Master of Science in Aerospace Engineering

Research Project S2 Progress Report

Trajectory Module for Launcher Multidisciplinary Design, Analysis and Optimization

Jorge L. valderrama

supervised by:

Dr. Annafederica urbano
DCAS, ISAE-SUPAERO

Dr. Mathieu balesdent
DTIS, ONERA

Dr. Loïc brevault
DTIS, ONERA

Starting date of project: January 27, 2020
Submission date: April 8, 2020

Contents

1 Introduction

In the design of a launch vehicle it is desired to obtain the maximum performance at the lowest cost possible for a given objective mission. In the pursuit of these aims, the interactions between multiple disciplines must be taken into account, for example: trajectory optimization, structures, propulsion and aerodynamics.

In a traditional design approach, individual optimization of each discipline is performed and its outputs are transferred to the design offices in charge of the other disciplines in an iterative process. This approach does not take into account all of the complex interactions between the different disciplines thus yielding non-optimal results.

In the quest for designing more performant and lower cost launch vehicles, Multidisciplinary Design Analysis and Optimization (MDAO) approaches have been been used to better assess the complex interactions among disciplines. NASA's OpenMDAO is one of the platforms enabling this integration of disciplines through efficient gradient computations for the optimization process, allowing parallel computing in multiple processors.

Trajectory optimization is a central approach to launch vehicle MDAO. Due to the high complexity of the mathematical models describing the different disciplines, it is important to integrate an efficient trajectory optimization method into the MDAO model to keep low computing times. Packages as DYMOS integrate highly accurate and efficient direct pseudospectral methods with the OpenMDAO platform to achieve this goal.

The Launcher Analysis Sizing Tool (LAST) is being developed at ISAE-SUPAERO's SACLAB using OpenMDAO. Structure and propulsion design are being integrated as disciplines into it as well as a 2D trajectory module. The subject of this research project is going to be the integration of a 3D trajectory module for expendable and reusable launchers using pseudospectral methods with the Dymos package into the LAST tool.

2 Aim

To develop and integrate an efficient and accurate 3D trajectory optimization module for a launcher into the LAST tool.

3 Objectives

- To implement trajectories of several level of complexity , including 2D and 3D
- To include different command laws for expendable and reusable launchers under different stage recovery scenarios
- To provide the analytic derivatives required for the MDAO for each case

4 Literature Review

As a first topic, this literature review will asses the governing equations of motion (EoM) for launcher trajectories for 2D and 3D models in spherical coordinates and some strategies to cope with the singularities of the 3D case, followed by the control laws used for launcher ascent and some recovery scenarios. Then, a review of some optimal control strategies used for trajectory optimization will be presented, justifying the advantage of using direct pseudospectral methods. Some case studies demonstrating the advantages of using OpenMDAO in trajectory optimization problems will be discussed and finally the use of optimal control control strategies within the MDAO frame will be treated.

4.1 Equations of Motion (EoM)

The model to simulate launcher trajectories is commonly simplified by neglecting rotation dynamics in order to keep low computation times. This simplification is based on the assumption that attitude control system of the launcher ensures the right attitude at every time.

Additional simplifications for the model can be done by assuming motion within the equatorial plane resulting in the 2D model represented by the set of 4 differential equations shown below in spherical coordinates and whose demonstration is included in the appendix A with a graphical explanation of the variables in figure 1.

$$\dot{r} = v \sin(\phi) \tag{4.1}$$

$$\dot{\lambda} = \frac{v \cos(\phi)}{r} \tag{4.2}$$

$$\dot{v} = \frac{-D + T\cos(\theta - \phi)}{m} + \left(-g + \omega^2 r\right)\sin(\phi) \tag{4.3}$$

$$\dot{\phi} = \frac{L}{mv} + \frac{T\sin(\theta - \phi)}{mv} + \frac{(\omega^2 r - g)\cos(\phi)}{v} + 2\omega + \frac{v\cos(\phi)}{r} \tag{4.4}$$

Where ω is Earth's angular velocity, λ is the longitude, r is the distance from Earth's center to a fixed point in the launcher (Ex.: the geometric center), θ is the pitch angle, ϕ is the flight path angle, α is the angle of attack, m is the mass of the launcher at any given time, g is the acceleration of gravity, D is the drag force, L is the lift force, T is the thrust force and v the speed. Given that launchers are a variable mass system, an additional equation must be solved to account for the change in mass. This 2D simplification will be used in the early stage of the project as it allows a faster implementation and verification.

