LAST project-Reusable earth Launcher

July 24, 2019

1 Introduction

Rocket design is one of the most complex engineering projects Humankind has ever tackle. Indeed a launcher is subjected to numerous disciplines involving almost every aspects of mechanical physics: Aerodynamics, structures, trajectory, propulsion, control and automation...
In recent years, after the rise of private companies providing "cheap" access to space, Europe and other space companies have set their objective to reduce their cost to reach orbit. In order to full fill that goal, to reduce the cost of rocket design is a mandatory requirement. However, the task is far from easy as these disciplines previously mentioned are highly dependent and connected to each other. In the past, the different disciplines have been optimized individually, so multiple iterations in the design were necessary and not cost effective.
Nonetheless, in the last 10 years or so, a new method of design engineering systems has emerged. Especially in the pre-design phase. It is called Multidisciplinary Analysis Optimization or MDAO, at which an objective function is to be minimized with respect of a set of design variables while the outputs of the disciplines are to respect some constrains. This method allows for the disciplines involved in rocket design to be optimize at the same time. If you want to know more about MDAO please see [6] Among all the different softwares available to do MDAO, it was decided to use OpenMDAO, which is an open source python based framework that gathers all the necessary tools to solve MDAO problems. This software was used to encourage cooperation between different researchers. Please refer to [7] to see more details about this tool.
It is in this context that LAST was created. LAST or Launcher Analysis Sizing Tool is a tool to allow ISAE-SUPAERO to be able to design a Launcher from scratch, to accomplish a certain mission, taking into account all the disciplines involved in rocket design.

2 Modules

In this section the different models used for LAST are explained. For now LAST has 4 disciplines: Propulsion, trajectory, mass and structures, and Aerodynamics. The idea of this first iteration of LAST was to have a complete description of the main disciplines involved in rocket design, and to obtain some early results in the optimization process.

2.1 Propulsion

The propulsion module was computed based on [2] and [1]. The inputs and outputs of the modules are the following:

inputs	ouputs
P_c	Isp
P_e	ε
OF	c^*

Assuming that there was an equilibrium in the chamber, we were able to use the python package RocketCEA (coming from NASA CEA calculations), the temperature of the chamber, the molecular mass at combustion (kg/kmol) and the isentropic coefficient at throat are calculated. The expansion ratio of the nozzle ε can then be computed by:

$$\varepsilon = \frac{\Gamma}{\sqrt{\frac{2\gamma}{\gamma-1}(\frac{pe}{pc})^{\frac{2}{\gamma}}(1-(\frac{P_e}{P_c})^{\frac{\gamma-1}{\gamma}})}} \tag{1}$$

where Γ is calculated by:

$$\Gamma = \sqrt{\gamma(\frac{2}{\gamma+1})^{\frac{\gamma+1}{\gamma}}} \tag{2}$$

The combustion characteristic speed c^* is calculated by:

$$c^* = \eta_c \frac{\sqrt{T_c R}}{\Gamma} \tag{3}$$

where $\eta_c = 0.98$ is the combustion efficiency. The thrust coefficient c_f is calculated by:

$$c_f = \Gamma\sqrt{\frac{2\gamma}{\gamma-1}(1-(\frac{p_e}{p_c})^{\frac{\gamma-1}{\gamma}})} + \frac{\varepsilon}{p_c}(p_e - p_a(r)) \tag{4}$$

Where $p_a(r) = p_a(0) = 1$ atm for now. Using (3) and (4), the Isp can be computed as:

$$Isp = \frac{\eta_n c_f c^*}{g(r)} \tag{5}$$

where η_n is the efficiency of the nozzle and $g(r) = g_0$ for now.

2.2 Aerodynamics

For now, in the aerodynamics module it was assumed that either the drag coefficient is constant c_d=0.3 for the whole trajectory, or that it was obtained from data coming from [3] from actual missile values. In both cases it is an approximation, and different values of c_d will give a different trajectory, which will have an impact over the others modules. For that reason, a more complex approximation of this coefficient (basic CFD) should be performed using the geometry of the rocket in the future.

2.3 Mass and geometry estimations

The mass modules has the following inputs-outputs:

inputs	outputs
P_c	m_d
ε	A_t
OF	A_e
c^*	
m_d	
m_p	
n_{axial}	

The rocket modeled is made by 2 stages with cryogenic propulsion in both stages. As for this moment only one propellant has been implemented: The propellant used was LOX/H2. All the models and assumptions are done for this propellant, and only this one. As it will be seen later, out model was only validated for LOX/H2 as it did not behave correctly for other propellants. Every stage is made of the following structures:

- The tanks: one for the fuel and one for the oxidizer.

• The turbopumps: to supply the propellant in the combustion chamber.

• The pressurized gas and its tanks, to allow that the pressure in the tanks will be constant during combustion, for which we used Helium.

