KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY, KUMASI

GROUP 7 AIRCRAFT VEHICLE DESIGN PROJECT

PYTHON PROGRAMMING PROJECT 2

Project Description

The project tasked us to take the user’s inputs on range, endurance, payloads, and aircraft type. The use those values to estimate the weight of the aircraft, RDT&E + flyaway cost, the wing sizing, the fuselage sizing, maximum thrust, and engine sizing.

Disclaimer: All values and assumptions were based on Daniel Raymer’s textbook Aircraft Design: A conceptual approach. However, it must be treated as an academic exercise.

Our solution approaches.

1. Defining all needed constants in a json file.
2. Defining a python class called “auxi\_func” (auxiliary functions) that contained all the functions needed to perform the final computations.
3. Defined a child class of the “auxi\_func” called “aeroplane” that performs the calculations on the python class.
4. Lastly, we defined the graphic user interface of the software so that the user could easily interact with it.

DETAILS ON EACH OF THE SOLUION

Defining the constants (constants.json file):

We scripted a json file which looks like a python dictionary. It had four keys: jet transport, fighter jet, seaplane, and uni\_constants (universal constants).

Each key had a corresponding dictionary value. The values associated with the following keys, jet transport, fighter jet, and seaplane, had the following constants: weight fractions of a predefined flight profile, A, c, Kvc, allowance, a, b, C1, C2, C3, C4, C5, Mach number (M), Thrust to weight (T\_W), lift to drag ratio (L\_D) for cruise and loiter, specific heat capacity (Csp) for cruise and loiter, aspect ratio (AR), flying altitude, angle of approach during landing (), distance travelled before touch down from 50ft height (Sa), cost of avionics (C\_avionics).

Note that for the weight fractions, each flight leg was shown as a list with the weight fraction value of that leg or specified as a cruise (c) or loiter (l) followed by a Boolean (True, or False). The Boolean is to determine if it is accelerated or not accelerated flight.

For the universal constant key, its dictionary value had the following sub-keys: souls\_weight (estimated weight of an individual and their personal luggage), maximum co-efficient of lift (Cl\_max), speed of sound (V\_sound), number of engines per aircraft (N\_engines), air density (air\_density), initially guessed take-off weight (W0\_guess), Y\_sscg, Y\_ma, cost rate of engineering an aircraft (R\_eng), cost rate of tooling (R\_tooling), cost rate of quality control (R\_quality), cost rate of manufacturing (R\_mfg), quantity of airplanes (Q), flight test aircraft (FTA), temperature at inlet of engine (T\_inlet).

Python class auxi\_function (AVD\_functions.py file)

The class relies on 2 python libraries, math library and the json library.

The class contains the following methods,

|  |  |
| --- | --- |
| Class methods | Functions of the methods |
| \_\_init\_\_ | initializing the class with values such as plane, cruise\_range, endurance, n\_souls, fixed\_payload, dropable\_payload, dp |
| calc\_We\_W0 | calculates the empty weight fraction and make use of the range and endurance equations.  Eqn 1 – crude method  Eqn 2 – refined method |
| Calc\_T\_W | Calculates the trust to weight ratio of the aircraft.  Eqn:  Or  The maximum value is selected after both equations are computed. |
| Calc\_W\_S | Calculates the wing loading.  Eqn: |
| Calc\_Wf\_W0 | Calculates the fuel fraction of the aircraft across various legs.  Eqn:  Where , are calculated from a different method, calc\_W\_fraction. |
| Calc\_Wf | Calculates the weight of the fuel.  Eqn: |
| Calc\_W\_fraction | Calculates the weight fraction of the entire flight profile.  Eqn:  Note that, if any of the weight fraction represents cruise, loiter or accelerated climb then eternal methods will be called to perform the calculations. |
| acc\_climb | Calculates the weight fraction if the climb of the aircraft is going to be accelerated.  Eqn 1 – from Mach 0.1 to a given subsonic Mach number, M.  Eqn 2 – from Mach 0.1 to a given supersonic Mach number, M.  Note, if the accelerated climb does not start from 0.1 but let’s say to another Mach number, then the weight fraction is the ratio of the weight fraction from 0.1 to to the weight fraction from 0.1 to . |
| Range\_eqn | The range equation is used to calculate the weight fraction at cruise.  Eqn  Note that the “e” in python is imported from the math library as exp( ) function. The Csp for cruise is different from Csp for loiter. |
| Endurance eqn | The endurance equation is used to calculate the weight fraction at cruise.  Eqn  Note that the “e” in python is imported from the math library as exp( ) function. The Csp for loiter is different from Csp for cruise. |
| \_\_main | This method was created for testing the class and its methods. |

After running the class auxi\_func, the following attributes are created that are associated with the object (self): plane, const, uni\_const, payload, dp, cruise\_range, endurance, weight, We\_W0, T\_W, Wf\_W0, and W\_fraction.