To describe the motion of the Launcher in 3D space, a set of 6 differential equations is commonly used in the literature. In the work of Briese *et al.* [1], the following coordinate systems are defined to obtain the set of equations in spherical coordinates:

- Earth Centered Inertial (ECI)
- Earth Centered Earth Fixed (ECEF)
- North-East-Down (N)
- Kinematic (K)
- Body fixed (B)

The different transformations between the coordinates systems are presented and a transformation matrix to express the launcher's position according to the *World Geodetic System 1984* (WGS'84) is given. Also in this work, a transformation between the ECEF and N coordinate systems is used to avoid the singularity presented when the velocity is near $0m/s$ and the flight path angle near 90°. The EoM in spherical coordinates according to the *Aerospace Standard ISO 1511* are presented in the work of Castellini [2] in the context of MDAO for expendable launch vehicles. Cartesian coordinates to express the EoM were used by Volo [3] to perform trajectory optimization using pseudospectral methods, the EoM in these coordinates do not present the singularity of their spherical counterpart.

All of the aforementioned models neglect the rotation dynamics to reduce the number of equations and variables and thus the computing time, in their 3D version this simplification is said to have 3 degrees of freedom (3DOF). Berend an Talbot [4] described CNES's packages ORAGE and OPTAX, the former being used to perform atmospheric reentry optimization using 3DOF models and the later to optimize Ariane's trajectories and having the possibility of interconnection with 6DOF Ariane simulators. 6DOF models account for rotational dynamics thus requiring information about the inertia tensor of the launcher generating a system of 12 differential equations. Wang and Wu [5] compared the performance of 3DOF and 6DOF models for reusable launcher optimization using pseudospectral methods, finding the later to require 46% more time to be solved.

4.2 Forces

The EoM are function of gravity, thrust and aerodynamic force terms. In this subsection some approaches found in the literature referring to these topics will be described.

A simplified model for gravity is Newton's law of universal gravitation, this will be used in the early stage of the project. A more complete model accounting for Earth's elliptical shape is used in [2], by default it includes only the main gravitational field harmonic, but there exists the possibility to include up to the J_4 term too. The aforementioned models are broadly explained by Tewari [6]. Finally, the *Earth Gravitational Model 1996* (EGM96) is used in [1].

Aerodynamic data is usually provided in the form of tables where drag an lift coefficients are given as functions of the angle of attack and Mach number. Any interpolation used to integrate this tabular data into the trajectory optimization model must be C_2 continuous if Non-Linear-Programming (NLP) solvers are to be used as expressed by Betts in [7], he suggests to use B-splines for this purpose. The use of NLP solvers will be explained in the section covering Optimal Control. Variations on atmospheric properties were modeled with the *U.S. Standard Atmosphere 1976* in [2], [8] and [9], and in [3] the *COSPAR International Reference Atmosphere 2012* was used.

In the early stage of the project a very simplified thrust model will be used based on the specific impulse of the engine, a standard gravity parameter and the mass flow rate. A MDAO tool for optimizing the thrust system has been designed in another master thesis within the SACLAB and its integration within LAST and the trajectory module to be developed in this work could be considered.

4.3 Control laws

The control laws used to steer the launcher through a certain trajectory at a desired speed will be first described for the ascent phase and later for different recovery scenarios. To steer, a launcher can perform maneuvers with its gimbaling engines or its reaction control system (RCS) composed by thrusters with thrust components perpendicular to its longitudinal axis. The effect of these steering strategies is reflected on the pitch and yaw angles of the vehicle. In their works, Castellini [2] and Pagano [9] used similar approaches for the control strategy during the ascent phase of the launcher. This strategy considers the following sequence for the control of the pitch angle:

1. Vertical lift-off: The vehicle ascends vertically at least up to the minimum height constrained by launchpad safety criteria.

2. Linear pitch-over: The vehicle modifies its attitude and generates angle of attack

3. Exponential decay of pitch: The vehicle returns to its zero angle of attack position with a pitch angle different from the vertical position

4. Gravity turn: The gravity force helps steer the vehicle towards its orbital attitude

5. Bilinear tangent law: Insertion of vehicle to its orbital attitude and motion

6. Coast phases: Engines are off and trajectory is ballistic. It happens after stage and fairings separation.

7. Stage burns: After stage separation the mass of the launcher changes

The yaw angle control laws are also described by the aforementioned authors as a function of vehicle latitude and orbit inclination.

There exist many different strategies for the total or partial recovery of launchers. The recovery off the Liquid Fly-back Booster stage of a Vertical Take-off Horizontal Landing (VTHL) vehicle was described in [1] and the control strategy for the Vertical Take-off Vertical Landing (VTVL) Callisto demonstrator was described by Sagliano *et al.* [10] covering two recovery scenarios for the reusable first stage, namely Downrange Landing (DR) and Return-to-Launch-site (RTLS). The use of thrust magnitude, thrust pointing, RCS and aerodynamic control surfaces was also described by these authors during the take-off, ascent, boost-back, aerodynamic descent and landing phases.

