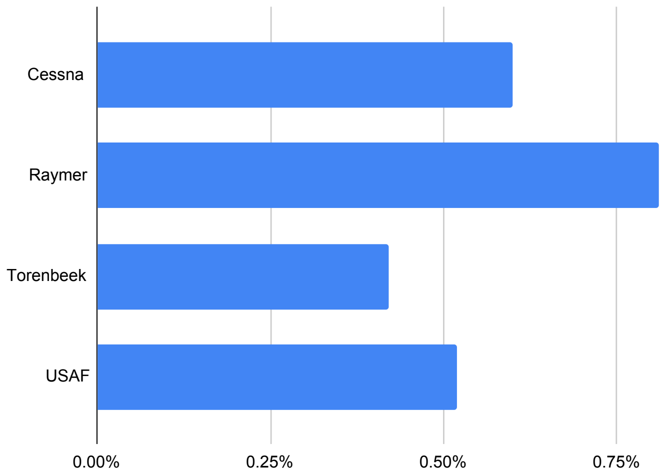


FIGURE 4.12: Comparison of mass percentage from each methods on ATR 42-600

4.4 Saab 340 Data Analysis

From table 4.3 we can see that For the total predicted weight component for Raymer method has the biggest value from the other method with 26714 lb. The overall predicted weight of an aircraft frequently involves a number of distinct components, including the weight of the aircraft's structure, the weight of its fuel, the weight of its payload, and any other relevant operational items. The total weight of an airplane is the weight of all of its basic parts, like the airframe, engines, landing gear, and other systems. The maximum takeoff weight (MTOW) of the aircraft shall not be greater than or equal to the total anticipated weight of the aircraft. With the value of MTOW from Saab 340 which is 28999.60 lb, means Raymer's method is the method that has the closest results to the MTOW results.

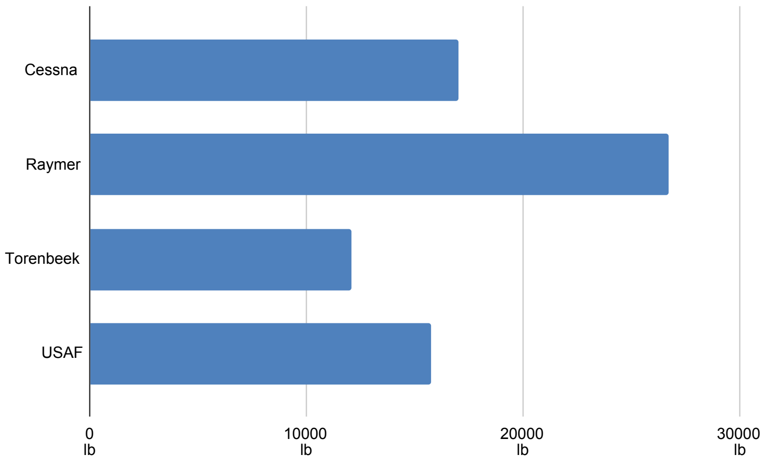


FIGURE 4.13: Comparison of total component weight from each method

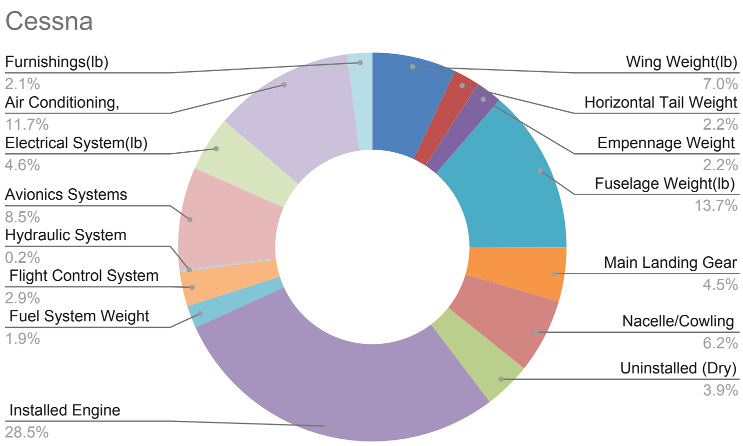


FIGURE 4.14: Cessna chart

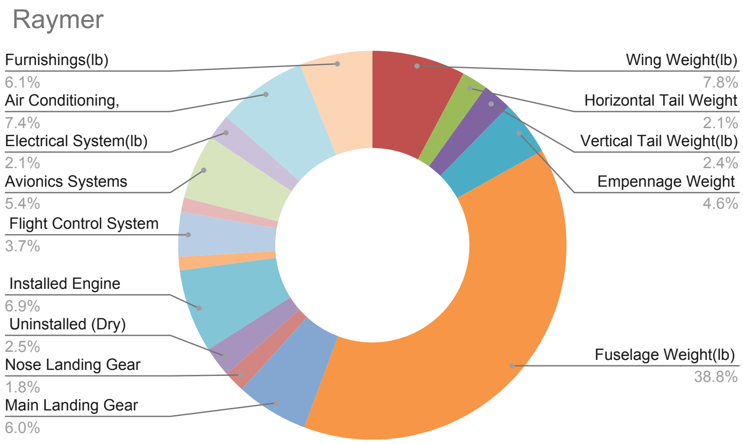


FIGURE 4.15: Raymer chart

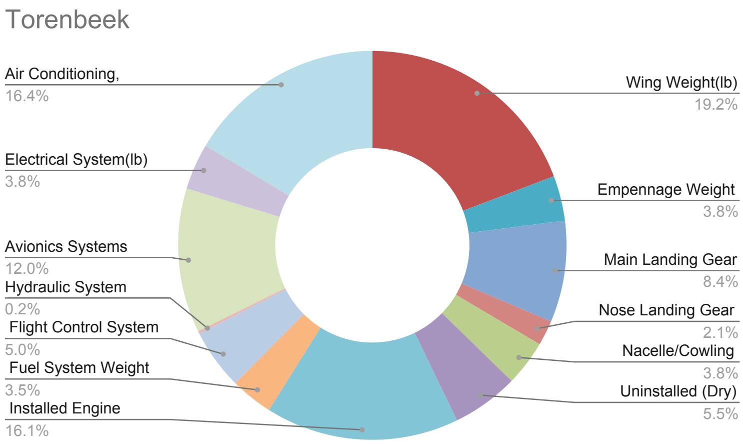


FIGURE 4.16: Torenbeek chart

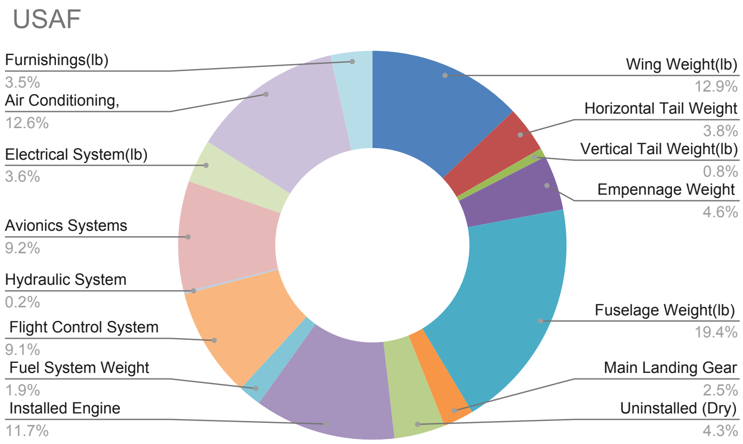


FIGURE 4.17: USAF chart

4.4.1 Effectiveness in Estimating Total Mass (%)

With a mass percentage of 0.42%, the Torenbeek Method gives a relatively larger amount of the total mass to the parts that are taken into account see table 4.3. This means that the Torenbeek Method might be a better way to figure out how much something weights. After the USAF Method with 0.54%, the Cessna Method with 0.59%, and the Raymer Method with 0.92%, the Raymer Method has the highest mass percentage. The contributions made by these methods to the overall mass are not as good as those made by the Torenbeek Method.

4.4.2 Impact on the Total Mass (%)

Based on the data that was provided from table 4.3 , the Torenbeek Method yields the greatest mass percentage, which comes in at 0.42%. This suggests that the Torenbeek approach credits the largest amount of the total mass to the components evaluated in the approach, which hints at the possibility of a major impact on the aircraft's overall weight. After the USAF Method with 0.54%, the Cessna Method with 0.59%, and the Raymer Method with 0.92%, the Raymer Method has the highest mass percentage. When compared to the Torenbeek Method, these approaches contribute a highest percentage to the overall mass, means The contributions made by these methods to the overall mass are not as good as those made by the Torenbeek Method. Therefore, based on the data that has been presented, the Torenbeek Method is expected to have the most substantial impact, in comparison to the other methods that have been described, on the overall mass of the aircraft.

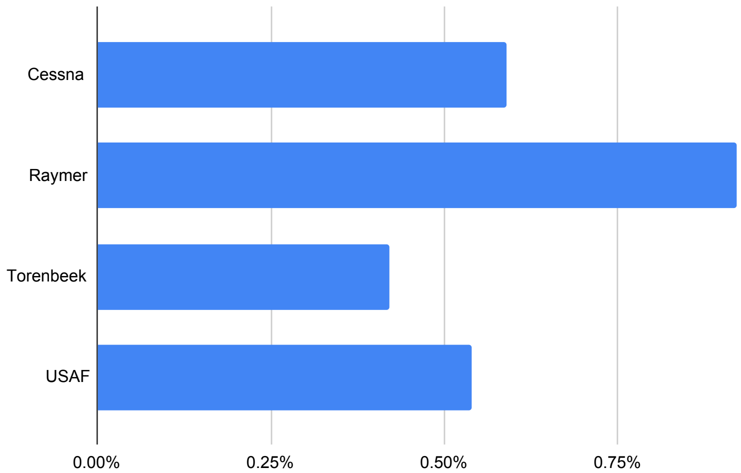


FIGURE 4.18: Comparison of mass percentage from each methods on Saab 340

4.4.3 Final Results

Aircraft	Actual EMTOW(lb)	Raymer(lb)	Torenbeek(lb)	USAF(lb)	AVG RTU(lb)	% Different of AVG With Actual EMTOW	Cessna(lb)
CN-235	21605	29752	14502	19259	21171	2.01	22299
ATR 42-600	25904	33097	17022	21224	23781	8.20	24569
Saab 340	18999	26714	12087	15756	18186	4.28	16992

TABLE 4.4: Comparison Between The Actual EMTOW and the Average From Each Method.

From the table 4.4 we can see the comparison between actual EMTOW and the results from each method. From the raymer method we can see that raymer method estimates a significantly higher empty weight compared to the actual Empty weight. Torenbeek estimates a much lower empty weight compared to the actual Empty weight. And USAF provides an empty weight that is somewhat close to the actual empty but still different. While the average calculation of

Raymer, Torenbeek, and USAF has the result that almost similar to the real actual empty weight. The average calculation from CN-235 is only 2.01% different from actual Empty weight. The ATR 42-600 is 8.20% and for Saab 340 is 4.28%. Means the methods used are reasonably accurate is estimating the empty weight of each aircraft.

CHAPTER 5
SUMMARY, CONCLUSION, AND RECOMMENDATION

5.1 Summary

Based on what have been shown and described in this thesis, this thesis can be summarized as the following:

1. The author had accomplished the task of acquiring and extracting the necessary reference values of data from a variety of sources that were available on the internet.

2. The python programming tools that were utilized in order to estimate the weight component for each type of aircraft performed excellently. It is adequate on a level that allows the author to move forward with this thesis.

3. The purpose of this research was to evaluate and contrast the performance of several approaches to weight estimate for regional turboprop aircraft. It's because the author couldn't find the real weight of each component of an aircraft. Raymer, Cessna, Torenbeek, and USAF were some of the methods that were taken into consideration. The analysis centered on the effect that they had on the total mass, the mass percentages, and the estimated empty masses (EMTOM) in comparison to the total mass (MTOM) of an aircraft.

5.2 Conclusion

On the basis of what has been demonstrated and discussed throughout this thesis, the following can be deduced regarding this thesis:

1. The author did a comparison and analysis of different methods for estimating the weight of regional turboprop aircraft, focusing particular attention to the Raymer, Cessna, Torenbeek, and USAF methods. Based on the information that was readily available, the analysis took into account the impact on the total mass, the mass percentages, and the estimated empty masses (EMTOM) in proportion to the total mass (MTOM).

2. The provided data for 3 aircraft indicates that there is variability among the different estimation methods. We can see that the raymer method has an overestimated calculation, and the torenbeek has an underestimates calculation, while the USAF is in the middle between the raymer and torenbeek because it has results that are slightly close to actual EMTOW

3. The average estimation is the one who almost similar to the actual empty weight. With a percentage difference that does not exceed 10%. This suggests that the methods used are providing reasonably accurate predictions of the aircraft's empty weight.

4. According to the data, the Torenbeek Method consistently demonstrated the lowest mass percentage when compared to the other methods, indicating that a larger proportion of the aircraft's total weight is available for carrying fuel, passengers, and cargo. This can result in better performance characteristics such as longer range, higher payload capacity, and improved fuel efficiency.

5. In addition to this, we need to take into account the correlation that exists between the estimated empty masses, also known as EMTOM, and the overall mass, which is denoted by MTOM. The method has a greater chance of successfully estimating weight in proportion to the degree to which the projected empty mass approaches the total mass. This likelihood increases as the distance between the two values decreases. In spite of this, it is difficult to determine how accurate the estimates are because we do not have access to any genuine measurements.

5.3 Recommendation

The author is able to draw several lines for future works that can be developed from this research based on the results of comparing different statistical weighting methods. These lines are as follows:

1. It is recommended to further validate and calibrate the weight estimation methods using actual measurements from regional turboprop aircraft. This will help assess the accuracy and reliability of the methods and ensure their applicability to real-world scenarios. Collaborations with aircraft manufacturers or operators can provide access to data for validation purposes.

2. In future study, it may be possible to investigate the applicability of these methods to a greater variety of aircraft types and to think about other weight components as a way to further improve the accuracy of weight estimation. In addition, the utilization of sophisticated data analysis procedures and optimization algorithms may result in weight estimating strategies that are both more accurate and more productive.

3. The aviation industry is always seeing the development of new technology, materials, and design techniques; as a result, weight estimating methodologies should be continuously developed and assessed. The efficiency of the approaches, as well as their continued relevance in the rapidly changing landscape of regional turboprop aircraft, can be ensured through the implementation of regular updates and enhancements to those methods.

Bibliography

[1] S. Gudmundsson, *General Aviation Aircraft Design: Applied Methods and Procedures*, Second edition. Oxford Cambridge, MA: Butterworth-Heinemann, 2022, ISBN: 978-0-12-818465-3.

[2] *Aircraft Design*, https://www.cranfield.ac.uk/academic-disciplines/aircraft-design. (visited on 06/30/2023).

[3] C. You, M. Yasaee, S. He, *et al.*, "Identification of the key design inputs for the FEM-based preliminary sizing and mass estimation of a civil aircraft wing box structure," *Aerospace Science and Technology*, vol. 121, p. 107 284, Feb. 2022, ISSN: 12709638. DOI: 10.1016/j.ast.2021.107284. (visited on 06/30/2023).

[4] K. Welding, *What are the Five Main Parts of an Airplane?* Mar. 2022. (visited on 06/05/2023).

[5] *Aircraft Components & Structure*, https://www.cfinotebook.net/notebook/aerodynamics-and-performance/aircraft-components-and-structure. (visited on 06/05/2023).

[6] *ATR predicts a need for at least 2,450 new turboprops to address a global demand for lower emission and affordable connectivity*, https://www.atr-aircraft.com/presspost/atr-predicts-a-need-for-at-least-2450-new-turboprops-to-address-a-global-demand-for-lower-emission-and-affordable-connectivity/. (visited on 06/30/2023).

[7] *Turboprop Aircraft*, https://nbaa.org/business-aviation/business-aircraft/turboprop-aircraft/. (visited on 06/30/2023).

[8] *Maximum Take-Off Mass (MTOM) — SKYbrary Aviation Safety*, https://www.skybrary.aero/art take-mass-mtom. (visited on 07/02/2023).

[9] aviator, *MTOW - Maximum Takeoff Weight - aviationfile*, https://www.aviationfile.com/mtow-maximum-takeoff-weight/, Apr. 2023. (visited on 07/02/2023).

[10] "List of regional airliners," *Wikipedia*, May 2023. (visited on 06/30/2023).

[11] "CASA/IPTN CN-235," *Wikipedia*, Jun. 2023. (visited on 07/02/2023).

[12] "Antonov An-140," *Wikipedia*, Jun. 2023. (visited on 07/02/2023).

[13] "De Havilland Canada Dash 8," *Wikipedia*, Jun. 2023. (visited on 07/02/2023).

