Master thesis and internship[BR]- [BR]-

Auteur : Azouar, Mohamed
Promoteur(s) : Kerschen, Gaëtan
Faculté : Faculté des Sciences appliquées
Diplôme : Master en ingénieur civil en aérospatiale, à finalité spécialisée en "aerospace engineering"
Année académique : 2025-2026
URI/URL : http://hdl.handle.net/2268.2/25233

UNIVERSITY OF LIÈGE
FACULTY OF APPLIED SCIENCES

Development of Physical Models for Launch Vehicle Dynamics Simulation

---

Mohamed Azouar

Thesis presented to obtain the degree of:

Master of Science in Aerospace Engineering

Thesis supervisors:

Gaëtan Kerschen
Daniel Blanquez

Academic year: 2025 – 2026

Acknowledgements

This work would not have been possible without the support and guidance of several people, to whom I would like to express my sincere thanks.

I am grateful to my academic supervisor, Prof. Gaëtan Kerschen, for his guidance and constructive feedback throughout this project.

I would like to thank my industrial supervisor, Mr. Daniel Blanquez, for his constant availability, his technical support, and the many discussions that helped shape this work. His expertise and advice were essential to the progress of this project.

I also wish to thank all the members of the Spacebel team for their support, their kindness, and the stimulating working environment they provided during my internship.

I would also like to express my gratitude to my family and friends for their continuous support and encouragement throughout my studies.

This manuscript also benefited from the use of artificial intelligence tools for improving the clarity and quality of the writing. These tools were used solely for language refinement and stylistic improvements. The scientific content, technical developments, analyses, and conclusions remain entirely my own work and responsibility.

Abstract

This thesis presents the development and integration of a set of physical models for launch vehicle dynamics simulation within an existing simulation framework used at Spacebel. The work focuses on improving the representation of the physical environment and the associated forces acting on the vehicle, with particular emphasis on atmosphere, aerodynamics, and gravity.

A modular modeling approach is adopted, in which new environment and force computation blocks are implemented and coupled with the vehicle dynamics. The developments include the implementation of an atmospheric model with stochastic turbulence based on the Dryden formulation, the computation of aerodynamic angles and loads, and improvements in the handling of kinematics, reference frames, attitude dynamics, and gravity modeling. The implemented models are verified through unit tests and consistency checks in order to ensure their physical validity and numerical robustness.

In a second step, the impact of atmospheric disturbances on the vehicle response is investigated. Deterministic sensitivity analyses are first carried out to assess the influence of key parameters such as turbulence intensity and mean wind magnitude. Then, Monte Carlo simulations are performed to quantify the statistical dispersion of critical aerodynamic quantities, including the angle of attack, sideslip angle, dynamic pressure, and lateral aerodynamic force.

The results show that the turbulence intensity mainly affects fluctuation-related metrics, while steady wind components and stochastic realizations can significantly modify the flight conditions at which peak loads occur. In particular, lateral aerodynamic quantities exhibit a very large variability across realizations, and rare but more severe cases are observed. These results highlight the strongly nonlinear nature of the coupled atmosphere–aerodynamics–dynamics system.

Overall, this work provides a consistent extension of Spacebel's simulation framework and demonstrates its capability to support physically sound modeling and uncertainty-aware analyses for launch vehicle dynamics simulation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Launch vehicle simulation plays a central role in the design, analysis, and verification of modern space systems. Before a launcher ever flies, its behaviour must be predicted with sufficient accuracy to guarantee mission success and to ensure that safety and performance requirements are met. Numerical simulations allow engineers to analyse trajectories, estimate loads, evaluate aerodynamic performance, design guidance strategies, and assess the robustness of the system under various environmental and operational conditions. As space missions become increasingly ambitious and as performance margins tighten, simulation tools must incorporate physical models that are both reliable and computationally efficient.

A launch vehicle is subjected to a wide range of physical phenomena throughout its ascent. These include gravitational variations, aerodynamic forces and moments, propulsion effects and wind profiles among others. Representing these phenomena requires a combination of models that must interact coherently within an integrated simulation framework. Ensuring consistency between these models while maintaining modularity, maintainability and numerical stability constitutes a significant engineering challenge.

In the industrial context, companies such as Spacebel develop modular simulation environments used for system design, mission analysis, and verification activities. These simulators typically rely on a collection of physical models that interact through well-defined interfaces. Improving the fidelity of such models, or introducing new ones, directly enhances the predictive capability of the tool and contributes to the overall robustness of the simulation chain. This requires careful attention to model formulation, numerical implementation, data handling, and integration within the global architecture.

The present work contributes to this effort by developing and integrating several physical models relevant to the early phases of launcher flight within the simulation environment used at Spacebel. In particular, this thesis focuses on the implementation of an atmospheric model, an aerodynamic model, and a gravity model. These new components are combined with existing modules within a unified simulation framework. To account for environmental uncertainties, turbulence modeling based on the Dryden model and Monte Carlo analyses are also addressed. A special attention is given to model coupling, interface design, numerical robustness and validation strategies.

1.1 Company presentation

Overview

Spacebel is a Belgian company specializing in software engineering and space systems development. Founded in 1988, the company has grown into a recognized provider of critical software solutions for the European space sector. Its activities span the design, development, integration, and validation of software systems for spacecraft, launchers, and ground infrastructures, as well as the exploitation and management of geospatial and Earth observation data. With offices in Liège, Hoeilaart, and Toulouse, Spacebel plays an important role in the European space industry.

Core Activities

Spacebel's activities cover the entire software chain involved in space missions. The company develops embedded flight software for spacecraft and launchers, ground control software for mission operations, and tools dedicated to the processing and dissemination of space and geospatial data. In parallel, Spacebel designs high-fidelity simulators used for system design, flight software validation, operator training, and mission preparation. These simulators reproduce the dynamics of spacecraft or launchers, as well as the behavior of their subsystems, enabling the injection of failures and realistic operational scenarios.

Operational Units

The company is structured into three main operational units. The *Flight Segment* unit focuses on on-board software, ensuring reliable execution of mission functions throughout the spacecraft or launcher life cycle. The *Ground Segment* unit develops mission control systems, data processing chains, and operator interfaces used to supervise, command, and monitor space assets. Finally, the *Modeling and Simulation* unit, in which this thesis was carried out, develops physics-based simulators that emulate the behavior of space systems and their environment. These simulators may interface with real hardware, external software components, or operational infrastructures, and form a key element of verification and validation activities. Since relatively few European companies operate in this niche, Spacebel is considered a significant actor in the simulation domain, with expertise comparable to internal teams of major aerospace primes.

International Links and Partnerships

Spacebel plays an active role in numerous European space programs and collaborates closely with institutions such as the European Space Agency (ESA) and the Centre National d'Études Spatiales (CNES). The company is strongly involved in various European programs and has contributed to numerous space missions including PROBA-V, Hera, ALTIUS, Euclid, Sentinel-1, and RAMSES. Spacebel also maintains several international links. It owns N7 Space, a Polish subsidiary specialized in flight and ground support software, and holds shares in ConstellR, a company active in thermal infrared Earth observation for agricultural and environmental monitoring.

1.2 Objectives

The goal of this thesis is to enhance the physical fidelity and robustness of the launcher simulation environment used at Spacebel by developing, integrating, and validating several new models relevant to the early phases of flight. The specific objectives of the work are summarized below.

- **Implement an atmospheric model suitable for launcher simulations.** This includes selecting an appropriate standard atmosphere formulation, implementing the model in a modular structure, and ensuring that variables such as density and speed of sound are computed consistently. Wind and turbulence effects, including a Dryden-based turbulence model, are also incorporated to represent environmental variability.
- **Implement an aerodynamic model providing forces and moments acting on the launcher.** The objective is to compute aerodynamic coefficients based on user-provided tables, perform the required interpolations, and determine aerodynamic forces and moments in a coherent and numerically stable manner. The model must integrate properly with the existing vehicle geometry and the dynamics modules.
- **Improve the attitude computation to ensure accurate coupling with aerodynamic effects.** Although not originally planned as a stand-alone objective, improvements to the attitude integration are required to guarantee a physically consistent interaction between the aerodynamic model and the dynamics. The objective is therefore to refine the attitude integration method and ensure stable, accurate attitude propagation.
- **Implement a gravity model to improve the representation of gravitational forces within the simulator.** The objective is to replace the previous fixed-gravity assumption and provide a more realistic representation of gravitational forces.
- **Perform sensitivity analyses and Monte Carlo simulations to assess the impact of atmospheric uncertainties, in particular wind turbulence, on the launcher trajectory.** These analyses evaluate how variations in atmospheric parameters influence the launcher trajectory, providing insight into the sensitivity and robustness of the implemented model.
- **Ensure proper integration of all new models within the existing simulation architecture.** The goal is to connect the newly developed components to the current modular framework, respect the defined interfaces, and guarantee that the models interact coherently with the rest of the simulation chain.
- **Validate the developed models through dedicated tests.** This consists in verifying the physical correctness, numerical stability, and expected behavior of each model, first individually and then collectively in an integration test once assembled within the full simulator.

Chapter 2

State of the Art

The simulation of launch vehicles relies on a combination of physical models describing the surrounding environment, the vehicle's dynamics, and the interactions between external forces and internal actuators. Over the past decades, a wide range of atmospheric, aerodynamic, gravity, and turbulence models has been developed in the aerospace community, each offering different levels of fidelity, complexity, and computational cost. Understanding these existing approaches is essential to properly position the models implemented in this thesis within the broader context of launcher simulation.

Beyond individual physical models, several integrated multi-physics simulation frameworks for launch vehicles have also been proposed in the literature. For instance, the R2M2 tool described in [1] provides a modular environment combining dynamics, atmosphere, aerodynamics, and guidance models for reusable launch vehicle studies. Such frameworks illustrate the general architectural philosophy adopted in modern launcher simulation tools and provide useful reference points for the present work.

This chapter provides an overview of the modelling strategies most commonly used in industry and in the literature. It introduces reference atmospheric standards, classical and modern formulations for aerodynamic forces and moments, the main families of stochastic wind turbulence representations, and the gravity models used in trajectory simulation. The objective is not to detail the specific implementation choices of this work, but rather to summarize the established methods on which contemporary launcher simulators typically rely.

2.1 Atmospheric Models

The atmosphere plays a central role in the simulation of launcher trajectories, as it directly influences the aerodynamic forces during the ascent phase. A wide range of atmospheric models has been developed in the aerospace community, differing in scope, fidelity, and intended operational use. They can generally be classified into two main families: standard empirical atmospheres and measurement-based atmospheric profiles, which are often used in a complementary manner in operational simulations.

2.1.1 Standard Empirical Atmospheres

Standard atmosphere models provide tabulated or analytical profiles of temperature, pressure, density, and speed of sound as a function of altitude under reference atmospheric conditions. They offer a deterministic reference environment and are therefore widely used for preliminary analyses, trajectory design, and verification.

- **U.S. Standard Atmosphere (COESA)** One of the most commonly used reference models for launch vehicle studies. COESA provides temperature, pressure, density, and related quantities up to about 86 km. Its simplicity, reproducibility, and widespread industrial adoption make it a standard choice for early-phase launcher simulations [2].
- **NRLMSISE-00** A more advanced empirical model extending to the thermosphere and exosphere. It provides density, temperature, and constituent species up to 1000 km and depends on additional parameters such as solar flux and geomagnetic activity. Although primarily used for satellites rather than launchers, it is sometimes employed when modelling upper-atmosphere effects above the COESA limit [3].
- **Harris–Priester and Jacchia Models** Older semi-empirical formulations such as Harris-Priester [4] and Jacchia [5] were historically used for orbital decay predictions. They provide coarse thermospheric density estimates but are less common in launcher applications due to their limited fidelity near the stratosphere and mesosphere.

2.1.2 Measured Atmospheric Profiles

In operational contexts, launch providers often complement standard atmosphere models with measured atmospheric profiles when available. These measurements are typically acquired shortly before launch using meteorological balloons. They supply altitude-dependent wind components as well as temperature and density profiles [6].

Launcher simulations therefore often incorporate:

- user-provided wind tables (mean wind profiles),
- standard atmosphere models for nominal temperature and density,
- site-specific atmospheric deviations from the nominal standard atmosphere.

These data inputs allow mission planners to assess day-of-launch conditions.

2.1.3 Turbulence and Wind Disturbance Models

Beyond mean winds, high-frequency turbulent fluctuations can play a significant role in launcher structural loads and trajectory dispersions. Several stochastic turbulence models exist:

- **Dryden Model** A widely used linear stochastic model providing velocity fluctuations with specific spectral properties. It is commonly employed in aerospace simulations due to its simplicity, analytical filters, and compatibility with discrete-time numerical integration [7].
- **von Kármán Model** Offers a more realistic turbulence spectrum, particularly at low frequencies. However, the lack of simple analytical filters makes it less practical for real-time or embedded simulations [8].
- **Kaimal and Davenport Models** Often used in the wind energy sector to model atmospheric boundary-layer turbulence, these models are uncommon in launcher simulations because their underlying assumptions break down for fast-moving launch vehicles, which exit the atmospheric boundary layer within seconds [9] [10].

2.1.4 Use in Launcher Simulation

In operational launcher simulation environments, atmospheric models are primarily used to provide realistic boundary conditions for aerodynamic force computation during ascent. Rather than relying on a single monolithic atmosphere representation, practical implementations typically combine complementary data sources in order to balance realism, flexibility, and computational efficiency.

In practice, launch simulations typically combine:

- a **deterministic mean wind profile** (measured or user-provided), and
- a **stochastic turbulence component** superimposed on top of it.

2.2 Aerodynamic Models

Aerodynamic modelling plays a central role in launcher simulation, as aerodynamic forces and moments dominate the early phases of atmospheric ascent. These quantities determine the vehicle's lateral loads, structural constraints, attitude evolution, and overall trajectory shaping. A wide variety of models exist, ranging from simplified analytical expressions to high-fidelity databases derived from wind-tunnel testing or computational fluid dynamics (CFD). This section reviews the main approaches commonly used in industry and in the aerospace literature.

2.2.1 Coefficient-Based Formulation

Most launch vehicle simulators rely on a coefficient-based representation of aerodynamics. In this framework, forces and moments are expressed as non-dimensional coefficients multiplied by dynamic pressure and a reference area chosen according to the launcher geometry. The aerodynamic force components in the body frame are written as:

$$\mathbf{F}_{\text{aero}} = q_\infty S \begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix},$$

where $q_\infty = \frac{1}{2}\rho V^2$ is the dynamic pressure, S the reference area, and C_X, C_Y, C_Z the axial and lateral force coefficients.

Similarly, aerodynamic moments are expressed as:

$$\mathbf{M}_{\text{aero}} = q_\infty S L_{\text{ref}} \begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix},$$

with L_{ref} a reference length also chosen according to the launcher geometry.