4.4 Optimal Control

Optimal control problems are sometimes called also trajectory optimization problems. Betts introduced a compendium of methods for trajectory optimization in [11] and [7]. In the other hand, Rao did the same for optimal control in [12] and defined trajectory optimization problems as a subset of Optimal Control where the input parameters are static, thus dealing with the optimization of functions, whereas in the more general case Optimal Control is related to functional optimization problems.

Three main classifications can be done for Optimal Control methods: Heuristic Optimization Methods, Indirect methods and Direct methods. Heuristic Optimization methods are global, that implies that they are good at finding solutions in search spaces with many local minimums. Usually they are easy to implement but they are based on stochastic approaches rather than in gradient information thus having as output non-optimal solutions and being considered inferior for the application discussed in this work.

Indirect methods are founded on the calculus of variations and the formulation of first-order optimality conditions for the original problem, forming a Hamiltonian boundary value problem (HBVP) whose solutions are determined numerically. This strong formal background can ensure that solutions, when obtained, are indeed optimal. In [11] the application of indirect methods for low thrust orbit analysis was demonstrated. Berend and Talbot [4] described its use within CNES and Ariane programs and addressed some difficulties at implementing aerodynamic controls. In general these methods require experience to provide an initial guess due to its sensibility, making really difficult to explore changes on the launcher design variables.

On the other hand, Direct Methods are less sensible to the quality of the initial guess, and for some specific methods the mapping between the direct and indirect formulation has been done. Herman and Conway [13] used these relations to determine the accuracy of a Direct high-order Gauss Lobatto Collocation scheme. Garg *et al.* [14] demonstrated this mapping for a Radau Pseudospectral Method. The importance of this notion is that some direct methods benefit from a low sensibility to the initial guess, and their outputs can be verified with the mapping with the indirect formulation, thus guaranteeing high accuracy. For these reasons, the remaining part of this work will be centered on direct methods.

Direct methods use two classes of numerical methods as a basis, the first one dealing with Numerical Solution of differential Equations and Integration and the second one with Non-Linear Optimization. The first class is used in a transcription process to formulate a Non-Linear Optimization problem which then will look for the values of the inputs that minimize the objective function satisfying some given constraints. These classes will be described in the following sections together with the transcription methods.

4.4.1 Numerical Solution of differential Equations and Integration

Time marching methods are used to approximate numerically the solution of a differential equation at a given time step using information of current or previous information about the solution. Adams methods and Runge-Kutta methods including Euler and Hermite-Simpson formulations are covered in [7] and [12].

In collocation methods coefficients of an interpolating polynomial are calculated simultaneously to "collocate" it at the collocation points when approximating the solution of the equation. The solution itself can be discretized into intervals. Different formulations as Gauss, Radau and Lobatto methods exist and their differences are based on the inclusion or not of endpoints on the current interval. A subset of these methods is orthogonal collocation which uses orthogonal polynomials to approximate the functions, typically Chebyshev or Legendre polynomials as explained in [12], but also Jacobi polynomials were used for a High-Order Gauss Lobatto method implemented in [13].

As the objective function in the optimal control formulation includes often an integral term, numeric integration is also required. The approximation of the objective function and of the differential equations must be consistent.

4.4.2 Non-Linear Optimization

To find the values of the inputs that minimize the objective function and satisfy the given constraints a Non-Linear Programming (NLP) problem is formulated with a transcription method. NLP solvers are mostly based on Newton's method and formulate the problem in the Karush-Khun-Tucker (KKT) form. This matrix formulation contains the Jacobian and Hessian matrices of the problem that can be calculated analytically to obtain the best efficiency but also can be numerically approximated. These formulations are widely covered in [7]. One of the biggest challenges of the transcription process and the NLP solvers is to take advantage of the sparsity of the matrix formulation, specially when solutions for the equations are discretized in multiple segments, thus increasing the number of NLP variables. Different NLP solvers where found to be used in the literature, as SNOPT in [8] [15] [12] , IPOPT in [12] and NPSOL in [11].

4.4.3 Direct Transcription Methods for Solving Optimal Control Problems

The simplest direct method is called single shooting. In the case of a two point boundary value, it uses an initial guess for the initial boundary conditions and a numerical method to propagate the solution accordingly. The obtained value at the final time is then compared whit the boundary conditions to calculate an error. Then the NLP solver is used to find the set of initial conditions that take the error to zero. A more accurate method is multiple shooting, it divides time into intervals and requires an initial guess for all of them. To guarantee continuity of the solution this methods adds some constraints that increase the size of the NLP problem. Both methods are covered in [11] and [12]. Due to the low number of control parameters, multiple shooting has been used effectively for launcher ascent trajectories.