• The engine: made by the combustion chamber and the nozzle.

• An additional mass that takes into account the errors made in mass estimation and everything else that was not considered, such as the structures, plumbing and connections ...

The approximation of the masses that were used were taken from both [2] and [1]. First, basic computations that are used throughout the calculations will be shown:
Mass of fuel and oxidizer:

$$m_{p_{LOX}} = \frac{m_p OF}{OF + 1}$$

$$m_{p_{H2}} = \frac{m_p}{OF + 1}$$

Volumes of propellant:

$$V_{LOX} = \frac{m_{p_{LOX}}}{\rho_{LOX}}$$

$$V_{H_2} = \frac{m_{p_{H_2}}}{\rho_{H_2}}$$

where $\rho_{LOX} = 1141 kg/m^3$ and $\rho_{H_2} = 71 kg/m^3$.
For a given material, let's define a parameter Φ:

$$\Phi = \frac{\sigma_{max_m}}{\rho_m k_T n_S g_0}$$

where k_T is the "shape factor", $k_T = 1.5$ for spherical shape and $k_T = 2$ for our tanks shape, $n_S = 2$ is the security factor. In our case the material used was aluminum ($\rho_{alu} = 2800 kg/m^3$ and $\sigma_{max_{alu}} = 400 MPa$).
Knowing the global flow rate, the corresponding fuel and oxidizer's flow rate are written:

$$\dot{m}_{LO_X} = \frac{\dot{m} OF}{OF + 1}$$

$$\dot{m}_{H_2} = \frac{\dot{m}}{OF + 1}$$

From the propulsion and trajectory modules, the surface of the throat A_t and exit of the nozzle A_e are computed, under isentropic conditions:

$$A_t = \frac{c^* \dot{m}}{P_c}$$

$$A_e = \varepsilon A_e$$

The mass approximations are the following:

2.3.1 tanks

The dry mass of a tank made by a cylinder with two spherical callotes is given by:

$$m_{tank} = \frac{V_p P_p}{g_0 \Phi} \tag{6}$$

where V_p the volume of the propellant (LOX or H_2) and P_p the pressure of the propellant (LOX or H_2) and it is calculated with the following regression linking the pressure and the volume:

$$P_p = (10^{-0.1068(\ln V_p - 0.2588)})10^6 \tag{7}$$

2.3.2 turbo pumps

The required power of a pump is given by:

$$W_p = \dot{m}_p \frac{(P_p - P_c)}{\rho_p \eta_{tp}} \tag{8}$$

where $\eta_{tp} = 0.75$ corresponds to the turbo pump efficiency. The speed of ration of each pump is calculated by:

$$\omega_p = 2\pi \frac{n_p}{60} \tag{9}$$

where n_p is the number of laps per minute of each pump and $n_{LOX} = 10^4$, $n_{H_2} = 3 \times 10^4$. The dry mass of the turbo pumps are estimated by:

$$m_{tp} = A(\frac{W_p}{\omega_p})^B \tag{10}$$

Where $A = 1.5$ and $B = 0.6$ [1] for LOX/H2.

2.3.3 combustion chamber

For the combustion chamber, it was estimated that inside the chamber the flow is at a certain Mach number before going through the convergent. Using this assumption, the surface of the combustion chamber can be estimated using the empirical equation:

$$A_c = A_t(8(100D_t)^{-0.6} + 1.25) \tag{11}$$

The length of the chamber can be estimated by:

$$l_c = l^* \frac{A_t}{A_c} \tag{12}$$

Where $l^* = 1.02$ is the equivalent length for LOX/H_2 [1]. The height of the conical section is:

$$h_{cone} = \frac{D_c - D_t}{2 \tan\theta} \tag{13}$$

Where $\theta = 45°$.Finally the dry mass of the combustion chamber is computed:

$$m_{cc} = (D_c l_c + \frac{R_c^2 - R_t^2}{\sin\theta}) \frac{\pi P_c D_c \rho_s n_s}{2\sigma_s} \tag{14}$$

where $\sigma_s = 310MPa$ and $\rho_s = 8000kg/m^3$.

2.3.4 nozzle

The nozzle is assumed to be a cone with two circular bases. The area of the cone is given by:

$$A_n = \pi \frac{R_e^2 - R_t^2}{\sin\theta} \tag{15}$$

The dry mass of the cone is computed:

$$m_{cone} = \frac{\rho_s A_n P_c D_c n_s}{2\sigma_s} \tag{16}$$

where $\sigma_s = 310MPa$ and $\rho_s = 8000kg/m^3$.