[14] "ATR 42," *Wikipedia*, Jun. 2023. (visited on 07/02/2023).

[15] "Xian MA600," *Wikipedia*, Apr. 2023. (visited on 07/02/2023).

[16] "Saab 340," *Wikipedia*, Jun. 2023. (visited on 07/02/2023).

[17] "De Havilland Canada Dash 7," *Wikipedia*, May 2023. (visited on 07/02/2023).

[18] "Fokker 50," *Wikipedia*, Jun. 2023. (visited on 07/02/2023).

[19] "British Aerospace ATP," *Wikipedia*, Jun. 2023. (visited on 07/02/2023).

[20] "Antonov An-24," *Wikipedia*, Jun. 2023. (visited on 07/02/2023).

[21] "Hawker Siddeley HS 748," *Wikipedia*, May 2023. (visited on 07/02/2023).

[22] "Fokker F27 Friendship," *Wikipedia*, May 2023. (visited on 07/02/2023).

[23] "Saab 2000," *Wikipedia*, Jun. 2023. (visited on 07/02/2023).

[24] K. Towell, "Aircraft Design – a Conceptual Approach – Fifth edition D. P. Raymer American Institute of Aeronautics and Astronautics, 1801 Alexander Bell Drive, Suite 500, Reston, VA 20191-4344, USA. 2012. Distributed by Transatlantic Publishers Group, 97 Greenham Road London N10 1LN, UK (Tel: 020-8815 5994; e-mail: Mark.chaloner@tpgltd.co.uk). 2012. 1044pp. Illustrated. £85. [20% discount available to RAeS members on request. ISBN 978-1-60086-911-2.," *The Aeronautical Journal*, vol. 117, no. 1192, pp. 649–650, 2013, ISSN: 2059-6464. DOI: 10.1017/S0001924000008289.

[25] E. Torenbeek, *Synthesis of Subsonic Airplane Design: An Introduction to the Preliminary Design of Subsonic General Aviation and Transport Aircraft, with Emphasis on Layout, Aerodynamic Design, Propulsion and Performance*. Delft: Delft University Press, 1976, ISBN: 978-90-298-2505-4.

[26] S. Corda, *Introduction to Aerospace Engineering with a Flight Test Perspective*. Chichester, West Sussex, United Kingdom: Wiley, 2017, ISBN: 978-1-118-95336-5.

[27] *Jane's Aircraft - Google Drive*, https://drive.google.com/drive/folders/1QmBKXku4yzjpotmyv
(visited on 08/09/2023).

[28] *What is NumPy? — NumPy v1.25 Manual*, https://numpy.org/doc/stable/user/whatisnumpy.h
(visited on 06/24/2023).

Appendices

Appendix A: Python Codes

A. Cessna Method

```python
#!/usr/bin/env python

# Statistical method for estimating aircraft weight components using
# formulas developed by Cessna.
# See:


import numpy as np


## -------------------------1. WING
    ----------------------------------------------
def wing_weight(nz, w0, sw, arw, wing_type):
    """
    Estimate wing weight using cessna formula;
    valid only VH <= 200 KTAS (Maximum level airspeed at S-L in KEAS).

    Keyword Arguments:
    nz         -- Ultimate load factor (= 1.5 x limit load factor);
    w0         -- Design gross weight (lb);
    sw         -- Trapezoidal wing area in (ft2);
    arw        -- Aspect Ratio of wing;
    wing type -- "cantilever" or "strut-braced"
    """

    if wing_type == "cantilever":
        w_w = 0.04674 * (nz * w0) ** 0.397 * sw**0.360 * arw**1.712
    elif wing_type == "strut-braced":
        w_w = 0.002933 * nz**0.611 * sw**1.018 * arw**2.473

```

```python
    return w_w


## -------------------------2. Horizontal Tail -----------------------------------
def ht_weight(w0, sht, arht, tht_max):
    """
    Estimate the horizontal tail weight using Cessna formula.;
    valid for vh <= 200 KTAS.

    Keyword Arguments:
    w0       -- Design gross weight (lbf);
    sht      -- Trapezoidal HT area in ft2;
    arht     -- Aspect Ratio of HT;
    tht_max -- Max root chord thickness of HT in ft;
    """

    w_ht = (
        3.184 * w0**0.887 * sht**0.101 * arht**0.138 / (174.04 * tht_max**0.223)
    )

    return w_ht


## -------------------------3. Vertical Tail -------------------------------------
def vt_weight(f_tail, w0, svt, arvt, tvt_max, sweep4_vt):
    """
    Estimate the horizontal tail weight using Cessna formula.

    Keyword Arguments:
    f_tail      -- 0 for conventional tail, 1 for T-tail;
    w0          -- Design gross weight (lbf);
    svt         -- Trapezoidal VT area in ft2;
    ar_vt       -- AR of VT;
    tvt_max     -- Max root chord thickness of VT in ft;
    vt_sweep_4 -- VT sweep at 25% MGC.
    """

```

```
    w_vt = (
        (1 + 0.2 * f_tail)
        * (1.68 * w0**0.567 * svt**0.1249 * arvt**0.482)
        / (639.95 * tvt_max**0.747 * np.cos(sweep4_vt) ** 0.882)
    )

    return w_vt


## -------------------------4. Fuselage ------------------------------------------
def fus_weight(w0, rmax, lfs, wing_pos, nocc=1):
    """
    Estimate fuselage weight using cessna formula;
    valid for vh <= 200 KTAS.

    Keyword Arguments:
    w0       -- Design gross weight (lbf);
    rmax     -- Fuselage maximum perimeter in ft;
    lfs      -- Length of fuselage structure (forward bulkhead to aft frame) in ft;
    wing_pos -- Wing position "low" or "high";
    nocc     -- (default 1 for UAV) Number of occupants (crew and passengers).
    """

    if wing_pos == "low":
        w_fus = 0.04682 * w0**0.692 * nocc**0.374 * lfs ** (0.590 / 100)
    elif wing_pos == "high":
        w_fus = (
            14.86 * w0**0.144 * (lfs / rmax) ** 0.778 * lfs**0.383 * nocc**0.455
        )

    return w_fus


## -------------------------5. Main Landing Gear --------------------------------
```

```
def mnlg_weight(w0, wl, lm, nz, wing_pos, nl=4.5):
    """
    Estimate main LG weight using usaf Cesna formula;
    valid for vh <= 200 KTAS.


    Keyword Arguments:
    w0       -- Design gross weight (lbf);
    wl       -- Design landing weight in lbf;
    lm       -- Length of the main landing gear shock strut in ft;
    nz       -- Ultimate load factor (= 1.5 x limit load factor);
    wing_pos -- "low" or "high";
    nl       -- (default 4.5) Ultimate landing load factor (typical
   range 3.5-5.5).
    """
    w_mnlg = (
        6.2
        + 0.0143 * w0
        + 0.362 * wl**0.417 * nl**0.950 * lm**0.183
        + 0.007157 * wl**0.749 * nz * lm * 0.788
    )

    return w_mnlg


## -------------------------6. Nose/Tail Landing Gear
   ----------------------------
## NONE
# # Included in the main landing gear


## -------------------------7. Nacelle/Cowling Weight
   ----------------------------
## NONE
def nac_weight(pmax, n_eng, piston_engine_type):
    """
    Estimate the weight of nacelles or cowlings using Cessna formula.

    Keyword Arguments:
    pmax         -- Maximum rated power per engine in BHP or ESHP
```

```
    n_eng       -- Number of engines
    engine_type -- "rpe" (radial piston engine) or
                   "hop" (horizontally opposed piston engine)
    """
    if piston_engine_type == "rpe":
        w_nac = 0.37 * pmax * n_eng
    elif piston_engine_type == "hop":
        w_nac = 0.24 * pmax * n_eng

    return w_nac


## -------------------------8. Uninstalled (Dry) Engine --------------------------
def engine_dry_weight(p_or_t_max, engine_type):
    """
    Estimate uninstalled engine weight when the actual weight are not known.

    Keyword Arguments:
    p_or_t_max  -- If prop engine, then Pmax (BHP), if jet then Tmax (lbf)
    engine_type -- "piston", "prop", "jet"
    """

    if engine_type == "piston":
        w_eng = 50.56 + 1.352 * p_or_t_max
    elif engine_type == "prop":
        w_eng = 71.65 + 0.3658 * p_or_t_max
    elif engine_type == "jet":
        w_eng = 295.5 + 0.1683 * p_or_t_max

    return w_eng


## -------------------------9. Installed Engine ----------------------------------
def engine_installed_weight(pmax, wprop, n_eng, w_nac):
    """
    Estimate installed engine weight using Cessna formula.
```

```

    Keyword Arguments:
    pmax  -- Maximum rated power per engine in BHP;
    wprop -- Propeller weight (set wprop=0 for jet);;
    n_eng -- Number of engines;
    w_nac -- Predicted weight of all engine nacelles in lbf.
    """
    w_ei = (1.3 * pmax + wprop) * n_eng + w_nac

    return w_ei


## ------------------------10. Fuel System
    --------------------------------------
def fuel_sys_weight(qtot, fuel_sys_type):
    """
    Estimate installed fuel weight using Cessna formula.

    Keyword Arguments:
    qtot          -- Total fuel quantity in US gallons;
    fuel_sys_type -- "avgas-no-tip", "jeta-no-tip",
                     "avgas-tip", "jet-a-tip"
    """
    if fuel_sys_type == "avgas-no-tip":
        w_fs = 0.40 * qtot
    elif fuel_sys_type == "jeta-no-tip":
        w_fs = 0.4467 * qtot
    elif fuel_sys_type == "avgas-tip":
        w_fs = 0.70 * qtot
    elif fuel_sys_type == "jet-a-tip":
        w_fs = 0.7817 * qtot

    return w_fs


## ------------------------11. Flight Control System
    ----------------------------
def fcs_weight(w0):
    """
    Estimate flight control system using cessna formula
```

```
    (manual control system).

    Keyword Arguments:
    w0 -- Design gross weight (lbf);
    """
    w_ctrl = 0.0168 * w0

    return w_ctrl


## -------------------------12. Hydraulic
    -----------------------------------------
def hydraulic_weight(w0):
    """
    Estimate hydraulic system.

    The weight of the hydraulic systems for the flight controls
    is usually included in the Flight Control System,
    so the following expression is for the other components.

    Keyword Arguments:
    w0 -- Design gross weight (lbf);
    """

    w_hyd = 0.001 * w0

    return w_hyd


## -------------------------13. Avionics Systems
    ----------------------------------
def avionics_weight(w_uav):
    """
    The expression below assumes analog dials and overpredicts
    the weight of modern electronic flight instrument
    system (EFIS).

    Keyword Arguments:
    w_uav -- Weight of the uninstalled avionics in lbf
    """
```

```python

    w_av = 2.11 * w_uav**0.933

    return w_uav


## -------------------------14. Electrical Systems
    --------------------------------
def electric_weight(w0):
    """
    Comprises all electric wiring for lights, instruments,
    avionics, fuel system, climate control, and so forth.

    Using Cessna formula.

    Keyword Arguments:
    w0 -- Design gross weight (lbf);
    """

    w_el = 0.0268 * w0

    return w_el


## ---------15. Air Conditioning, Pressurization, and Antiicing
    -------------------
def aircond_weight(w0, n_occ, wav, mach):
    """
    Air conditioning includes both cooling and heating of
    the cabin volume. Pressurization system usually consists
    of various equipment (outflow and relief valves, pressure
    regulators, compressors, heat exchangers, and ducting).
    Antiicing systems included are either pneumatic inflat-
    able boots or bleed air heated elements.

    Keyword Arguments:
    w0    -- Design gross weight (lbf);
    n_occ -- Number of occupants (crew and passengers);
    wav   -- Predicted weight of the avionics installation;
    mach  -- Mach number.
```

```
    """

    w_ac = 0.265 * w0**0.52 * n_occ**0.68 * wav**0.17 * mach**0.08

    return w_ac


## -------------------------16. Furnishing
   ----------------------------------------
def furn_weight(n_occ, w0):
    """
    Includes seats, insulation, sound proofing, lighting,
    galley, lavatory, overhead hat-racks, emergency equip-
    ment, and associated electric systems.

    Using Cessna formula.

    Keyword Arguments:
    n_occ -- Number of occupants (crew and passengers);
    w0    -- Design gross weight (lbf).
    """

    w_furn = 0.0412 * n_occ**1.145 * w0**0.489

    return w_furn
```

A. Raymer Method

```
#!/usr/bin/env python

# Statistical method for estimating aircraft weight components using
# formulas developed by Raymer.
# See: Aircraft Design: A Conceptual Approach by Daniel P. Raymer,
# Section 15.3.3
#
# General Aviation Weights (British Units, Results i n Pounds).

import numpy as np