In Global Orthogonal Collocation or Pseudospectral methods the state or solution is approximated using global polynomials collocated at the collocation points. Even so, the state is still discretized into segments. In these methods the number of segments or meshes is fixed and the degree of the polynomial is varied. One remarkable advantage of these methods is the spectral convergence (exponential rate) as a function of the number of collocation points. In the survey of methods done by Rao [12] the following pseudospectral methods of interest for this project because of the possibility to integrate them with MDAO methods are covered:

- Gauss-Lobatto Pseudoespectral Methods
- Pseudoespectral method Using Legendre-Gauss-Radau points

As mentioned before, Herman and Conway [13] worked with High Order Gauss Lobatto methods. They showed the better convergence characteristics for these methods when compared to Low order methods of the same family thanks to the reduction of rounding errors associated to the lower number of segments required to approximate a function with high order polynomials. Garg *et al.* [14] demonstrated the spectral convergence of a Radau Pseudospectral method and developed the mapping with its indirect formulation. In a case study integrating a propulsion and a trajectory optimization module into an MDAO formulation, Hendricks *et al.* [8] showed that Legendre-Gauss-Radau have better performance over Legendre-Gauss-Lobatto when parallelization is used.

4.5 MDAO and its integration with Optimal Control

Multidisciplinary Design, Analysis and Optimization (MDAO) is an engineering strategy to optimize complex systems involving multiple disciplines. This process is done with gradient based methods that can involve analytic and numerically approximated derivatives. OpenMDAO is a MDAO open source framework developed at NASA Glenn Research Center written in Python. In this tool, the problem can be divided into components that must provide the derivatives of its outputs with respect to its inputs, and then the software will compute the total derivative of the system to perform the optimization of the involved variables .

In [8] a coupled modeling approach of trajectories and propulsion systems of an aircraft was proposed using OpenMDAO. The strategy was validated using the trajectory optimization for minimum time-to-climb of an F-4 supersonic jet. Falck and Gray [15] proposed a Legendre-Gauss-Lobatto based collocation scheme with analytic derivatives for the parallel trajectory optimization of an aircraft using OpenMDAO. The same authors presented in [16] the Dymos tool, a package designed to solve Optimal control problems in conjunction with MDAO. Initially, it implements Legendre-Gauss-Radau (LGR) and Legendre-Gauss-Lobatto pseudospectral methods, but the possibility to integrate other transcription methods exists. The package has also built-up interfaces with NLP solvers like SNOPT.

5 Research Project Plan and Description of Tasks

Task	Description	Due date
1	Implement Single and Multiple Shooting Direct Methods for trajectory optimization in Python but outside OpenMDAO for 2D and 3D trajectories	15 / 05 / 2020
2	Implement Direct Pseudospectral Methods for trajectory optimization in Dymos and OpenMDAO for 3D trajectories, this may be Legendre-Gauss-Radau or Legendre-Gauss-Lobatto. Implement different control laws for these methods	15 / 11 / 2020
3	Integrate the trajectory optimization method into the LAST tool in OpenMDAO with the other disciplines (Structures, Propulsion, etc.)	15 / 02 / 2021
4	Document the process and results	15 / 03 / 2021

References

[1] L. E. Briese, K. Schnepper, and P. Acquatella B., "Advanced modeling and trajectory optimization framework for reusable launch vehicles," Mar. 2018.

[2] F. Castellini, "MULTIDISCIPLINARY DESIGN OPTIMIZATION FOR EXPENDABLE LAUNCH VEHICLES," 2012.

[3] G. D. C. B. de Volo, "Vega Launchers' Trajectory Optimization Using a Pseudospectral Transcription," 2017.

[4] N. Berend and C. Talbot, "Overview of some optimal control methods adapted to expendable and reusable launch vehicle trajectories," *Aerospace Science and Technology*, Apr. 2006.

[5] Z. Wang and Z. Wu, "Six-DOF trajectory optimization for reusable launch vehicles via Gauss pseudospectral method," *Journal of Systems Engineering and Electronics*, Apr. 2016.

[6] A. Tewari, *Atmospheric and Space Flight Dynamics: Modeling and Simulation with MATLAB and Simulink*. MSSET - Modeling and Simulation in Science, Engineering & Technology, Boston, Mass.: Birkhäuser, 2007. OCLC: 180887853.

[7] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Advances in Design and Control, Society for Industrial and Applied Mathematics, Jan. 2010.

[8] E. S. Hendricks, R. D. Falck, and J. S. Gray, "Simultaneous Propulsion System and Trajectory Optimization," in *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA AVIATION Forum, American Institute of Aeronautics and Astronautics, June 2017.