2.3.5 pressurizing gas

The mass of the pressurizing gas accounts for the gas itself as well as for the tanks (one for the fuel tank and one for the oxidizer tank). It is as follows:

$$m_{pg} = \frac{\gamma_g V_p P_p}{1 - \frac{P_p}{P_g}} \left(\frac{1}{R_g T_g} + \frac{1}{g_0 \Phi}\right) \tag{17}$$

where $\gamma_g = 1.67$, $R_g = 2078 J/kg.K$, $T_g = 300K$ and $P_g = 5P_p$.

2.3.6 additional mass

The additional mass was computed in order to take into account the everything that was not considered in the mass estimations [2]. It is assumed to be directly linked to the propellant mass and that it is mainly composed by the structure of the rocket:

$$m_{add} = m_p(-2.3 \times 10^{-7} m_p + 0.07) \tag{18}$$

The fairing and the skirt of the launcher are not implemented in the calculations yet.

2.3.7 dry mass

The dry mass of the system is the sum of all the components previously described:

$$m_d = c_t(m_{add} + m_{eng} + m_{sys_{LOX}} + m_{sys_{H_2}}) \tag{19}$$

where
$m_{eng} = m_{cc} + m_{cone}$
$m_{sys_{LOX}} = m_{t_{LOX}} + m_{tp_{LOX}} + m_{pg_{LOX}}$
$m_{sys_{H_2}} = m_{t_{H_2}} + m_{tp_{H_2}} + m_{pg_{H_2}}$
and c_t is a coefficient coming directly from the trajectory module linking the maximal load factor during ascent and the structure module [2]:

$$c_t = 0.02 n_{max} + 1 \tag{20}$$

2.3.8 propellant mass

In order to estimate the propellant mass, we use the inert fraction f:

$$f = \frac{m_d}{m_d + m_p} \tag{21}$$

Knowing the dry mass and the inert fraction of the rocket, the propellant mass is easily obtained.

2.4 Trajectory

The trajectory module has the following inputs-outputs:

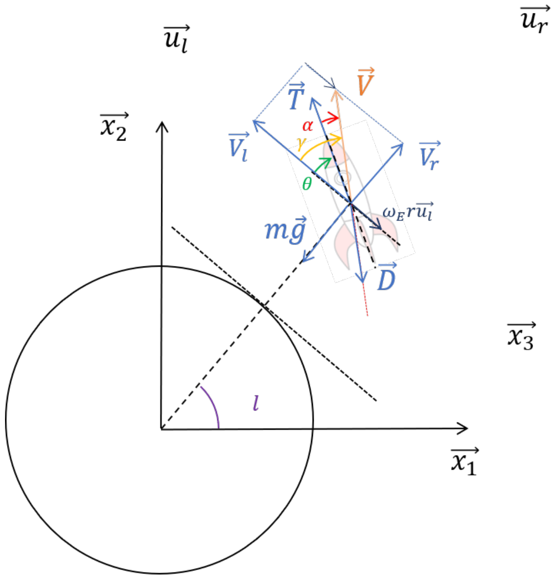

Figure 1: Earth Centered, Earth-non fixed reference frame

inputs	outputs
C_X	A_{s1}
m_{p1}	T_1
ISP_1	T_2
c_1^*	A_{s2}
m_{d1}	n_{max}
A_{e1}	r
TW_1	V
C_1	$\dot{m}_1$
m_{p2}	$\dot{m}_2$
ISP_2	α
C_2	θ
c_2^*	γ
m_{d2}	
A_{e2}	
TW_2	

Where the index 1 refers to the first stage and 2 to the second stage. For the trajectory of a double stage rocket, two different models with different parametrizations of the trajectory were used. Until today, no decision on which parametrization to use was decided so both models will be presented.

2.4.1 Planar trajectory in Earth Centered, Earth non fixed reference frame

The first trajectory parametrization used is derived from the following classical 3D dynamics equations, written in an Earth-centered, inertial referential as seen in figure 1

$$\begin{cases} \dot{r} = V_r \\ \dot{l} = \frac{V_r}{r^2} \\ \dot{V_r} = \frac{V_t^2}{r} + \frac{T}{m}\sin l - \frac{\mu}{r^2} - D_f V_r \\ \dot{V_l} = -V_t \frac{V_r}{r} + \frac{T}{m}\cos l - D_f (V_l - \omega_E r) \\ \dot{m} = -\frac{T}{g_0 Isp} \end{cases}$$

where: r: altitude (m), V_r and V_l: are the two components of the velocity (m/s), l: longitude from the ground (rad), m: mass (kg), θ: pitch angle (rad), T: thrust (N), $\dot{m}$: mass flow rate (kg/m^3), μ: standard gravitational parameter (m^3/s^2), ω_E: Rotation rate of the Earth (rad/s), Isp: Specific impulse of the rocket (s), g_0: acceleration of gravity at r = 0m (m/s^2) D_f: the 'drag factor' (s^{-1}):

$$D_f = \frac{1}{2}\rho u C_f \frac{A}{m} \tag{22}$$

and V the relative speed with respect to the air:

$$V = \sqrt{V_r^2 + (V_\theta - \omega_E r)^2} \tag{23}$$

The pitch angle θ (of the first stage) is computed through a simplified command law:

$$\theta(t) = \frac{\pi}{2}, 0 \leq t \leq t_v \tag{24}$$

$$\theta(t) = \frac{\pi}{2} \frac{1}{t_v - t_a}(t - t_a), t_v \leq t \tag{25}$$

where t_a is the duration of the pitch over maneuver, t_v is the time when the rocket begins the pitch over maneuver.