```

```
## -------------------------1. WING
    ----------------------------------------------
def wing_weight(sw, wfw, arw, sweep4_wing, q, tr_wing, t2c_w, nz, w0):
    """
    Estimate wing weight (lb) using Raymer formula (Eq. 15.46);

    NOTE: ignore second term if wfw = 0;

    Keyword Arguments:
    sw            -- Trapezoidal wing area in (ft2);
    wfw           -- Weight of fuel in wing in lb. (If wfw=0 then let
                     w^0.0035=1);
    arw           -- Aspect Ratio of wing;
    sweep4_wing   -- Wing sweep at 25% MGC;
    q             -- Dynamic pressure at cruise (lbf/ft2);
    tr_wing       -- Taper ratio of wing;
    t2c_w         -- Wing thickness-to-chord ratio (maximum);
    nz            -- Ultimate load factor (= 1.5 x limit load factor);
    w0            -- Design gross weight (lb).
    """
    if wfw == 0:
        wfw_ = 1
    else:
        wfw_ = wfw
    w_w = (
        0.036
        * sw**0.758
        * wfw_**0.0035
        * (arw / np.cos(sweep4_wing) ** 2) ** 0.6
        * q**0.006
        * tr_wing**0.004
        * (100 * t2c_w / np.cos(sweep4_wing)) ** (-0.3)
        * (nz * w0) ** 0.49
    )

    return w_w


## -------------------------2. Horizontal Tail
    -----------------------------------
```

```python
def ht_weight(nz, w0, q, sht, t2c_wing, arw, sweep4_ht, sweep4_wing,
    tr_ht):
    """
    Estimate the horizontal tail weight using Raymer formula.

    Keyword Arguments:
    nz          -- Ultimate load factor (= 1.5 x limit load factor);
    w0          -- Design gross weight (lb);
    q           -- Dynamic pressure at cruise (lbf/ft2);
    sht         -- Trapezoidal HT area in ft2;
    t2c_wing    -- Wing thickness-to-chord ratio (maximum);
    arw         -- Aspect Ratio of wing;
    sweep4_ht   -- HT sweep ratio at 25% MGC;
    tr_ht       -- HT taper ratio.
    """

    w_ht = (
        0.016
        * (nz * w0) ** 0.414
        * q**0.168
        * sht**0.896
        * (100 * t2c_wing / np.cos(sweep4_wing)) ** -0.12
        * (arw / np.cos(sweep4_ht) ** 2) ** 0.043
        * tr_ht**-0.02
    )

    return w_ht


## -------------------------3. Vertical Tail
    -------------------------------------
def vt_weight(f_tail, nz, w0, q, svt, t2c_wing, sweep4_vt, arw, tr_vt)
    :
    """
    Estimate the horizontal tail weight using Raymer formula.

    Keyword Arguments:
    f_tail      -- 0 for conventional tail, 1 for T-tail;
    nz          -- Ultimate load factor (= 1.5 x limit load factor);
    w0          -- Design gross weight (lb);
```

```
    q           -- Dynamic pressure at cruise (lbf/ft2);
    svt         -- Trapezoidal VT area in ft2;
    t2c_wing    -- Wing thickness-to-chord ratio (maximum);
    sweep4_vt   -- VT sweep at 25% MGC.
    ar_vt       -- AR of VT;
    tr_vt       -- VT taper ratio.
    """
    if tr_vt < 0.2:
        tr_vt_ = 0.2
    else:
        tr_vt_ = tr_vt

    w_vt = (
        0.073
        * (1 + 0.2 * f_tail)
        * (nz * w0) ** 0.376
        * q**0.122
        * svt**0.873
        * (100 * t2c_wing / np.cos(sweep4_vt)) ** (-0.49)
        * (arw / np.cos(sweep4_vt) ** 2) ** 0.357
        * tr_vt_**0.039
    )

    return w_vt


## -------------------------4. Fuselage
   ------------------------------------------
def fus_weight(sfus, nz, w0, lht, lfs, dfs, q, vp, delta_p=8):
    """
    Estimate fuselage weight using Raymer formula.

    Keyword Arguments:
    sfus    -- Fuselage wetted area in ft2
    nz      -- Ultimate load factor (= 1.5 x limit load factor);
    w0      -- Design gross weight (lb);
    lht     -- Horizontal tail arm, from wing c/4 to HT c/4 in ft;
    lfs     -- Length of fuselage structure (forward bulkhead to aft
   frame) in ft;
    dfs     -- Depth of fuselage structure in ft;
```

```
    q       -- Dynamic pressure at cruise (lbf/ft2);
    vp      -- Volume of pressurized cabin section in ft3;
    delta_p -- (default 8 psi) Cabin pressure differential, in psi (
   typically 8 psi).
    """

    w_fus = (
        0.052
        * sfus**1.086
        * (nz * w0) ** 0.177
        * lht**-0.051
        * (lfs / dfs) ** -0.072
        * q**0.241
        + 11.9 * (vp * delta_p) ** 0.271
    )

    return w_fus


## -------------------------5. Main Landing Gear
   ---------------------------------
def mnlg_weight(wl, lm, nl=4.5):
    """
    Estimate main LG weight using Raymer formula.

    Keyword Arguments:
    wl -- Design landing weight in lb;
    lm -- Length of the main landing gear shock strut in ft;
    nl -- (default 4.5) Ultimate landing load factor (typical range
   3.5-5.5).
    """

    w_mnlg = 0.095 * (nl * wl) ** 0.768 * lm**0.409

    return w_mnlg


## -------------------------6. Nose/Tail Landing Gear
   ----------------------------
def nlg_weight(wl, ln, nl=4.5):
```

```
    """
    Estimate nose LG weight using Raymer formula.

    TODO: (reduce total landing gear weight by 1.4%
    of TOGW if nonretractable )

    Keyword Arguments:
    wl -- Design landing weight in lb;
    ln -- Length of the nose landing gear shock strut in ft;
    nl -- (default 4.5) Ultimate landing load factor (typical range
    3.5-5.5).
    """

    w_nlg = 0.125 * (nl * wl) ** 0.566 * ln**0.845

    return w_nlg


## -------------------------7. Nacelle/Cowling Weight
    ----------------------------
def nac_weight():
    """
    Dummy equation to calculate nacelle weight;
    Nacelle weight included in installed engine.
    """

    return 0


## -------------------------8. Uninstalled (Dry) Engine
    --------------------------
def engine_dry_weight(p_or_t_max, engine_type):
    """
    Estimate uninstalled engine weight when the actual weight are not
    known.

    Keyword Arguments:
    p_or_t_max  -- If prop engine, then Pmax (BHP), if jet then Tmax (
    lbf)
    engine_type -- "piston", "prop", "jet"
```

```
    """

    if engine_type == "piston":
        w_eng = 50.56 + 1.352 * p_or_t_max
    elif engine_type == "prop":
        w_eng = 71.65 + 0.3658 * p_or_t_max
    elif engine_type == "jet":
        w_eng = 295.5 + 0.1683 * p_or_t_max

    return w_eng


## -------------------------9. Installed Engine
    ----------------------------------
def engine_installed_weight(w_eng, n_eng):
    """
    Estimate installed engine weight using Raymer formula.
    (includes propeller and engine mounts).

    Keyword Arguments:
    w_eng -- Weight of each uninstalled engine in lb;
    n_eng -- Number of engines.
    """

    w_ei = 2.575 * w_eng**0.922 * n_eng

    return w_ei


## ------------------------10. Fuel System
    ---------------------------------------
def fuel_sys_weight(qtot, qint, n_tank, n_eng):
    """
    Estimate fuel system weight using Raymer formula.

    Keyword Arguments:
    qtot   -- Total fuel quantity in US gallons;
    qint   -- Fuel quantity in integral tanks in US gallons;
    n_tank -- Number of fuel tanks;
    n_eng  -- Number of engines.
```

```
    """

    w_fs = (
        2.49
        * qtot**0.726
        * (qtot / (qtot + qint)) ** 0.363
        * n_tank**0.242
        * n_eng**0.157
    )

    return w_fs


## -------------------------11. Flight Control System
    -----------------------------
def fcs_weight(lfs, bw, nz, w0):
    """
    Estimate flight control system using Raymer formula.

    Keyword Arguments:
    lfs -- Length of fuselage structure (forward bulkhead to aft frame
    ) in ft;
    bw  -- Wingspan in ft;
    nz  -- Ultimate load factor (= 1.5 x limit load factor);
    w0  -- Design gross weight (lb).
    """
    w_ctrl = 0.053 * lfs**1.536 * bw**0.371 * (nz * w0 * 1e-4) ** 0.80

    return w_ctrl


## -------------------------12. Hydraulic
    -----------------------------------------
def hydraulic_weight(w0, hyd_type, mach_max):
    """
    Estimate hydraulic system using Raymer formula.

    The weight of the hydraulic systems for the flight controls
    is usually included in the Flight Control System,
    so the following expression is for the other components.
```

```

    Keyword Arguments:
    w0        -- Design gross weight (lb);
    hyd_type -- "low", "medium", "high", "light" in term of speed
    in subsonic regime;
    mach      -- Mach number (design maximum)
    """

    # w_hyd = 0.001 * w0 (from Snorri)

    if hyd_type == "low":
        kh = 0.05
    elif hyd_type == "medium":
        kh = 0.11
    elif hyd_type == "high":
        kh = 0.12
    elif hyd_type == "light":
        kh = 0.013
        mach_max = 0.1

    w_hyd = kh * w0**0.8 * mach_max**0.5

    return w_hyd


## -------------------------13. Avionics Systems
    ----------------------------------
def avionics_weight(w_uav):
    """
    The expression below assumes analog dials and overpredicts
    the weight of modern electronic flight instrument
    system (EFIS).

    Keyword Arguments:
    w_uav -- Weight of the uninstalled avionics in lb
             (typically = 800-1400 lb)
    """

    w_av = 2.117 * w_uav**0.933

```

```
    return w_uav


## -------------------------14. Electrical Systems
    --------------------------------
def electric_weight(wfs, wav):
    """
    Comprises all electric wiring for lights, instruments,
    avionics, fuel system, climate control, and so forth.

    Using Raymer/USAF formula.

    Keyword Arguments:
    wfs -- Predicted fuel system weight;
    wav -- Predicted weight of the avionics installation;
    """

    w_el = 12.57 * (wfs + wav) ** 0.51

    return w_el


## ---------15. Air Conditioning, Pressurization, and Antiicing
    -------------------
def aircond_weight(w0, n_occ, wav, mach_max):
    """
    Air conditioning includes both cooling and heating of
    the cabin volume. Pressurization system usually consists
    of various equipment (outflow and relief valves, pressure
    regulators, compressors, heat exchangers, and ducting).
    Antiicing systems included are either pneumatic inflat-
    able boots or bleed air heated elements.

    Keyword Arguments:
    w0        -- Design gross weight (lb);
    n_occ     -- Number of occupants (crew and passengers);
    wav       -- Predicted weight of the avionics installation;
    mach_max  -- Maximum design Mach number.
    """

```

```
    w_ac = 0.265 * w0**0.52 * n_occ**0.68 * wav**0.17 * mach_max**0.08

    return w_ac


## -------------------------16. Furnishing
    ----------------------------------------
def furn_weight(w0):
    """

    Includes seats, insulation, sound proofing, lighting,
    galley, lavatory, overhead hat-racks, emergency equip-
    ment, and associated electric systems.

    Using Raymer formula.

    Keyword Arguments:
    w0     -- Design gross weight (lb).
    """

    w_furn = 0.0582 * w0 - 65

    return w_furn
```

A. Torenbeek Method

```
#!/usr/bin/env python

# Statistical method for estimating aircraft weight components using
# formulas developed by Torrenbeek.
# See:

import numpy as np


## -------------------------1. WING
    ----------------------------------------------
def wing_weight(w0, bw, sw, sweep2_wing, nz, tw_max):
    """
    Estimate wing weight using Torrenbeek formula;
```

```

    Eq. 8.12

    Keyword Arguments:
    w0            -- Design gross weight (lbf);
    bw            -- Wingspan in (ft);
    SW            -- Trapezoidal wing area in ft2
    wing_sweep_2 -- Wing sweep at 50% MGC;
    nz            -- Ultimate load factor (= 1.5 x limit load factor);
    tw_max        -- Max thickness of the wing root chord in ft.
    """

    w_w = (
        0.00125
        * w0
        * (bw / np.cos(sweep2_wing)) ** 0.75
        * (1 + np.sqrt(6.3 * np.cos(sweep2_wing) / bw))
        * nz**0.55
        * (bw * sw / (tw_max * w0 * np.cos(sweep2_wing))) ** 0.30
    )

    return w_w


## -------------------------2. Horizontal Tail
   -----------------------------------
def emp_weight(nz, sht, svt):
    """
    Estimate the emmpenage weight using Torenbeek formula.

    Keyword Arguments:
    nz  -- Ultimate load factor (= 1.5 x limit load factor);
    sht -- Trapezoidal HT area in ft2;
    svt -- Trapezoidal VT area in ft2;
    """

    w_emp = 0.04 * (nz * (sht + svt) ** 2) ** 0.75

    return w_emp

```

```

## -------------------------3. Vertical Tail -------------------------------------
# Included in empenage

## -------------------------4. Fuselage ------------------------------------------
# Torenbeek: No expression given for GA aircraft


## -------------------------5. Main Landing Gear ---------------------------------
def mnlg_weight(w0, wing_pos, lg_type, ac_class):
    """
    Estimate main LG weight using torrenbeek formula.

    Keyword Arguments:
    w0        -- Design gross weight (lbf);
    wing_pos -- "low" or "high"
    lg_type  -- "fixed" or "retract"
    ac_class -- "bizjet" or "civil"
    """
    if ac_class == "bizjet":
        A = 33
        B = 0.04
        C = 0.021
        D = 0
    elif ac_class == "civil":
        if lg_type == "fixed":
            A = 20
            B = 0.10
            C = 0.019
            D = 0
        elif lg_type == "retract":
            A = 40
            B = 0.16
            C = 0.019
            D = 1.5 * 1e-5

    if wing_pos == "low":
```

```
        w_mnlg = A + B * w0**0.75 + C * w0 + D * w0**1.5
    elif wing_pos == "high":
        w_mnlg = 1.08 * (A + B * w0**0.75 + C * w0 + D * w0**1.5)

    return w_mnlg


## -------------------------6. Nose/Tail Landing Gear
    ----------------------------
def nlg_weight(w0, wing_pos, lg_type, ac_class):
    """
    Estimate tail LG weight using torrenbeek formula.

    Keyword Arguments:
    w0       -- Design gross weight (lbf);
    wing_pos -- "low" or "high"
    lg_type  -- "fixed" or "retract"
    ac_class -- "bizjet" or "civil"
    """
    if ac_class == "bizjet":
        A = 12
        B = 0.06
        C = 0
        D = 0
    elif ac_class == "civil":
        if lg_type == "fixed":
            A = 25
            B = 0
            C = 0.0024
            D = 0
        elif lg_type == "retract":
            A = 20
            B = 0.10
            C = 0
            D = 2.0 * 1e-6

    if wing_pos == "low":
        w_nlg = A + B * w0**0.75 + C * w0 + D * w0**1.5
    elif wing_pos == "high":
        w_nlg = 1.08 * (A + B * w0**0.75 + C * w0 + D * w0**1.5)
```

```

    return w_nlg


def tlg_weight(w0, wing_pos, lg_type, ac_class):
    """
    Estimate tail LG weight using torrenbeek formula.

    Keyword Arguments:
    w0       -- Design gross weight (lbf);
    wing_pos -- "low" or "high"
    lg_type  -- "fixed" or "retract"
    ac_class -- "bizjet" or "civil"
    """
    if ac_class == "bizjet":
        A = 0
        B = 0
        C = 0
        D = 0
    elif ac_class == "civil":
        if lg_type == "fixed":
            A = 9
            B = 0
            C = 0.0024
            D = 0
        elif lg_type == "retract":
            A = 5
            B = 0
            C = 0.0031
            D = 0

    if wing_pos == "low":
        w_tlg = A + B * w0**0.75 + C * w0 + D * w0**1.5
    elif wing_pos == "high":
        w_tlg = 1.08 * (A + B * w0**0.75 + C * w0 + D * w0**1.5)

    return w_tlg

```

```
## -------------------------7. Nacelle/Cowling Weight
    ----------------------------
def nac_weight(p_or_t_max, n_eng, nac_engine_type):
    """
    Estimate the weight of nacelles or cowlings using Torrenbeek
    formula.
    For prop engines set tmax = 0, and for jet engines, set pmax=0.

    Keyword Arguments:
    p_or_tmax       -- Maximum rated power per engine in BHP or ESHP/
                       Maximum rated thrust per engine in lbf;;
    n_eng           -- Number of engines
    nac_engine_type -- "stp" (Single-engine tractor propeller), "
    multihop", "rp" (radial piston),
                       "turboprop", "podjet", "hbpr"
    """
    if nac_engine_type == "stp":
        w_nac = 2.5 * np.sqrt(p_or_t_max)
    elif nac_engine_type == "multihop":
        w_nac = 0.32 * p_or_t_max * n_eng
    elif nac_engine_type == "multirp":
        w_nac = 0.045 * p_or_t_max**1.25 * n_eng
    elif nac_engine_type == "turboprop":
        w_nac = 0.14 * p_or_t_max * n_eng
    elif nac_engine_type == "podjet":
        w_nac = 0.055 * p_or_t_max
    elif nac_engine_type == "hbpr":
        w_nac = 0.065 * p_or_t_max

    return w_nac


## -------------------------8. Uninstalled (Dry) Engine
    --------------------------
def engine_dry_weight(p_or_t_max, engine_type):
    """
    Estimate uninstalled engine weight when the actual weight are not
    known.

    Keyword Arguments:
```

```
    p_or_t_max  -- If prop engine, then Pmax (BHP), if jet then Tmax (
    lbf)
    engine_type -- "piston", "prop", "jet"
    """

    if engine_type == "piston":
        w_eng = 50.56 + 1.352 * p_or_t_max
    elif engine_type == "prop":
        w_eng = 71.65 + 0.3658 * p_or_t_max
    elif engine_type == "jet":
        w_eng = 295.5 + 0.1683 * p_or_t_max

    return w_eng


## -------------------------9. Installed Engine
    ----------------------------------
def engine_installed_weight(w_eng, wprop, n_eng, pmax, w_nac):
    """
    Estimate installed engine weight using Torrenbeek formula.

    Table 8.9 of Torrenbeek's book.

    Keyword Arguments:
    w_eng -- Weight of each uninstalled engine in lbf;
    wprop -- Propeller weight (set wprop=0 for jet);
    n_eng -- Number of engines;
    pmax  -- Maximum rated power per engine in BHP;
    w_nac -- Predicted weight of all engine nacelles in lbf.
    """

    w_ei = (w_eng + wprop) * n_eng + 1.03 * n_eng**0.3 * pmax**0.7 +
    w_nac

    return w_ei


## ------------------------10. Fuel System
    ---------------------------------------
def fuel_sys_weight(
```

```
    qtot,
    engine_conf,
    n_tank=1,
    n_eng=1,
):
    """
    Estimate fuel system weight using torrenbeek formula.

    See Torrenbeek (P 286).

    Keyword Arguments:
    qtot        -- Total fuel quantity in US gallons;
    engine_conf -- "single-piston", "multi-piston", "turbo-integral"
                   "turbo-bladder"
    n_tank      -- (default 1) Number of fuel tanks;
    n_eng       -- (default 1) Number of engines.
    """

    if engine_conf == "single-piston":
        w_fs = 2 * qtot**0.667
    elif engine_conf == "multi-piston":
        w_fs = 4.5 * qtot**0.60
    elif engine_conf == "turbo-integral":
        w_fs = 80 * (n_eng + n_tank - 1) + 15 * n_tank**0.5 * qtot**0.333
    elif engine_conf == "turbo-bladder":
        w_fs = 3.2 * qtot**0.727

    return w_fs


## -------------------------11. Flight Control System -----------------------------
def fcs_weight(w0, ctrl_sys_type):
    """
    Estimate flight control system using Torrenbeek formula.

    Keyword Arguments:
    w0            -- Design gross weight (lbf);
    ctrl_sys_type -- "manual-single", "manual", "powered".
```

```
    """

    if ctrl_sys_type == "manual-single":
        w_ctrl = 0.23 * w0**0.667
    elif ctrl_sys_type == "manual":
        w_ctrl = 0.44 * w0**0.667
    elif ctrl_sys_type == "powered":
        w_ctrl = 0.64 * w0**0.667

    return w_ctrl


## -------------------------12. Hydraulic
    -----------------------------------------
def hydraulic_weight(w0):
    """
    Estimate hydraulic system.

    Torrenbeek doesn't provide estimation
    of hydraulic so here snorri's method used.

    The weight of the hydraulic systems for the flight controls
    is usually included in the Flight Control System,
    so the following expression is for the other components.

    Keyword Arguments:
    w0 -- Design gross weight (lbf);
    """

    w_hyd = 0.001 * w0

    return w_hyd


## -------------------------13. Avionics Systems
    ----------------------------------
def avionics_weight(w_uav):
    """
    The expression below assumes analog dials and overpredicts
    the weight of modern electronic flight instrument
```

```
    system (EFIS).

    Torrenbeek doesn't provide estimation
    of hydraulic so here snorri's method used.

    Keyword Arguments:
    w_uav -- Weight of the uninstalled avionics in lbf
    """

    w_av = 2.11 * w_uav**0.933

    return w_uav


## -------------------------14. Electrical Systems
    --------------------------------
def electric_weight(w0, wu, whyd):
    """
    Comprises all electric wiring for lights, instruments,
    avionics, fuel system, climate control, and so forth.

    Using Torrenbeek.

    Keyword Arguments:
    w0   -- Design gross weight (lb);
    wu   -- Target useful load in lb;
    whyd -- Predicted weight of the hydraulics system in lb.
    """

    w_el = 0.0078 * (w0 - wu) ** 1.2 - whyd

    return w_el


## ---------15. Air Conditioning, Pressurization, and Antiicing
    -------------------
def aircond_weight(w0, n_occ, wav, mach):
    """
    Air conditioning includes both cooling and heating of
    the cabin volume. Pressurization system usually consists
```

```
    of various equipment (outflow and relief valves, pressure
    regulators, compressors, heat exchangers, and ducting).
    Antiicing systems included are either pneumatic inflat-
    able boots or bleed air heated elements.

    Torrenbeek doesn't provide estimation
    of hydraulic so here snorri's method used.

    Keyword Arguments:
    w0     -- Design gross weight (lbf);
    n_occ -- Number of occupants (crew and passengers);
    wav    -- Predicted weight of the avionics installation;
    mach   -- Mach number.
    """

    w_ac = 0.265 * w0**0.52 * n_occ**0.68 * wav**0.17 * mach**0.08

    return w_ac


## -------------------------16. Furnishing
   ----------------------------------------
# None
def furn_weight():
    """
    Torrenbeek doesn't provide estimation
    of hydraulic so here snorri's method used.
    """
    return 0
```