[9] A. Pagano and E. Mooij, "Global Launcher Trajectory Optimization for Lunar Base Settlement," in *AIAA/AAS Astrodynamics Specialist Conference*, (Toronto, Ontario, Canada), American Institute of Aeronautics and Astronautics, Aug. 2010.

[10] M. Sagliano, T. Tsukamoto, J. A. Macés-Hernández, D. Seelbinder, S. Ishimoto, and E. Dumont, "Guidance and Control Strategy for the CALLISTO Flight Experiment," July 2019.

[11] J. T. Betts, "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, 1998.

[12] A. V. Rao, "A SURVEY OF NUMERICAL METHODS FOR OPTIMAL CONTROL," 2010.

[13] A. L. Herman and B. A. Conway, "Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules," *Journal of Guidance, Control, and Dynamics*, 1996.

[14] D. Garg, M. Patterson, C. Darby, C. Francolin, G. Huntington, W. Hager, and A. Rao, "Direct Trajectory Optimization and Costate Estimation of General Optimal Control Problems Using a Radau Pseudospectral Method," in *AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, June 2012.

[15] R. D. Falck, J. S. Gray, and B. Naylor, "Parallel Aircraft Trajectory Optimization with Analytic Derivatives," in *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, June 2016.

[16] R. D. Falck and J. S. Gray, "Optimal Control within the Context of Multidisciplinary Design, Analysis, and Optimization," in *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics.

A Derivation of 2D EoM for a rotating planet

The launcher is modeled as a particle moving in the equatorial plane of a rotating planet according to Newton's Second Law of dynamics. An Inertial Earth Centered reference frame, $\mathcal{F}_I := \{x_I, y_I\}$, is defined as inertial reference frame. In this reference frame it can be written for a variable mass system

$$\mathbf{f} = m\mathbf{a_I} \tag{A.1}$$

Where m is the mass of the launcher at any given time, $\mathbf{a_I}$ is the inertial acceleration and $\mathbf{f}$ is the vector comprised of aerodynamic forces (drag ($\mathbf{D}$) and lift ($\mathbf{L}$)), thrust force ($\mathbf{T}$) and gravity force ($m\mathbf{g}$).

The position of the launcher is better expressed with respect to a fixed point on Earth, this is achieved with the Rotating Earth Centered reference frame, $\mathcal{F}_R := \{x_R, y_R, z_R\}$, this frame is the one used by the reader to know its own position in Earth and has unitary vectors $\hat{\mathbf{I}}$, $\hat{\mathbf{J}}$ and $\hat{\mathbf{K}}$. To give a notion of the attitude of the vehicle the Local Vertical - Local Horizontal reference frame, $\mathcal{F}_t := \{x_t, y_t, z_t\}$, is defined with its center lying at a fix point on the launcher (Ex.: the geometric center) and with unitary vectors $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$. Finally, a fourth reference frame orientated with respect to the launcher velocity component is defined to better express the aerodynamic forces and is called the Wind reference frame $\mathcal{F}_w := \{x_w, y_w\}$. All of the aforementioned reference frames are shown in figure 1.