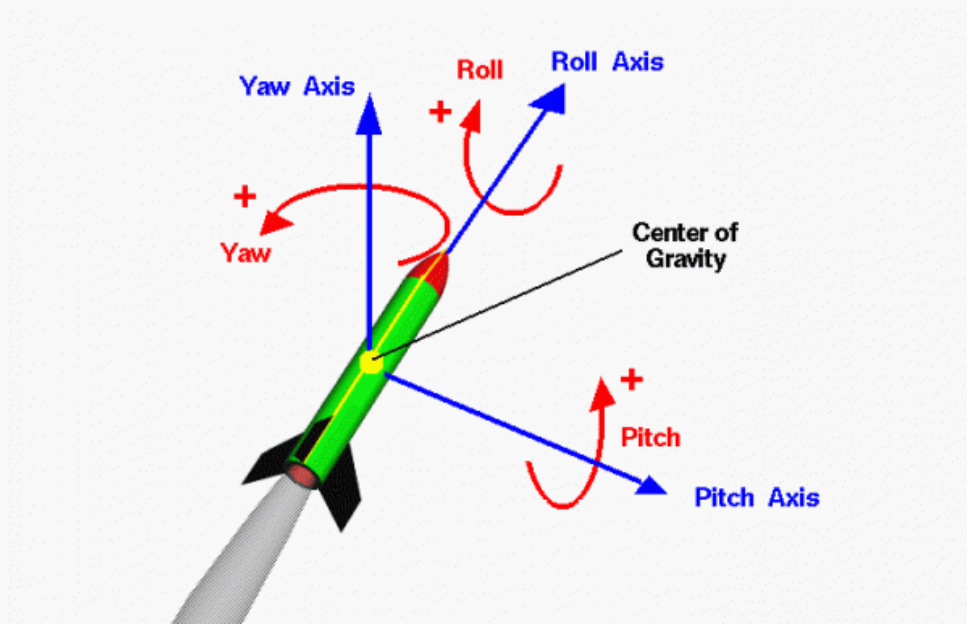


Figure 2.1: Body-axes convention for rockets: roll, pitch, yaw [11].

This formulation allows aerodynamic behaviour to be fully characterised through non-dimensional coefficients that depend primarily on the Mach number, angle of attack (α), sideslip angle (β), and possibly other parameters such as control surface deflections or angular rates. This coefficient-based formalism is widely used in launch vehicle simulations and is consistent with the aerodynamic conventions established in NASA reports such as [12].

Accurate aerodynamic modelling requires consistent definitions of the reference frames and aerodynamic angles. The aerodynamic angles are typically defined as:

$$\alpha = \arctan\left(\frac{V_Z}{V_X}\right), \qquad \beta = \arcsin\left(\frac{V_Y}{V}\right),$$

where the velocity components are expressed in the body frame, following the body-axes convention illustrated in Fig. 2.1.

The correct transformation from inertial or Earth-fixed frames to the body frame is therefore critical. Errors in attitude propagation or velocity transformation directly affect the computed aerodynamic forces and moments, especially during high dynamic pressure phases.

2.2.2 Sources of Aerodynamic Coefficients

Several approaches exist to obtain aerodynamic coefficients for launchers:

- **Wind-Tunnel Testing** Historically, wind-tunnel campaigns played a central role in launcher development. Scaled models are tested over a grid of Mach numbers and angles of attack, providing high-fidelity aerodynamic datasets. These campaigns remain essential for new launcher programs, though they are costly and time-consuming [13].
- **Computational Fluid Dynamics (CFD)** Modern launcher programs rely increasingly on CFD simulations for aerodynamic characterisation. High-fidelity solvers

allow the prediction of complex flow phenomena, including transonic shocks and separated flows. CFD-derived databases are often used to augment or replace wind-tunnel measurements for many flight regimes [14].

- **Engineering Methods** Before high-fidelity data are available, preliminary aerodynamic estimates may be obtained using semi-empirical methods such as:
  - Newtonian flow assumptions (for high angles of attack or hypersonic regimes),
  - slender-body theory,
  - DATCOM-like empirical formulations.

  These methods offer rapid estimates suitable for early design phases but are less accurate near critical regimes (transonic or separated flows), and are commonly summarized in engineering references such as USAF DATCOM [15].

2.2.3 Tabulated Aerodynamic Databases

Most operational simulators, including those used by industry, rely on tabulated aerodynamic coefficients. These tables typically provide $C_X, C_Y, C_Z, C_l, C_m, C_n$ as functions of Mach number and angle of attack, and sometimes sideslip angle.

Since aerodynamic coefficients are only available at discrete points of the flight envelope, interpolation is required to evaluate aerodynamic forces and moments continuously during trajectory propagation.

Interpolation strategies commonly used include:

- bilinear or bicubic 2D interpolation (Mach, α),
- trilinear interpolation when including sideslip (β),
- spline-based interpolation for smoother derivatives when required.

The use of tabulated aerodynamic data ensures reproducibility and enables fast, reliable evaluations within trajectory simulations, as commonly adopted in NASA trajectory analysis frameworks [16].

High-fidelity simulators may also include additional aerodynamic effects such as:

- rate-dependent damping derivatives (dependence on p, q, r),
- control surface effectiveness models (for launchers equipped with fins),
- base drag models,
- jet-interaction effects during thrust vectoring.

These extensions increase fidelity but also require more detailed flight data or CFD campaigns.

For most launcher ascent simulations, a coefficient-based, tabulated aerodynamic database remains the dominant approach. It provides a reliable compromise between fidelity and computational cost. High-fidelity CFD or wind-tunnel data are typically compressed into multidimensional tables, which are then interpolated in real time.

2.2.4 Use in Launcher Simulation

In launcher simulations, aerodynamic models are therefore primarily implemented as database-driven components. The role of the software is not to compute aerodynamic behavior from first principles, but to perform robust multi-dimensional interpolation of precomputed coefficient tables and to convert these coefficients into forces and moments consistent with the vehicle reference frames and dynamics. This approach ensures numerical efficiency, reproducibility, and compatibility with real-time or Monte Carlo trajectory simulations.

2.3 Gravity Models

The gravity field plays a central role in launcher trajectory simulation, as it represents the dominant force throughout most of the ascent phase. Several formulations exist, offering different levels of fidelity and computational cost. This section summarizes the gravity models most commonly employed in aerospace simulation environments.

2.3.1 Spherical Earth Gravity

The simplest representation assumes a spherical Earth with a gravitational acceleration defined as

$$g(r) = \frac{\mu}{r^2},$$

where μ is Earth's standard gravitational parameter and r the distance from Earth's center. The gravity vector is directed radially and decreases with altitude. This model is widely used for preliminary trajectory analyses due to its simplicity and low computational cost.

2.3.2 Ellipsoidal Gravity and Latitude Dependence

More refined gravity representations account for the Earth's oblateness and rotation by introducing an ellipsoidal Earth model and an associated reference gravity field. Geodetic reference systems such as WGS-84 [17] define the Earth's shape and fundamental constants, which are commonly used to derive latitude-dependent gravity formulations at the Earth's surface. Using the WGS-84 ellipsoid, the surface gravity magnitude is commonly approximated by

$$g(\phi) = g_e \left(1 + k \sin^2 \phi\right),$$

where ϕ is the geodetic latitude, g_e the equatorial gravity, and k a constant linked to Earth's flattening. Such simplified latitude-dependent gravity expressions are frequently used in guidance and navigation algorithms, where moderate accuracy and computational efficiency are required.

2.3.3 Zonal Harmonics and Perturbation Models

The Earth's gravitational field can be described more generally by expanding the gravitational potential in spherical harmonics. This formulation captures deviations from a purely central gravity field caused by the Earth's oblateness and mass distribution asymmetries.

In this framework, the gravitational potential is expressed as:

$$U(r,\theta,\lambda) = \frac{\mu}{r}\left[1 - \sum_{n=2}^{\infty}\left(\frac{R_e}{r}\right)^n \sum_{m=0}^{n} \bar{P}_{nm}(\sin\theta)\left(C_{nm}\cos m\lambda + S_{nm}\sin m\lambda\right)\right],$$

where θ is the geocentric latitude, λ the longitude, $\bar{P}_{nm}$ the normalized associated Legendre functions, and C_{nm}, S_{nm} the gravitational coefficients.

The gravitational acceleration vector is obtained as the gradient of the gravitational potential, such that

$$\mathbf{g} = -\nabla U.$$

For launcher ascent simulations, retaining only the dominant zonal term J_2 is generally sufficient to capture the primary effect of the Earth's oblateness. Higher-order zonal (J_3, J_4) or tesseral ($m \neq 0$) harmonic components generally provide only minor corrections and are therefore unnecessary for short-duration trajectories [18, 19].

The gravitational potential corrected for J_2 is commonly expressed as

$$U = -\frac{\mu}{r}\left[1 - J_2\left(\frac{R_e}{r}\right)^2 \frac{1}{2}(3\sin^2\theta - 1)\right].$$

2.3.4 Use in Launcher Simulation

For real-time or faster-than-real-time trajectory propagation, operational launcher simulators typically rely on gravity models that balance fidelity and computational efficiency. Common choices include:

- a latitude-dependent gravity formulation derived from a WGS-84 reference ellipsoid,
- or a spherical gravity model corrected with the J_2 harmonic.

Chapter 3

Simulation Framework and Workflow

This section presents the software environment and workflow used to develop, assemble, verify, and execute the launcher simulation model. The development relies on a combination of industry standards, internal tools, and automated code generation pipelines, which are orchestrated to produce components compliant with the Simulation Model Portability (SMP) standard.

3.1 Execution Environment

All developments were performed under a Linux environment executed through the Windows Subsystem for Linux (WSL). Linux is the preferred operating system in the space simulation industry due to its stability, its compatibility with common toolchains, and its alignment with SMP-compliant simulation environments. The runtime engine, the metadata generation tools used to produce the XML files (detailed later in 3.4), and the code generation tools are all optimized for Linux, making WSL a practical choice when working on Windows workstations.

3.2 SMP Standard

The simulation workflow used in this project follows the *Simulation Model Portability* (SMP) standard, formally defined as ECSS-E-ST-40-07 [20]. Over the past decades, various space organizations and industries have developed simulation software for engineering analysis, operations preparation, and training. These tools were often heterogeneous in language, architecture, and execution environment. SMP addresses these issues by defining a unified set of concepts, interfaces, and types that enable the reuse and portability of simulation models across missions, organizations, and simulation platforms.

SMP distinguishes between two categories of components:

- **Simulation Models**, which implement application-specific behaviour;
- **Simulation Environments**, which provide simulation services such as logging, time management, event handling, and scheduling.

An SMP-compliant environment exposes essential services, including the Logger, TimeKeeper, Scheduler, Event Manager, Resolver, and Link Registry. These services enable models to interact through standardised mechanisms such as interface-based communication, dataflow connections (Input/Output fields), and event-based communication.

SMP also defines a well-specified simulation lifecycle, including the *Building*, *Connecting*, *Initialising*, *Standby*, *Executing*, and *Storing* phases. This ensures that all components

are correctly instantiated, configured, connected, and synchronised before simulation time progresses.

By enforcing a strict separation between model logic and simulation services, SMP enables both source-level and binary-level reuse of models. Spacebel has historically been an active contributor to the development of SMP and continues to rely on it for large-scale simulation infrastructures.

3.3 Code Generation Using CDKs and XSMPcat

Several industrial partners provide *Code Development Kits* (CDKs) that generate SMP-compliant code from high-level model descriptions. The project presented in this thesis uses the XSMP CDK developed by Thales, although other organisations such as ESA provide alternative CDKs based on the same SMP principles.

The modelling workflow begins with an XSMP catalogue (`XSMPcat`), which describes the structure of the component in terms of its hierarchy, entry points (i.e., callbacks executed by the simulation environment at specific stages of the simulation lifecycle), and interfaces. From this catalogue, the XSMP toolchain automatically generates a set of C++ header and source files. Some of these files are intended to be extended by the developer, while others are generated by the toolchain and should not be modified.

The generated skeleton provides the structural framework required for SMP compliance: it defines the component class, the associated entry points, and the methods that integrate the model into the SMP simulation environment. The developer is then responsible for implementing the actual algorithmic behaviour inside the appropriate C++ files.

Once the implementation is complete, the component is compiled into a shared library (`.so`), which can be loaded by any SMP-compliant simulation environment.

3.4 Assembly and Metadata Generation

After compilation, individual model libraries must be assembled into a complete simulation. This step is performed using an internal Python-based assembly tool developed by Spacebel. The assembly process:

- specifies the shared libraries (.so) that the runtime environment will later load;
- instantiates and arranges the components in a hierarchical structure;
- generates the SMP metadata files required by the simulation environment.

These metadata files are all provided in XML format and include:

- **SMPASM**: structural description of the simulation assembly,
- **CFN**: configuration values for component fields,
- **SED**: description of the simulation scheduling.

The assembly ensures that the final simulation is consistent with the SMP model hierarchy and ready for execution.

3.5 Execution with BSLNG

The final model is executed using `BSLNG`, Spacebel's SMP-compliant simulation runtime environment. `BSLNG` provides the services required by SMP components, including time management, logging, scheduling, event handling, and model resolution (i.e., locating models in memory so that they can be accessed through commands or graphical tools).

When delivering a simulation model to a customer, Spacebel provides:

- the compiled shared libraries (`.so`),
- the `BSLNG` runtime environment (provided as an executable).

Internal tools used during development (`XSMPcat`, assembly scripts, and `SVStoolNG`, which will be described in 3.6) are not included in the delivery. `BSLNG` is responsible for loading the simulation bundle — a folder containing the shared libraries, assembly files, tests, and metadata — and managing the execution lifecycle, as well as providing the services required for model interaction. It therefore constitutes the final runtime context in which the models developed in this thesis are executed.

3.6 Verification Using SVStoolNG

Model verification is performed using the `SVStoolNG`, an automated testing tool. Test specifications are written in Markdown or JSON files, which the tool converts into Python test scripts. These scripts load the SMP components, execute their operations, and compare the results produced by the C++ implementation against reference values computed by the test.

Each test asserts the correctness of the model behaviour up to predefined tolerance thresholds. This validation workflow provides a systematic method for ensuring model compliance and detecting implementation errors early in the development cycle.