Python Class aeroplane (AVD\_aeroplane.py file).

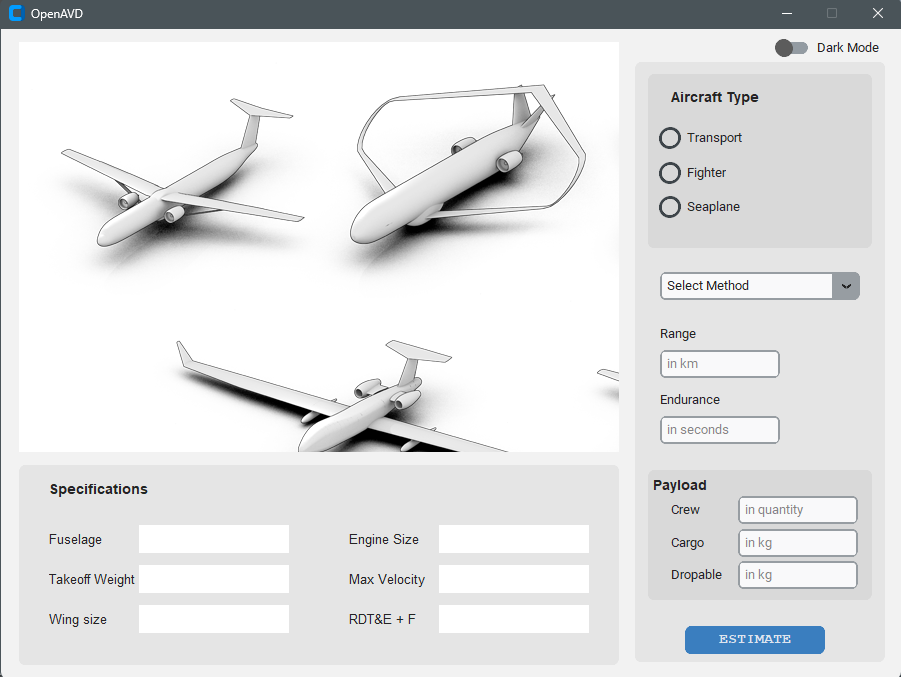
The class aeroplane is a child class of auxi\_func and it inherits all the method and attributes of the auxi\_func class.

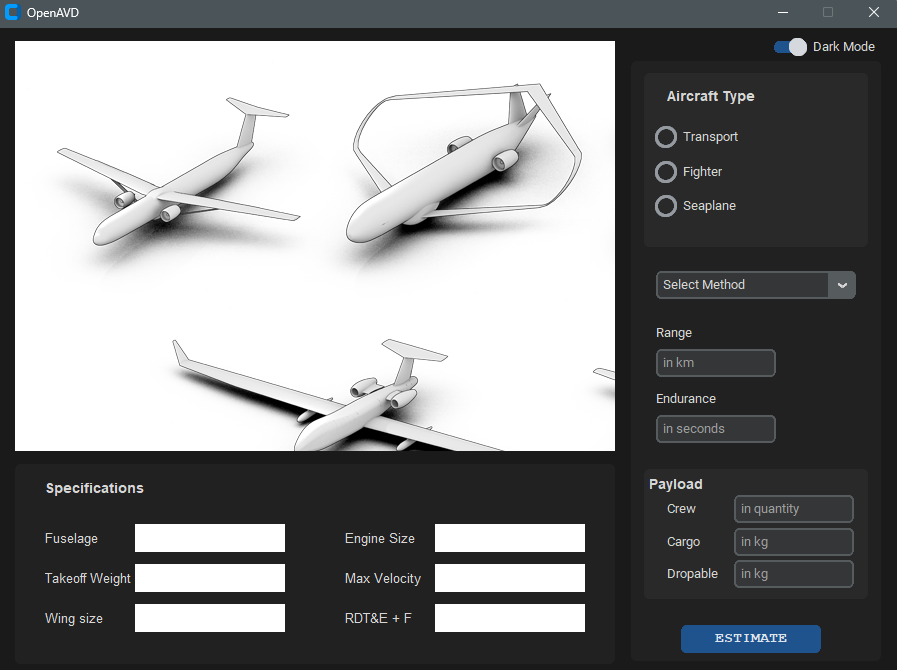
The inherited attributes are used in several methods associated with the aeroplane class. See the table below.