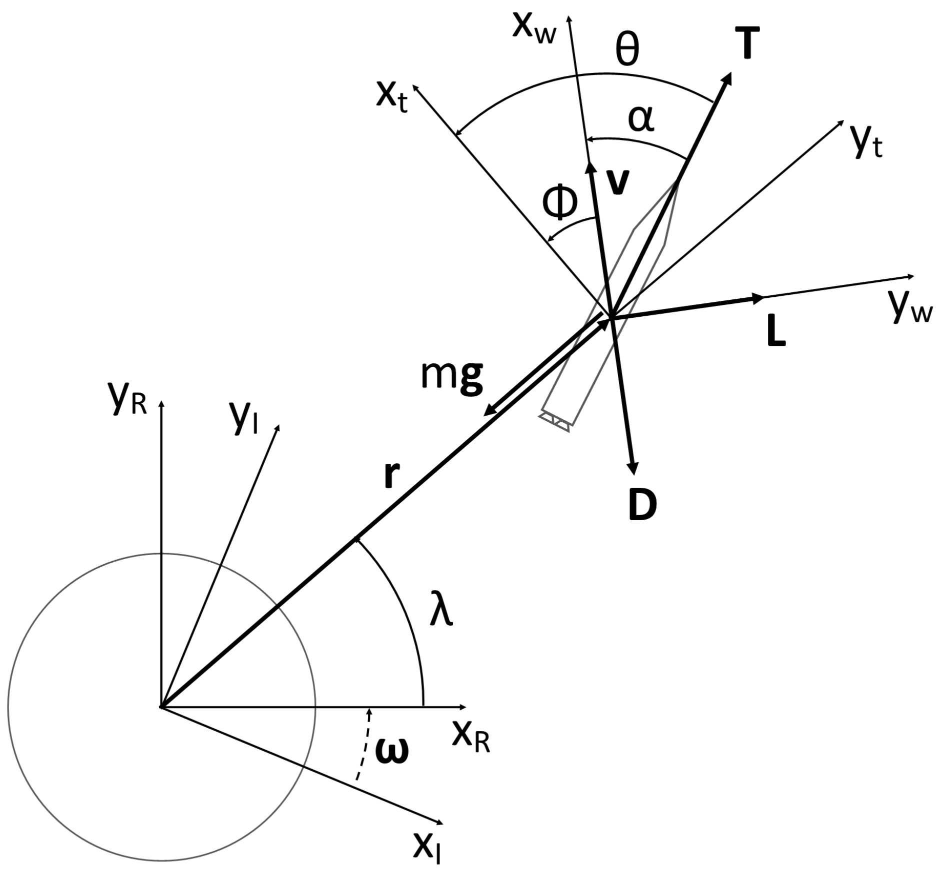

Figure 1: Reference frames and variables for 2D EoM

Where ω is Earth's angular velocity, λ is the longitude, $\mathbf{r}$ is the position vector from Earth's center to a fixed point in the launcher (Ex.: the geometric center), θ is the pitch angle, ϕ is the flight path angle, α is the angle of attack, m is the mass of the launcher at any given time, $\mathbf{g}$ is the acceleration of gravity, $\mathbf{D}$ is the drag force, $\mathbf{L}$ is the lift force, $\mathbf{T}$ is the thrust force and $\mathbf{v}$ the velocity vector.

For a vector $\mathbf{p}$ expressed in terms of the unitary vectors of the reference frame $\mathcal{F}_t$, the transformation matrix expressed in A.2 can be used to transform it and express it in terms of the unitary vectors of reference frame $\mathcal{F}_w$.

$$\mathbf{p_w} = \mathbf{R}'_\phi \mathbf{p_t} \quad , \quad \mathbf{R}_\phi = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{A.2}$$

To deduce the equations of motion some expressions for the change in time of the position $\dot{r}$ and the longitude $\dot{\lambda}$ will be find by using the velocity $\mathbf{v}$ as a firs step. Then the inertial velocity $\mathbf{v_I}$ will be used to find the inertial acceleration $\mathbf{a_I}$ and the necessary transformations will be done to use equation A.1 and finally find expressions for the change of velocity $\dot{v}$ and flight path angle $\dot{\phi}$.

From figure 1 a first expression for the velocity of the launcher relative to $\mathcal{F}_R$ but expressed in terms of the unitary vectors of $\mathcal{F}_t$ can be deduced.

$$\mathbf{v} = v(\cos(\phi)\mathbf{i} + \sin(\phi)\mathbf{j}) \tag{A.3}$$

A second expression for $\mathbf{v}$ is obtained by the relation with the linear change of the position vector $\dot{\mathbf{r}}$ and its angular rate $\mathbf{\Omega}$. Given that $\mathbf{r}$ can be expressed in terms of $\mathcal{F}_t$ unitary vectors as $r\mathbf{j}$, it can be written

$$\mathbf{v} = \dot{r}\mathbf{j} + \mathbf{\Omega} \times (r\mathbf{j}) \tag{A.4}$$

$\mathbf{\Omega}$ represents the angular velocity of the reference frame $\mathcal{F}_t$ with respect to $\mathcal{F}_r$ and can be written as $\Omega_x\mathbf{i} + \Omega_y\mathbf{j} + \Omega_z\mathbf{k}$. Solving the cross product it reads

$$\mathbf{v} = \dot{r}\mathbf{j} + \Omega_x r\mathbf{k} - \Omega_z r\mathbf{i} \tag{A.5}$$

By comparison of equations A.3 and A.5, it is obtained

$$\dot{r} = v\sin(\phi) \tag{A.6}$$

$$\Omega_z r = -v\cos(\phi) \tag{A.7}$$

As can be noticed in figure 1 the angular velocity of the reference frame $\mathcal{F}_t$ with respect to $\mathcal{F}_r$ is equal to the rate of change of λ, hence it can be expressed $\mathbf{\Omega} = \dot{\lambda}\mathbf{K}$ or whats is equivalent $\mathbf{\Omega} = -\dot{\lambda}\mathbf{k}$. Using this result and rewriting equation A.4 as

$$\mathbf{v} = \dot{r}\mathbf{j} + \dot{\lambda}r\mathbf{i} \tag{A.8}$$

and equating with A.7

$$\dot{\lambda} = \frac{v}{r}\cos(\phi) \tag{A.9}$$

the equations A.6 and A.9 are obtained and represent the first two EoM.