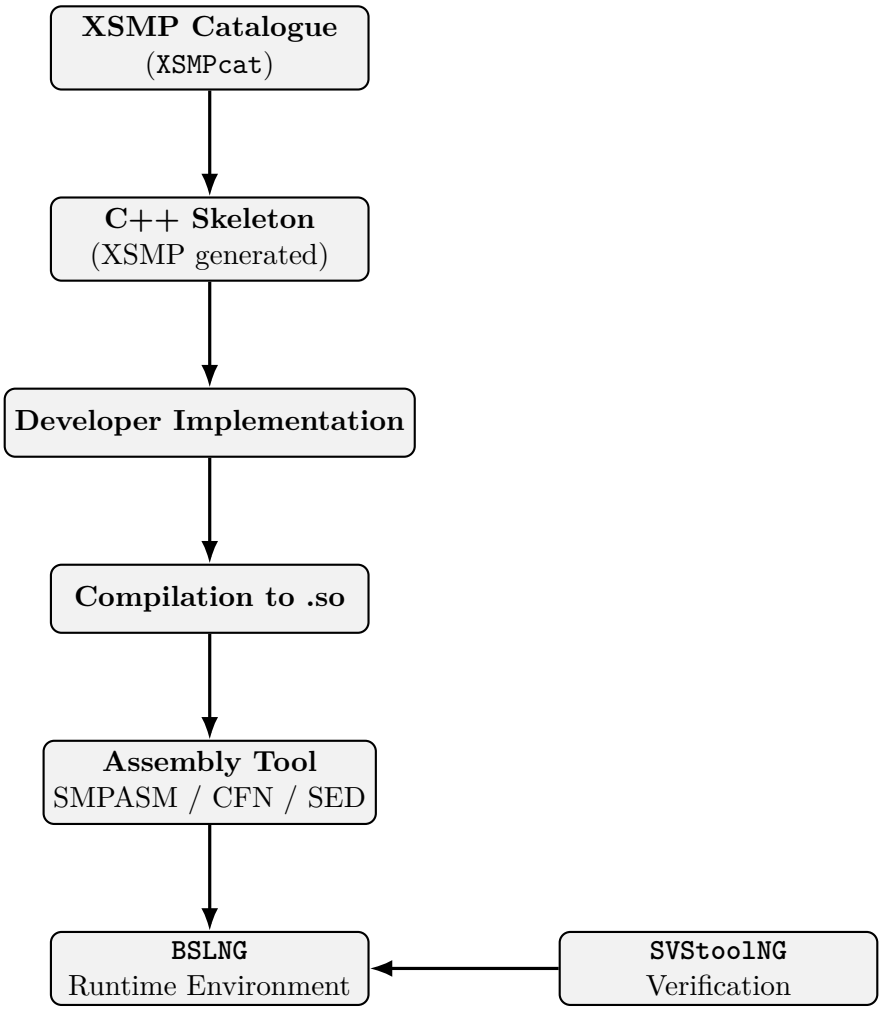


Figure 3.1: Overview of the simulation workflow.

Chapter 4

Models Overview and Pre-existing Models

4.1 Models Overview

The launcher dynamics simulator follows a modular, block-based architecture in which each physical phenomenon is represented by an independent model. These models interact through interfaces and are executed in a defined sequence within a time-stepped simulation loop. This architectural choice facilitates maintainability, extensibility, and progressive refinement of the simulation framework.

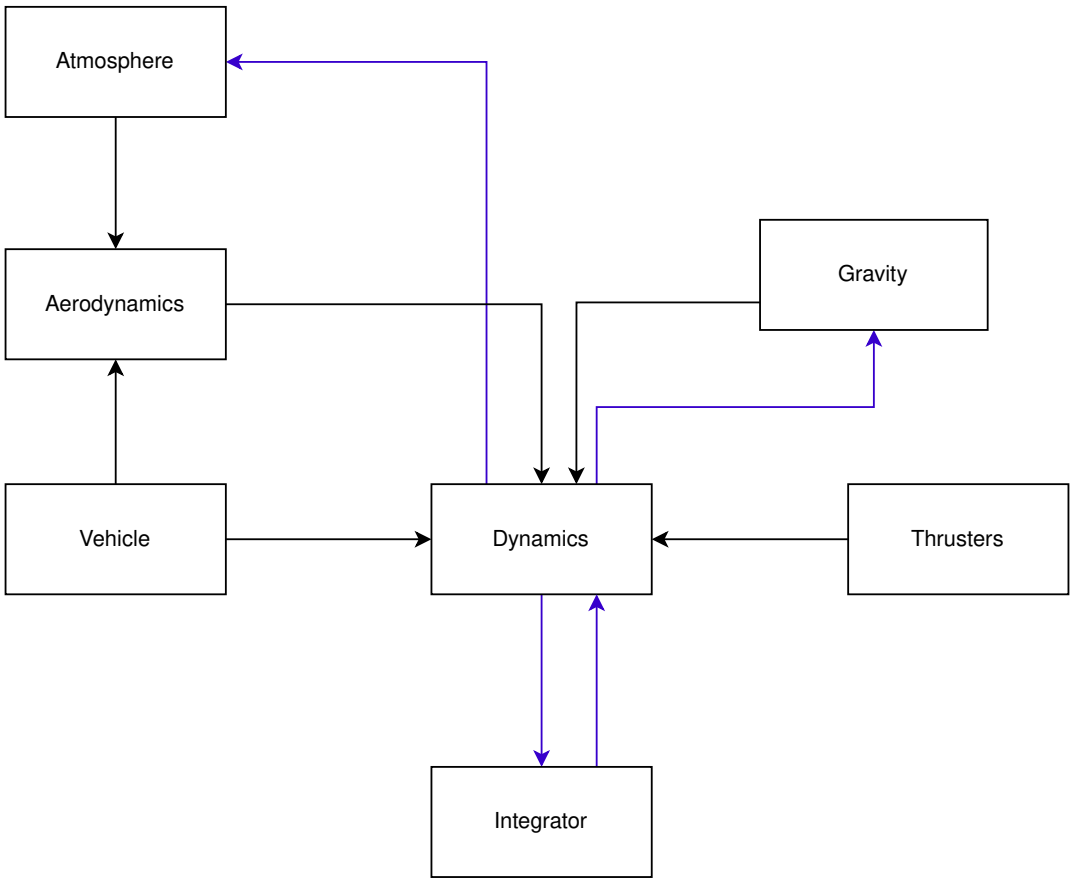


Figure 4.1: Block diagram of the launcher dynamics simulation framework. Purple arrows highlight exchanges related to the launcher state $\mathbf{x}$, while black arrows denote other model interactions.

Figure 4.1 presents a high-level overview of the simulation architecture and the interactions between the different models. At each simulation time step, the launcher state is updated through the numerical integration process and handled by the dynamics model, which acts as the central hub of the simulation.

The vehicle model stores the physical and geometrical properties of the launcher. Propulsion effects are computed by the thrusters model, which generates the thrust force applied to the launcher. Environmental effects acting on the launcher are handled through dedicated models. The atmospheric model computes local atmospheric properties, while the aerodynamic model evaluates the resulting aerodynamic loads. The gravity model computes the gravitational effects.

All model interactions contributing to the vehicle motion and attitude are collected by the dynamics model, which formulates the equations of motion and computes the state derivatives. These derivatives are then propagated in time by the numerical integrator, and the updated state is fed back to the dynamics model for the next simulation step.

This overview provides the necessary context for the detailed description of the individual models. The following sections will first present the models that were already available at the beginning of this thesis, before focusing on the models developed and improved as part of this work.

4.2 Dynamics Model

4.2.1 Role and Scope

The dynamics model constitutes the physical core of the launcher simulation framework. Its primary role is to compute the translational and rotational motion of the vehicle by formulating the equations of motion and aggregating the contributions provided by the different subsystems.

In the overall architecture, the dynamics model acts as a central hub between the physical subsystems and the numerical integration process. Based on the current state of the launcher and on the inputs received from the connected models, it evaluates the state derivatives required to propagate the vehicle motion over time.

4.2.2 State Vector Definition

The dynamics model relies on a state description that characterizes the instantaneous motion of the launcher. In the present framework, the state vector exchanged with the numerical integrator is composed of the translational position of the vehicle center of mass expressed in the inertial frame, the corresponding translational velocity, and the angular velocity.

The state vector handled by the dynamics model and exchanged with the numerical integrator can therefore be written as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix}, \tag{4.1}$$

where $\mathbf{r}$ denotes the position of the vehicle center of mass in the inertial frame, $\mathbf{v}$ the translational velocity, and $\boldsymbol{\omega}$ the angular velocity.

Although the vehicle attitude is part of the physical state of a six-degree-of-freedom rigid-body system, it is not included in the state vector handled by the numerical integrator. Instead, the attitude is stored internally within the dynamics model and updated

separately. This distinction results in a clear separation between the variables propagated through the generic integration process and internal quantities managed using dedicated numerical schemes. The attitude representation and integration strategy are discussed in more detail in 5.1, where dedicated improvements are introduced.

4.2.3 Reference Frames and Equations of Motion

The dynamics model describes the motion of the launcher as a rigid body using a set of well-defined reference frames. An inertial reference frame is used to express the translational motion of the vehicle center of mass, while a body-fixed frame is attached to the launcher and used to express forces, moments, and inertia properties. The orientation of the body frame with respect to the inertial frame is represented using a unit quaternion $\mathbf{q}$.

The translational motion of the launcher is governed by Newton's second law applied to the center of mass:

$$m\,\dot{\mathbf{v}} = \mathbf{F}_{\text{ext}}, \tag{4.2}$$

where m denotes the vehicle mass and $\mathbf{F}_{\text{ext}}$ is the resultant external force expressed in the inertial frame. In the initial implementation, the external force represented only the thrust contribution, which is expressed in the body frame and rotated into the inertial frame using the current vehicle attitude. In the final framework, aerodynamic forces are also included and follow the same transformation process. The gravitational force is then added after being computed internally using a central-body point-mass model.

The rotational motion of the launcher is described by the rigid-body rotational equations expressed in the inertial frame:

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}\left(\mathbf{M}_{\text{ext}} - \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega})\right), \tag{4.3}$$

where $\boldsymbol{\omega}$ is the angular velocity vector, $\mathbf{I}$ is the inertia tensor, and $\mathbf{M}_{\text{ext}}$ is the resultant external moment, all expressed in the inertial reference frame. The inertia tensor is provided by the vehicle model in the body frame and rotated into the inertial frame using the current attitude. Similarly, thrust-induced and aerodynamic moments are computed in the body frame and subsequently transformed into the inertial frame before being applied in the rotational equations. Since gravity is modeled as a central force acting at the vehicle center of mass, no gravitational torque is considered. This formulation follows a classical six-degree-of-freedom rigid-body dynamics approach commonly adopted in launcher and spacecraft simulation frameworks.

4.2.4 Interfaces with Other Models

The interactions between the dynamics model and the other simulation models are illustrated in Figure 4.1 and are described in more detail below.

The dynamics model interfaces with several other models within the simulation architecture in order to compute the translational and rotational motion of the launcher. It considers the different contributions provided by the different subsystems and supplies the state information required for numerical propagation.

The vehicle model provides the mass properties of the launcher, including the total mass, the center of mass location, and the inertia tensor expressed in the body frame. These properties are required to evaluate both the translational and rotational equations of motion and are accessed by the dynamics model at each simulation step.

The thrusters model supplies the thrust forces acting on the launcher. Each thruster provides its force magnitude, direction, and application point expressed in the body frame.

The dynamics model considers the contributions of all active thrusters to compute the resulting thrust force and thrust-induced moment, which are subsequently transformed into the inertial frame using the current vehicle attitude.

In the final framework, the aerodynamics model provides the aerodynamic forces and moments acting on the launcher. These quantities are expressed in the body frame and are transferred to the dynamics model through a dedicated interface. The dynamics model incorporates these contributions into the total external force and moment before evaluating the equations of motion.

In the initial implementation, the gravitational contribution was computed internally within the dynamics model based on the current position of the vehicle center of mass. In the final framework, gravity is provided by a dedicated gravity model through an interface, which returns the gravitational acceleration $\mathbf{g}$ expressed in the inertial frame. The dynamics model then accounts for gravity by adding the corresponding contribution to the equations of motion.

In addition to receiving force and moment contributions, the dynamics model provides state information required by environment-related models, such as the atmospheric and gravity models, to evaluate environment-dependent effects.

Finally, the dynamics model interfaces with the numerical integrator through a bidirectional exchange. Based on the current state vector, the dynamics model computes the corresponding state derivatives, which are provided to the integrator. The integrator then propagates these derivatives in time and returns the updated state vector to the dynamics model for the next simulation step.

4.3 Integrator Model

4.3.1 Role and Scope

The integrator model is responsible for the numerical propagation of the launcher state in time. It operates as a generic numerical component that is independent of the physical modeling and does not embed any knowledge of the launcher dynamics itself. Its role is limited to integrating the system of ordinary differential equations defined by the dynamics model and acts as a dedicated time-integration tool within the simulation loop.

While the integrator propagates the translational state variables, the attitude propagation is handled separately within the dynamics model using a dedicated numerical scheme. This aspect is further discussed in 5.1.

4.3.2 State Propagation Principle

The launcher motion is formulated as a first-order system of ordinary differential equations of the form

$$\dot{\mathbf{x}} = f(\mathbf{x}, t), \tag{4.4}$$

where $\mathbf{x}$ denotes the state vector provided by the dynamics model and $f(\mathbf{x}, t)$ represents the corresponding state derivatives.

State propagation is performed using a classical fourth-order Runge–Kutta (RK4) integration scheme. Given the state vector $\mathbf{x}_n$ at time t_n and a fixed time step Δt, the

scheme computes the updated state $\mathbf{x}_{n+1}$ as:

$$\mathbf{k}_1 = f(\mathbf{x}_n, t_n), \tag{4.5}$$

$$\mathbf{k}_2 = f\left(\mathbf{x}_n + \tfrac{\Delta t}{2}\mathbf{k}_1, t_n + \tfrac{\Delta t}{2}\right), \tag{4.6}$$

$$\mathbf{k}_3 = f\left(\mathbf{x}_n + \tfrac{\Delta t}{2}\mathbf{k}_2, t_n + \tfrac{\Delta t}{2}\right), \tag{4.7}$$

$$\mathbf{k}_4 = f\left(\mathbf{x}_n + \Delta t\,\mathbf{k}_3, t_n + \Delta t\right), \tag{4.8}$$

leading to the state update:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{\Delta t}{6}\left(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4\right). \tag{4.9}$$

A fixed integration time step corresponding to an update rate of 60 Hz is used in the simulator. This value is configurable and was kept unchanged in this work, as it provides a suitable compromise between numerical accuracy and computational cost, while remaining consistent with typical real-time simulation frequencies used in industrial launcher simulators.

The internal numerical implementation of the integrator is not modified nor analyzed in this work and is treated as a black box from the modeling perspective.

4.3.3 Interaction with the Dynamics Model

The interaction between the integrator and the dynamics model follows a bidirectional exchange, as illustrated in Figure 4.1. At each simulation step, the dynamics model computes the state derivatives based on the current state and the physical contributions provided by the other subsystem models. These derivatives are then passed to the integrator for numerical propagation. The integrator advances the state vector in time and returns the updated state to the dynamics model, which uses it as the starting point for the next simulation step.

4.4 Vehicle Model

4.4.1 Role and Scope

The vehicle model provides the mass properties required by the dynamics model, namely the total mass, the position of the center of mass (COM), and the inertia tensor expressed in the vehicle structural reference frame (the body frame).

From an architectural perspective, the vehicle component acts as an interface between the dynamics model and a set of internal sub-components contributing to the overall mass properties. These sub-components are treated as rigid bodies and may represent, for instance, structural elements or propellant tanks. The vehicle is therefore responsible for aggregating their contributions and exposing consistent global mass properties to the dynamics model.

The vehicle component constitutes the only entry point through which the dynamics model accesses mass and inertia information, making it a central element for the computation of translational and rotational motion.

4.4.2 Mass Properties Computation Principle

The vehicle mass properties are computed using a composite approach, in which the vehicle may combine its own mass properties with those of referenced internal bodies.

The vehicle model maintains a list of referenced rigid bodies, each characterized by its own mass, center of mass, inertia tensor, and position expressed in the body frame.