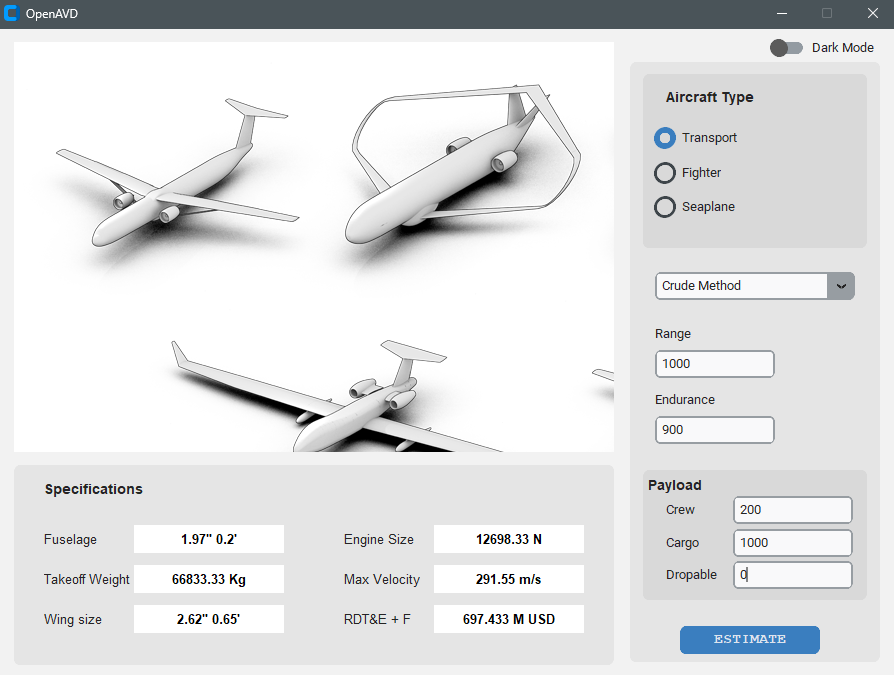
|  |  |
| --- | --- |
| methods | Functions |
| Take-off weight | Calculates the take-off weight of the aircraft given that the user prefers the crude method or the refined method.  Eqn 1 – crude method  Eqn 2 – refined method |
| Engine\_size | Calculates the minimum thrust required for the take-off weight.  Eqn: |
| Wing\_sizing | Calculates the length and area of the wing respectively.  Eqn 1: length of the wing  Eqn 2: Area of the wing |
| Fuselage\_sizing | Calculates the length and diameter of the fuselage.  Eqn 1: Length of the fuselage  Eqn 2: Fuselage diameter |
| V\_max | Calculates the Maximum velocity of the aircraft.  Eqn:  Where M is the Mach number of the aircraft. |
| RDT\_cost | Calculates the RTD&E + flyaway cost of the aircraft using the following equations.  Eqns |

Graphic user interface

We used the libraries, tkinter and customtkinter to implement a simple graphic user interface. The GUI was rendered in both light and dark themes. Taking the required inputs and making the required calculations.







The image above shows our software performing computations using the crude method.

A screenshot of a computer

Description automatically generated

Given the same inputs the software performed a more refined computation reducing the initial take-off weight from 66833.33kg to 34001.23 Kg (49.13 % decrease). This also affects the minimum thrust required and any other parameter that is trust dependent.

RECOMMENDATIONS

Future developers should focus on,

1. Including a 3D modeling feature to replace the image where users could design their own aircraft.
2. Include more functionalities such as calculating certain assumed values like coefficient of lift max and aspect ratio of the aircraft.
3. Allow the user to select their preferred flight profile instead of a predetermined profile.

Contact the developers via

Email 1 [kobinaadu10@gmail.com](mailto:kobinaadu10@gmail.com)

Email 2 [pytech.hub@gmail.com](mailto:pytech.hub@gmail.com)

WhatsApp: <https://wa,me/233209130538>