2.4.2 Planar trajectory in Earth-centered, Earth-fixed reference frame

The second model is based on [3], where a planar trajectory on the equatorial plane in a non rotatory Earth is considered, as seen in figure 2 . For that description the equations of Motion are the following:

$$\begin{cases} \dot{r} = V \sin(\gamma) \\ \dot{V} = -\frac{1}{2m}\rho S_{ref} C_X V^2 - g(r)\sin(\gamma) + \frac{T}{m}\cos(\theta - \gamma) \\ \dot{\gamma} = \left(\frac{V}{r} - \frac{g(r)}{V}\right)\cos(\gamma) + \frac{T}{mV}\sin(\theta - \gamma) \\ \dot{l} = \frac{V}{r}\cos(\gamma) \\ \dot{m} = -q \end{cases}$$

where: r: altitude (m), V: norm of velocity (m/s), γ: flight path angle (rad), l: longitude (rad), m: mass (kg), θ: pitch angle (rad), S_{ref}: surface of reference (m^2), T: thrust (N), ρ: air density (kg/m^3), C_X: drag coefficient, $g(r)$: Earth gravity (m/s^2), q: mass flow rate (kg/m^3) and α: angle of attack (rad).
The command laws for the trajectory phases are as follows [3]:

- Lift-off: $\theta = \pi/2$

- Pitch-over maneuver: $\theta(t) = \gamma - \frac{\Delta_\theta * t}{\Delta_t}$, where Δ_θ is the variation (in rad) of θ during the maneuver and Δ_t is the duration of the maneuver (in s)

- Gravity turn: return to angle of attack of zero and then $\theta = \gamma$ ($\alpha = 0.$)

- Exo-atmospheric command for first stage: θ is optimized by a linear interpolation control law (parameters are way-points of θ as a function of time, 2 parameters are used: θ at the time at which the exo-atmospheric phase begins and θ at the end of the first stage flight).

- Second stage flight (bi-linear tangent command): $\theta(t) = \frac{a^\xi \tan\theta_i + (\tan\theta_f - a^\xi \tan\theta_i)t}{a^\xi + (1-a^\xi)t}$ where a is a constant, ξ a parameter defining the shape of the bi-linear tangent law, θ_i and θ_f the pitch angles at the beginning and at the end of the phase.

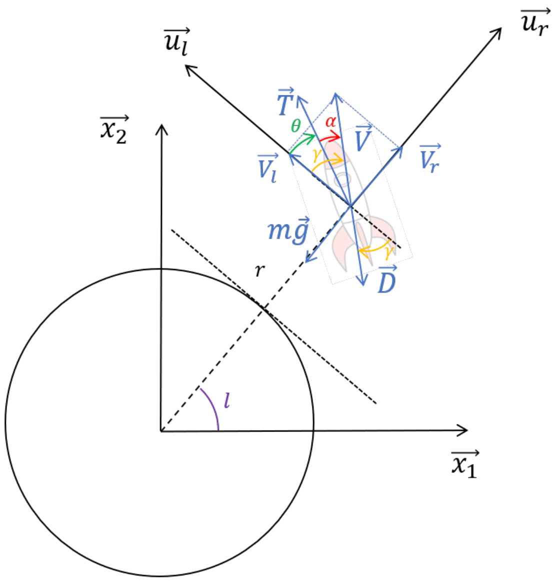

Figure 2: Earth Centered, Earth-fixed reference frame

Different events are simulated in this trajectory:

- Main Engine Cut Off (MECO) occurs when the propellant mass is used.

- The jettison of the fairing will occur when the flux is below 1135 $kg.m^{-2}.s^{-1}$.

- Impact, stops the integration if the altitude of the stage is 0m.

- The command law for the pitch angle θ during exo-atmospheric flight takes place when the dynamic pressure is below 1 kPa.