Let body i have mass m_i, a position vector $\mathbf{r}_i$ in the body frame, and a local center of mass $\mathbf{r}_{\mathrm{COM},i}$. The center of mass of body i expressed in the body frame is given by

$$\mathbf{p}_i = \mathbf{r}_i + \mathbf{r}_{\mathrm{COM},i}. \tag{4.10}$$

The total vehicle mass and global center of mass are computed as

$$m = \sum_i m_i, \qquad \mathbf{r}_{\mathrm{COM}} = \frac{1}{m} \sum_i m_i \mathbf{p}_i. \tag{4.11}$$

The total inertia tensor of the vehicle, expressed about the global center of mass and in the body frame, is obtained by summing the inertia tensors of all referenced bodies translated to the global center of mass using the parallel-axis theorem:

$$\mathbf{I}_{\mathrm{tot}} = \sum_i \left(\mathbf{I}_i + m_i \left(\|\mathbf{d}_i\|^2 \mathbf{I}_3 - \mathbf{d}_i \mathbf{d}_i^\top \right) \right), \tag{4.12}$$

where $\mathbf{I}_i$ denotes the inertia tensor of body i expressed about its own center of mass, $\mathbf{I}_3$ is the 3×3 identity matrix, and

$$\mathbf{d}_i = \mathbf{r}_{\mathrm{COM}} - \mathbf{p}_i \tag{4.13}$$

is the displacement vector between the center of mass of body i and the global vehicle center of mass.

These computations are performed whenever updated mass properties are requested by the dynamics model.

4.4.3 Tank

In the current simulation configuration, the vehicle references a single sub-body corresponding to a propellant tank. The tank submodel represents a rigid cylindrical body whose mass may vary over time due to propellant consumption.

The tank mass is updated by integrating propellant mass flow rates provided by the propulsion subsystem. Assuming a mass flow $\dot{m}$ and a simulation time step Δt, the mass evolution is given by

$$m(t + \Delta t) = m(t) + \dot{m}\, \Delta t, \tag{4.14}$$

with the convention that $\dot{m} < 0$ corresponds to propellant consumption. After each mass update, the inertia tensor of the tank is recomputed consistently with the new mass.

In the present implementation, the tank geometry is approximated as a uniform solid cylinder of length L and radius R, with its center of mass located on the symmetry axis at mid-length. Under this assumption, the principal moments of inertia are

$$I_{xx} = I_{yy} = \frac{m}{12} \left(3R^2 + L^2 \right), \qquad I_{zz} = \frac{1}{2} m R^2, \tag{4.15}$$

and the products of inertia are set to zero.

4.4.4 Interface with Other Models

The dynamics model interacts with the vehicle exclusively through a dedicated interface that provides the current mass properties. At each computation of the state derivatives and state outputs, the dynamics model invokes the vehicle method returning the total mass m, the global center of mass $\mathbf{r}_{\mathrm{COM}}$ expressed in the body frame, and the inertia tensor $\mathbf{I}_{\mathrm{tot}}$ expressed about the global center of mass.

The center of mass position and inertia tensor are then transformed into the inertial reference frame within the dynamics model and used to compute linear accelerations, angular accelerations, and attitude evolution.

In addition to its interface with the dynamics model, the vehicle model also provides geometric reference quantities to the aerodynamics model. In particular, the reference length l_{ref} and reference area A_{ref}, derived from the vehicle geometry, are used for the computation of aerodynamic forces and moments.

4.5 Thruster Model

4.5.1 Role and Scope

The thruster model represents a propulsion actuator providing thrust forces applied at a fixed location on the launcher structure. Within the simulation framework, its role is to supply an external force contribution to the dynamics model and to provide a propellant mass flow rate that can be consumed by the tank model to update the vehicle mass properties over time.

In the current framework, the thruster model is intentionally kept simple: it does not introduce internal dynamic states and does not model combustion processes, throttle laws, ignition transients, or torque injection.

4.5.2 Thrust and Mass Flow Modelling Principle

The thruster provides a thrust force expressed in the body frame through a force description defined by an application point $\mathbf{r}_T$, a unit direction vector $\hat{\mathbf{d}}_T$, and an amplitude T. The thrust force applied to the vehicle is expressed as

$$\mathbf{F}_T = T\,\hat{\mathbf{d}}_T. \tag{4.16}$$

The associated moment about the vehicle center of mass is generated implicitly by the lever arm between the thrust application point and the current center of mass; its computation is handled within the dynamics model.

The propellant mass flow rate associated with the thrust is derived using the standard rocket relation:

$$\dot{m} = -\frac{T}{g_0 I_{sp}}, \tag{4.17}$$

where g_0 denotes the standard gravity constant and I_{sp} the specific impulse. The negative sign follows the convention adopted in the tank model, such that a negative mass flow corresponds to propellant consumption.

4.5.3 Interfaces with Other Models

The dynamics model retrieves thrust force information from each thruster at every evaluation of the equations of motion and aggregates these contributions to compute the total

propulsion forces and moments acting on the vehicle. The computation of thrust-induced moments is performed within the dynamics model.

In parallel, the tank component, which is part of the vehicle model, receives the propellant mass flow rate provided by the thruster model to account for propellant consumption over time.

Chapter 5

Models Developed

5.1 Attitude Improvements

5.1.1 Motivation

In the initial version of the launcher simulation framework, the vehicle attitude was already represented and propagated within the dynamics model. However, this attitude information was not explicitly used by the pre-existing subsystem models. In particular, propulsion forces were applied in the body frame using fixed thrust directions, and gravitational effects were computed independently of the vehicle orientation. As a result, inaccuracies in attitude propagation did not affect the simulation results and remained largely unnoticed during the early development stages.

This situation changed with the introduction of environment-dependent models, namely the atmospheric and aerodynamics models. These models rely directly on the vehicle attitude to perform reference frame transformations. In the atmospheric model, the attitude is required to express wind velocities and relative airflow quantities in the launcher body axes. In the aerodynamics model, the body-frame relative airflow velocity constitutes a key input for the computation of angles of attack, sideslip angles, and the resulting aerodynamic forces and moments.

During the integration of these models, physically inconsistent simulation results were observed, in particular unstable attitude-dependent quantities. These anomalies indicated that the existing attitude propagation scheme did not provide sufficient accuracy or consistency once the attitude became an active input to the simulation pipeline rather than a passive internal state.

As described in the presentation of the pre-existing dynamics model 4.2, the vehicle attitude was propagated using a dedicated internal update mechanism that was decoupled from the numerical integration scheme used for the translational dynamics. This approach relied on a simplified treatment of rotational kinematics and was initially considered sufficient, as the resulting attitude evolution appeared smooth and numerically well-behaved under the validation procedures in place at the time.

For these reasons, improving the attitude propagation scheme became a necessary prerequisite for the reliable implementation of the atmospheric and aerodynamics models. The objective of the work presented in this section is therefore to refine the attitude propagation scheme, assess its accuracy using a robust attitude error metric, and ensure that the resulting attitude information can be safely used by all downstream models within the simulation framework.

5.1.2 Attitude Representation Choice

Several mathematical representations can be used to describe the orientation of a rigid body in three-dimensional space, including Euler angles, rotation matrices, and quaternions. Euler-angle formulations offer an intuitive interpretation but suffer from kinematic singularities (gimbal lock) and can lead to numerical difficulties when large or coupled rotations are involved. Rotation matrices provide a singularity-free representation but require nine parameters subject to orthogonality constraints, which must be preserved numerically.

Representation	Parameters	Singularities	Numerical constraints
Euler angles	3	Yes (gimbal lock)	None
Rotation matrices	9	No	Orthogonality preservation
Quaternions	4	No	Unit-norm constraint

Table 5.1: Comparison of common attitude representations used in rigid-body simulations.

In the present framework, the vehicle attitude is represented using unit quaternions, a choice that was already adopted in the initial implementation of the simulation framework and therefore retained in this work. Quaternions provide a compact and singularity-free representation that is well suited for numerical integration and repeated reference frame transformations. This representation requires the quaternion to remain normalized, which is enforced through systematic renormalization during the attitude propagation, but avoids the more restrictive orthogonality constraints associated with rotation matrices.

These properties make quaternion-based representations particularly suitable for time-marching six-degree-of-freedom rigid-body simulations.

5.1.3 Quaternion Representation and Conventions

The vehicle attitude is represented using a unit quaternion $\mathbf{q} = [q_w,\ q_x,\ q_y,\ q_z]^\top$, where q_w denotes the scalar part and $[q_x,\ q_y,\ q_z]$ the vector part.

In the present framework, the quaternion $\mathbf{q}$ represents the orientation of the launcher body frame with respect to the inertial reference frame.

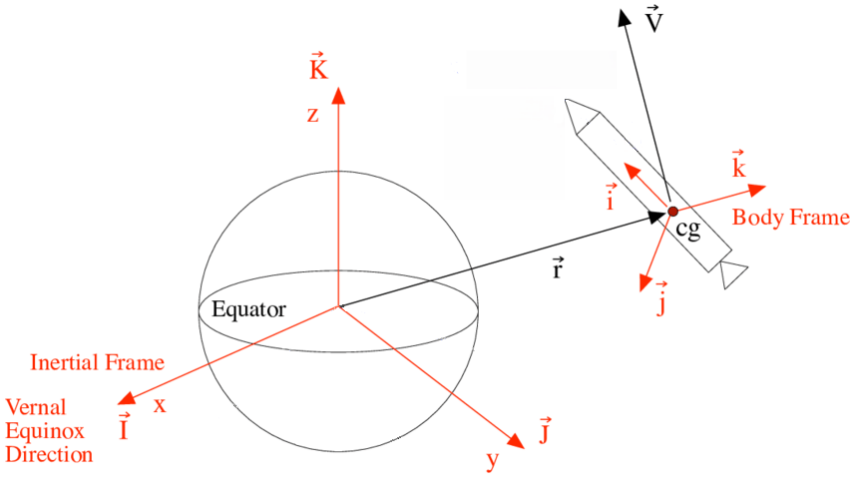


Figure 5.1: Schematic representation of the inertial reference frame and the launcher body reference frame. The attitude quaternion $\mathbf{q}$ describes the orientation of the body frame with respect to the inertial frame. Adapted from [21].

The corresponding rotation matrix $\mathbf{R}(\mathbf{q})$, which provides an equivalent representation of the same orientation, is given by

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} 1-2(q_y^2+q_z^2) & 2(q_xq_y - q_zq_w) & 2(q_xq_z + q_yq_w) \\ 2(q_xq_y + q_zq_w) & 1-2(q_x^2+q_z^2) & 2(q_yq_z - q_xq_w) \\ 2(q_xq_z - q_yq_w) & 2(q_yq_z + q_xq_w) & 1-2(q_x^2+q_y^2) \end{bmatrix}. \tag{5.1}$$

This matrix is orthogonal by construction when $\mathbf{q}$ is normalised.

Quaternion algebra exhibits a double-cover property, such that the quaternions $\mathbf{q}$ and $-\mathbf{q}$ describe the same physical orientation. To ensure numerical consistency and continuity of the attitude time history, the quaternion is systematically normalised and its sign is chosen such that $q_w \geq 0$.

Attitude propagation is performed through quaternion composition. Over a simulation time step Δt, the rotational motion of the vehicle is represented by an incremental rotation quaternion $\Delta\mathbf{q}$, which is constructed from the angular velocity integrated over the time step. This incremental quaternion represents the rotation undergone by the body during the current step.

The updated attitude is then obtained by composing this incremental rotation with the current orientation according to

$$\mathbf{q}_{k+1} = \Delta\mathbf{q} \otimes \mathbf{q}_k, \tag{5.2}$$

where $\otimes$ denotes quaternion multiplication. This operation corresponds to the composition of successive rotations and is equivalent to multiplying the associated rotation matrices, while preserving numerical robustness.

These conventions are consistently applied throughout the simulation framework and constitute the basis for all subsequent reference frame transformations used by the atmosphere and aerodynamics models.

5.1.4 Previous Attitude Implementation

In the initial implementation of the simulation framework, the vehicle attitude was stored in quaternion form but updated through an intermediate rotation-matrix-based procedure executed outside the main numerical integration loop. While the state variables were propagated using a fourth-order Runge–Kutta scheme, the attitude was updated separately using a custom kinematic approximation.

At each simulation step, the current attitude quaternion was first converted into a rotation matrix. The rotational motion over the time step was then approximated by updating this matrix using a truncated series expansion of the matrix exponential based on the instantaneous angular velocity. The update took the form

$$\mathbf{R}_{k+1} = \left(\mathbf{I}_3 + \boldsymbol{\Omega}\,\Delta t + \frac{1}{2}\boldsymbol{\Omega}^2\Delta t^2\right)\mathbf{R}_k, \tag{5.3}$$

where $\boldsymbol{\Omega}$ denotes the antisymmetric matrix associated with the angular velocity vector $\boldsymbol{\omega}$,

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \tag{5.4}$$

and Δt is the simulation time step.

After this update, the resulting rotation matrix was converted back into a quaternion and renormalised. Although the attitude was stored as a quaternion, this approach did

not fully exploit the benefits of quaternion-based representations, as the propagation itself was performed in matrix form. An equivalent implementation could therefore have relied directly on rotation matrices without altering the numerical behaviour.

This attitude update scheme produced a continuous orientation evolution but was only weakly coupled to the numerical integration of the dynamics, as the attitude was updated once per time step, independently of the intermediate stages of the Runge–Kutta scheme. As a consequence, the attitude propagation effectively operated at a lower integration fidelity than the Runge–Kutta scheme used for the other state variables, leading to inconsistencies once the attitude became tightly coupled with environment-dependent models.

The accuracy of the initial attitude propagation scheme was first assessed using a component-wise comparison of the attitude quaternion with a reference solution. This approach consisted in monitoring the differences between individual quaternion components over time and was initially deemed acceptable, as these differences remained numerically small.

Using this component-wise validation approach, the attitude appeared to be accurately propagated, with relative errors on individual quaternion components remaining on the order of 10^{-5} for the scalar component and 10^{-3} for selected vector components.

Although these figures suggested a high level of accuracy, such component-wise comparisons do not reflect the true angular separation between orientations and may lead to a misleading assessment of the attitude error. Indeed, due to the geometric nature of quaternions and their nonlinear relationship with rotation angles, small differences in individual components may still correspond to large orientation errors.

To address this limitation, a quaternion-based angular distance metric was introduced. Given a reference attitude quaternion $\mathbf{q}_{\text{ref}}$ and a simulated attitude quaternion $\mathbf{q}$, the relative error quaternion is defined as

$$\mathbf{q}_{\text{err}} = \mathbf{q}_{\text{ref}}^{-1} \otimes \mathbf{q}. \tag{5.5}$$

The corresponding attitude error angle is then computed from the scalar part of $\mathbf{q}_{\text{err}}$ as

$$\theta_{\text{err}} = 2 \arccos\left(|q_{\text{err},w}|\right), \tag{5.6}$$

which represents the minimal rotation required to align the two orientations.

Applying this metric to the initial attitude implementation revealed significant orientation discrepancies over short simulation durations. While the component-wise differences remained small, the angular error exhibited a rapid growth, demonstrating that the previous validation approach masked substantial attitude inaccuracies. This observation confirmed that the original propagation scheme did not provide the level of accuracy required once the attitude became an active input to environment-dependent models.