A. USAF Method

```
#!/usr/bin/env python

# Statistical method for estimating aircraft weight components using
# formulas developed by USAF.
# See:

import numpy as np

```

```

## -------------------------1. WING ----------------------------------------------
def wing_weight(nz, w0, arw, sweep4_wing, sw, tr_wing, t2c_w, vh):
    """
    Estimate wing weight using USAF formula;
    valid for vh <= 300 KTAS.

    Keyword Arguments:
    nz            -- Ultimate load factor (= 1.5 x limit load factor);
    w0            -- Design gross weight (lbf);
    arw           -- Aspect Ratio of wing;
    sweep4_wing   -- Wing sweep at 25% MGC;
    sw            -- Trapezoidal wing area in (ft2);
    tr_wing       -- Taper ratio of wing;
    t2c_w         -- Wing thickness-to-chord ratio (maximum);
    vh            -- Maximum level airspeed at S-L in KEAS;
    """

    if vh > 300:
        vh = 300
        # raise Exception("USAF formula nly valid for vh <= 300 KTAS")

    w_w = (
        96.948
        * (
            (nz * w0 / 1e5) ** 0.65
            * (arw / np.cos(sweep4_wing) ** 2) ** 0.57
            * (sw / 100) ** 0.61
            * ((1 + tr_wing) / (2 * t2c_w)) ** 0.36
            * np.sqrt(1 + vh / 500)
        )
        * 0.993
    )

    return w_w


## -------------------------2. Horizontal Tail -----------------------------------
```

```
def ht_weight(nz, w0, sht, lht, bht, tht_max):
    """
    Estimate the horizontal tail weight using USAF formula.

    Keyword Arguments:
    nz      -- Ultimate load factor (= 1.5 x limit load factor);
    w0      -- Design gross weight (lbf);
    sht     -- Trapezoidal HT area in ft2;
    lht     -- Horizontal tail arm, from wing c/4 to HT c/4 in ft;
    bht     -- HT span in ft;
    tht_max -- Max root chord thickness of HT in ft.
    """

    w_ht = (
        71.927
        * (
            (nz * w0 / 1e5) ** 0.87
            * (sht / 100) ** 1.2
            * (lht / 10) ** 0.483
            * np.sqrt(bht / tht_max)
        )
        * 0.458
    )

    return w_ht


## -------------------------3. Vertical Tail
    -------------------------------------
def vt_weight(f_tail, nz, w0, svt, bvt, tvt_max):
    """
    Estimate the horizontal tail weight using USAF formula.

    Keyword Arguments:
    f_tail  -- 0 for conventional tail, 1 for T-tail;
    nz      -- Ultimate load factor (= 1.5 x limit load factor);
    w0      -- Design gross weight (lbf);
    svt     -- Trapezoidal VT area in ft2;
    bvt     -- VT span in ft;
    tvt_max -- Max root chord thickness of VT in ft.
```

```python
    """

    w_vt = (
        55.786
        * (1 + 0.2 * f_tail)
        * ((nz * w0 / 1e5) ** 0.87 * (svt / 100) * np.sqrt(bvt /
    tvt_max)) ** 0.458
    )

    return w_vt


## -------------------------4. Fuselage
    -----------------------------------------
def fus_weight(nz, w0, lf, wf, df, vh):
    """
    Estimate fuselage weight using usaf formula.

    Keyword Arguments:
    nz -- Ultimate load factor (= 1.5 x limit load factor);
    w0 -- Design gross weight (lbf);
    lf -- Fuselage length in ft;
    wf -- Fuselage max width in ft;
    df -- Fuselage max depth in ft;
    vh -- Maximum level airspeed at S-L in KEAS.
    """

    w_fus = (
        200
        * (
            (nz * w0 / 1e5) ** 0.286
            * (lf / 10) ** 0.857
            * ((wf + df) / 10)
            * (vh / 100) ** 0.338
        )
        ** 1.1
    )

    return w_fus

```

```

## -------------------------5. Main Landing Gear ---------------------------------
def mnlg_weight(wl, lm, nl=4.5):
    """
    Estimate main LG weight using usaf formula;
    valid for vh <= 200 KTAS.

    Keyword Arguments:
    wl -- Design landing weight in lbf;
    lm -- Length of the main landing gear shock strut in ft;
    nl -- (default 4.5) Ultimate landing load factor (typical range 3.5-5.5).
    """

    w_mnlg = 0.054 * (nl * wl) ** 0.684 * lm**0.501

    return w_mnlg


## -------------------------6. Nose/Tail Landing Gear ----------------------------
## NONE
## Included in the main landing gear

## -------------------------7. Nacelle/Cowling Weight ----------------------------
## NONE
## Included in the installed engine


## -------------------------8. Uninstalled (Dry) Engine --------------------------
def engine_dry_weight(p_or_t_max, engine_type):
    """
    Estimate uninstalled engine weight when the actual weight are not known.

    Keyword Arguments:
```

```
    p_or_t_max  -- If prop engine, then Pmax (BHP), if jet then Tmax (lbf)
    engine_type -- "piston", "prop", "jet"
    """

    if engine_type == "piston":
        w_eng = 50.56 + 1.352 * p_or_t_max
    elif engine_type == "prop":
        w_eng = 71.65 + 0.3658 * p_or_t_max
    elif engine_type == "jet":
        w_eng = 295.5 + 0.1683 * p_or_t_max

    return w_eng


## -------------------------9. Installed Engine ----------------------------------
def engine_installed_weight(w_eng, n_eng):
    """
    Estimate installed engine weight using USAF formula.

    Keyword Arguments:
    w_eng -- Weight of each uninstalled engine in lbf;
    n_eng -- Number of engines;
    """

    w_ei = 2.575 * w_eng * 0.922 * n_eng

    return w_ei


## ------------------------10. Fuel System ---------------------------------------
def fuel_sys_weight(qtot, qint, n_tank, n_eng):
    """
    Estimate fuel system weight using USAF formula.

    Keyword Arguments:
    qtot         -- Total fuel quantity in US gallons;
    qint         -- Fuel quantity in integral tanks in US gallons;
```

```python
    n_tank        -- Number of fuel tanks;
    n_eng         -- Number of engines.
    """

    w_fs = 2.49 * (
        qtot**0.6 * (qtot / (qtot + qint)) ** 0.3 * n_tank**0.2 * n_eng**0.13
    )

    return w_fs


## -------------------------11. Flight Control System -----------------------------
def fcs_weight(w0, ctrl_sys_type):
    """
    Estimate flight control system using USAF formula.

    Keyword Arguments:
    w0             -- Design gross weight (lbf).
    ctrl_sys_type -- "manual", "powered".
    """
    if ctrl_sys_type == "manual":
        w_ctrl = 1.066 * w0**0.626
    elif ctrl_sys_type == "powered":
        w_ctrl = 1.08 * w0**0.7

    return w_ctrl


## -------------------------12. Hydraulic -----------------------------------------
def hydraulic_weight(w0):
    """
    Estimate hydraulic system.

    The weight of the hydraulic systems for the flight controls
    is usually included in the Flight Control System,
    so the following expression is for the other components.

```

```
    Keyword Arguments:
    w0 -- Design gross weight (lbf);
    """

    w_hyd = 0.001 * w0

    return w_hyd


## -------------------------13. Avionics Systems
    ---------------------------------
def avionics_weight(w_uav):
    """
    The expression below assumes analog dials and overpredicts
    the weight of modern electronic flight instrument
    system (EFIS).

    Keyword Arguments:
    w_uav -- Weight of the uninstalled avionics in lbf
    """

    w_av = 2.11 * w_uav**0.933

    return w_uav


## -------------------------14. Electrical Systems
    --------------------------------
def electric_weight(wfs, wav):
    """
    Comprises all electric wiring for lights, instruments,
    avionics, fuel system, climate control, and so forth.

    Using Raymer/USAF formula.

    Keyword Arguments:
    wfs -- Predicted fuel system weight;
    wav -- Predicted weight of the avionics installation;
    """

```

```
    w_el = 12.57 * (wfs + wav) ** 0.51

    return w_el


## ---------15. Air Conditioning, Pressurization, and Antiicing
    -------------------
def aircond_weight(w0, n_occ, wav, mach):
    """
    Air conditioning includes both cooling and heating of
    the cabin volume. Pressurization system usually consists
    of various equipment (outflow and relief valves, pressure
    regulators, compressors, heat exchangers, and ducting).
    Antiicing systems included are either pneumatic inflat-
    able boots or bleed air heated elements.

    Keyword Arguments:
    w0    -- Design gross weight (lbf);
    n_occ -- Number of occupants (crew and passengers);
    wav   -- Predicted weight of the avionics installation;
    mach  -- Mach number.
    """

    w_ac = 0.265 * w0**0.52 * n_occ**0.68 * wav**0.17 * mach**0.08

    return w_ac


## -------------------------16. Furnishing
    ----------------------------------------
def furn_weight(n_crew, qh):
    """
    Includes seats, insulation, sound proofing, lighting,
    galley, lavatory, overhead hat-racks, emergency equip-
    ment, and associated electric systems.

    Using usaf formula.

    Keyword Arguments:
    n_crew -- Number of crew;
```

```
    qh      -- Dynamic pressure at max level airspeed, lbf/ft2.
    """

    w_furn = 34.5 * n_crew * qh**0.25

    return w_furn
```

A. CN-235 Data

```
#!/usr/bin/env python

import numpy as np

# ----------------------------------------------#
# ATR-42 Data #
# ----------------------------------------------#


## ------------------------1. WING
    ---------------------------------------------


# Area (ft^2)
SW = 636.14

# Weight of fuel in wing (lb)
WFW = 11508.13

# Wing Aspect Ratio
ARW = 11.27

# Wing sweep at 25% MGC (radian)
SWEEP4_WING = np.radians(2)  # radian

# Wing sweep at 50% MGC (radian)
SWEEP2_WING = np.radians(0)  # radian

# Dynamic pressure at cruise (lbf/ft2)
VC = 415
RHO_ATR42 = 0.03427
```

```
Q = 0.5 * RHO_ATR42 * VC**2

# Wing Taper Ratio
TR_WING = 0.36

# Wing thickness-to-chord ratio (maximum)
T2C_WING = 0.18

# Design gross weight (lb)
W0 = 35494.424

# Limit Load Factor
LLF = 2.5
# Ultimate load factor (1.5 x limit load factor)
NZ = 3.5

# Max thickness of the wing root chord (ft)
CR = 8.202099738  # ft
TW_MAX = CR * T2C_WING

# Wing type (only for Cessna Method): "cantilever" or  "strut-braced"
WING_TYPE = "strut-braced"

# Wingspan (ft)
BW = 80.61023622  # np.sqrt(ARW * SW)

# Maximum level airspeed at S-L (knot/KEAS)
VH = 300  # KTAS

# For USAF
# Thickness to chord ratio
T2C_W = 0.15


## ------------------------2. Horizontal Tail
   ----------------------------------
# Horizontal Tail Area (ft^2)
SHT = 123.7849698

# HT Aspect Ratio (About half the aspect ratio of the wing)
```

```python
AR_HT = 5.535

# HT sweep at 25% MGC (radian)
SWEEP4_HT = np.radians(2)

# HT Taper ratio
TR_HT = 0.3

# Max root chord thickness of HT (ft)
THT_MAX = 0.16 * 6.56

# For USAF
# Span of HT (ft)
BHT = 34.7
# Span of VT (ft)
BVT = 15.19
# Max root chord thickness of HT (ft)
# THT_MAX = 22 + 4 / 12


# ## -------------------------3. Vertical Tail
    -----------------------------------

# FTail: 0 for conventional tail, 1 for T-tail;
F_TAIL = 0


# Vertical Tail Area (ft^2)
SVT = 119.58

# VT sweep at 25% MGC (radian)
SWEEP4_VT = np.radians(2)

# VT Taper ratio
TR_VT = 0.22

# VT Aspect Ratio
AR_VT = 1.92

# Max root chord thickness of VT (ft)
```

```
TVT_MAX = 0.16 * 7.84

# For USAF
# Max root chord thickness of VT (ft)
# TVT_MAX = 25 + 10 / 12


## -------------------------4. Fuselage
    ------------------------------------------
# Fuselage wetted area in ft2
SFUS = 2271

# Horizontal tail arm, from wing c/4 to HT c/4 (ft)
LHT = 36

# Length of fuselage structure (forward bulkhead to aft frame) (ft)
LFS = 70.20

# Depth of fuselage structure (ft)
DFS = 8.7

# Fuselage Diameter
DFUS = 8.43

# Volume of pressurized cabin section (ft^3)
VP = 0.25 * np.pi * DFUS**2 * LFS

# Cabin pressure differential (psi)
DELTA_P = 7.78

RMAX = np.pi * DFUS

# For USAF
# Fuselage lenght; here we assume LF = LFS, they are usually different
    .
LF = LFS
# Fuselage width (ft)
WF = 9.6
# Fuselage max depth in ft
DF = DFS
```

```

## -------------------------5. Main Landing Gear
    ---------------------------------
# Design landing weight in lb;
WL = W0

# Length of the main landing gear shock strut (ft);
# Shock strut length is the distance between the upper
# attachment point and the center of the wheel axis
LM = 8.2

# (Default 4.5) the ultimate landing factor (typical range 3.5 - 5.5)
NL = 4.5

# For Torrenbeek
# Wing position: "low" or "high"
WING_POS = "high"

# Landing gear type: "fixed" or "retract"
LG_TYPE = "retract"

# Class of a/c: "bizjet" or "civil"
AC_CLASS = "civil"


# For USAF
DL = W0


## -------------------------6. Nose/Tail Landing Gear
    ----------------------------
# Length of the nose landing gear shock strut (ft)
LN = 6.5

## -------------------------7. Nacelle/Cowling Weight
    ----------------------------
# For Cessna
# engine_type -- "rpe" (radial piston engine) or "hop" (horizontally
    opposed piston engine)
PISTON_ENGINE_TYPE = "rpe"
```

```

# Nacelle weight included in installed engine.
# For Torrenbeek
# "stp" (Single-engine tractor propeller), "multihop", "rp" (radial
    piston),
# "turboprop" (Multi-engine turboprop), "podjet" (Podded turbojet or-
    fan), "hbpr" (HBPR turbofan on a pylon)
NAC_ENGINE_TYPE = "turboprop"


## -------------------------8. Uninstalled (Dry) Engine
    --------------------------
# If prop engine, then Pmax (BHP), if jet then Tmax (lbf)
P_OR_T_MAX = 1589.5
ENGINE_TYPE = "prop"
# Ref value= 12200 lb

## -------------------------9. Installed Engine
    ----------------------------------
# Number of engines
N_ENG = 2
# For Torrenbeek
WPROP = 48  # Sin it's a turbofan
PMAX = P_OR_T_MAX * (1.15078 * VH) / 375  # Conert lbf to HP


# Weight of each uninstalled engine in lb
W_ENG = 71.65 + 0.3658 * PMAX


## ------------------------10. Fuel System
    ---------------------------------------
# For Cessna
FUEL_SYS_TYPE = "jeta-no-tip"
#

# Fuel quantity in integral tanks (US gallons)
# An integral fuel tank is defined as primary aircraft structure,
# usually wing or fuselage, that is sealed to contain fuel,
# as opposed to a rubberized fuel cell mounted in aircraft structure
```

```
QINT = 1438.5
# Total fuel quantity in (US gallons)
# QTOT = QINT + Additional Fuel Tank
QTOT = 1438.5

# Number of fuel tanks;
# e.g. wing, center, surge, and vent
N_TANK = 2

# For Torreenbeek
# "single-piston", "multi-piston", "turbo-integral", "turbo-bladder"
ENGINE_CONF = "turbo-bladder"


## -------------------------11. Flight Control System
    -----------------------------
# No new input is needed
# For Torreenbeek
# ctrl_sys_type -- "manual-single", "manual", "powered".
CTRL_SYS_TYPE = "powered"

## -------------------------12. Hydraulic
    -----------------------------------------
# Air speed in subsonic regime: "low", "medium", "high", "light"
HYD_TYPE = "high"

# Mach number (design maximum)
MACH_MAX = 0.45


## -------------------------13. Avionics Systems
    ----------------------------------
# Weight of the uninstalled avionics in lb (typically = 800-1400 lb);
# For smaller a/c for smaller aircraft to weigh 45 to 50 lb.
# This can be found from the avionics manufacturer data sheet.
W_UAV = 1100

## -------------------------14. Electrical Systems
    --------------------------------
# No new input is needed
```

```
# For Torrenbeek
# Target useful load (lb)
WU = 21605.302


## ---------15. Air Conditioning, Pressurization, and Antiicing
    -------------------
# N_CABIN_CREW = 8
# N_FLIGHT_CREW = 2
# N_PAX = 400
N_OCC = 51

# For USAF
N_CREW = 2
## -------------------------16. Furnishing
    ----------------------------------------
# No new input is needed
# For USAF
# Dynamic pressure at max level airspeed, lbf/ft2.
QH = Q
```