Besides the trajectory itself, this module allowed us to compute different quantities required for other modules, some assumptions were made along the way. The thrust (N) is assumed to be constant and equal to the initial thrust of the rocket:

$$T = TW \times GLOW \times g_0 \tag{26}$$

where TW is the initial Thrust to weight ratio, $GLOW$ is the initial mass of the rocket or Gross Lift-Off Weight. The flow rate (kg/s) of the stage is considered also constant and equal to the initial flow rate:

$$\dot{m} = n_e \frac{T}{ISPg_0} \tag{27}$$

where n_e is the number of engines of the stage. And lastly, the surface of the stage (m^2) is assumed to be directly related to the surface of the exit surface of the nozzle through the coefficient C:

$$A_s = A_e C \tag{28}$$

Finally, the axial load factor on the first stage is computed as follows:

$$n_x = \frac{T_1 - P_{dyn} C_x A_{s1}}{mg(r)} \tag{29}$$

In the end a trajectory with the V_r, V_l parametrization was used, but in a non rotational earth centered frame as seen in figure 2

2.5 Constraints on the modules

As the intention of the project is to perform an optimization on the design of the rocket, the disciplines and their components are subjected to a certain amount of constraints. For each module each constraint will be specify:

2.5.1 Propulsion

The Summerfield criteria gives a boundary on the exit pressure on the nozzle to avoid flow separation in an over-expanded nozzle [2]:

$$P_e \leq 0.4 P_a(r) \tag{30}$$

2.5.2 Trajectory

From the trajectory we have some constraints on the maximum values of the trajectory [3]:

$$\alpha_{max} \leq 15^\circ \tag{31}$$

$$n_{max} \leq 4.5 \tag{32}$$

$$P_{dyn_{max}} \leq 40kPa \tag{33}$$

$$\Phi_{max} \leq 100W/m^2 \tag{34}$$

Where Φ is the heat flux over the surface of the stage.

2.5.3 Mass and structures

The Diameter of the nozzle is constraint not to be bigger that the diameter of the stage [1]:

$$D_e \leq 0.8 D_{stage} \tag{35}$$

The pressure of the tanks are bounded by conception limits [1]:

$$0.2MPa \leq P_p \leq 0.5MPa \tag{36}$$

The Mach at the exit of the combustion chamber is also constraint [3]:

$$0.2 \leq Ma_c \leq 0.4 \tag{37}$$

where the Mach can be calculated using the isentropic relations at the chamber and the throat [3]:

$$\frac{A_c}{A_t} = \frac{1}{Ma_c}[\frac{2}{\gamma+1}(1+\frac{\gamma-1}{2}Ma_c^2)]^{\frac{\gamma+1}{2(\gamma+1)}} \tag{38}$$

The ratio of the chamber diameter and length must be described as [3]:

$$0.5 \leq \frac{L_c}{D_c} \leq 2.5 \tag{39}$$

3 Results

3.1 XDSM diagramm

Figure 3 represents the XDSM diagram of the LAST. It can be seen how all the variables are connected to the other disciplines. For now, the desgin variables are the following: P_c, f, OF, C, $\theta(t)$ (the command law parameters) and TW.