5.1.5 Improved Attitude Propagation Scheme

To address the limitations identified in the previous attitude update mechanism, a new propagation scheme was implemented based on direct quaternion integration. The objective was to ensure a numerically robust attitude evolution that remains consistent with the time integration of the translational and rotational dynamics.

In the improved formulation, the attitude quaternion is still propagated explicitly once per simulation time step, as in the previous implementation. Ideally, the attitude quaternion could be included directly in the Runge–Kutta state vector and integrated alongside the translational dynamics. However, since high-fidelity attitude dynamics were not a primary objective of this work, such an approach was not adopted in order to limit

both implementation complexity and computational cost. The selected scheme was found to provide sufficient accuracy for the intended applications, as will be confirmed through numerical validation.

The attitude update is based on the incremental rotation undergone over a single time step. The angular velocity $\boldsymbol{\omega}$ and angular acceleration $\boldsymbol{\alpha}$ provided by the dynamics model are used to approximate the angular rate at the midpoint of the time step,

$$\boldsymbol{\omega}_{\text{mid}} = \boldsymbol{\omega} + \frac{1}{2}\boldsymbol{\alpha}\,\Delta t. \tag{5.7}$$

The corresponding rotation angle

$$\theta = \|\boldsymbol{\omega}_{\text{mid}}\|\Delta t \tag{5.8}$$

represents the physical angle of rotation of the vehicle over the time step.

For rotation angles larger than a numerical threshold ($\theta > 10^{-12}\,\text{rad}$), the incremental rotation is constructed using an axis–angle representation,

$$\Delta\mathbf{q} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \mathbf{u}\,\sin\left(\frac{\theta}{2}\right) \end{bmatrix}, \qquad \mathbf{u} = \frac{\boldsymbol{\omega}_{\text{mid}}}{\|\boldsymbol{\omega}_{\text{mid}}\|}. \tag{5.9}$$

For very small angular increments, a first-order approximation, based on $\sin(\theta/2) \approx \theta/2$ and $\cos(\theta/2) \approx 1$, is adopted. This allows the update to be expressed without explicit axis normalisation and avoids the associated numerical instabilities,

$$\Delta\mathbf{q} \approx \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega}_{\text{mid}}\,\Delta t \end{bmatrix}. \tag{5.10}$$

The updated attitude is then obtained through quaternion composition,

$$\mathbf{q}_{k+1} = \Delta\mathbf{q} \otimes \mathbf{q}_k, \tag{5.11}$$

followed by normalisation to preserve unit norm.

5.1.6 Validation and Comparison of Attitude Propagation Schemes

The improved attitude propagation scheme was validated using a controlled test case designed to isolate attitude integration errors and allow comparison against an analytical reference solution. The test scenario consisted of a rigid body subjected to a constant torque, resulting in a uniform angular acceleration about a principal axis.

The simulated vehicle was defined as a rigid body with a mass of 1000 kg and a diagonal inertia tensor corresponding to a uniform body of characteristic length 2 m,

$$\mathbf{I} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}, \qquad I = \frac{1}{6}mL^2. \tag{5.12}$$

A constant thrust force was applied at an offset from the center of mass, generating a constant torque about the body z-axis. The initial attitude was set to the identity quaternion, and the initial angular velocity was zero.

Under these conditions, the rotational motion admits a closed-form solution. For a constant angular acceleration α_z, the rotation angle about the z-axis after a duration t is given by

$$\theta_z(t) = \frac{1}{2}\alpha_z t^2. \tag{5.13}$$

The corresponding reference attitude quaternion is therefore

$$\mathbf{q}_{\text{ref}} = \begin{bmatrix} \cos(\theta_z/2) \\ 0 \\ 0 \\ \sin(\theta_z/2) \end{bmatrix}. \tag{5.14}$$

The attitude error was evaluated using the quaternion-based angular distance metric introduced previously 5.1.4. Identical simulation scenarios, corresponding to the test case described above, were executed using both the previous attitude propagation scheme and the improved formulation.

After a simulation duration of 3 s, the previous implementation exhibited an attitude error of approximately 2.3°, despite component-wise quaternion errors remaining small. Under identical conditions, the improved propagation scheme reduced the attitude error to approximately 0.01° (see Table 5.2).

To further assess long-term error accumulation, the simulation was extended to 120 s. In this case, the previous implementation exhibited a rapid divergence, leading to an attitude error of approximately 114°. In contrast, the improved scheme maintained an attitude error below 1° over the same duration. Such a small error level over several minutes has a negligible impact on the computation of aerodynamic angles and loads when compared to other sources of uncertainty, such as atmospheric variability and aerodynamic modelling assumptions.

These results demonstrate that the improved attitude propagation scheme provides a level of accuracy sufficient for reference-frame transformations and environment-dependent computations. The validated attitude information therefore constitutes a reliable basis for the atmospheric modelling introduced in the following section.

Propagation scheme	Simulation time	Angular error (deg)
Previous implementation	3 s	2.33
Improved implementation	3 s	0.012
Previous implementation	120 s	114
Improved implementation	120 s	< 1

Table 5.2: Comparison of attitude propagation accuracy for the constant-torque validation test case.

5.2 Atmosphere Model

5.2.1 Motivation and Role

The introduction of an atmosphere model constitutes a key step in the evolution of the launcher dynamics simulation framework. In its initial configuration, the simulation was primarily driven by propulsion and gravity effects, and the surrounding environment did not explicitly influence the vehicle motion. Under these conditions, the launcher trajectory and attitude evolution were insensitive to atmospheric parameters.

This situation changes fundamentally with the integration of aerodynamic modelling. Aerodynamic forces and moments depend directly on the interaction between the vehicle and the surrounding air mass. In particular, the computation of aerodynamic loads requires knowledge of the local air density, the speed of sound, and the relative velocity of the airflow with respect to the vehicle. These quantities cannot be derived solely from

the vehicle state propagated by the dynamics model and therefore require a dedicated environmental representation.

The atmosphere model introduced in this work is designed to fulfill this role within the simulation framework. It provides a physically consistent description of the atmospheric environment encountered by the launcher during its ascent phase, based on the current vehicle position, velocity, and attitude. By combining standard atmosphere properties, wind effects, and Earth rotation, the model delivers the environmental quantities required by the aerodynamics model to evaluate aerodynamic forces and moments.

From an architectural perspective, the atmosphere model acts as an intermediate component between the dynamics model and the aerodynamics model. It retrieves the instantaneous vehicle state from the dynamics model and processes this information to compute local atmospheric conditions. The resulting quantities are then exposed through a dedicated interface and used downstream by the aerodynamics model. This separation of concerns preserves the modular structure of the simulation framework and allows each physical phenomenon to be modelled independently.

Beyond nominal aerodynamic load computation, the atmosphere model also plays an important role in the analysis of dispersion mechanisms and unsteady effects. Wind profiles and stochastic atmospheric turbulence directly affect the relative airflow experienced by the launcher and may induce variations in aerodynamic forces, moments, and attitude response. A coherent atmospheric representation is therefore essential to capture these effects in a consistent manner within the simulation pipeline.

The atmosphere model developed in this work is intended to support launcher dynamics and aerodynamic analyses within the scope of this thesis by providing a physically consistent and computationally efficient representation of the atmospheric environment.

5.2.2 Modelling Scope and Assumptions

The atmosphere model developed in this work is intended to provide a simplified yet physically consistent representation of the environmental conditions encountered by the launcher during its ascent. Its scope is deliberately limited to phenomena that have a first-order impact on aerodynamic load computation and trajectory dispersion, while avoiding unnecessary modelling complexity that would not be justified within the context of this thesis.

Atmospheric properties are assumed to depend solely on altitude. This assumption is consistent with the use of standard atmosphere representations. As a consequence, horizontal inhomogeneities, seasonal effects, and transient weather phenomena are not explicitly modelled.

The atmosphere is considered stationary in an Earth-fixed reference frame and is assumed to co-rotate with the Earth. As a consequence, the relative airflow experienced by the vehicle depends not only on its inertial motion, but also on the rotational motion of the Earth.

Mean wind effects are represented through externally defined vertical wind profiles, which specify the wind velocity components as functions of altitude. These profiles are treated as deterministic inputs during a simulation run and are assumed to remain constant over time. This approach allows the influence of wind on the launcher trajectory and aerodynamic loads to be analysed in a controlled and repeatable manner.

In addition to mean wind effects, high-frequency atmospheric disturbances are represented using a stochastic turbulence model based on a classical engineering gust formulation. The turbulence is assumed to be locally homogeneous and statistically stationary, and its characteristics depend on the current altitude and vehicle airspeed. The result-

ing turbulent velocity components are superimposed on the mean wind field and directly affect the relative airflow experienced by the launcher.

The atmosphere model does not introduce internal dynamic states that influence the numerical stability of the global simulation. All atmospheric quantities are computed instantaneously from the current vehicle state and a set of configuration parameters. This design choice ensures a clear separation between environmental modelling and vehicle dynamics, while maintaining computational efficiency and numerical robustness throughout the simulation.

Finally, the modelling assumptions adopted in this work are chosen to strike a balance between physical realism and implementation complexity. While they limit the fidelity of the atmospheric representation, they are considered appropriate for the objectives of this thesis and provide a consistent basis for the computation of aerodynamic loads. More advanced atmospheric representations, including time-varying or site-dependent models, could be incorporated in future developments without modifying the interface between the atmosphere and aerodynamics models.

5.2.3 Atmospheric Properties

The atmosphere model provides the local air density and speed of sound used to evaluate dynamic pressure and Mach number for aerodynamic load computation.

Standard atmosphere representation

The atmosphere model implemented in this work is based on the Committee on Extension to the Standard Atmosphere (COESA) reference atmosphere. The COESA model constitutes the basis of the U.S. Standard Atmosphere [2], which defines a reference vertical distribution of atmospheric properties for engineering and aerospace applications. In the present framework, the COESA-based atmosphere is available as a reference model and does not require additional development or configuration effort from the user.

Although the COESA reference atmosphere provides a comprehensive set of atmospheric quantities as functions of altitude, including temperature, pressure, and composition, only air density ρ and speed of sound a are required for the purposes of this work. These two quantities are sufficient to support the computation of dynamic pressure, Mach number, and aerodynamic coefficients within the simulation framework.

Atmospheric properties are provided in the form of precomputed tabulated data derived from the COESA reference atmosphere. These tables are loaded automatically during the initialization phase of the simulation and contain discrete values of ρ and a defined at successive altitude levels. The tabulated data extend from sea level up to an altitude of 100 km, with a vertical resolution of one meter obtained by evaluating the COESA model over the altitude range of interest. At each simulation step, the vehicle altitude is computed from its Earth-centered position, and the corresponding atmospheric properties are obtained by direct indexing of the tabulated data at the nearest lower altitude level.

This tabulated approach ensures computational efficiency and numerical robustness while remaining fully consistent with the COESA standard. Since atmospheric properties are accessed exclusively through altitude-indexed tables, alternative reference atmospheres or mission-specific models can be incorporated by replacing the tabulated data, without modifying the interface between the atmosphere and aerodynamics models.

Altitude (m)	Air density ρ ($\mathrm{kg\,m^{-3}}$)	Speed of sound a ($\mathrm{m\,s^{-1}}$)
58643	3.4245×10^{-4}	316.492
58644	3.4241×10^{-4}	316.490
58645	3.4237×10^{-4}	316.488
58646	3.4233×10^{-4}	316.486
58647	3.4228×10^{-4}	316.485

Table 5.3: Excerpt of the tabulated atmospheric properties derived from the COESA reference atmosphere. Air density ρ and speed of sound a are provided at one-meter altitude intervals.

Although the values are reported in Table 5.3 with limited significant digits for clarity, atmospheric properties are stored and accessed internally at full double-precision resolution.

Mach number

The speed of sound a is required for the computation of the Mach number,

$$M = \frac{\|\mathbf{v}_{\mathrm{rel}}\|}{a}. \tag{5.15}$$

The Mach number is a dimensionless quantity commonly used in aerodynamics to characterize the flow regime. The Mach number governs the transition between subsonic, transonic, and supersonic flight conditions and plays a key role in the selection of appropriate aerodynamic coefficients. Accurate evaluation of the Mach number is therefore essential to ensure consistency between the atmospheric and aerodynamic models.

5.2.4 Reference Frames

The computation of atmospheric quantities relevant to aerodynamic modelling involves several reference frames, each of which is associated with a specific physical interpretation. While the launcher translational dynamics are formulated in an inertial reference frame, atmospheric properties such as wind and air mass motion are more naturally described in Earth-fixed frames. Furthermore, aerodynamic forces and moments must ultimately be expressed in the vehicle body frame. A consistent framework for handling reference frame transformations is therefore required to relate these quantities and compute the relative airflow experienced by the launcher.

The atmosphere model relies on four main reference frames: the Earth-centred inertial frame (denoted ECI), the Earth-centred, Earth-fixed frame (ECEF), the launcher body frame, and the local North–East–Down (NED) frame. Their respective roles and interactions are described in the following subsections.

Overview of Reference Frames

The ECI frame is Earth-centered and assumed to be non-rotating with respect to distant stars, providing a practical approximation of an inertial frame over the time scales considered. It is used as the primary frame for the numerical integration of the equations of motion within the dynamics model. The vehicle position and velocity propagated by the integrator are expressed in this frame.

The ECEF frame rotates with the Earth about its spin axis. This frame provides a natural reference for describing the atmospheric environment, which is assumed to be

stationary with respect to the rotating Earth. Atmospheric wind profiles and air mass motion are therefore defined in this frame.

The body frame is rigidly attached to the launcher and moves with it. This frame is used to express aerodynamic forces, moments, and relative airflow quantities. The orientation of the body frame with respect to the inertial frame is represented by a unit quaternion provided by the dynamics model.