The inertial velocity $\mathbf{v_i}$ is comprised by the velocity $\mathbf{v}$ and the tangential component due to Earth's angular velocity ω. As the later can be expressed as $\omega\mathbf{K}$ or the equivalent $-\omega\mathbf{k}$, it reads

$$\mathbf{v_I} = \mathbf{v} + \omega r\mathbf{i} \tag{A.10}$$

The inertial acceleration can be find by derivating $\mathbf{v_I}$. This time the angular rate of change is $\omega + \mathbf{\Omega}$ or the equivalent $-(\omega + \dot{\lambda})\mathbf{k}$. Hence it can be written

$$\mathbf{a_I} = \frac{d\mathbf{v_I}}{dt} = \dot{\mathbf{v}} + \omega\dot{r}\mathbf{i} - (\omega + \dot{\lambda})\omega r\mathbf{j} \tag{A.11}$$

taking the derivative of $\mathbf{v}$ as expressed in A.8 and again using $-(\omega+\dot{\lambda})\mathbf{k}$ as the angular rate of change, it reads

$$\dot{\mathbf{v}} = \ddot{r}\mathbf{j} + \ddot{\lambda}r\mathbf{i} + \dot{\lambda}\dot{r}\mathbf{i} - (\omega+\dot{\lambda})\mathbf{k}\times(\dot{r}\mathbf{j} + \dot{\lambda}r\mathbf{i}) \tag{A.12}$$

By using A.12 and simplifying, equation A.11 can thus be rewritten as

$$\mathbf{a_I} = [\ddot{\lambda}r + 2\dot{r}(\dot{\lambda}+\omega)]\mathbf{i} + [\ddot{r} - r(\omega+\dot{\lambda})^2]\mathbf{j} \tag{A.13}$$

By finding the second derivatives of $\mathbf{r}$ and λ

$$\ddot{r} = v\cos(\phi)\dot{\phi} + \sin(\phi)\dot{v} \tag{A.14}$$

$$\ddot{\lambda} = -\frac{v\sin(\phi)\dot{\phi}}{r} + \frac{\cos(\phi)\dot{v}}{r} - \frac{v^2\sin(\phi)\cos(\phi)}{r^2} \tag{A.15}$$

and by replacing in equation A.13, we can rewrite $\mathbf{a_I}$ once again but in matrix form

$$\mathbf{a_I} = \begin{bmatrix} 2\omega v\sin(\phi) - v\sin(\phi)\dot{\phi} + \cos(\phi)\dot{v} + \frac{v^2\sin(2\phi)}{2r} \\ -\omega^2 r - 2\omega v\cos(\phi) + v\cos(\phi)\dot{\phi} + \sin(\phi)\dot{v} - \frac{v^2\cos^2(\phi)}{r} \end{bmatrix} \tag{A.16}$$

So far $\mathbf{a_I}$ is expressed in terms of the unitary vectors $\mathbf{i}$ and $\mathbf{i}$ that are the basis of $\mathcal{F}_t$. A final transformation to the Wind reference frame $\mathcal{F}_w$ must be done using the matrix transformation expressed in equation A.2. The new unitary vectors were not explicitly defined but $\mathbf{a_I}$ as a function of them reads

$$\mathbf{a_I} = \begin{bmatrix} -\omega^2 r\sin(\phi) + \dot{v} \\ -\omega^2 r\cos(\phi) - 2\omega v + v\dot{\phi} - \frac{v^2\cos(\phi)}{r} \end{bmatrix} \tag{A.17}$$

Also in the $\mathcal{F}_w$ reference frame, the forces exerted by the launcher as can be seen in figure 1 can be expressed in matrix form as

$$\mathbf{f} = \begin{bmatrix} -D + T\cos(\theta-\phi) - gm\sin(\phi) \\ L + T\sin(\theta-\phi) - gm\cos(\phi) \end{bmatrix} \tag{A.18}$$

Finally, by applying Newton's second law as expressed in equation A.1, using A.17 and A.18 and solving for $\dot{v}$ and $\dot{\phi}$ we can find the two remaining equations of motion

$$\dot{v} = \frac{-D + T\cos(\theta-\phi)}{m} + \left(-g + \omega^2 r\right)\sin(\phi) \tag{A.19}$$

$$\dot{\phi} = \frac{L}{mv} + \frac{T\sin(\theta-\phi)}{mv} + \frac{(\omega^2 r - g)\cos(\phi)}{v} + 2\omega + \frac{v\cos(\phi)}{r} \tag{A.20}$$

B Technical progress to date

The implementation of the 2D EoM for a rotating planet has been done in Python. So far, the script propagates the EoM with the solver *solve_ivp* which is also used in the LAST tool. The aerodynamic model consists of an interpolation with cubic splines of the drag coefficient, which is tabulated as a function of Mach number. The atmospheric model is based on the *U.S. Standard Atmosphere 1976*. The gravity model is Newton's. This script is part of Task 1 as described in section 5. Work is being done for the implementation of different phases. Figure 2 contains the simulation for the Lift-off phase up to $5km$ height above the launch pad according to the input parameters and initial conditions shown below.