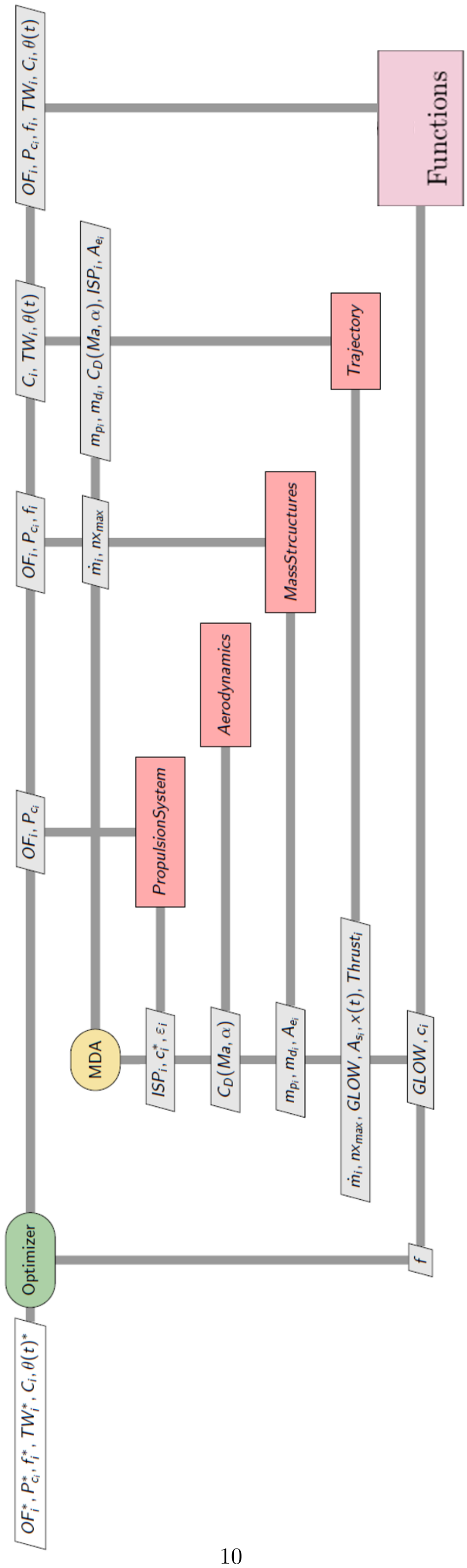

Figure 3: XDSM diagramm of LAST

3.2 Model validations

In order to validate mass and propulsion models, we try to obtain the data on multiple existing rockets in order to obtain the interesting data we needed to compare it to LAST models. The idea was to match the design variables with the values of real rockets and see the results we would obtain. We perform this comparison for only the first stage of each rocket, where with a simplified command law. This of course was done in the MDA calculations where no optimization occurs. Data from the rockets was directly obtained from [4] and [5] The first rocket we compare to was the first stage of the DELTA IV with the RS-68 engine. Then it was performed the same for Falcon 9 first stage with its 9 Merlin engines using (LOX –RP1) propellant. A comparison now with the second stage of Atlas V and Atlas III rockets was also made. Results can be seen in figure **??**.

We can see that for rockets using LOX/H2 propellant, the mass estimations of our models compares rather well to the real values. However, when the propellant is changed (Falcon 9 uses H2/RP1 propellant), the estimations of LAST differ. The model is then only validated with LOX/H2 propellants. In order to augment the range of work of LAST models taking into account have to be implemented.

3.3 Geometry generation

As the main objective of this project is to create a first version of the program for other students to work on, it was thought that it could be interesting to be able to create a geometry directly from the results of the MDA. Indeed, a more accurate estimation of the masses, or even FEM analysis could be performed on the structures to add to the complexity the model. In order to generate the geometry, the software OpenVSP (open source) was used. The geometry results from OpenMDAO were compiled in an .as file that would generate the geometry in OpenVSP. The idea is that having the mass and volumes of all the components of the structure, to calculate the thickness of those structures. For most components their thickness is directly linked to the stress endured by the structure as following:

$$e = \frac{n_s P_p R}{\sigma} \tag{40}$$

The problem is that without the length of the tanks we can not compute the thickness of the stage, and without the thickness of the stage we do not know the inner diameter of the tank. As it can be seen in figure 4 the following geometry needs to be respected at any time :

$$R_{tank} + e_{tank} + gap = R_{stage} - e_{stage} \tag{41}$$

To achieve it, the process is to give an initial length of both the tanks, to fix a gap between the outer surface of the tank and the inner surface of the stage, which will give us a first value for the R_{tank} which then will give us a specific length of the tanks, and by iterating will converge to the final geometry.

- The tanks are considered to be made by a cylinder with a volume of structure V_{tank} of length L_{tank} and with two hemi-spherical bases of an inner diameter D_{tank}. Knowing the volume of propellant and the "shape" of the tank, the length of the tank can be computed by:

$$L_{tank} = (V_{prop} - \frac{4}{3}\pi(\frac{D_{tank}}{2})^3)\frac{4}{\pi D_{tank}^2} \tag{42}$$

The thickness of the walls e of the tanks can be computed using (40) solving:

$$e = \frac{n_s P_{tank} R_{tank}}{\sigma_{tank}} \tag{43}$$

- The stage is made by a cylinder of length $L_{stage}$with circular bases of external diameter D_{stage}.It is considered that the total of the additional mass exclusively accounts for the stage. As an approximation the length is computed by:

$$L_{stage} = L_{LO_X} + L_{H_2} + 1m \tag{44}$$

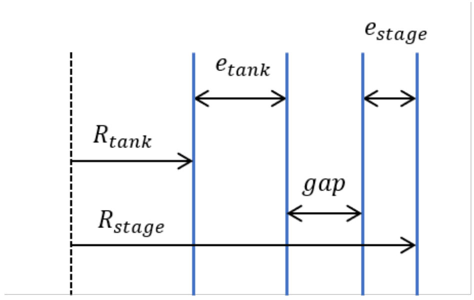

Figure 4: OpenVSP visualization of the LAST

As the mass of the stage is considered to come directly from the additional mass, no physical thickness was used to compute its mass. However we can get a "geometry" thickness using the following relation:

$$\rho_{st} = \frac{m_{st}}{V_{st}} \tag{45}$$

which, applied to the geometry of the tank (cylinder and circular bases), the thickness is obtained solving the following equation:

$$-2\pi L_{stage} e^2 + (\pi D_{stage} L_{stage} + \pi \frac{D_s tage^2}{2})e - \frac{m_{stage}}{\rho_{stage}} = 0 \tag{46}$$

- The nozzle's thickness is computed as follows

$$e_{nozzle} = \frac{P_c D_c n_s}{2\sigma} \tag{47}$$

- The combustion chamber's thickness is computed as follows

$$e_{cchamber} = \frac{P_c R_c n_s}{\sigma} \tag{48}$$

As for now, for each stage, only the tanks, the engine and the structures have been implemented. An additional skirt that joins the two stages and a fairing where also simulated. Once the geometry is generated,the code will tell the name of the file that was generated. This file script has to be run in the shell command using the following command line:

vsp -script NameOfTheScriptGenerated.as

This will generate the geometry of the launcher, as it can be seen in figure 5. To launch the geometry, it can be done either by clicking in the application file "vsp" and by opening the geometry file generated (.vsp3) or directly on the shell command line by typing:

vsp NameOfTheGeometryGenerated.vsp3

Note: to run these commands you have to be in the folder where the (.as) script is located.