A. ATR 42-600 Data

```
#!/usr/bin/env python

import numpy as np

# ----------------------------------------------#
# ATR-42 Data #
# ----------------------------------------------#


## -------------------------1. WING
    ----------------------------------------------


# Area (ft^2)
SW = 586.63

# Weight of fuel in wing (lb)
```

```
WFW = 9920.8

# Wing Aspect Ratio
ARW = 11.07

# Wing sweep at 25% MGC (radian)
SWEEP4_WING = np.radians(2)  # radian

# Wing sweep at 50% MGC (radian)
SWEEP2_WING = np.radians(0)  # radian

# Dynamic pressure at cruise (lbf/ft2)
VC = 487.569
RHO_ATR42 = 0.041184
Q = 0.5 * RHO_ATR42 * VC**2

# Wing Taper Ratio
TR_WING = 0.54

# Wing thickness-to-chord ratio (maximum)
T2C_WING = 0.15

# Design gross weight (lb)
W0 = 41005.981

# Limit Load Factor
LLF = 2.5
# Ultimate load factor (1.5 x limit load factor)
NZ = 3.5

# Max thickness of the wing root chord (ft)
CR = 6.692913386  # ft
TW_MAX = CR * T2C_WING

# Wing type (only for Cessna Method): "cantilever" or  "strut-braced"
WING_TYPE = "strut-braced"

# Wingspan (ft)
BW = 84.67847769  # np.sqrt(ARW * SW)

```

```
# Maximum level airspeed at S-L (knot/KEAS)
VH = 248  # KTAS

# For USAF
# Thickness to chord ratio
T2C_W = 0.18


## -------------------------2. Horizontal Tail -----------------------------------
# Horizontal Tail Area (ft^2)
SHT = 228.19

# HT Aspect Ratio (About half the aspect ratio of the wing)
AR_HT = 4.37

# HT sweep at 25% MGC (radian)
SWEEP4_HT = np.radians(2)

# HT Taper ratio
TR_HT = 0.4

# Max root chord thickness of HT (ft)
THT_MAX = 0.16 * 5.31

# For USAF
# Span of HT (ft)
BHT = 23.26
# Span of VT (ft)
BVT = 14.76
# Max root chord thickness of HT (ft)
# THT_MAX = 22 + 4 / 12


# ## -------------------------3. Vertical Tail -----------------------------------

# FTail: 0 for conventional tail, 1 for T-tail;
F_TAIL = 1

```

```

# Vertical Tail Area (ft^2)
SVT = 136.7016623

# VT sweep at 25% MGC (radian)
SWEEP4_VT = np.radians(2)

# VT Taper ratio
TR_VT = 0.29

# VT Aspect Ratio
AR_VT = 1.6

# Max root chord thickness of VT (ft)
TVT_MAX = 0.16 * 11.15

# For USAF
# Max root chord thickness of VT (ft)
# TVT_MAX = 25 + 10 / 12


## -------------------------4. Fuselage
    ------------------------------------------
# Fuselage wetted area in ft2
SFUS = 2271

# Horizontal tail arm, from wing c/4 to HT c/4 (ft)
LHT = 37

# Length of fuselage structure (forward bulkhead to aft frame) (ft)
LFS = 74.47506562

# Depth of fuselage structure (ft)
DFS = 9.32

# Fuselage Diameter
DFUS = 8.43

# Volume of pressurized cabin section (ft^3)
VP = 0.25 * np.pi * DFUS**2 * LFS
```

```python

# Cabin pressure differential (psi)
DELTA_P = 7.78

RMAX = np.pi * DFUS

# For USAF
# Fuselage lenght; here we assume LF = LFS, they are usually different
    .
LF = LFS
# Fuselage width (ft)
WF = 8.530183727
# Fuselage max depth in ft
DF = DFS

## -------------------------5. Main Landing Gear
    ---------------------------------
# Design landing weight in lb;
WL = W0

# Length of the main landing gear shock strut (ft);
# Shock strut length is the distance between the upper
# attachment point and the center of the wheel axis
LM = 7.21


# (default 4.5) Ultimate landing load factor (typical range 3.5-5.5)..
NL = 4.5

# For Torrenbeek
# Wing position: "low" or "high"
WING_POS = "high"

# Landing gear type: "fixed" or "retract"
LG_TYPE = "retract"

# Class of a/c: "bizjet" or "civil"
AC_CLASS = "civil"


```

```
# For USAF
DL = W0


## -------------------------6. Nose/Tail Landing Gear ----------------------------
# Length of the nose landing gear shock strut (ft)
LN = 5.31

## -------------------------7. Nacelle/Cowling Weight ----------------------------
# For Cessna
# engine_type -- "rpe" (radial piston engine) or "hop" (horizontally opposed piston engine)
PISTON_ENGINE_TYPE = "rpe"

# Nacelle weight included in installed engine.
# For Torrenbeek
# "stp" (Single-engine tractor propeller), "multihop", "rp" (radial piston),
# "turboprop" (Multi-engine turboprop), "podjet" (Podded turbojet or-fan), "hbpr" (HBPR turbofan on a pylon)
NAC_ENGINE_TYPE = "turboprop"


## -------------------------8. Uninstalled (Dry) Engine --------------------------
# If prop engine, then Pmax (BHP), if jet then Tmax (lbf)
P_OR_T_MAX = 2337.5
ENGINE_TYPE = "prop"
# Ref value= 12200 lb

## -------------------------9. Installed Engine ----------------------------------
# Number of engines
N_ENG = 2
# For Torrenbeek
WPROP = 44  # Sin it's a turbofan
PMAX = P_OR_T_MAX * (1.15078 * VH) / 375  # Conert lbf to HP

```

```

# Weight of each uninstalled engine in lb
W_ENG = 71.65 + 0.3658 * PMAX


## ------------------------10. Fuel System ---------------------------------------
# For Cessna
FUEL_SYS_TYPE = "jeta-no-tip"
#

# Fuel quantity in integral tanks (US gallons)
# An integral fuel tank is defined as primary aircraft structure,
# usually wing or fuselage, that is sealed to contain fuel,
# as opposed to a rubberized fuel cell mounted in aircraft structure
QINT = 1241.125
# Total fuel quantity in (US gallons)
# QTOT = QINT + Additional Fuel Tank
QTOT = 1241.125

# Number of fuel tanks;
# e.g. wing, center, surge, and vent
N_TANK = 2

# For Torreenbeek
# "single-piston", "multi-piston", "turbo-integral", "turbo-bladder"
ENGINE_CONF = "turbo-bladder"


## -------------------------11. Flight Control System -----------------------------
# No new input is needed
# For Torreenbeek
# ctrl_sys_type -- "manual-single", "manual", "powered".
CTRL_SYS_TYPE = "powered"

## -------------------------12. Hydraulic -----------------------------------------
# Air speed in subsonic regime: "low", "medium", "high", "light"
HYD_TYPE = "high"
```

```

# Mach number (design maximum)
MACH_MAX = 0.5


## -------------------------13. Avionics Systems
    ----------------------------------
# Weight of the uninstalled avionics in lb (typically = 800-1400 lb);
# For smaller a/c for smaller aircraft to weigh 45 to 50 lb.
# This can be found from the avionics manufacturer data sheet.
W_UAV = 1100

## -------------------------14. Electrical Systems
    --------------------------------
# No new input is needed
# For Torrenbeek
# Target useful load (lb)
WU = 25904.316


## ---------15. Air Conditioning, Pressurization, and Antiicing
    -------------------
# N_CABIN_CREW = 8
# N_FLIGHT_CREW = 2
# N_PAX = 400
N_OCC = 48

# For USAF
N_CREW = 2
## -------------------------16. Furnishing
    ----------------------------------------
# No new input is needed
# For USAF
# Dynamic pressure at max level airspeed, lbf/ft2.
QH = Q
```

A. Saab 340 Data

```
#!/usr/bin/env python

```

```
import numpy as np

# ----------------------------------------------#
# Saab 340 Data #
# ----------------------------------------------#


## -------------------------1. WING
    ---------------------------------------------


# Area (ft^2)
SW = 450.03

# Weight of fuel in wing (lb)
WFW = 5687.9

# Wing Aspect Ratio
ARW = 11.0

# Wing sweep at 25% MGC (radian)
SWEEP4_WING = np.radians(2)  # radian

# Wing sweep at 50% MGC (radian)
SWEEP2_WING = np.radians(0)  # radian

# Dynamic pressure at cruise (lbf/ft2)
VC = 478.5
RHO_ATR42 = 0.03427
Q = 0.5 * RHO_ATR42 * VC**2

# Wing Taper Ratio
TR_WING = 0.4

# Wing thickness-to-chord ratio (maximum)
T2C_WING = 0.16

# Design gross weight (lb)
W0 = 28999.606

```

```
# Limit Load Factor
LLF = 2.5
# Ultimate load factor (1.5 x limit load factor)
NZ = 3.5

# Max thickness of the wing root chord (ft)
CR = 6.397637795  # ft
TW_MAX = CR * T2C_WING

# Wing type (only for Cessna Method): "cantilever" or  "strut-braced"
WING_TYPE = "strut-braced"

# Wingspan (ft)
BW = 70.34120735  # np.sqrt(ARW * SW)

# Maximum level airspeed at S-L (knot/KEAS)
VH = 283  # KTAS

# For USAF
# Thickness to chord ratio
T2C_W = 0.16


## -------------------------2. Horizontal Tail
   -----------------------------------
# Horizontal Tail Area (ft^2)
SHT = 156.8301748

# HT Aspect Ratio (About half the aspect ratio of the wing)
AR_HT = 5.87

# HT sweep at 25% MGC (radian)
SWEEP4_HT = np.radians(2)

# HT Taper ratio
TR_HT = 0.3

# Max root chord thickness of HT (ft)
THT_MAX = 0.16 * 5.16

```

```
# For USAF
# Span of HT (ft)
BHT = 30.34
# Span of VT (ft)
BVT = 12.76246719
# Max root chord thickness of HT (ft)
# THT_MAX = 22 + 4 / 12


# ## -------------------------3. Vertical Tail
    -----------------------------------

# FTail: 0 for conventional tail, 1 for T-tail;
F_TAIL = 0


# Vertical Tail Area (ft^2)
SVT = 113.3439767

# VT sweep at 25% MGC (radian)
SWEEP4_VT = np.radians(2)

# VT Taper ratio
TR_VT = 0.22

# VT Aspect Ratio
AR_VT = 1.44

# Max root chord thickness of VT (ft)
TVT_MAX = 0.16 * 8.85

# For USAF
# Max root chord thickness of VT (ft)
# TVT_MAX = 25 + 10 / 12


## -------------------------4. Fuselage
    ------------------------------------------
# Fuselage wetted area in ft2
SFUS = 2421
```

```

# Horizontal tail arm, from wing c/4 to HT c/4 (ft)
LHT = 31.26

# Length of fuselage structure (forward bulkhead to aft frame) (ft)
LFS = 64.73097113

# Depth of fuselage structure (ft)
DFS = 9.25

# Fuselage Diameter
DFUS = 8.82

# Volume of pressurized cabin section (ft^3)
VP = 0.25 * np.pi * DFUS**2 * LFS

# Cabin pressure differential (psi)
DELTA_P = 7.78

RMAX = np.pi * DFUS

# For USAF
# Fuselage lenght; here we assume LF = LFS, they are usually different
    .
LF = LFS
# Fuselage width (ft)
WF = 7.578740157
# Fuselage max depth in ft
DF = DFS

## -------------------------5. Main Landing Gear
    ---------------------------------
# Design landing weight in lb;
WL = W0

# Length of the main landing gear shock strut (ft);
# Shock strut length is the distance between the upper
# attachment point and the center of the wheel axis
LM = 5.41

```

```

# (default 4.5) Ultimate landing load factor (typical range 3.5-5.5).,
NL = 4.5

# For Torrenbeek
# Wing position: "low" or "high"
WING_POS = "low"

# Landing gear type: "fixed" or "retract"
LG_TYPE = "retract"

# Class of a/c: "bizjet" or "civil"
AC_CLASS = "civil"


# For USAF
DL = W0


## -------------------------6. Nose/Tail Landing Gear
    ----------------------------
# Length of the nose landing gear shock strut (ft)
LN = 6.4

## -------------------------7. Nacelle/Cowling Weight
    ----------------------------
# For Cessna
# engine_type -- "rpe" (radial piston engine) or "hop" (horizontally
    opposed piston engine)
PISTON_ENGINE_TYPE = "rpe"

# Nacelle weight included in installed engine.
# For Torrenbeek
# "stp" (Single-engine tractor propeller), "multihop", "rp" (radial
    piston),
# "turboprop" (Multi-engine turboprop), "podjet" (Podded turbojet or-
    fan), "hbpr" (HBPR turbofan on a pylon)
NAC_ENGINE_TYPE = "turboprop"


```

```
## -------------------------8. Uninstalled (Dry) Engine
    --------------------------
# If prop engine, then Pmax (BHP), if jet then Tmax (lbf)
P_OR_T_MAX = 1636.25
ENGINE_TYPE = "prop"
# Ref value= 12200 lb

## -------------------------9. Installed Engine
    ----------------------------------
# Number of engines
N_ENG = 2
# For Torrenbeek
WPROP = 51.3  # Sin it's a turbofan
PMAX = P_OR_T_MAX * (1.15078 * VH) / 375  # Conert lbf to HP


# Weight of each uninstalled engine in lb
W_ENG = 71.65 + 0.3658 * PMAX


## ------------------------10. Fuel System
    ---------------------------------------
# For Cessna
FUEL_SYS_TYPE = "jeta-no-tip"
#

# Fuel quantity in integral tanks (US gallons)
# An integral fuel tank is defined as primary aircraft structure,
# usually wing or fuselage, that is sealed to contain fuel,
# as opposed to a rubberized fuel cell mounted in aircraft structure
QINT = 710.98
# Total fuel quantity in (US gallons)
# QTOT = QINT + Additional Fuel Tank
QTOT = 710.98

# Number of fuel tanks;
# e.g. wing, center, surge, and vent
N_TANK = 2

# For Torreenbeek
```

```
# "single-piston", "multi-piston", "turbo-integral", "turbo-bladder"
ENGINE_CONF = "turbo-integral"


## -------------------------11. Flight Control System
    -----------------------------
# No new input is needed
# For Torreenbeek
# ctrl_sys_type -- "manual-single", "manual", "powered".
CTRL_SYS_TYPE = "powered"

## -------------------------12. Hydraulic
    -----------------------------------------
# Air speed in subsonic regime: "low", "medium", "high", "light"
HYD_TYPE = "high"

# Mach number (design maximum)
MACH_MAX = 0.5


## -------------------------13. Avionics Systems
    ----------------------------------
# Weight of the uninstalled avionics in lb (typically = 800-1400 lb);
# For smaller a/c for smaller aircraft to weigh 45 to 50 lb.
# This can be found from the avionics manufacturer data sheet.
W_UAV = 1100

## -------------------------14. Electrical Systems
    --------------------------------
# No new input is needed
# For Torrenbeek
# Target useful load (lb)
WU = 18999.438


## ---------15. Air Conditioning, Pressurization, and Antiicing
    -------------------
# N_CABIN_CREW = 8
# N_FLIGHT_CREW = 2
# N_PAX = 400
```

```
N_OCC = 34

# For USAF
N_CREW = 2
## -------------------------16. Furnishing ----------------------------------------
# No new input is needed
# For USAF
# Dynamic pressure at max level airspeed, lbf/ft2.
QH = Q
```

