Research project meeting summary: Trajectory Module for Launcher MDAO

Jorge L. Valderrama [1]
Dr. Annafederica Urbano [2] Dr. Mathieu Balesdent [3] Dr. Loïc Brevault [4]

[1]ISAE-SUPAERO, MSc. in Aerospace Engineering
[2]ISAE-SUPAERO, DCAS [3]ONERA, DTIS [4]ONERA, DTIS

April 24, 2020

Plan:

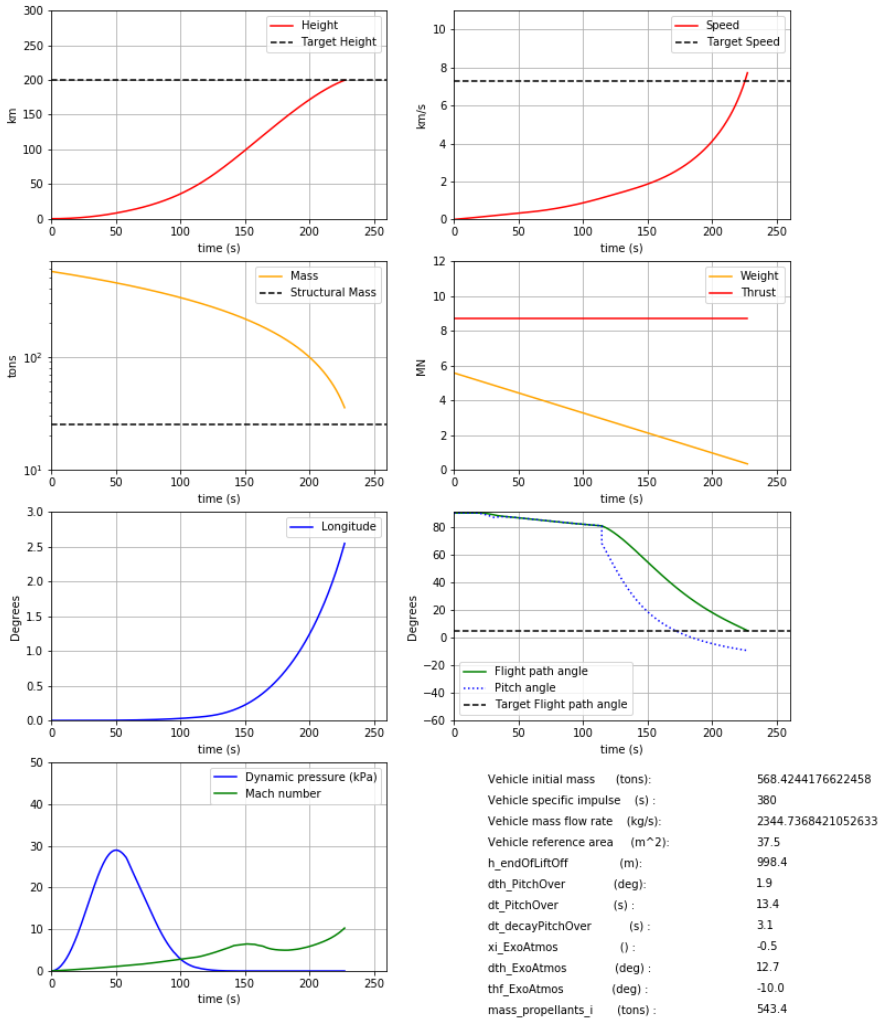
1. Review of previous work
2. Key points discussed
3. Future actions

Optimization of 2D trajectories

- I've implemented the optimization procedure using NLopt's COBYLA algorithm in Python. As suggested by Dr. Balesdent I'm now using the initial mass of propellants as an optimization parameter and my objective function is to minimize the consumed mass of propellants.
- The algorithm struggles to converge to feasible solutions when the initial guess is not feasible itself. But by running multiple optimization procedures and using the feasible solution they output as inputs for the next simulation, the program minimizes trajectories with flight path target angles up to 5 degrees. For values of zero degrees it still doesn't find feasible trajectories.

Review of previous work

Output of optimization



Vehicle initial mass	(tons):	568.4244176622458
Vehicle specific impulse	(s) :	380
Vehicle mass flow rate	(kg/s):	2344.7368421052633
Vehicle reference area	(m^2):	37.5
h_endOfLiftOff	(m):	998.4
dth_PitchOver	(deg):	1.9
dt_PitchOver	(s) :	13.4
dt_decayPitchOver	(s) :	3.1
xi_ExoAtmos	() :	-0.5
dth_ExoAtmos	(deg) :	12.7
thf_ExoAtmos	(deg) :	-10.0
mass_propellants_i	(tons) :	543.4
Iteration	:	166

Key points discussed

- The optimal solutions that I'm having as output have a really vertical trajectory and start the pitch over phase at really high altitudes of above 1 km. In reality this phase starts around 200 m, so it would be convenient to fix this parameter at that value and not use it as an optimization variable. This should favor less vertical trajectories.
- I should normalize the optimization parameters before inputting them into the COBYLA solver. This will favor more homogeneous steps along the search space.
- Dr. Balesdent could find a feasible trajectory with flight path target angle of zero deg using initial mass of 700 tons. So it should be a matter of testing to reach feasible solutions with the modifications I implemented.

Key points discussed

- The new step of the work will involve the calculation of analytic derivatives. We are going to use Dymos as it receives as inputs the Jacobians of the disciplines and then it computes the total derivatives using chain rule. I can use symbolic derivative libraries as Sympy to find the Jacobians and OpenMDAO has a function to check the partial derivatives using numeric propagation.
- This work will be based on Dymos' SSTO tutorial and will start by the implementation of a single phase but will then evolve to multiple phases as explained in the tutorial of the Multiple phase cannon ball and an example that will be shared by Dr. Brevault. I will implement the phases and control laws that I've already programmed in Python.
- We still need to define which solver to use in Dymos for the moment. Dr Balesdent and Dr. Brevault will discuss about that.

Future actions

- Implement the fixed threshold height for the lift-off phase and the normalization of the parameters. Try different settings to reach feasible orbital trajectories. We are almost there :)
- Start the new phase of the project using Dymos.