3.4 Optimization

As the final goal of LAST is full-fill a certain mission while optimizing over the design variables, in this section we will discuss how the optimization is performed so far. The model describes both stages, but as for the trajectory, only the trajectory of the first stage with a simplified command law (25) is computed. In the future, a more complex trajectory is required to be able to optimize the full launcher.
As said before, the goal of the optimization is to reduce the GLOW or initial mass of the rocket. As for this version of LAST, the tool is able to optimize over the following variables:

- P_c, the pressure of the chamber (Bar) of the two stages. Where

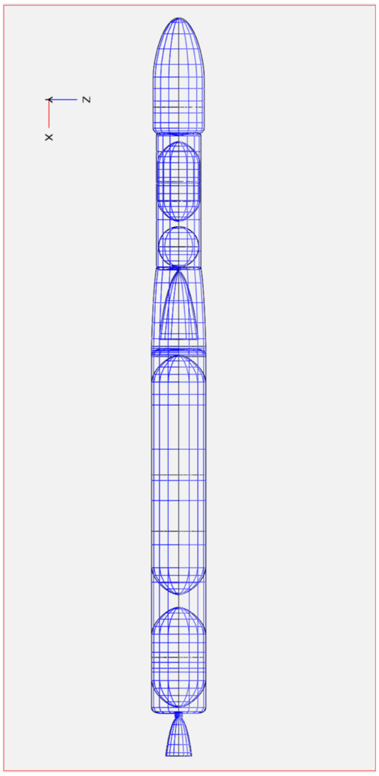

Figure 5: OpenVSP visualization of the LAST

$$99.2 \leq P_{c1} \leq 112.6$$
$$30 \leq P_{c2} \leq 34$$

- f, the inert fraction of each stage. Where:

$$0.08 \leq f_1 \leq 0.16$$
$$0.14 \leq f_2 \leq 0.17$$

- TW, the initial Thrust to Weight ratio of each stage. Where:

$$1.01 \leq TW_1 \leq 2$$
$$0.3 \leq TW_2 \leq 0.38$$

- C, the exit nozzle area to stage area ratio of each stage. Where:

$$0.2 \leq C_1 \leq 0.5$$
$$0.7 \leq C_2 \leq 0.8$$

- O/F, the mixture ratio of the propellant. Where:

$$5.92 \leq OF_1 \leq 6.05$$
$$5.83 \leq OF_2 \leq 6.05$$

- t_a, the pitch duration (s) of the maneuver of trajectory (25). Where:

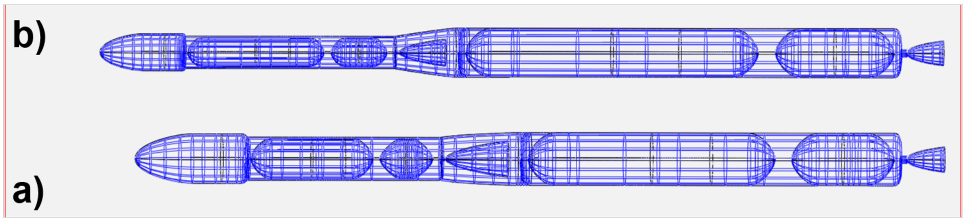

Figure 6: a) MDA, b) MDO

$$230 \leq t_a \leq 310$$

- t_v, the time of the beginning of the pitch maneuver (s) (25). Where:

$$10 \leq t_v \leq 40$$

Different constraints where applied to the different variables of the problem. However not all the constraints explained above have been coded yet. They are the following:

- Limit the minimum altitude of where stage separation occurs,

$$100km \leq h_{min}$$

- Limit the maximum axial load factor of the ascent [3],

$$n_{xmax} \leq 4.5$$

- Limit the maximum dynamic pressure during ascent trajectory [2]:

$$P_{dyn_{max}} \leq 40kPa$$

- Constraint the dimensions of the second stage wrt the first stage

$$1 \leq \frac{A_{s1}}{A_{s2}}$$

- Constraint the dimensions of the first stage Area, so it not too long with respect to its diameter (need a better description of this one, but the total length is calculated after the MDO)

$$11 \leq A_{s1}$$

The MDA (where all the calculations are performed) was solved using fix point iteration algorithm (NonLinearGaussSiedel) and a direct method algorithm. For the optimizer,the Sequential Least Squares Programming (SLSQP) solver was used. The optimizer used seemed not to be adapted to solve optimal control problems, so in the future other solvers using external packages (Dymos) should be implemented. Figure 6 shows the comparison between the MDA (representing a hybrid version the of Delta IV) and the MDO using the design variables and constraints previously mentioned, where the objective function was to minimize the GLOW. The MDO results in a longer and thinner both first and second stage, with larger nozzle and greater difference in the stage surfaces. Nonetheless, additional constraints need to be implemented to refine the optimization and the models.

4 Problems encountered and future work

4.1 Problems

The first set of problems was encountered when working in Windows' version of the machines at ISAE-Supaero. As a security measure, windows machines will not have administrator rights which complicates the download of EVERY package, software or tool of any kind. In my case, as my internship was lasting only 3 months, it was decided to use my personal computer. However, in the future, it is highly recommended to use ISAE Supaero's machines working in Linux environment. Indeed, optimizations' duration were increasing rapidly as the number of constraints and design variables were going up. (more than an hour for 11 design variables and 5 constraints).

Even if for the MDA (no optimization) a full description of the command law was implemented, the code seemed to have problems to be optimized. Indeed no solution was able to be found even if mathematically the system was solvable. It is for that reason that we scale down the trajectory to the description of only the first stage as explained in the previous section.
Moreover, the description of the design variables is far from trivial, as their range of variations might make the MDA not to converge or simply give an Error to the program. In the future, a Design of Experiment (DOE) should be implemented to know the reactivity of our model to the design variables variations. Apparently there is such a function already implemented in OpenMDAO to perform DOEs.
The implementation of vectors values as outputs of the trajectory module (such as V, α, γ ...) has a negative impact in the convergence of the MDA. Indeed the MDA will not converge when these outputs are declared. The state vectors were saved in a (.csv) file and plotted after the calculations were done. However, when the more full trajectory description was adopted, the state vectors were used and outputted to plot their corresponding trajectory, the optimization was impossible to perform.

To sum up the "coding" problems, we have in one side a program with a simple trajectory where the visualization of the trajectory is possible with a full optimization, and on the other hand a full description of the trajectory but with no optimization. However we believe that working on the simplified trajectory is the way forward as we can complexify the model as we go along and making sure that the optimization is still operational.

4.2 Future work

Here will be discussed non exhaustively some guideline for future development of LAST.
First in order to accelerate the code convergence we should declare analytically the partial derivative of the modules, instead of having the "approxtotals" function. Indeed the speed of convergence is presumably going to increase as the models complexify, so a robust and fast convergence is required.
For now the optimization of both the stages is done at the same time. However, as seen in [**?**], the "SWORD" have advantages in terms of overall efficiency of the optimization. This method consist in optimize each stage individually (with its corresponding modules and trajectories) which simplifies the global MDO problem.
Concerning now the trajectory module, having a description of the second stage is compulsory so this part has to be tackled within the shortest delays.

5 Conclusion

A first version of LAST has been implemented successfully. Multiple interconnected disciplines, essential to rocket design were coded. Moreover, successful optimizations of one objective function

subjected to various constraints have also been performed. At this point an expandable double stage LOX/H2 rocket has been created.
As every tool in its early days, there are things than work better than other, and it is still work in progress as of this day.
As said throughout the report, the models can be complexify to tackle a wider range of problems and missions: such as implement multiple propellants other than LOX/H2 and validate them, incorporate engine cycles (open and closed cycles) to each engine, or even create the reusable version of the first stage. Concerning the trajectory it is believed that the way to go is to incorporate optimal control laws.
Another point that has to be addressed will be the complete generation of the geometry of the rocket, with the implementation of all the components that were not addressed in this iteration.
Overall this project tackles every basic aspect of rocket design and it is thought to be built upon, giving ISAE-Supaero the tool to efficiently study rocket design and different concepts.

References

[1] Humble, R., Henry, G., and Larson, W. (1995). *Space Propulsion Analysis and Design.*, Mc-Graw Hill.

[2] M. Balesdent, *Multidisciplinary Design Optimization of Launch Vehicles*, Optimization and Control [math.OC]. Ecole Centrale de Nantes (ECN), 2011.

[3] FELIN project, 2019, Onera.

[4] F. Castellini, *Multidisciplinary Desgin Optimization for expandable launch vehicles*, Politecnico de Milano, 2012, Matr. 738531.

[5] www.astronautix.com

[6] Joaquim R. R. A. Martins, *A Short Course on Multidisciplinary Design Optimization*, University of Michigan, 2012.

[7] www.openmdao.org