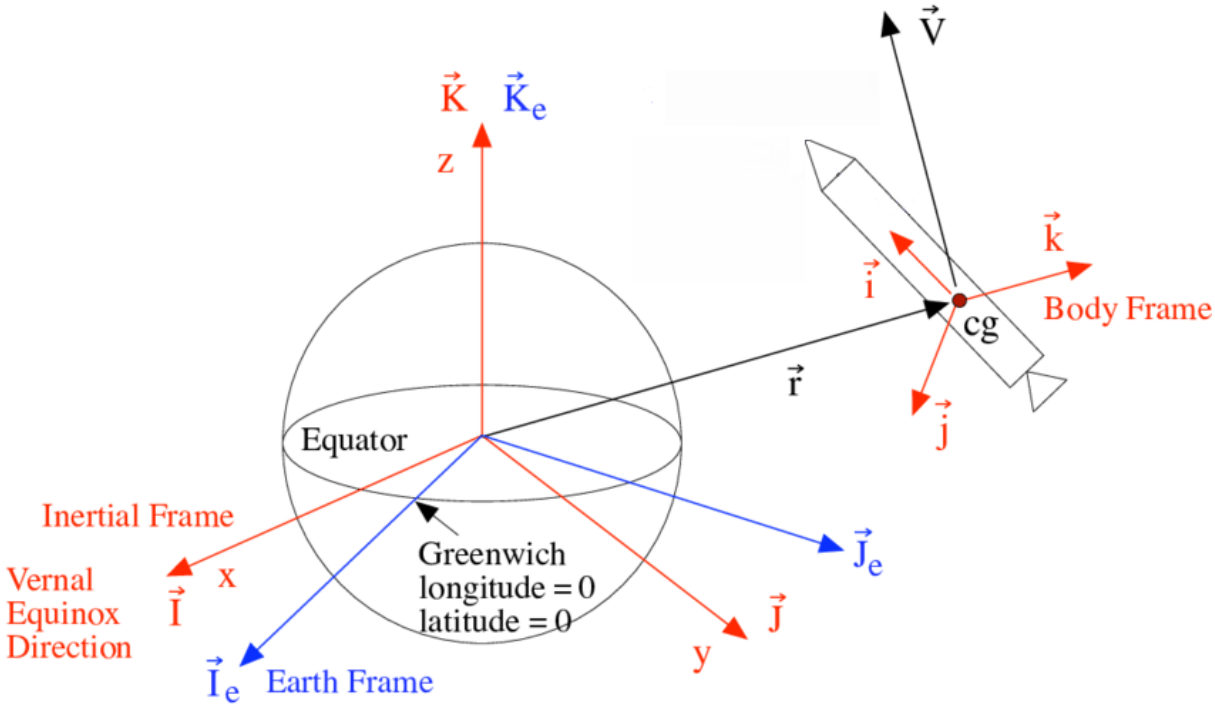


Figure 5.2: Overview of the global reference frames used in the atmosphere model. The Earth-centered inertial frame (ECI), defined with its X-axis pointing towards the vernal equinox, the Earth-fixed frame (ECEF), and the launcher body frame are illustrated. Adapted from [21].

In addition to these global reference frames, a local navigation frame is constructed at the vehicle position. Expressed in the ECEF frame, a local navigation frame following the North–East–Down (NED) convention is constructed. This local frame provides a convenient representation for wind profiles, which are commonly specified in terms of northward, eastward, and vertical components as functions of altitude.

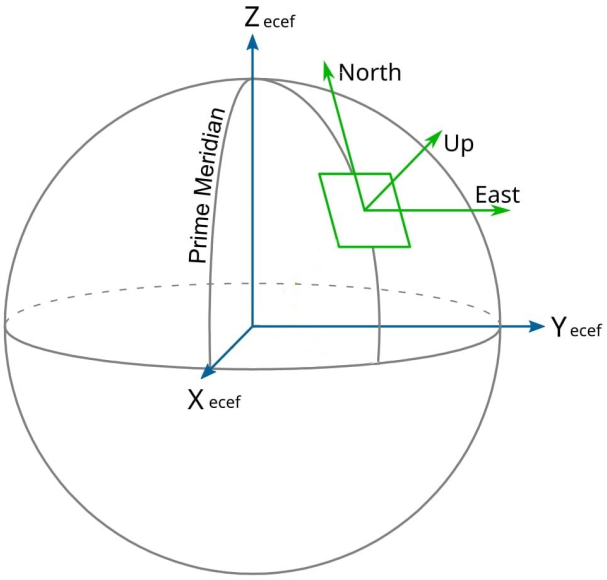


Figure 5.3: Local North–East–Down (NED) reference frame constructed at the vehicle position. The local frame is defined with respect to the ECEF frame and follows the northward, eastward, and downward directions associated with the local tangent plane [22].

Earth-Fixed Frame and Earth Rotation Modelling

The atmosphere is assumed co-rotate with the Earth and to be stationary in the ECEF frame. As a result, the computation of the relative airflow experienced by the launcher requires explicit modelling of Earth rotation and consistent transformations between the ECI, ECEF, and body frames.

The transformation between the ECI frame and the ECEF frame is modelled as a rotation about the Earth's spin axis. The corresponding rotation angle is computed from the simulation epoch using a simplified sidereal time formulation. Let JD denote the Julian Date associated with the current simulation time. The Julian Date is a continuous count of days elapsed since the reference epoch January 1, 4713 BC, commonly used in astrodynamics to provide a uniform and unambiguous representation of time. Let

$$d = JD - 2451545 \tag{5.16}$$

be the number of days elapsed since the J2000 epoch, corresponding to January 1, 2000 at 12:00 TT, where TT (Terrestrial Time) is a uniform time scale commonly used in astrodynamics for the definition of reference epochs and inertial frames. The Earth rotation angle is then approximated as

$$\theta = 2\pi \,\mathrm{frac}(0.7790572732640 + 1.00273781191135448\, d)\,, \tag{5.17}$$

where $\mathrm{frac}(\cdot)$ denotes the fractional part. This expression provides a computationally efficient approximation of the Greenwich sidereal angle suitable for engineering-level simulations.

The associated rotation matrix is defined as

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5.18}$$

which maps vectors expressed in the ECI frame into the ECEF frame. In particular, the vehicle position in the ECEF frame is obtained as

$$\mathbf{r}_{ECEF} = \mathbf{R}_\theta \, \mathbf{r}_{ECI}. \tag{5.19}$$

Due to Earth rotation, an air mass located at position $\mathbf{r}_{ECI}$ possesses a non-zero velocity with respect to the inertial frame. Assuming rigid co-rotation of the atmosphere with the Earth, this co-rotation velocity is given by

$$\mathbf{v}_{\text{corot},ECI} = \boldsymbol{\Omega}_E \times \mathbf{r}_{ECI}, \tag{5.20}$$

where $\boldsymbol{\Omega}_E = [0 \; 0 \; \Omega_E]^\top$ denotes the Earth angular velocity vector, with $\Omega_E = 7.292115 \times 10^{-5}\,\mathrm{rad\,s^{-1}}$. This co-rotation velocity is the kinematic consequence of the Earth rotation described by the ECI-to-ECEF transformation matrix $\mathbf{R}(\theta)$ and reflects the time variation of the Earth-fixed frame with respect to the inertial frame.

Local NED-to-ECEF Transformation

As illustrated in Figure 5.3, the local North–East–Down (NED) frame is defined at the vehicle position, with its axes aligned with the local tangent plane: North and East lie in the horizontal plane, while Down points towards the Earth's centre.

In this work, a local NED frame is constructed at the instantaneous vehicle position and related to the ECEF frame through a position-dependent rotation based on the local tangent plane.

Let $\mathbf{r}_{ECEF}$ denote the vehicle position expressed in the ECEF frame. Under a local tangent-plane approximation of the Earth, the local vertical direction is assumed to coincide with the radial direction, such that the upward unit vector is defined as

$$\hat{\mathbf{u}} = \frac{\mathbf{r}_{ECEF}}{\|\mathbf{r}_{ECEF}\|}. \tag{5.21}$$

The local East direction is obtained from the Earth spin axis unit vector $\hat{\mathbf{k}} = [0 \;\; 0 \;\; 1]^\top$ as

$$\hat{\mathbf{e}} = \frac{\hat{\mathbf{k}} \times \hat{\mathbf{u}}}{\left\|\hat{\mathbf{k}} \times \hat{\mathbf{u}}\right\|}, \tag{5.22}$$

and the local North direction follows from

$$\hat{\mathbf{n}} = \hat{\mathbf{u}} \times \hat{\mathbf{e}}. \tag{5.23}$$

The Down direction is defined as

$$\hat{\mathbf{d}} = -\hat{\mathbf{u}}. \tag{5.24}$$

By construction, the triad $\{\hat{\mathbf{n}}, \hat{\mathbf{e}}, \hat{\mathbf{d}}\}$ forms an orthonormal basis aligned with the local tangent plane at the vehicle position.

A wind velocity expressed in NED components $\mathbf{v}_{wind,NED} = [V_N \;\; V_E \;\; V_D]^\top$ can then be mapped to the ECEF frame as

$$\mathbf{v}_{wind,ECEF} = V_N \, \hat{\mathbf{n}} + V_E \, \hat{\mathbf{e}} + V_D \, \hat{\mathbf{d}}. \tag{5.25}$$

Equivalently, introducing the rotation matrix

$$\mathbf{R}_{NED \to ECEF} = \left[\hat{\mathbf{n}} \;\; \hat{\mathbf{e}} \;\; \hat{\mathbf{d}}\right], \tag{5.26}$$

the transformation may be written as

$$\mathbf{v}_{wind,ECEF} = \mathbf{R}_{NED \to ECEF}\, \mathbf{v}_{wind,NED}. \tag{5.27}$$

This formulation provides a consistent link between locally defined atmospheric quantities and the global ECEF frame, enabling subsequent transformations to the inertial and body frames.

5.2.5 Relative Airflow Velocity

Aerodynamic forces and moments acting on the launcher depend on the velocity of the surrounding air relative to the vehicle. The computation of this relative airflow velocity therefore constitutes a central output of the atmosphere model and provides the primary input to the aerodynamics model.

In the present framework, the vehicle translational velocity is provided by the dynamics model in the ECI frame. Atmospheric motion, on the other hand, is described through a combination of Earth rotation effects, mean wind profiles, and stochastic turbulence. In order to combine these contributions consistently, all velocity components are ultimately expressed in the launcher body frame.

Vehicle Velocity in the Body Frame

The inertial translational velocity of the launcher, denoted by $\mathbf{v}_{ECI}$, is obtained directly from the dynamics model. Using the attitude quaternion $\mathbf{q}$, the corresponding rotation matrix $\mathbf{R}(\mathbf{q})$ is constructed and used to express the vehicle velocity in the body frame as

$$\mathbf{v}_{Body} = \mathbf{R}(\mathbf{q})\, \mathbf{v}_{ECI}. \tag{5.28}$$

This body-frame velocity represents the inertial translational motion of the launcher, expressed in the body frame, and is used to compute the relative airflow velocity by comparison with the atmospheric motion.

Atmospheric Co-Rotation Velocity

As introduced in Section 5.2.4, the rigid co-rotation of the atmosphere with the Earth induces a non-zero inertial velocity of the surrounding air mass. This co-rotation velocity is expressed in the body frame as:

$$\mathbf{v}_{\mathrm{corot},Body} = \mathbf{R}(\mathbf{q})\, \mathbf{v}_{\mathrm{corot},ECI}. \tag{5.29}$$

This term represents the apparent atmospheric motion observed in the inertial frame due solely to Earth rotation.

Mean Wind Velocity

Mean wind profiles are provided to the atmosphere model as external inputs in the form of altitude-dependent tables. These tables are specified by the user prior to the simulation and define the components of the wind velocity as functions of altitude. The atmosphere model does not impose any specific origin for these data and remains agnostic to the method used to generate the wind profiles.

In practice, wind profiles may be obtained from pre-flight atmospheric measurements, such as radiosonde or weather balloon launches conducted prior to the mission. These

measurements provide vertical profiles of wind speed and direction and are commonly used in launcher operations to assess ascent dispersion and aerodynamic loading conditions. The resulting data can be post-processed and formatted as altitude-dependent wind tables compatible with the atmosphere model.

To ensure flexibility and robustness with respect to user-provided data, a configuration toggle is implemented in the atmosphere model to select the reference frame convention used for wind specification. While the model internally constructs and uses a local North–East–Down (NED) navigation frame as introduced in the previous section, wind tables may also be provided in the East–North–Up (ENU) convention, which is commonly used in meteorological datasets. In the ENU convention, the vertical axis is oriented upward, and the horizontal components are ordered as East–North, whereas the NED convention follows a North–East ordering with the vertical axis oriented downward.

Internally, the atmosphere model maps the selected input convention to a consistent North–East–Down representation prior to further processing. This avoids ambiguity in sign conventions and ensures compatibility with the subsequent reference frame transformations.

At each simulation step, the local wind velocity is obtained by linearly interpolating the tabulated wind components at the current vehicle altitude. Linear interpolation is used to ensure a continuous variation of the wind field along the ascent trajectory while maintaining numerical simplicity and robustness. Mean wind profiles are specified as altitude-dependent tables providing the three components of the wind velocity in the local navigation frame. While all three components are supported by the atmosphere model, the vertical component (D) is typically much smaller than the horizontal components in practical applications.

Altitude (m)	V_N ($\mathrm{m\,s^{-1}}$)	V_E ($\mathrm{m\,s^{-1}}$)	V_D ($\mathrm{m\,s^{-1}}$)
3500	-4.0	6.5	0.10
3750	-4.8	7.2	0.12
4000	-5.6	8.0	0.15
4250	-6.5	9.3	0.18
4500	-7.5	10.5	0.20

Table 5.4: Example of an altitude-dependent mean wind profile expressed in the North–East–Down (NED) convention.

The interpolated wind velocity vector expressed in the local NED frame, $\mathbf{v}_{wind,NED}$, is first mapped to the ECEF frame as

$$\mathbf{v}_{wind,ECEF} = \mathbf{R}_{NED\to ECEF}\,\mathbf{v}_{wind,NED}, \tag{5.30}$$

using the local NED-to-ECEF transformation introduced in Section 5.2.4. The resulting vector is then rotated into the ECI frame according to

$$\mathbf{v}_{wind,ECI} = \mathbf{R}_{\theta}^{\top}\,\mathbf{v}_{wind,ECEF}, \tag{5.31}$$

and finally expressed in the launcher body frame using the attitude quaternion,

$$\mathbf{v}_{wind,Body} = \mathbf{R}(\mathbf{q})\,\mathbf{v}_{wind,ECI}. \tag{5.32}$$

This formulation allows mission-specific wind conditions to be incorporated into the simulation while preserving a clear separation between environmental data definition and vehicle dynamics modelling.

Stochastic Turbulence

In addition to deterministic contributions arising from Earth rotation and mean wind profiles, the atmosphere model optionally accounts for stochastic atmospheric turbulence. Turbulence is introduced to represent high-frequency, random fluctuations of the local airflow that cannot be captured by altitude-dependent mean wind models alone, but which may significantly influence aerodynamic loads, attitude response, and trajectory dispersion during the ascent phase. In the present formulation, these fluctuations are represented by an additional turbulent velocity component, $\mathbf{v}_{\text{turb},Body}$, expressed in the launcher body frame and superimposed on the mean atmospheric flow.

The inclusion of turbulence is controlled through a configuration toggle, allowing the user to enable or disable stochastic effects depending on the objectives of the simulation. This design choice ensures flexibility: turbulence can be activated for dispersion analyses and robustness studies, or deactivated for nominal trajectory simulations where deterministic behaviour is preferred.

The detailed formulation of the Dryden turbulence model, including its statistical properties, parameter selection, and numerical implementation, is presented in Section 5.2.6.

Final Expression of the Relative Airflow Velocity

Combining the vehicle motion and atmospheric contributions, the relative airflow velocity experienced by the launcher and expressed in the body frame is given by

$$\mathbf{v}_{\text{rel},Body} = \mathbf{v}_{Body} - \left(\mathbf{v}_{\text{corot},Body} + \mathbf{v}_{\text{wind},Body} + \mathbf{v}_{\text{turb},Body}\right). \tag{5.33}$$

This relative airflow velocity constitutes the primary output of the atmosphere model and is provided to the aerodynamics model for the computation of aerodynamic angles, forces, and moments. By explicitly accounting for vehicle motion, Earth rotation, mean winds, and turbulence within a consistent reference frame formulation, the atmosphere model ensures that aerodynamic effects are evaluated in a physically coherent manner throughout the launcher ascent.

In the following, the relative airflow velocity $\mathbf{v}_{rel}$ is always expressed in the body frame unless stated otherwise.