A. Component Weight of CN-235

```
import numpy as np
import wanalysis.cessna as wcess
import wanalysis.raymer as wray
import wanalysis.torrenbeek as wtor
import wanalysis.usaf as wusaf

#from example_ac_data import *
#from ATR42_data import *
#from Saab340_data import *
from CN235_Data import *


def weight_components_cessna():
    """
    Estimate weight components using Cessna Method.
    """
    ## -------------------------1. WING ----------------------------------------------
    wing_weight = wcess.wing_weight_cessna(NZ, W0, SW, ARW, WING_TYPE)

    ## -------------------------2. Horizontal Tail -----------------------------------
    ht_weight = wcess.ht_weight_cessna(W0, SHT, AR_HT, THT_MAX)

    # -------------------------3. Vertical Tail -------------------------------------
```

```
 vt_weight = wcess.vt_weight_cessna(F_TAIL, W0, SVT, AR_VT, TVT_MAX, SWEEP4_VT)

 ## -------------------------4. Fuselage ------------------------------------------
 fus_weight = wcess.fus_weight_cessna(W0, RMAX, LFS, WING_POS, N_OCC)

 ## -------------------------5. Main Landing Gear ---------------------------------
 mnlg_weight = wcess.mnlg_weight_cessna(W0, WL, LM, NZ, WING_POS, NL)

 # ## -------------------------6. Nose/Tail Landing Gear ----------------------------
 # # Included in the main landing gear

 # ## -------------------------7. Nacelle/Cowling Weight ----------------------------
 nac_weight = wcess.nac_weight_cessna(PMAX, N_ENG, PISTON_ENGINE_TYPE)

 ## -------------------------8. Uninstalled (Dry) Engine --------------------------
 engine_dry_weight = wcess.engine_dry_weight(P_OR_T_MAX, ENGINE_TYPE)

 # ## -------------------------9. Installed Engine ----------------------------------
 W_NAC = nac_weight
 engine_installed_weight = wcess.install_engine_weight_cessna(PMAX, WPROP, N_ENG, W_NAC)

 ## ------------------------10. Fuel System ---------------------------------------
 fuel_sys_weight = wcess.fuel_sys_weight(QTOT, FUEL_SYS_TYPE)

 ## -------------------------11. Flight Control System -----------------------------
 fcs_weight = wcess.fligthcs_weight_cessna(W0)
```

```

    ## -------------------------12. Hydraulic -----------------------------------------
    hydraulic_weight = wcess.hydraulic_system_weight(W0)

    ## -------------------------13. Avionics Systems ----------------------------------
    avionics_weight = wcess.avionics_system_weight(W_UAV)

    # ## -------------------------14. Electrical Systems --------------------------------
    WAV = avionics_weight
    electric_weight = wcess.electrical_system_cessna(W0)

    ## ---------15. Air Conditioning, Pressurization, and Antiicing -------------------
    aircond_weight = wcess.aircon_pressurization_antiicing(W0, N_OCC, WAV, MACH_MAX)

    ## -------------------------16. Furnishing ----------------------------------------
    furn_weight = wcess.furnishings_weight_cessna(N_OCC, W0)

    emp_weight = ht_weight + vt_weight
    nlg_weight = 0

    weight_components = np.array(
        [
            wing_weight,
            ht_weight,
            vt_weight,
            emp_weight,
            fus_weight,
            mnlg_weight,
            nlg_weight,
            nac_weight,
            engine_dry_weight,
            engine_installed_weight,
            fuel_sys_weight,
            fcs_weight,
```

```
            hydraulic_weight,
            avionics_weight,
            electric_weight,
            aircond_weight,
            furn_weight,
        ]
    )

    return weight_components


def weight_components_raymer():
    """
    Estimate weight components using Raymer Method.
    """
    ## -------------------------1. WING
    ----------------------------------------------
    wing_weight = wray.wing_weight_raymer(
        SW, WFW, ARW, SWEEP4_WING, Q, TR_WING, T2C_WING, NZ, W0
    )

    ## -------------------------2. Horizontal Tail
    -----------------------------------
    ht_weight = wray.ht_weight(
        NZ, W0, Q, SHT, T2C_WING, ARW, SWEEP4_HT, SWEEP4_WING, TR_HT
    )

    ## -------------------------3. Vertical Tail
    -------------------------------------
    vt_weight = wray.vt_weight_raymer(F_TAIL, NZ, W0, Q, SVT, T2C_WING
    , SWEEP4_VT, ARW, TR_VT)

    ## -------------------------4. Fuselage
    ------------------------------------------
    fus_weight = wray.fus_weight_raymer(SFUS, NZ, W0, LHT, LFS, DFS, Q
    , VP, DELTA_P)

    ## -------------------------5. Main Landing Gear
    ---------------------------------
    mnlg_weight = wray.mnlg_weight_raymer(WL, LM, NL)
```

```

## -------------------------6. Nose/Tail Landing Gear ----------------------------
nlg_weight = wray.nslg_weight_raymer(WL, LN, nl=4.5)

## -------------------------7. Nacelle/Cowling Weight ----------------------------
nac_weight = wray.nac_weight()

## -------------------------8. Uninstalled (Dry) Engine --------------------------
engine_dry_weight = wray.engine_dry_weight(P_OR_T_MAX, ENGINE_TYPE
)

## -------------------------9. Installed Engine ----------------------------------
engine_installed_weight = wray.install_engine_weight_raymer(W_ENG,
 N_ENG)

## ------------------------10. Fuel System ---------------------------------------
fuel_sys_weight = wray.fuelsys_weight_raymer(QTOT, QINT, N_TANK,
N_ENG)

## -------------------------11. Flight Control System ----------------------------
fcs_weight = wray.flightcs_weight_raymer(LFS, BW, NZ, W0)

## -------------------------12. Hydraulic -----------------------------------------
hydraulic_weight = wray.hydraulic_weight(W0, HYD_TYPE, MACH_MAX)

## -------------------------13. Avionics Systems ---------------------------------
avionics_weight = wray.avionics_system_weight(W_UAV)

## -------------------------14. Electrical Systems --------------------------------
WFS = fuel_sys_weight
WAV = avionics_weight
```

```
    electric_weight = wray.electrical_system_raymer_usaf(WFS, WAV)

    ## ---------15. Air Conditioning, Pressurization, and Antiicing
    --------------------
    aircond_weight = wray.aircon_pressurization_antiicing(W0, N_OCC,
    WAV, MACH_MAX)

    ## -------------------------16. Furnishing
    ----------------------------------------
    furn_weight = wray.furnishings_weight_raymer(W0)

    emp_weight = ht_weight + vt_weight
    weight_components = np.array(
        [
            wing_weight,
            ht_weight,
            vt_weight,
            emp_weight,
            fus_weight,
            mnlg_weight,
            nlg_weight,
            nac_weight,
            engine_dry_weight,
            engine_installed_weight,
            fuel_sys_weight,
            fcs_weight,
            hydraulic_weight,
            avionics_weight,
            electric_weight,
            aircond_weight,
            furn_weight,
        ]
    )

    return weight_components


def weight_components_torrenbeek():
    ## -------------------------1. WING
    ----------------------------------------------
```

```
    wing_weight = wtor.wing_weight_torenbeek(W0, BW, SW, SWEEP2_WING, NZ, TW_MAX)

    ## -------------------------2. Empenage -----------------------------------

    emp_weight = wtor.emp_weight_torrenbeek(NZ, SHT, SVT)

    ## -------------------------3. Vertical Tail -------------------------------------

    # Vertical Tail + Horizontal Tail = Empenage

    ## -------------------------4. Fuselage ------------------------------------------

    ## -------------------------5. Main Landing Gear ---------------------------------

    mnlg_weight = wtor.mnlg_weight_torrenbeek(W0, WING_POS, LG_TYPE, AC_CLASS)

    ## -------------------------6. Nose/Tail Landing Gear ----------------------------

    nlg_weight = wtor.nslg_weight_torenbeek(W0, WING_POS, LG_TYPE, AC_CLASS)

    ## -------------------------7. Nacelle/Cowling Weight ----------------------------

    nac_weight = wtor.nac_weight(P_OR_T_MAX, N_ENG, NAC_ENGINE_TYPE)

    ## -------------------------8. Uninstalled (Dry) Engine --------------------------

    engine_dry_weight = wtor.engine_dry_weight(P_OR_T_MAX, ENGINE_TYPE)

    ## -------------------------9. Installed Engine ----------------------------------

    W_NAC = nac_weight
    engine_installed_weight = wtor.install_engine_weight_torrenbeek(
        W_ENG, WPROP, N_ENG, PMAX, W_NAC
    )

```

```
    ## ------------------------10. Fuel System ---------------------------------------
    fuel_sys_weight = wtor.fuel_sys_weight(QTOT, ENGINE_CONF, N_TANK, N_ENG)

    ## -------------------------11. Flight Control System -----------------------------
    fcs_weight = wtor.fcs_weight(W0, CTRL_SYS_TYPE)

    ## -------------------------12. Hydraulic -----------------------------------------
    hydraulic_weight = wtor.hydraulic_weight(W0)

    ## -------------------------13. Avionics Systems ----------------------------------
    avionics_weight = wtor.avionics_system_weight(W_UAV)

    ## -------------------------14. Electrical Systems --------------------------------
    WAV = avionics_weight
    WHYD = hydraulic_weight
    electric_weight = wtor.electrical_system_torenbeek(W0, WU, WHYD)

    ## ---------15. Air Conditioning, Pressurization, and Antiicing -------------------
    aircond_weight = wtor.aircon_pressurization_antiicing(W0, N_OCC, WAV, MACH_MAX)

    ## -------------------------16. Furnishing ----------------------------------------
    furn_weight = wtor.furn_weight()

    ht_weight = 0
    vt_weight = 0
    fus_weight = 0

    weight_components = np.array(
        [
            wing_weight,
            ht_weight,
```

```
            vt_weight,
            emp_weight,
            fus_weight,
            mnlg_weight,
            nlg_weight,
            nac_weight,
            engine_dry_weight,
            engine_installed_weight,
            fuel_sys_weight,
            fcs_weight,
            hydraulic_weight,
            avionics_weight,
            electric_weight,
            aircond_weight,
            furn_weight,
        ]
    )

    return weight_components


def weight_components_usaf():
    ## -------------------------1. WING
    ----------------------------------------------
    wing_weight = wusaf.wing_weight_usaf(NZ, W0, ARW, SWEEP4_WING, SW,
     TR_WING, T2C_W, VH)

    ## -------------------------2. Horizontal Tail
    -----------------------------------
    ht_weight = wusaf.ht_weight_usaf(NZ, W0, SHT, LHT, BHT, THT_MAX)

    # -------------------------3. Vertical Tail
    -------------------------------------
    vt_weight = wusaf.vt_weight_usaf(F_TAIL, NZ, W0, SVT, BVT, TVT_MAX
    )

    ## -------------------------4. Fuselage
    ------------------------------------------
    fus_weight = wusaf.fus_weight_usaf(NZ, W0, LF, WF, DF, VH)

```

```
## -------------------------5. Main Landing Gear ---------------------------------
mnlg_weight = wusaf.mnlg_weight_uasf(WL, LM, NL)

## -------------------------6. Nose/Tail Landing Gear ----------------------------
# Included in the main landing gear

# ## -------------------------7. Nacelle/Cowling Weight ----------------------------
# Included in the installed engine

## -------------------------8. Uninstalled (Dry) Engine --------------------------
engine_dry_weight = wusaf.engine_dry_weight(P_OR_T_MAX, ENGINE_TYPE)

## -------------------------9. Installed Engine ----------------------------------
# W_NAC = nac_weight
engine_installed_weight = wusaf.install_engine_weight_usaf(W_ENG, N_ENG)

## ------------------------10. Fuel System ---------------------------------------
fuel_sys_weight = wusaf.fuelsys_weight_usaf(QTOT, QINT, N_TANK, N_ENG)

## -------------------------11. Flight Control System -----------------------------
fcs_weight = wusaf.flightcs_weight_usaf(W0, CTRL_SYS_TYPE)

## -------------------------12. Hydraulic -----------------------------------------
hydraulic_weight = wusaf.hydraulic_system_weight(W0)

## -------------------------13. Avionics Systems ----------------------------------
avionics_weight = wusaf.avionics_system_weight(W_UAV)
```

```
    # ## -------------------------14. Electrical Systems --------------------------------
    WAV = avionics_weight
    WFS = fuel_sys_weight
    electric_weight = wusaf.electrical_system_raymer_usaf(WFS, WAV)

    ## ---------15. Air Conditioning, Pressurization, and Antiicing -------------------
    aircond_weight = wusaf.aircon_pressurization_antiicing(W0, N_OCC, WAV, MACH_MAX)

    ## -------------------------16. Furnishing ----------------------------------------
    furn_weight = wusaf.furnishings_usaf(N_CREW, QH)

    emp_weight = ht_weight + vt_weight
    nlg_weight = 0
    nac_weight = 0
    weight_components = np.array(
        [
            wing_weight,
            ht_weight,
            vt_weight,
            emp_weight,
            fus_weight,
            mnlg_weight,
            nlg_weight,
            nac_weight,
            engine_dry_weight,
            engine_installed_weight,
            fuel_sys_weight,
            fcs_weight,
            hydraulic_weight,
            avionics_weight,
            electric_weight,
            aircond_weight,
            furn_weight,
        ]
    )
    return weight_components
```

```


# Estimating all weights
weights_cessna = weight_components_cessna()
weights_raymer = weight_components_raymer()
weights_torrenbeek = weight_components_torrenbeek()
weights_usaf = weight_components_usaf()

weights = np.concatenate((weights_cessna, weights_raymer, weights_torrenbeek, weights_usaf,)).reshape(
    17, -1, order="F"
)

np.savetxt("weights_summary.csv", weights, delimiter=",", fmt="%i")
```

A. Component Weight of ATR 42-600

```
import numpy as np
import wanalysis.cessna as wcess
import wanalysis.raymer as wray
import wanalysis.torrenbeek as wtor
import wanalysis.usaf as wusaf

#from example_ac_data import *
from ATR42_data import *
#from Saab340_data import *
#from CN235_Data import *


def weight_components_cessna():
    """
    Estimate weight components using Cessna Method.
    """
    ## -------------------------1. WING ----------------------------------------------
    wing_weight = wcess.wing_weight_cessna(NZ, W0, SW, ARW, WING_TYPE)

    ## -------------------------2. Horizontal Tail -----------------------------------
    ht_weight = wcess.ht_weight_cessna(W0, SHT, AR_HT, THT_MAX)
```

```

 # -------------------------3. Vertical Tail -------------------------------------
 vt_weight = wcess.vt_weight_cessna(F_TAIL, W0, SVT, AR_VT, TVT_MAX, SWEEP4_VT)

 ## -------------------------4. Fuselage ------------------------------------------
 fus_weight = wcess.fus_weight_cessna(W0, RMAX, LFS, WING_POS, N_OCC)

 ## -------------------------5. Main Landing Gear ---------------------------------
 mnlg_weight = wcess.mnlg_weight_cessna(W0, WL, LM, NZ, WING_POS, NL)

 # ## -------------------------6. Nose/Tail Landing Gear ----------------------------
 # # Included in the main landing gear

 # ## -------------------------7. Nacelle/Cowling Weight ----------------------------
 nac_weight = wcess.nac_weight_cessna(PMAX, N_ENG, PISTON_ENGINE_TYPE)

 ## -------------------------8. Uninstalled (Dry) Engine --------------------------
 engine_dry_weight = wcess.engine_dry_weight(P_OR_T_MAX, ENGINE_TYPE)

 # ## -------------------------9. Installed Engine ----------------------------------
 W_NAC = nac_weight
 engine_installed_weight = wcess.install_engine_weight_cessna(PMAX, WPROP, N_ENG, W_NAC)

 ## ------------------------10. Fuel System ---------------------------------------
 fuel_sys_weight = wcess.fuel_sys_weight(QTOT, FUEL_SYS_TYPE)