Parameter	Value
ω	$7.2921159e-5$ (rad/s)
$\dot{m}$ (Mass flow rate)	3025 (kg/s)
I_{sp} (Specific impulse)	280 (s)
S (Reference Area)	37.5 (m^2)
g_0 (gravity at sea level)	9.806 (m/s^2)

Initial conditions	Value
r (Height)	$6378.135e3$ (m)
v (speed)	1 $(m/s$
m (mass)	$550 * 1e3$ (kg)
λ (longitude)	0 (deg)
ϕ (flight path angle)	90 (deg)

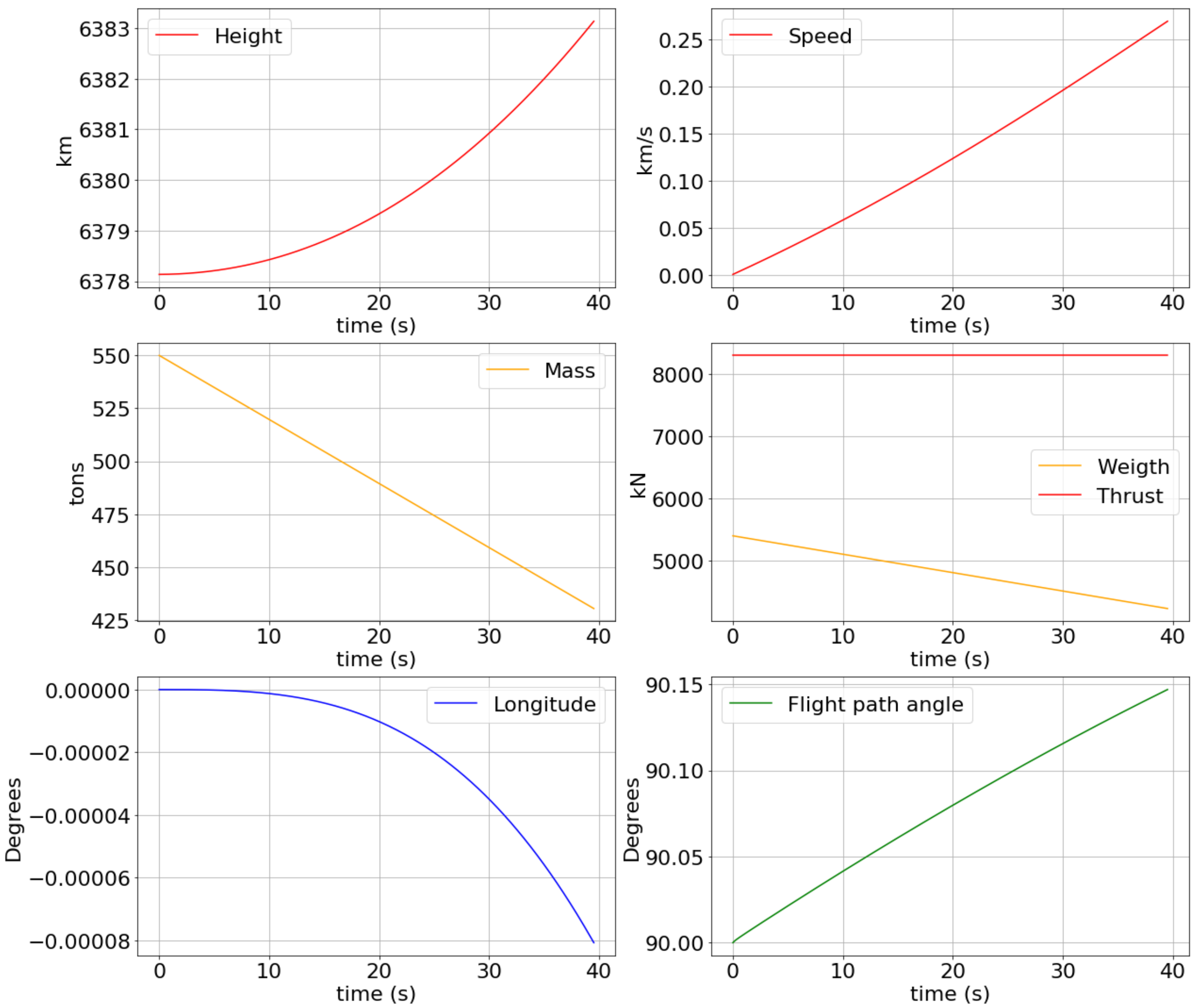

Figure 2: Lift-off phase simulation