5.2.6 Turbulence Model

The atmosphere model includes an optional stochastic turbulence contribution in order to represent high-frequency gusts superimposed on the mean atmospheric flow. These gusts correspond to small-scale random fluctuations of the wind velocity and may significantly affect the instantaneous relative airflow experienced by the launcher, particularly during the lower phases of ascent where atmospheric density and wind variability are higher.

The turbulence contribution is modelled as a body-frame gust velocity vector

$$\mathbf{v}_{\text{turb},Body} = \begin{bmatrix} u_g & v_g & w_g \end{bmatrix}^\top. \tag{5.34}$$

Activation logic and numerical safeguards

The turbulence model can be enabled or disabled through a toggle. When disabled, the gust components are set to zero and the atmosphere model reduces to a purely deterministic description based on mean wind and Earth rotation effects. When enabled,

turbulence generation is conditioned on the longitudinal body-axis airspeed. If the absolute forward velocity $|v_{Body,x}|$ falls below a threshold of 10 $\mathrm{m\,s^{-1}}$, turbulence is suppressed. This safeguard avoids numerical issues arising from turbulence time constants being inversely proportional to airspeed and reflects the limited physical relevance of gust models at very low air-relative velocities.

Dryden modelling principle

Atmospheric turbulence is generated using a filter-based stochastic approach inspired by the classical Dryden turbulence model [7]. In this formulation, gust velocities are obtained as the outputs of linear shaping filters driven by white Gaussian noise, i.e. zero-mean, unit-variance random processes with no temporal correlation, which are commonly used as excitation inputs for stochastic shaping filters. This approach allows the generation of coloured random processes with prescribed statistical properties while remaining well suited for time-domain simulation.

At each simulation step, three independent standard Gaussian noise samples are generated:

$$\eta_u,\ \eta_v,\ \eta_w \sim \mathcal{N}(0,1). \tag{5.35}$$

These samples are produced using a Mersenne Twister pseudo-random number generator and a normal distribution. A user-defined seed ensures repeatability of turbulence realisations, which is essential for sensitivity analyses and Monte-Carlo simulations.

Altitude-dependent turbulence parameters

The statistical properties of turbulence are governed by a characteristic length scale and a turbulence intensity, both defined as functions of altitude. The longitudinal turbulence length scale L_u is modelled as a piecewise function of altitude h:

$$L_u(h) = \begin{cases} 200 + 150\,(h/1000), & h < 2000, \\ 500 + 70\,((h/1000) - 2), & 2000 \le h < 10000, \\ 1060 + 100\,((h/1000) - 10), & 10000 \le h \le 20000, \\ 3000, & h > 20000. \end{cases} \tag{5.36}$$

The corresponding longitudinal turbulence intensity σ_u is defined as

$$\sigma_u(h) = \begin{cases} 4.0 - 1.5\,(h/1000), & h < 2000, \\ 1.0 - 0.1\,((h/1000) - 2), & 2000 \le h < 10000, \\ 0.2 - 0.01\,((h/1000) - 10), & 10000 \le h \le 20000, \\ 0.05, & 20000 < h \le 30000, \\ 0, & h > 30000. \end{cases} \tag{5.37}$$

The lateral and vertical turbulence parameters are defined as fixed ratios of the longitudinal quantities:

$$L_v = 0.7\,L_u, \quad \sigma_v = 0.8\,\sigma_u, \qquad L_w = 0.5\,L_u, \quad \sigma_w = 0.6\,\sigma_u. \tag{5.38}$$

This introduces anisotropy in the turbulence field, with dominant longitudinal fluctuations. The adopted turbulence length scales and intensities are based on commonly used engineering-order assumptions for atmospheric gust modelling.

Discrete-time filter formulation

Let $V = |v_{Body,x}|$ denote the forward body-axis airspeed and Δt the simulation time step. For each axis, a characteristic turbulence time constant is defined as

$$\tau = \frac{L}{V}, \tag{5.39}$$

leading to the dimensionless parameter

$$r = \frac{2\tau}{\Delta t}. \tag{5.40}$$

Let k denote the discrete-time index associated with the simulation time step Δt. The longitudinal gust component u_g is generated using a first-order recursive filter:

$$u_g[k] = -\frac{B_u}{A_u} u_g[k-1] + b_{0u}\,\eta_u[k] + b_{1u}\,\eta_u[k-1], \tag{5.41}$$

with

$$A_u = 1 + r, \qquad B_u = 1 - r, \tag{5.42}$$

and

$$b_{0u} = b_{1u} = \frac{\sigma_u \sqrt{\frac{2L_u}{V}}}{A_u}. \tag{5.43}$$

The lateral and vertical gust components v_g and w_g are generated using second-order recursive filters of the form

$$y[k] = -a_1 y[k-1] - a_2 y[k-2] + b_0 \eta[k] + b_1 \eta[k-1] + b_2 \eta[k-2], \tag{5.44}$$

where

$$a_1 = 2\frac{B}{A}, \qquad a_2 = \left(\frac{B}{A}\right)^2, \tag{5.45}$$

$$\beta = \frac{r}{\sqrt{3}}, \qquad C = 1 + \beta, \qquad D = 1 - \beta, \tag{5.46}$$

$$b_0 = \text{scale}\, C, \quad b_1 = \text{scale}\,(C + D), \quad b_2 = \text{scale}\, D, \tag{5.47}$$

with

$$\text{scale} = \frac{\sigma \sqrt{\frac{2L}{V}}}{A^2}. \tag{5.48}$$

For the lateral axis, $L = L_v$ and $\sigma = \sigma_v$; for the vertical axis, $L = L_w$ and $\sigma = \sigma_w$.

Intended use and limitations

The implemented turbulence model is intended as an engineering-level representation of atmospheric gusts. It provides physically plausible, repeatable high-frequency flow variations suitable for sensitivity analyses and dispersion studies. It is not intended to reproduce site-specific or meteorologically accurate turbulence statistics, which would require more complex boundary-layer and stability modelling. The adopted approach offers a favourable compromise between realism, numerical robustness, and computational efficiency. Standard references on statistical turbulence modelling and Dryden gust formulations may be consulted for further background [7].

5.2.7 Interfaces with Other Models

The atmosphere model is integrated into the launcher simulation framework as an environmental component providing local atmospheric conditions required for aerodynamic load computation. From an architectural perspective, it does not introduce additional dynamic states coupled to the global equations of motion. Instead, all atmospheric quantities are computed instantaneously from the current launcher state and configuration parameters.

Interface with the dynamics model

The atmosphere model retrieves the launcher kinematic state from the dynamics model at each simulation step. This includes the vehicle position and translational velocity expressed in the inertial reference frame, as well as the vehicle attitude represented by a unit quaternion. These quantities constitute the primary inputs required to evaluate altitude-dependent atmospheric properties, reference frame transformations, and relative airflow velocity.

The vehicle position is used to compute the geocentric altitude, which serves as the independent variable for the evaluation of atmospheric properties and wind profiles. The inertial velocity and attitude are used to perform the necessary transformations between the ECI, ECEF, and body reference frames. The dynamics model therefore acts as the sole provider of state information to the atmosphere model, ensuring a clear and unidirectional dependency.

Interface with the aerodynamics model

The primary role of the atmosphere model is to supply the aerodynamics model with the local atmospheric quantities required for aerodynamic force and moment computation. At each evaluation, the atmosphere model provides the air density ρ, the speed of sound a, and the relative airflow velocity vector $\mathbf{v}_{\text{rel}}$ expressed in the launcher body frame. These quantities are computed consistently using the same reference frame conventions and are guaranteed to correspond to the same simulation time and vehicle state.

The aerodynamics model relies on these outputs to compute dynamic pressure, Mach number, aerodynamic coefficients, and the resulting aerodynamic forces and moments. No additional assumptions regarding atmospheric properties are required within the aerodynamics model itself.

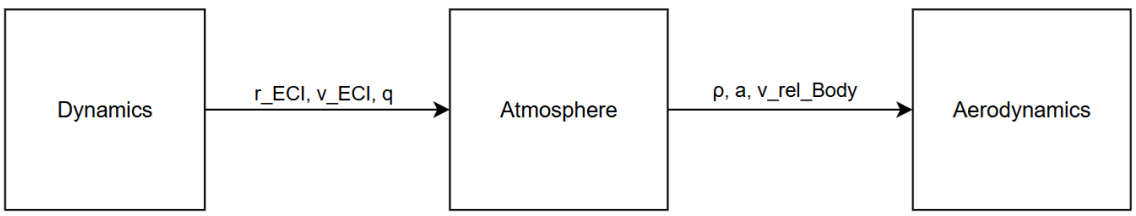


Figure 5.4: Block diagram illustrating the interfaces between the atmosphere model and the other subsystems of the launcher simulation framework.

All interactions shown in Fig. 5.4 occur within the main simulation loop and are synchronized with the numerical integration process. The atmosphere model does not

introduce internal dynamic states and is evaluated instantaneously from the current vehicle state. This design avoids algebraic loops and preserves the modular structure of the simulation framework, allowing alternative atmospheric representations to be integrated without modifying the dynamics or aerodynamics models.

5.2.8 Verification and Validation

The atmosphere model was verified through a combination of unit-level and integrated tests covering the full computation chain, from the evaluation of flow-derived quantities to the reconstruction of dimensional aerodynamic forces and moments. The objective of this verification effort was to ensure the correctness of the numerical implementation and the robustness of the model when coupled to the other simulation components.

Given the scope of this thesis and the use of tabulated aerodynamic coefficients as input data, the verification activities focused on code correctness and internal consistency. Validation against experimental or flight-test aerodynamic data was not within the scope of this work and is therefore limited to qualitative consistency checks.

Unit-level verification

Several core functions of the atmosphere model were verified through unit tests by comparing the outputs of the C++ implementation against independent reference computations performed in Python. All numerical comparisons were performed using a relative tolerance of 10^{-4}. These tests were designed to isolate individual functionalities and to verify that the implemented equations were correctly evaluated.

The altitude computation routine was verified by comparing the altitude derived from the ECI position vector with an analytical reference expression based on the Euclidean norm of the position. This test confirmed the correct evaluation of altitude in meters over a range of representative positions.

The implementation of the COESA-based atmospheric properties was verified by evaluating the air density and speed of sound at selected altitudes and comparing the results against reference tabulated values. Agreement within numerical tolerance was obtained, confirming the correct indexing and retrieval of atmospheric properties from the precomputed tables.

The stochastic turbulence generation was also verified at unit level. For a prescribed altitude and forward airspeed, the turbulence velocity components generated by the C++ model were compared against an independent Python implementation reproducing the same Dryden-based filter equations and using identical random inputs, obtained by enforcing the same pseudo-random seed. This test confirmed the correct implementation of the discrete-time filters and the altitude-dependent turbulence parameters.

Integrated verification within the simulation framework

In addition to unit-level tests, several integrated tests were performed to verify the correct interaction of the atmosphere model with the dynamics and aerodynamics components once connected within the full simulation framework.

The loading and interpolation of mean wind profiles were verified by comparing the wind velocity computed by the atmosphere model against a Python reference implementation. This test included the full processing chain: altitude computation, ECEF frame transformation, construction of the local navigation frame, interpolation of user-provided

wind tables, and transformation of the resulting wind velocity into the appropriate reference frame.

The computation of the relative airflow velocity was verified by constructing reference scenarios combining inertial velocity, Earth co-rotation effects, mean wind velocity, and vehicle attitude. For these scenarios, the relative airflow velocity computed by the C++ model was compared against an independent Python reconstruction using the same input data and reference frame transformations. Consistent results were obtained within numerical tolerances.

An additional integrated test was performed with the turbulence model enabled. In this case, the turbulence contribution generated by the Dryden model was included in the relative airflow computation. By enforcing a fixed random seed, the stochastic turbulence realisation was made deterministic, allowing direct comparison between the C++ and Python implementations. This test verified the correct integration of turbulence effects into the relative airflow velocity computation.

Validation considerations

The validation of the atmosphere model is inherently limited by the use of a reference standard atmosphere and simplified wind and turbulence representations. Nevertheless, the orders of magnitude of the computed atmospheric quantities, wind velocities, and turbulence-induced fluctuations were verified to be physically plausible and consistent with typical values encountered in launcher ascent scenarios.

Within the objectives of this thesis, the performed verification activities provide a high level of confidence in the correctness, numerical robustness, and internal consistency of the atmosphere model. The adopted approach ensures that any observed behaviour in the integrated launcher simulations can be attributed to modelling assumptions rather than to implementation errors.

5.3 Aerodynamics Model

5.3.1 Motivation and Role

During the atmospheric phase of a launch vehicle trajectory, aerodynamic effects play a central role in shaping the vehicle dynamics, structural loads, and control requirements. As the launcher ascends through the dense layers of the atmosphere, the interaction between the surrounding airflow and the vehicle geometry generates forces and moments that may dominate the overall dynamics, particularly during the early ascent phase and around the region of maximum dynamic pressure.

The primary objective of the aerodynamics model developed in this work is to compute the aerodynamic forces and moments acting on the launcher as a function of its instantaneous flight conditions. These quantities are subsequently provided to the dynamics model, where they contribute directly to the translational and rotational equations of motion. Accurate and consistent aerodynamic modelling is therefore essential to ensure a realistic simulation of vehicle behaviour during atmospheric flight.

The aerodynamics model follows a classical engineering formulation in which aerodynamic forces and moments are expressed in a non-dimensional form through aerodynamic coefficients. These coefficients are functions of the Mach number and of characteristic aerodynamic angles and are supplied to the model in the form of tabulated data. At

runtime, the aerodynamics model interpolates these tables based on the instantaneous flight conditions and computes the resulting aerodynamic loads.

The aerodynamics model is designed to integrate seamlessly with the atmosphere and dynamics models developed within the same simulation framework. The relative airflow velocity, computed by the atmosphere model and expressed in the launcher body frame, constitutes the primary kinematic input to the aerodynamics model. From this relative velocity, the aerodynamic angles required for load computation are derived internally. This tight coupling ensures that atmospheric effects, vehicle motion, and aerodynamic responses are treated coherently across the simulation chain.

Within this context, the aerodynamics model fulfills a well-defined role: it transforms the instantaneous atmospheric and kinematic state of the vehicle into aerodynamic forces and moments expressed in the launcher body frame. These outputs constitute a critical interface between the atmospheric environment and the vehicle dynamics, enabling the coupled simulation of atmospheric flight phases in a physically meaningful and numerically robust manner.

5.3.2 Modelling Scope and Assumptions

The aerodynamics model developed in this work is intended as an engineering-level representation of aerodynamic effects suitable for system-level launcher simulations. Its scope is deliberately limited in order to strike a balance between physical fidelity, numerical robustness, and computational efficiency.

Aerodynamic loads are assumed to be quasi-stationary and are fully characterised by tabulated aerodynamic coefficients as functions of Mach number and aerodynamic angles. Unsteady aerodynamic effects, flow separation dynamics, and transient phenomena are not explicitly modelled. Similarly, the influence of structural flexibility, aeroelastic effects, surface ablation, and vehicle deformation is neglected.