```

```
    ## -------------------------11. Flight Control System -----------------------------
    fcs_weight = wcess.fligthcs_weight_cessna(W0)

    ## -------------------------12. Hydraulic -----------------------------------------
    hydraulic_weight = wcess.hydraulic_system_weight(W0)

    ## -------------------------13. Avionics Systems ----------------------------------
    avionics_weight = wcess.avionics_system_weight(W_UAV)

    # ## -------------------------14. Electrical Systems --------------------------------
    WAV = avionics_weight
    electric_weight = wcess.electrical_system_cessna(W0)

    ## ---------15. Air Conditioning, Pressurization, and Antiicing -------------------
    aircond_weight = wcess.aircon_pressurization_antiicing(W0, N_OCC, WAV, MACH_MAX)

    ## -------------------------16. Furnishing ----------------------------------------
    furn_weight = wcess.furnishings_weight_cessna(N_OCC, W0)

    emp_weight = ht_weight + vt_weight
    nlg_weight = 0

    weight_components = np.array(
        [
            wing_weight,
            ht_weight,
            vt_weight,
            emp_weight,
            fus_weight,
            mnlg_weight,
            nlg_weight,
            nac_weight,
            engine_dry_weight,
```

```
            engine_installed_weight,
            fuel_sys_weight,
            fcs_weight,
            hydraulic_weight,
            avionics_weight,
            electric_weight,
            aircond_weight,
            furn_weight,
        ]
    )

    return weight_components


def weight_components_raymer():
    """
    Estimate weight components using Raymer Method.
    """
    ## -------------------------1. WING ----------------------------------------------
    wing_weight = wray.wing_weight_raymer(
        SW, WFW, ARW, SWEEP4_WING, Q, TR_WING, T2C_WING, NZ, W0
    )

    ## -------------------------2. Horizontal Tail -----------------------------------
    ht_weight = wray.ht_weight(
        NZ, W0, Q, SHT, T2C_WING, ARW, SWEEP4_HT, SWEEP4_WING, TR_HT
    )

    ## -------------------------3. Vertical Tail -------------------------------------
    vt_weight = wray.vt_weight_raymer(F_TAIL, NZ, W0, Q, SVT, T2C_WING, SWEEP4_VT, ARW, TR_VT)

    ## -------------------------4. Fuselage ------------------------------------------
    fus_weight = wray.fus_weight_raymer(SFUS, NZ, W0, LHT, LFS, DFS, Q, VP, DELTA_P)

```

```
 ## -------------------------5. Main Landing Gear
---------------------------------
 mnlg_weight = wray.mnlg_weight_raymer(WL, LM, NL)

 ## -------------------------6. Nose/Tail Landing Gear
----------------------------
 nlg_weight = wray.nslg_weight_raymer(WL, LN, nl=4.5)

 ## -------------------------7. Nacelle/Cowling Weight
----------------------------
 nac_weight = wray.nac_weight()

 ## -------------------------8. Uninstalled (Dry) Engine
--------------------------
 engine_dry_weight = wray.engine_dry_weight(P_OR_T_MAX, ENGINE_TYPE
)

 ## -------------------------9. Installed Engine
----------------------------------
 engine_installed_weight = wray.install_engine_weight_raymer(W_ENG,
 N_ENG)

 ## ------------------------10. Fuel System
---------------------------------------
 fuel_sys_weight = wray.fuelsys_weight_raymer(QTOT, QINT, N_TANK,
N_ENG)

 ## -------------------------11. Flight Control System
-----------------------------
 fcs_weight = wray.flightcs_weight_raymer(LFS, BW, NZ, W0)

 ## -------------------------12. Hydraulic
-----------------------------------------
 hydraulic_weight = wray.hydraulic_weight(W0, HYD_TYPE, MACH_MAX)

 ## -------------------------13. Avionics Systems
----------------------------------
 avionics_weight = wray.avionics_system_weight(W_UAV)

```

```
    ## -------------------------14. Electrical Systems--------------------------------
    WFS = fuel_sys_weight
    WAV = avionics_weight
    electric_weight = wray.electrical_system_raymer_usaf(WFS, WAV)

    ## ---------15. Air Conditioning, Pressurization, and Antiicing-------------------
    aircond_weight = wray.aircon_pressurization_antiicing(W0, N_OCC, WAV, MACH_MAX)

    ## -------------------------16. Furnishing----------------------------------------
    furn_weight = wray.furnishings_weight_raymer(W0)

    emp_weight = ht_weight + vt_weight
    weight_components = np.array(
        [
            wing_weight,
            ht_weight,
            vt_weight,
            emp_weight,
            fus_weight,
            mnlg_weight,
            nlg_weight,
            nac_weight,
            engine_dry_weight,
            engine_installed_weight,
            fuel_sys_weight,
            fcs_weight,
            hydraulic_weight,
            avionics_weight,
            electric_weight,
            aircond_weight,
            furn_weight,
        ]
    )

    return weight_components
```

```

def weight_components_torrenbeek():
    ## -------------------------1. WING ----------------------------------------------
    wing_weight = wtor.wing_weight_torenbeek(W0, BW, SW, SWEEP2_WING, NZ, TW_MAX)

    ## -------------------------2. Empenage -----------------------------------
    emp_weight = wtor.emp_weight_torrenbeek(NZ, SHT, SVT)

    ## -------------------------3. Vertical Tail -------------------------------------
    # Vertical Tail + Horizontal Tail = Empenage

    ## -------------------------4. Fuselage ------------------------------------------

    ## -------------------------5. Main Landing Gear ----------------------------------
    mnlg_weight = wtor.mnlg_weight_torrenbeek(W0, WING_POS, LG_TYPE, AC_CLASS)

    ## -------------------------6. Nose/Tail Landing Gear ----------------------------
    nlg_weight = wtor.nslg_weight_torenbeek(W0, WING_POS, LG_TYPE, AC_CLASS)

    ## -------------------------7. Nacelle/Cowling Weight ----------------------------
    nac_weight = wtor.nac_weight(P_OR_T_MAX, N_ENG, NAC_ENGINE_TYPE)

    ## -------------------------8. Uninstalled (Dry) Engine --------------------------
    engine_dry_weight = wtor.engine_dry_weight(P_OR_T_MAX, ENGINE_TYPE)

    ## -------------------------9. Installed Engine ----------------------------------
    W_NAC = nac_weight
```

```
    engine_installed_weight = wtor.install_engine_weight_torrenbeek(
        W_ENG, WPROP, N_ENG, PMAX, W_NAC
    )

    ## -------------------------10. Fuel System ---------------------------------------
    fuel_sys_weight = wtor.fuel_sys_weight(QTOT, ENGINE_CONF, N_TANK, N_ENG)

    ## -------------------------11. Flight Control System -----------------------------
    fcs_weight = wtor.fcs_weight(W0, CTRL_SYS_TYPE)

    ## -------------------------12. Hydraulic -----------------------------------------
    hydraulic_weight = wtor.hydraulic_weight(W0)

    ## -------------------------13. Avionics Systems ----------------------------------
    avionics_weight = wtor.avionics_system_weight(W_UAV)

    ## -------------------------14. Electrical Systems --------------------------------
    WAV = avionics_weight
    WHYD = hydraulic_weight
    electric_weight = wtor.electrical_system_torenbeek(W0, WU, WHYD)

    ## ---------15. Air Conditioning, Pressurization, and Antiicing -------------------
    aircond_weight = wtor.aircon_pressurization_antiicing(W0, N_OCC, WAV, MACH_MAX)

    ## -------------------------16. Furnishing ----------------------------------------
    furn_weight = wtor.furn_weight()

    ht_weight = 0
    vt_weight = 0
    fus_weight = 0
```

```
    weight_components = np.array(
        [
            wing_weight,
            ht_weight,
            vt_weight,
            emp_weight,
            fus_weight,
            mnlg_weight,
            nlg_weight,
            nac_weight,
            engine_dry_weight,
            engine_installed_weight,
            fuel_sys_weight,
            fcs_weight,
            hydraulic_weight,
            avionics_weight,
            electric_weight,
            aircond_weight,
            furn_weight,
        ]
    )

    return weight_components


def weight_components_usaf():
    ## -------------------------1. WING ---------------------------------------------
    wing_weight = wusaf.wing_weight_usaf(NZ, W0, ARW, SWEEP4_WING, SW, TR_WING, T2C_W, VH)

    ## -------------------------2. Horizontal Tail -----------------------------------
    ht_weight = wusaf.ht_weight_usaf(NZ, W0, SHT, LHT, BHT, THT_MAX)

    # -------------------------3. Vertical Tail -------------------------------------
    vt_weight = wusaf.vt_weight_usaf(F_TAIL, NZ, W0, SVT, BVT, TVT_MAX)

```

```
    ## -------------------------4. Fuselage ------------------------------------------
    fus_weight = wusaf.fus_weight_usaf(NZ, W0, LF, WF, DF, VH)

    ## -------------------------5. Main Landing Gear ---------------------------------
    mnlg_weight = wusaf.mnlg_weight_uasf(WL, LM, NL)

    ## -------------------------6. Nose/Tail Landing Gear ----------------------------
    # Included in the main landing gear

    # ## -------------------------7. Nacelle/Cowling Weight ----------------------------
    # Included in the installed engine

    ## -------------------------8. Uninstalled (Dry) Engine --------------------------
    engine_dry_weight = wusaf.engine_dry_weight(P_OR_T_MAX, ENGINE_TYPE)

    ## -------------------------9. Installed Engine ----------------------------------
    # W_NAC = nac_weight
    engine_installed_weight = wusaf.install_engine_weight_usaf(W_ENG, N_ENG)

    ## ------------------------10. Fuel System ---------------------------------------
    fuel_sys_weight = wusaf.fuelsys_weight_usaf(QTOT, QINT, N_TANK, N_ENG)

    ## -------------------------11. Flight Control System -----------------------------
    fcs_weight = wusaf.flightcs_weight_usaf(W0, CTRL_SYS_TYPE)

    ## -------------------------12. Hydraulic -----------------------------------------
    hydraulic_weight = wusaf.hydraulic_system_weight(W0)

```

```
    ## -------------------------13. Avionics Systems ----------------------------------
    avionics_weight = wusaf.avionics_system_weight(W_UAV)

    # ## -------------------------14. Electrical Systems --------------------------------
    WAV = avionics_weight
    WFS = fuel_sys_weight
    electric_weight = wusaf.electrical_system_raymer_usaf(WFS, WAV)

    ## ---------15. Air Conditioning, Pressurization, and Antiicing -------------------
    aircond_weight = wusaf.aircon_pressurization_antiicing(W0, N_OCC, WAV, MACH_MAX)

    ## -------------------------16. Furnishing ----------------------------------------
    furn_weight = wusaf.furnishings_usaf(N_CREW, QH)

    emp_weight = ht_weight + vt_weight
    nlg_weight = 0
    nac_weight = 0
    weight_components = np.array(
        [
            wing_weight,
            ht_weight,
            vt_weight,
            emp_weight,
            fus_weight,
            mnlg_weight,
            nlg_weight,
            nac_weight,
            engine_dry_weight,
            engine_installed_weight,
            fuel_sys_weight,
            fcs_weight,
            hydraulic_weight,
            avionics_weight,
            electric_weight,
            aircond_weight,
```

```
                furn_weight,
            ]
        )
        return weight_components


# Estimating all weights
weights_cessna = weight_components_cessna()
weights_raymer = weight_components_raymer()
weights_torrenbeek = weight_components_torrenbeek()
weights_usaf = weight_components_usaf()

weights = np.concatenate((weights_cessna, weights_raymer,
    weights_torrenbeek, weights_usaf)).reshape(
    17, -1, order="F"
)

np.savetxt("weights_summary.csv", weights, delimiter=",", fmt="%i")
```

A. Component Weight of Saab 340

```
import numpy as np
import wanalysis.cessna as wcess
import wanalysis.raymer as wray
import wanalysis.torrenbeek as wtor
import wanalysis.usaf as wusaf

#from example_ac_data import *
#from ATR42_data import *
from Saab340_data import *
#from CN235_Data import *


def weight_components_cessna():
    """
    Estimate weight components using Cessna Method.
    """
    ## ------------------------1. WING
    --------------------------------------------
    wing_weight = wcess.wing_weight_cessna(NZ, W0, SW, ARW, WING_TYPE)
```

```

 ## -------------------------2. Horizontal Tail -----------------------------------
 ht_weight = wcess.ht_weight_cessna(W0, SHT, AR_HT, THT_MAX)

 # -------------------------3. Vertical Tail -------------------------------------
 vt_weight = wcess.vt_weight_cessna(F_TAIL, W0, SVT, AR_VT, TVT_MAX, SWEEP4_VT)

 ## -------------------------4. Fuselage ------------------------------------------
 fus_weight = wcess.fus_weight_cessna(W0, RMAX, LFS, WING_POS, N_OCC)

 ## -------------------------5. Main Landing Gear ---------------------------------
 mnlg_weight = wcess.mnlg_weight_cessna(W0, WL, LM, NZ, WING_POS, NL)

 # ## -------------------------6. Nose/Tail Landing Gear ----------------------------
 # # Included in the main landing gear

 # ## -------------------------7. Nacelle/Cowling Weight ----------------------------
 nac_weight = wcess.nac_weight_cessna(PMAX, N_ENG, PISTON_ENGINE_TYPE)

 ## -------------------------8. Uninstalled (Dry) Engine --------------------------
 engine_dry_weight = wcess.engine_dry_weight(P_OR_T_MAX, ENGINE_TYPE)

 # ## -------------------------9. Installed Engine ----------------------------------
 W_NAC = nac_weight
 engine_installed_weight = wcess.install_engine_weight_cessna(PMAX, WPROP, N_ENG, W_NAC)

```

```
    ## ------------------------10. Fuel System ---------------------------------------
    fuel_sys_weight = wcess.fuel_sys_weight(QTOT, FUEL_SYS_TYPE)

    ## -------------------------11. Flight Control System -----------------------------
    fcs_weight = wcess.fligthcs_weight_cessna(W0)

    ## -------------------------12. Hydraulic -----------------------------------------
    hydraulic_weight = wcess.hydraulic_system_weight(W0)

    ## -------------------------13. Avionics Systems ----------------------------------
    avionics_weight = wcess.avionics_system_weight(W_UAV)

    # ## -------------------------14. Electrical Systems --------------------------------
    WAV = avionics_weight
    electric_weight = wcess.electrical_system_cessna(W0)

    ## ---------15. Air Conditioning, Pressurization, and Antiicing -------------------
    aircond_weight = wcess.aircon_pressurization_antiicing(W0, N_OCC, WAV, MACH_MAX)

    ## -------------------------16. Furnishing ----------------------------------------
    furn_weight = wcess.furnishings_weight_cessna(N_OCC, W0)

    emp_weight = ht_weight + vt_weight
    nlg_weight = 0

    weight_components = np.array(
        [
            wing_weight,
            ht_weight,
            vt_weight,
            emp_weight,
            fus_weight,
```

```python
            mnlg_weight,
            nlg_weight,
            nac_weight,
            engine_dry_weight,
            engine_installed_weight,
            fuel_sys_weight,
            fcs_weight,
            hydraulic_weight,
            avionics_weight,
            electric_weight,
            aircond_weight,
            furn_weight,
        ]
    )

    return weight_components


def weight_components_raymer():
    """
    Estimate weight components using Raymer Method.
    """
    ## -------------------------1. WING ----------------------------------------------
    wing_weight = wray.wing_weight_raymer(
        SW, WFW, ARW, SWEEP4_WING, Q, TR_WING, T2C_WING, NZ, W0
    )

    ## -------------------------2. Horizontal Tail -----------------------------------
    ht_weight = wray.ht_weight(
        NZ, W0, Q, SHT, T2C_WING, ARW, SWEEP4_HT, SWEEP4_WING, TR_HT
    )

    ## -------------------------3. Vertical Tail -------------------------------------
    vt_weight = wray.vt_weight_raymer(F_TAIL, NZ, W0, Q, SVT, T2C_WING, SWEEP4_VT, ARW, TR_VT)

