Universidad Carlos III de Madrid

Flight Mechanics

Aerospace Engineering

Laboratory Session 2: Aircraft Performance

Alejandro García Laguna 100345800
Jorge Martínez Garrido 100349480

Contents

1 Abstract

The objective of this laboratory session is to acquire and develop a certain knowledge, properties and conclusions about the performance equations coming from the theoretical sessions. Through this report they will be evaluated not only in physical, but also in dimensionless variables. Calculations will be performed for different phases of a flight (such as the **cruise**, **climb** and **turning** manoeuvres). Aircraft characteristics are provided by the laboratory session's statement and are the following:

Parameter	Value
Wingspan: b	47.574 [m]
Surface: S	286.15 [m^2]
Aspect Ratio: AR	7.9
Oswald Efficiency: e_0	0.8
Zero-lift drag coefficient: C_{D0}	0.018
T max @ sea level: $T_{max}\|_{SL}$	444822 [N]
SFC_T	$1.67 \cdot 10^{-4}$ [1/s]
MTOW	$1.33 \cdot 10^6$[N]

Methodology employed for the calculation will include certain assumptions and formulas that are detailed below:

- The **Drag** is assumed to follow what is called a polar.
- Calculations for the **ambient conditions of the airplane** follow the ISA (International Standard Atmosphere) from a subroutine provided in the laboratory session.
- The employed thrust model is given by the following equation:

$$T(\Pi, h) = \Pi \cdot T_{max}|_{SL} \cdot \left(\frac{\rho(h)}{\rho|_{SL}}\right)^{0.7} \quad (1)$$

All the different parameters, functions and scripts can be downloaded at any time at the official repository.

2 Cruise maneuver

This is the first phase of the laboratory session. Equations used for this part are introduced in subsection 1.1. The state vector is $u_s = [v, x, h, W]$ being the control vector $x_c = [\Pi]$. Note that during cruise, the variation of vertical speed as well as vertical position has to be 0 in order to keep a constant altitude, being the vertical forces equal. Therefore $L = W$, obtaining that $C_L = 2W/\rho S v^2$.

2.1 State variables evolution

For this first part, a vector $u_s(0) = [v_0 = 1.2 \cdot v|_{EM}, x_0 = 0, H_0 = 10000m, W_0 = MTOW]$ containing the initial conditions to allow the built-in ODE Matlab solver "ode45" do its job. One the function "ODE_Cruise" integrates the equations of motion of the aircraft, four plots are obtained for velocity, distance, height and weight versus the time elapsed. These figures are displayed below:

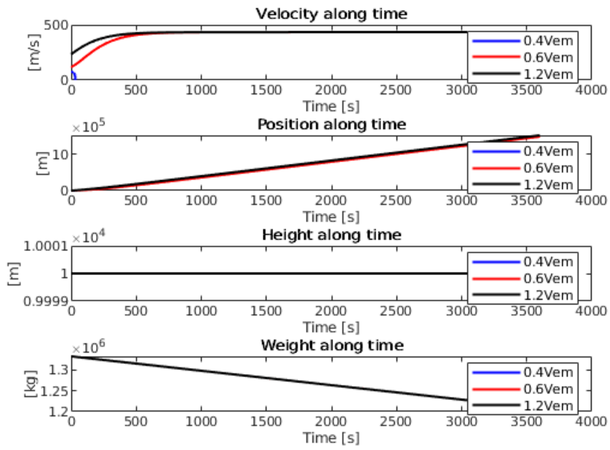

Figure 1: State variables at cruise conditions

As established in the statement, three initial conditions for the velocity are tested. Although the ones for 0.6 and 1.2 reach an stable solution, this does not happen for the velocity $0.4V_{Em}$, since it is not enough to maintain this maneuver along time.

2.2 Performance analysis

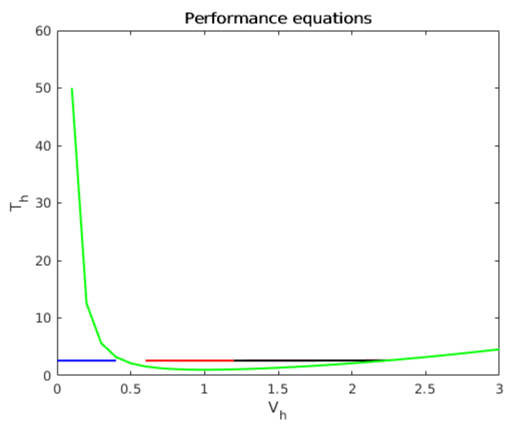

Figure 2: Performance analysis for cruise condition

As previously, it can be seen that an stability solution is reached for the two last cases, since their evolution along time follows the ideal plot. However, for the 0.4 case a solution is reached in the unstable part of the graph, meaning that this configuration is not valid in real world as previous state plot was showing.

