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, DECAS [3]ONERA, DTIS [4]ONERA, DCPS

April 6, 2020

Plan:

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

Review of previous work

S2 Progress report feedback

- EoM: Confusing ”-” sign because of inverted terms in for α

$$\dot{r} = v \sin(\phi) \tag{1}$$

$$\dot{\lambda} = \frac{v \cos(\phi)}{r} \tag{2}$$

$$\dot{v} = \frac{-D + T \cos(\phi - \theta)}{m} + \left(-g + \omega^2 r\right) \sin(\phi) \tag{3}$$

$$\dot{\phi} = \frac{L}{mv} - \frac{T \sin(\phi - \theta)}{mv} + \frac{(\omega^2 r - g)\cos(\phi)}{v} + 2\omega + \frac{v \cos(\phi)}{r} \tag{4}$$

$$\alpha = \theta - \phi = -(\phi - \theta) \tag{5}$$

- Include indirect methods
- Command laws for Reusable Launch Vehicles

Review of previous work

Implementation of 2D EoM in Python

- Python's solve_ivp : Runge-Kutta in Dymos, precision
- Lift model
- Events for Single-Stage-to-orbit: Vertical ascent, pitch over, gravity turn, Bi linear tangent law

Key points discussed

On S2 progress report feedback

- Include 2-3 sentences summarizing the importance of indirect methods as analytic solution
- Dr. Balesdent shared with me an article about control laws for reusable launch vehicles. Expand information on ”control laws” section

On Python implementation of 2D EoM

- zero lift model is good approximation at the beginning as pitch over maneuvers occur at low speeds and lift coefficient is small.
- solve_ivp: option ”Dense Output = True” allows access to interpolated solutions for every time input. This solver allows different phases.
- Propulsion: Losses up to 10% in the nozzle must be modeled. use Castellini’s model.
- Predefined sequence of the different phases for the beginning of the project

Future actions

- Read and follow Dymos tutorial for Single Stage to Orbit. It may serve as validation for my implementation.
- Check for possible error on my code causing weird behavior when Earth is rotating. Implement phases and try non-gradient-based optimizers from SciPy as COBYLA as a fast and easy to implement approach (Keep in mind that objective is to use gradient-based optimizers with analytic derivatives).
- Implement the given feedback on S2 progress report
- Next meeting on 2020/04/14 at 14h:00