```

```
## -------------------------4. Fuselage -------------------------------------------
fus_weight = wray.fus_weight_raymer(SFUS, NZ, W0, LHT, LFS, DFS, Q, VP, DELTA_P)

## -------------------------5. Main Landing Gear ---------------------------------
mnlg_weight = wray.mnlg_weight_raymer(WL, LM, NL)

## -------------------------6. Nose/Tail Landing Gear ----------------------------
nlg_weight = wray.nslg_weight_raymer(WL, LN, nl=4.5)

## -------------------------7. Nacelle/Cowling Weight ----------------------------
nac_weight = wray.nac_weight()

## -------------------------8. Uninstalled (Dry) Engine --------------------------
engine_dry_weight = wray.engine_dry_weight(P_OR_T_MAX, ENGINE_TYPE)

## -------------------------9. Installed Engine ----------------------------------
engine_installed_weight = wray.install_engine_weight_raymer(W_ENG, N_ENG)

## ------------------------10. Fuel System ---------------------------------------
fuel_sys_weight = wray.fuelsys_weight_raymer(QTOT, QINT, N_TANK, N_ENG)

## -------------------------11. Flight Control System ----------------------------
fcs_weight = wray.flightcs_weight_raymer(LFS, BW, NZ, W0)

## -------------------------12. Hydraulic -----------------------------------------
hydraulic_weight = wray.hydraulic_weight(W0, HYD_TYPE, MACH_MAX)

```

```
    ## -------------------------13. Avionics Systems
----------------------------------
    avionics_weight = wray.avionics_system_weight(W_UAV)

    ## -------------------------14. Electrical Systems
--------------------------------
    WFS = fuel_sys_weight
    WAV = avionics_weight
    electric_weight = wray.electrical_system_raymer_usaf(WFS, WAV)

    ## ---------15. Air Conditioning, Pressurization, and Antiicing
-------------------
    aircond_weight = wray.aircon_pressurization_antiicing(W0, N_OCC,
WAV, MACH_MAX)

    ## -------------------------16. Furnishing
----------------------------------------
    furn_weight = wray.furnishings_weight_raymer(W0)

    emp_weight = ht_weight + vt_weight
    weight_components = np.array(
        [
            wing_weight,
            ht_weight,
            vt_weight,
            emp_weight,
            fus_weight,
            mnlg_weight,
            nlg_weight,
            nac_weight,
            engine_dry_weight,
            engine_installed_weight,
            fuel_sys_weight,
            fcs_weight,
            hydraulic_weight,
            avionics_weight,
            electric_weight,
            aircond_weight,
            furn_weight,
        ]
```

```
    )

    return weight_components


def weight_components_torrenbeek():
    ## -------------------------1. WING ----------------------------------------------
    wing_weight = wtor.wing_weight_torenbeek(W0, BW, SW, SWEEP2_WING, NZ, TW_MAX)

    ## -------------------------2. Empenage -----------------------------------
    emp_weight = wtor.emp_weight_torrenbeek(NZ, SHT, SVT)

    ## -------------------------3. Vertical Tail -------------------------------------
    # Vertical Tail + Horizontal Tail = Empenage

    ## -------------------------4. Fuselage ------------------------------------------

    ## -------------------------5. Main Landing Gear ---------------------------------
    mnlg_weight = wtor.mnlg_weight_torrenbeek(W0, WING_POS, LG_TYPE, AC_CLASS)

    ## -------------------------6. Nose/Tail Landing Gear ----------------------------
    nlg_weight = wtor.nslg_weight_torenbeek(W0, WING_POS, LG_TYPE, AC_CLASS)

    ## -------------------------7. Nacelle/Cowling Weight ----------------------------
    nac_weight = wtor.nac_weight(P_OR_T_MAX, N_ENG, NAC_ENGINE_TYPE)

    ## -------------------------8. Uninstalled (Dry) Engine --------------------------
    engine_dry_weight = wtor.engine_dry_weight(P_OR_T_MAX, ENGINE_TYPE)
```

```
    ## -------------------------9. Installed Engine
---------------------------------
    W_NAC = nac_weight
    engine_installed_weight = wtor.install_engine_weight_torrenbeek(
        W_ENG, WPROP, N_ENG, PMAX, W_NAC
    )

    ## ------------------------10. Fuel System
---------------------------------------
    fuel_sys_weight = wtor.fuel_sys_weight(QTOT, ENGINE_CONF, N_TANK,
N_ENG)

    ## -------------------------11. Flight Control System
-----------------------------
    fcs_weight = wtor.fcs_weight(W0, CTRL_SYS_TYPE)

    ## -------------------------12. Hydraulic
-----------------------------------------
    hydraulic_weight = wtor.hydraulic_weight(W0)

    ## -------------------------13. Avionics Systems
---------------------------------
    avionics_weight = wtor.avionics_system_weight(W_UAV)

    ## -------------------------14. Electrical Systems
--------------------------------
    WAV = avionics_weight
    WHYD = hydraulic_weight
    electric_weight = wtor.electrical_system_torenbeek(W0, WU, WHYD)

    ## ---------15. Air Conditioning, Pressurization, and Antiicing
-------------------
    aircond_weight = wtor.aircon_pressurization_antiicing(W0, N_OCC,
WAV, MACH_MAX)

    ## -------------------------16. Furnishing
----------------------------------------
    furn_weight = wtor.furn_weight()

```

```python
    ht_weight = 0
    vt_weight = 0
    fus_weight = 0

    weight_components = np.array(
        [
            wing_weight,
            ht_weight,
            vt_weight,
            emp_weight,
            fus_weight,
            mnlg_weight,
            nlg_weight,
            nac_weight,
            engine_dry_weight,
            engine_installed_weight,
            fuel_sys_weight,
            fcs_weight,
            hydraulic_weight,
            avionics_weight,
            electric_weight,
            aircond_weight,
            furn_weight,
        ]
    )

    return weight_components


def weight_components_usaf():
    ## -------------------------1. WING ----------------------------------------------
    wing_weight = wusaf.wing_weight_usaf(NZ, W0, ARW, SWEEP4_WING, SW, TR_WING, T2C_W, VH)

    ## -------------------------2. Horizontal Tail -----------------------------------
    ht_weight = wusaf.ht_weight_usaf(NZ, W0, SHT, LHT, BHT, THT_MAX)

```

```
# -------------------------3. Vertical Tail -------------------------------------
vt_weight = wusaf.vt_weight_usaf(F_TAIL, NZ, W0, SVT, BVT, TVT_MAX)

## -------------------------4. Fuselage ------------------------------------------
fus_weight = wusaf.fus_weight_usaf(NZ, W0, LF, WF, DF, VH)

## -------------------------5. Main Landing Gear ---------------------------------
mnlg_weight = wusaf.mnlg_weight_uasf(WL, LM, NL)

## -------------------------6. Nose/Tail Landing Gear ----------------------------
# Included in the main landing gear

# ## -------------------------7. Nacelle/Cowling Weight ----------------------------
# Included in the installed engine

## -------------------------8. Uninstalled (Dry) Engine --------------------------
engine_dry_weight = wusaf.engine_dry_weight(P_OR_T_MAX, ENGINE_TYPE)

## -------------------------9. Installed Engine ----------------------------------
# W_NAC = nac_weight
engine_installed_weight = wusaf.install_engine_weight_usaf(W_ENG, N_ENG)

## ------------------------10. Fuel System ---------------------------------------
fuel_sys_weight = wusaf.fuelsys_weight_usaf(QTOT, QINT, N_TANK, N_ENG)

## -------------------------11. Flight Control System -----------------------------
fcs_weight = wusaf.flightcs_weight_usaf(W0, CTRL_SYS_TYPE)
```

```
    ## -------------------------12. Hydraulic
-----------------------------------------
    hydraulic_weight = wusaf.hydraulic_system_weight(W0)

    ## -------------------------13. Avionics Systems
----------------------------------
    avionics_weight = wusaf.avionics_system_weight(W_UAV)

    # ## -------------------------14. Electrical Systems
--------------------------------
    WAV = avionics_weight
    WFS = fuel_sys_weight
    electric_weight = wusaf.electrical_system_raymer_usaf(WFS, WAV)

    ## ---------15. Air Conditioning, Pressurization, and Antiicing
-------------------
    aircond_weight = wusaf.aircon_pressurization_antiicing(W0, N_OCC,
WAV, MACH_MAX)

    ## -------------------------16. Furnishing
----------------------------------------
    furn_weight = wusaf.furnishings_usaf(N_CREW, QH)

    emp_weight = ht_weight + vt_weight
    nlg_weight = 0
    nac_weight = 0
    weight_components = np.array(
        [
            wing_weight,
            ht_weight,
            vt_weight,
            emp_weight,
            fus_weight,
            mnlg_weight,
            nlg_weight,
            nac_weight,
            engine_dry_weight,
            engine_installed_weight,
            fuel_sys_weight,
```

```
            fcs_weight,
            hydraulic_weight,
            avionics_weight,
            electric_weight,
            aircond_weight,
            furn_weight,
        ]
    )
    return weight_components


# Estimating all weights
weights_cessna = weight_components_cessna()
weights_raymer = weight_components_raymer()
weights_torrenbeek = weight_components_torrenbeek()
weights_usaf = weight_components_usaf()

weights = np.concatenate((weights_cessna, weights_raymer,
    weights_torrenbeek, weights_usaf)).reshape(
    17, -1, order="F"
)

np.savetxt("weights_summary.csv", weights, delimiter=",", fmt="%i")
```

Turnitin Report

COMPARISON OF STATISTICAL WEIGHT METHODS APPLIED FOR REGIONAL TURBOPROP AIRCRAFT

ORIGINALITY REPORT

15%	10%	8%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Snorri Gudmundsson. "Aircraft Weight Analysis", Elsevier BV, 2022 Publication	3%
2	docero.net Internet Source	2%
3	wikimili.com Internet Source	1%
4	en.unionpedia.org Internet Source	1%
5	www.dtic.mil Internet Source	1%
6	Snorri Gudmundsson. "Initial Sizing", Elsevier BV, 2022 Publication	<1%
7	archive.org Internet Source	<1%
8	Submitted to Glyndwr University Student Paper	<1%

9	en.wikipedia.org Internet Source	<1%
10	ntrs.nasa.gov Internet Source	<1%
11	Submitted to University of Hertfordshire Student Paper	<1%
12	www.cfinotebook.net Internet Source	<1%
13	Mohammad H. Sadraey. "Aircraft Design", Wiley, 2012 Publication	<1%
14	uvidok.rcub.bg.ac.rs Internet Source	<1%
15	Submitted to University of Witwatersrand Student Paper	<1%
16	Submitted to Emirates Aviation College, Aerospace & Academic Studies Student Paper	<1%
17	Submitted to RMIT University Student Paper	<1%
18	www.livingwarbirds.com Internet Source	<1%
19	www.emerald.com Internet Source	<1%

20	Submitted to University of Sydney Student Paper	<1 %
21	hdl.handle.net Internet Source	<1 %
22	epdf.pub Internet Source	<1 %
23	www.govinfo.gov Internet Source	<1 %
24	www.traveloka.com Internet Source	<1 %
25	Submitted to Blackpool and The Fylde College, Lancashire Student Paper	<1 %
26	Submitted to Group Colleges Australia Student Paper	<1 %
27	Seulki Kim, Kyunghoon Lee. "Constraint Analysis for Aircraft Initial Sizing Using Reliability-Based Design Optimization", 2018 AIAA Non-Deterministic Approaches Conference, 2018 Publication	<1 %
28	eprints.kfupm.edu.sa Internet Source	<1 %
29	iopscience.iop.org Internet Source	<1 %

30	mafiadoc.com Internet Source	<1 %
31	www.so.vccs.edu Internet Source	<1 %
32	Grant E. Carichner, Leland M. Nicolai. "Fundamentals of Aircraft and Airship Design: Volume 2—Airship Design and Case Studies", American Institute of Aeronautics and Astronautics (AIAA), 2013 Publication	<1 %
33	wiki2.org Internet Source	<1 %
34	www.ae.metu.edu.tr Internet Source	<1 %
35	Mario Asselin. "Operational Aircraft Performance and Flight Test Practices", American Institute of Aeronautics and Astronautics (AIAA), 2021 Publication	<1 %
36	Submitted to University of Northampton Student Paper	<1 %
37	Submitted to Walsall College Student Paper	<1 %
38	www.flightliteracy.com Internet Source	<1 %

39	Submitted to Massey University Student Paper	<1 %
40	Steven A. Brandt, Randall J. Stiles, John J. Bertin, Ray Whitford. "Introduction to Aeronautics: A Design Perspective, Third Edition", American Institute of Aeronautics and Astronautics (AIAA), 2015 Publication	<1 %
41	dokumen.pub Internet Source	<1 %
42	C. You, M. Yasaee, S. He, D. Yang, Y. Xu, I. Dayyani, H. Ghasemnejad, S. Guo, P. Webb, J. Jennings, G. Federico. "Identification of the key design inputs for the FEM-based preliminary sizing and mass estimation of a civil aircraft wing box structure", Aerospace Science and Technology, 2021 Publication	<1 %
43	MARK SCHWABACHER, THOMAS ELLMAN, HAYM HIRSH. "Learning to set up numerical optimizations of engineering designs", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 1998 Publication	<1 %
44	www.dochub.com Internet Source	<1 %
	wikivividly.com	

45	Internet Source	<1%
46	Submitted to Bahrain Polytechnic Student Paper	<1%
47	Submitted to Imperial College of Science, Technology and Medicine Student Paper	<1%
48	Submitted to Preston College Student Paper	<1%
49	Submitted to Embry Riddle Aeronautical University Student Paper	<1%
50	Submitted to Cranfield University Student Paper	<1%
51	James I. Hileman, Russell W. Stratton, Pearl E. Donohoo. "Energy Content and Alternative Jet Fuel Viability", Journal of Propulsion and Power, 2010 Publication	<1%
52	Submitted to Singapore Institute of Technology Student Paper	<1%
53	Siwakorn Anosri, Natee Panagant, Pakin Champasak, Sujin Bureerat et al. "A Comparative Study of State-of-the-art Metaheuristics for Solving Many-objective	<1%

Optimization Problems of Fixed Wing Unmanned Aerial Vehicle Conceptual Design", Archives of Computational Methods in Engineering, 2023
Publication

54	www.atr-aircraft.com Internet Source	<1 %
55	"Handbook of Small Satellites", Springer Science and Business Media LLC, 2020 Publication	<1 %
56	www.mdpi.com Internet Source	<1 %
57	www.scmp.com Internet Source	<1 %
58	Submitted to Universidad Carlos III de Madrid Student Paper	<1 %
59	etd.aau.edu.et Internet Source	<1 %
60	J-Ch Angls dAuriac. "A classification of four-state spin edge Potts models", Journal of Physics A Mathematical and General, 11/08/2002 Publication	<1 %
61	Julius W. Quitter, Matthew Marino, J.-Michael Bauschat. "A Surrogate Model Approach to	<1 %

the Trimming of Joined Wing Configurations", AIAA SCITECH 2022 Forum, 2022
Publication

62	Pat Piperni, Mohammed Abdo, Fassi Fafyeke. "The Application of Multi-Disciplinary Optimization Technologies to the Design of a Business Jet", 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2004 Publication	<1 %
63	core.ac.uk Internet Source	<1 %
64	liu.diva-portal.org Internet Source	<1 %
65	www.coursehero.com Internet Source	<1 %
66	www.idsemergencymanagement.com Internet Source	<1 %
67	www.oldjets.net Internet Source	<1 %
68	Jay Gundlach. "Designing Unmanned Aircraft Systems", American Institute of Aeronautics and Astronautics (AIAA), 2012 Publication	<1 %
69	Spyridon Kilimtzidis, Efthymios Giannaros, Athanasios Kotzakolios, Angelos Kafkas et al.	<1 %

"Modeling, analysis and validation of the structural response of a large-scale composite wing by ground testing", Composite Structures, 2023
Publication

70	dspace.cvut.cz Internet Source	<1 %
71	lup.lub.lu.se Internet Source	<1 %
72	Marinus K. Okoronkwo, Raymond Alsaif, Neil Farvolden, Dominic G. Geneau et al. "Aeroacoustic Characterization of the 30P30N High-Lift Airfoil", AIAA AVIATION 2023 Forum, 2023 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

Curriculum Vitae

	Basic Information
Name	Raka Pradana Sanferdi
Place of Birth	Padang
Date of Birth	16, 05, 1997
Address	Perumahan Mahkota Simprug Blokd D3 No.18, RT/RW, 005/009, Larangan, Larangan Selatan, Kota Tanggerang
Year	**Education**
2016 - present	International University Liaison Indonesia
2012 - 2015	Kartini Senior High School
2009 - 2012	Kartini Junior High School
2003 - 2009	Kartini Elementary School
Languages	**Level**
Bahasa Indonesia	Master
German	B1
Japanese	N5
Year	**Work Experiences**
2018	Student intern at PTDI, Bandung, Indonesia