3 Climbing Analysis

For this simulation is supposed that the aircraft is having a constant elevation angle. In particular, the value of this angle is -2^o, which means the aircraft is in descending configuration.

3.1 State variables evolution

For a vertical maneuver, the control input vector needs two variables: Π and γ, where this last one is the climbing angle. In particular, two cases are simulated depending on different initial velocities configuration:

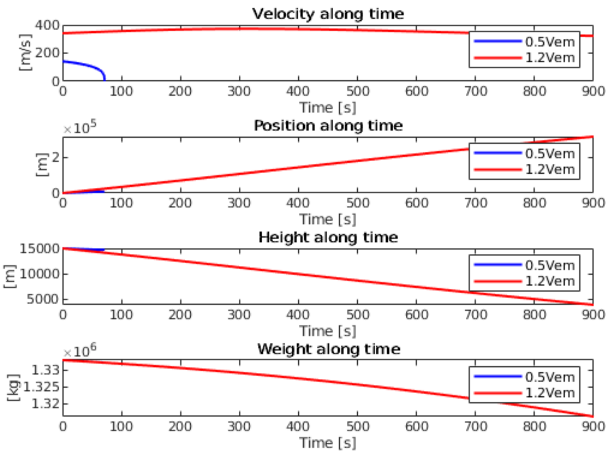

Figure 3: State variables at climbing conditions

It can be seen that for the case where the velocity is $0.5 \cdot V|_{EM}$, no convergence is reached and therefore and optimal solution is found. Velocity quickly decays to 0 after arround 70 seconds (maybe due to drag overcoming the other aerodynamical forces and resulting in an airplane's stall). This result is closely related with the stability behavior of an ODE which, depending on the constants that relate with the variables, can behave in very different ways. In fact, depending on the arrangement of the result of the integration (ode45 solves the equation numerically so the integrated equation cannot be seen), if under a certain constant an exponent becomes negative, the solution will decay over time (which is this case). Since velocity falls to 0, the evolution of the other state variables stops (as it can be seen in the three other graphs below that of the velocity vs. time).

3.2 Performance analysis

For the case of the performance analysis, different isolines at several climbing angles are plotted in a $\hat{v} - \hat{T}$ diagram:

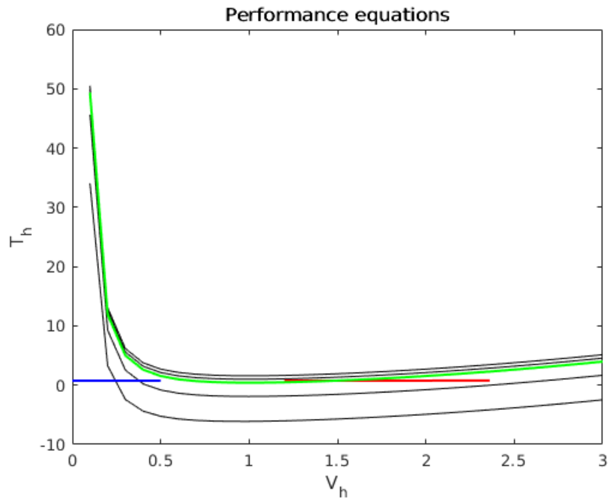

Figure 4: Performance analysis for climbing conditions

As we can see, the elevation angle strongly affects the diagram, and this indeed makes sense: the steeper the descent angle is (since it starts at +2^o and ends in -25^o) the less thrust is needed to maintain a certain velocity. This is why as the vertical angle gets each time more negative, the isolines displace downwards meaning that each time less thrust is needed (since during a descent an airplane is transforming potential energy into kinetic energy, thus increasing its velocity).

4 Turning analysis

For this last simulation, a turning maneuver is analyzed. These maneuvers are mainly characterized by the μ angle. Along this maneuver it is imposed that the bank angle is the same and altitude is maintained. Different factors can be studied depending on this angle.

4.1 Contour plots analysis

In order to see all the possible turning configuration, we can make a contour plot as suggested by the statement.

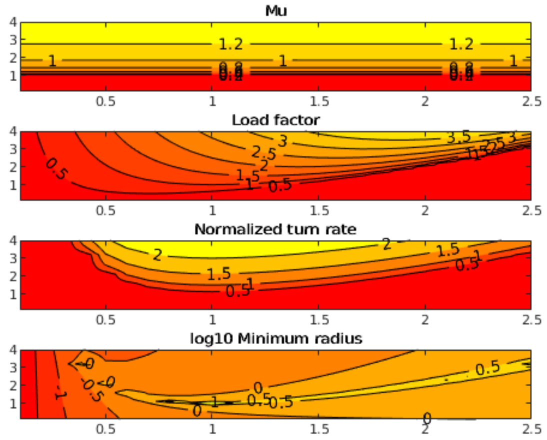

Figure 5: Contour plots for different μ conditions

4.2 State variables evolution

If we take a look to the state variables evolution figure, we can notice several things. Weight decreases since fuel is being consumed by the aircraft in a linear way due to a constant thrust. Also, velocity reaches a maximumm value along the maneuver.

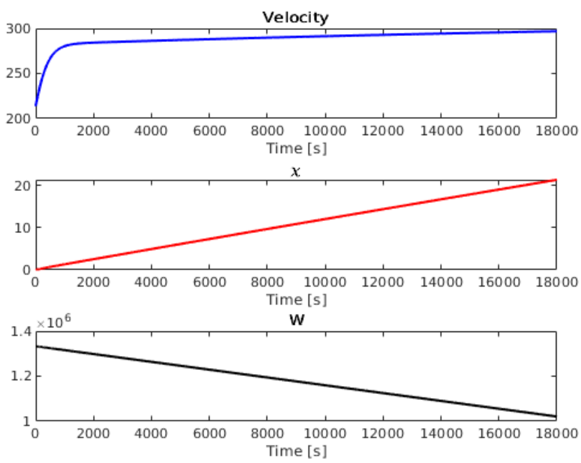

Figure 6: State variables evolution for turning conditions

More interesting is to make a plot of the two coordinates at which the aircraft is located along the maneuver:

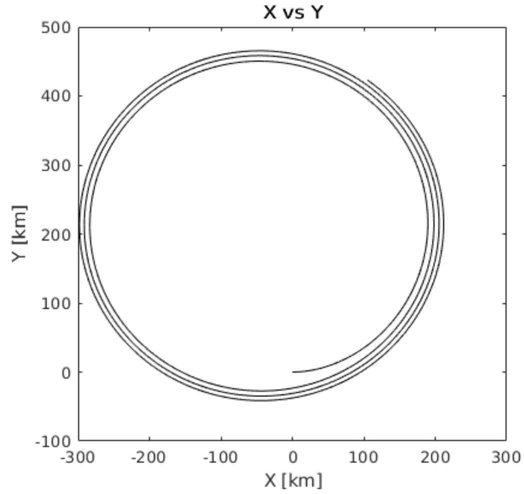

Figure 7: Trajectory along the turn

In previous graph it is easy to check that the radius of the turn is not constant, evolving as an spiral along time. This can be understood because of the loading factor: as pilot starts to begin the maneuver, speed increases. This increment in the velocity affects directly the dimension of the radius, which can be checked in the last subplot of the contour figures. As a result, an spiral trajectory is obtained.

4.3 Performance analysis

Finally and as done before, it can be checked that the analytical solution coincides with the ideal one for this case, since both curves are smooth and converge to the stable solution.

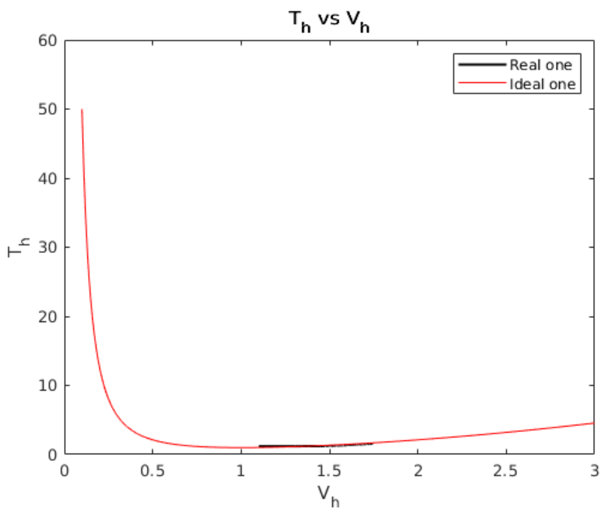

Figure 8: Performance at turn maneuver

5 Simulink results

After building this model with Simulink, results are shown in the following figures:

Although Matlab and Simulink end up by performing the same calculations / operations, Matlab's assembly is carried out via text build-up. On the other hand, Simulink is arranged in a pure graphical way, making the development of the laboratory session much more visual and easy-to-comprehend. Whereas a Matlab code consist on a series of code arrays combining variables, functions and mathematical calculations, a Simulink model is built by connecting with arrows different functions with their corresponding relationships, so that the same output is achieved but in a visual arrangement, making this last procedure more suitable for someone to evaluate and understand the operations carried out during the development of the laboratory session.

For simple projects like this one, the use of Simulink is simply a lost of time, since the GUI is slow. However, if we were analyzing the complete behaviour and state of the aircraft all along the different flight phases along the time it would be extremely useful. The different blocks would help to see in a clearly way how the model is built, however a strong base on how to use Simulink is requiered.

6 Conclusion

This laboratory show the different between directly solving the state equations and compare them with the performance ones. Furthermore, by using programming functionalities we could run up different simulations for initial conditions which gave us a deep understanding of the behaviour along all the flight conditions and different maneuver phases.