Each aerodynamic coefficient is assumed to depend on Mach number and on a single aerodynamic angle. This angle may be chosen among the angle of attack, the sideslip angle, or the total aerodynamic angle, depending on the dominant physical effects represented by the coefficient. While higher-fidelity aerodynamic databases may introduce dependencies on multiple angular parameters simultaneously, such formulations would require higher-dimensional coefficient tables and more complex interpolation schemes and are therefore considered beyond the scope of the present modelling approach.

In addition, the present formulation does not explicitly include aerodynamic damping terms or control surface effects. Contributions such as rate-dependent aerodynamic derivatives or control-induced aerodynamic loads are therefore not modelled explicitly within the current framework. These effects may be incorporated in future developments through extended aerodynamic coefficient tables or additional modelling layers, without modifying the core structure of the aerodynamics model.

The aerodynamic coefficients provided to the model are treated as externally generated data. The aerodynamics model does not embed assumptions regarding the vehicle geometry, the method used to derive the coefficients, or the fidelity of the underlying aerodynamic analyses. Coefficient tables may originate from wind tunnel testing, semi-empirical methods, or high-fidelity numerical simulations performed offline.

This modelling approach is not intended to replace detailed aerodynamic analyses or computational fluid dynamics simulations. Instead, it provides a computationally efficient and numerically stable means of capturing the dominant dependencies of aerody-

namic forces and moments on flight conditions within the context of coupled trajectory, dynamics, and control simulations.

5.3.3 Physical Modelling and Aerodynamic Quantities

The aerodynamics model relies on a classical quasi-steady formulation, in which aerodynamic forces and moments are expressed as functions of the instantaneous relative airflow conditions experienced by the launcher. This section details the physical quantities computed by the model, the underlying assumptions, and the sequence of operations leading from the relative airflow velocity to the final aerodynamic loads.

The model operates entirely in the launcher body frame. All aerodynamic forces and moments are therefore computed and returned in this frame, ensuring direct compatibility with the rotational and translational equations of motion used by the dynamics model.

Relative Airflow Velocity

The fundamental input to the aerodynamics model is the relative airflow velocity $\mathbf{v}_{\text{rel}}$, expressed in the launcher body frame and provided by the atmosphere model. This vector defines the instantaneous direction and magnitude of the incoming flow as seen by the vehicle and constitutes the kinematic basis for aerodynamic load computation.

Let

$$\mathbf{v}_{\text{rel}} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{5.49}$$

denote the components of the relative airflow velocity along the body-frame axes. These components are used to derive the aerodynamic angles that govern the selection of aerodynamic coefficients.

The magnitude of the relative airflow velocity,

$$V = \|\mathbf{v}_{\text{rel}}\|, \tag{5.50}$$

is used in the aerodynamic load formulation through the dynamic pressure and constitutes a scaling quantity for the resulting forces and moments.

Aerodynamic Angles

The orientation of the relative airflow with respect to the launcher body frame is characterized through a set of aerodynamic angles. These angles describe how the airflow approaches the vehicle and directly influence the aerodynamic coefficients.

The angle of attack α is defined as the angle between the projection of the relative airflow onto the longitudinal–vertical plane and the body longitudinal axis. It is computed as

$$\alpha = \arctan 2(w, u). \tag{5.51}$$

This definition ensures a continuous and numerically robust evaluation of the angle of attack over the full range of flight conditions.

The sideslip angle β represents the lateral deviation of the relative airflow with respect to the vehicle longitudinal axis and is represented as

$$\beta = \arcsin\left(\frac{v}{V}\right). \tag{5.52}$$

The sideslip angle plays a key role in the generation of lateral forces and yawing moments, particularly during asymmetric flight conditions or in the presence of crosswinds.

In addition to α and β, a total aerodynamic angle α_{tot} is introduced to characterize the overall deviation of the relative airflow from the body longitudinal axis. It is defined as

$$\alpha_{\text{tot}} = \arccos\left(\frac{u}{\|\mathbf{v}_{\text{rel}}\|}\right). \tag{5.53}$$

This definition is equivalent to $\alpha_{\text{tot}} = \arccos(\cos\alpha\cos\beta)$. This quantity provides a convenient scalar measure of the combined effects of angle of attack and sideslip and is sometimes used as an alternative independent variable in aerodynamic databases.

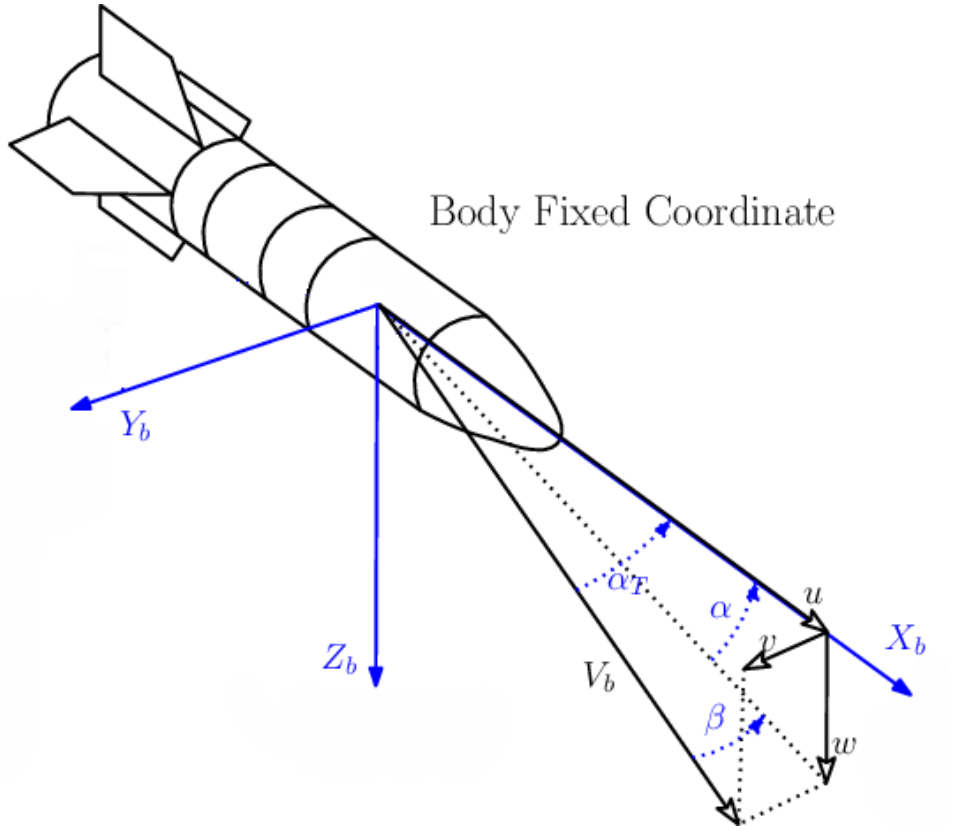


Figure 5.5: Launcher body-fixed coordinate system and representation of the relative airflow velocity components (u, v, w), the angle of attack α, the sideslip angle β, and the total aerodynamic angle, denoted α_T in the figure (corresponding to α_{tot} in the text). Adapted from [23].

Mach Number

The Mach number M, computed in the atmosphere model as $M = V/a$, is supplied to the aerodynamics model and constitutes a primary independent variable for the aerodynamic coefficient tables used to evaluate forces and moments.

Dynamic Pressure

The air density ρ provided by the atmosphere model is used by the aerodynamics model to compute the dynamic pressure according to

$$q_{\text{dyn}} = \frac{1}{2}\rho\|\mathbf{v}_{\text{rel}}\|^2, \tag{5.54}$$

where $\mathbf{v}_{\text{rel}}$ denotes the relative airflow velocity expressed in the launcher body frame. The dynamic pressure q_{dyn}, expressed in pascals (Pa), constitutes a primary driver of aerodynamic forces and moments and strongly influences the structural and control loads experienced by the vehicle during ascent.

Reference Quantities

Aerodynamic forces and moments are expressed using non-dimensional coefficients scaled by reference geometric quantities. Two reference parameters are introduced in the model.

The reference area A_{ref} is provided by the vehicle model and corresponds to the frontal cross-sectional area of the launcher. It is computed from the launcher radius R as

$$A_{\text{ref}} = \pi R^2. \tag{5.55}$$

The reference length l_{ref} is also provided by the vehicle model and corresponds to the overall length of the launcher.

These choices are consistent with the reference quantities commonly used to generate the aerodynamic coefficient tables.

Aerodynamic Forces

The aerodynamic force vector $\mathbf{F}_{\text{aero}}$ is expressed in the body frame as

$$\mathbf{F}_{\text{aero}} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}. \tag{5.56}$$

Using the aerodynamic force coefficients C_x, C_y, and C_z, the forces are computed according to

$$\begin{aligned} F_x &= -q_{\text{dyn}} A_{\text{ref}} C_x, \\ F_y &= -q_{\text{dyn}} A_{\text{ref}} C_y, \\ F_z &= -q_{\text{dyn}} A_{\text{ref}} C_z. \end{aligned} \tag{5.57}$$

The negative sign reflects the convention that aerodynamic forces oppose the direction of the relative airflow.

Aerodynamic Moments

Similarly, the aerodynamic moment vector $\mathbf{M}_{\text{aero}}$ expressed in the body frame is given by

$$\mathbf{M}_{\text{aero}} = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}. \tag{5.58}$$

Using the non-dimensional moment coefficients C_l, C_m, and C_n, the aerodynamic moments are computed as

$$\begin{aligned} M_x &= q_{\text{dyn}} A_{\text{ref}} l_{\text{ref}} C_l, \\ M_y &= q_{\text{dyn}} A_{\text{ref}} l_{\text{ref}} C_m, \\ M_z &= q_{\text{dyn}} A_{\text{ref}} l_{\text{ref}} C_n. \end{aligned} \tag{5.59}$$

These moments represent rolling, pitching, and yawing effects generated by the interaction between the airflow and the vehicle.

5.3.4 Aerodynamic Coefficient Tables

The aerodynamic loads computed by the model are derived from non-dimensional aerodynamic coefficients supplied by the user in the form of tabulated data. This design choice reflects the fact that aerodynamic coefficients are typically obtained through external processes, such as wind tunnel testing, semi-empirical methods, or high-fidelity numerical simulations, and are therefore not computed directly within the simulation framework.

The aerodynamics model developed in this work is deliberately agnostic to the origin of the aerodynamic data. Instead, it provides a generic and flexible mechanism to ingest user-defined coefficient tables and to evaluate them at runtime as functions of the instantaneous flight conditions. This approach enables the model to be reused across different launcher configurations and levels of aerodynamic fidelity without modification of its internal structure.

General Structure of the Coefficient Tables

Aerodynamic coefficients are provided to the model through an external data file, loaded during the initialization phase of the simulation. Each row of the table corresponds to a set of aerodynamic coefficients associated with a specific Mach number and a specific aerodynamic angle.

The Mach number constitutes the primary independent variable of all aerodynamic coefficients. Each coefficient is therefore defined as a function of Mach and of a single angular parameter. The angular dependency may be expressed using one of the following quantities:

- the angle of attack α,
- the sideslip angle β,
- the total aerodynamic angle α_{tot}.

The use of a single angular dependency per coefficient reflects common practice in engineering-level launcher aerodynamics, where dominant physical effects are often captured through one primary aerodynamic angle. While dependencies on multiple angular parameters may be introduced in higher-fidelity aerodynamic databases, such formulations are beyond the scope of the present model and would require higher-dimensional coefficient tables and interpolation schemes.

The model intentionally leaves a high degree of freedom to the user regarding the choice of aerodynamic parameterization. Each aerodynamic coefficient, namely C_x, C_y, C_z, C_l, C_m, and C_n, may be defined as a function of Mach and of either α, β, or α_{tot}, depending on the structure of the provided table.

At least one angular column must be present for each coefficient. Beyond this constraint, no assumptions are imposed on the shape, resolution, or origin of the coefficient tables. This design allows the user to employ simplified datasets during early design phases, as well as more refined aerodynamic databases when available. In practice, this allows sparse coefficient tables to be used, in which undefined entries (NaN values) indicate coefficients or angular dependencies that are not provided, as illustrated in Table 5.5

Internally, the model automatically identifies the angular dependency associated with each coefficient by detecting the first non-empty column corresponding to an angular variable. This mechanism eliminates the need for additional configuration parameters and reduces the risk of user error when defining aerodynamic datasets.

M	α (deg)	β (deg)	α_{tot} (deg)	C_x	C_y	C_z	C_l	C_m	C_n
1.05	-0.5	NaN	NaN	-0.0263	NaN	-0.0956	NaN	NaN	NaN
1.10	-0.5	NaN	NaN	-0.0262	NaN	-0.0964	NaN	NaN	NaN
1.25	-0.5	NaN	NaN	-0.0136	NaN	-0.0620	NaN	NaN	NaN
1.50	-0.5	NaN	NaN	-0.0153	NaN	-0.0697	NaN	NaN	NaN
1.80	-0.5	NaN	NaN	-0.0164	NaN	-0.0750	NaN	NaN	NaN
1.05	NaN	2.0	NaN	NaN	0.0120	NaN	NaN	NaN	-0.0040
1.50	NaN	2.0	NaN	NaN	0.0105	NaN	NaN	NaN	-0.0035
1.80	NaN	2.0	NaN	NaN	0.0090	NaN	NaN	NaN	-0.0030
1.05	NaN	NaN	1.0	NaN	NaN	NaN	0.0020	-0.0100	NaN
1.50	NaN	NaN	1.0	NaN	NaN	NaN	0.0016	-0.0085	NaN
1.80	NaN	NaN	1.0	NaN	NaN	NaN	0.0013	-0.0075	NaN

Table 5.5: Excerpt of a user-defined aerodynamic coefficient table. Each row is associated with a Mach number and one aerodynamic angle (α, β, or α_{tot}).

Runtime Evaluation and Interpolation Strategy

At each simulation time step, the instantaneous Mach number and aerodynamic angles computed by the model are used to evaluate the aerodynamic coefficients through interpolation within the user-provided tables.

For a given flight condition, the model first identifies the two tabulated Mach numbers M_1 and M_2 that bound the current Mach number M. For each of these Mach levels, the aerodynamic coefficient is evaluated as a function of the associated aerodynamic angle. To this end, the angular data points corresponding to the selected Mach number are first sorted to ensure a monotonic ordering, and a one-dimensional linear interpolation is then performed with respect to the aerodynamic angle. This yields two intermediate coefficient values, $C(M_1, \theta)$ and $C(M_2, \theta)$.

In a second step, a linear interpolation is performed between these two intermediate values with respect to Mach number, resulting in the aerodynamic coefficient evaluated at the current flight condition, $C(M, \theta)$. This two-stage interpolation procedure ensures continuity of the aerodynamic coefficients across both angular and Mach dimensions.

If either the Mach number or the aerodynamic angle lies outside the bounds of the tabulated data, a clamping strategy is applied and the nearest available boundary value is used. This choice prevents extrapolation beyond the validity domain of the aerodynamic data and ensures stable behavior of the model in extreme flight conditions.

The interpolation strategy is designed to gracefully handle incomplete datasets. Table entries that do not contain valid coefficient values are ignored, allowing sparse or partially defined aerodynamic tables to be used without compromising the integrity or numerical robustness of the model.

The use of tabulated aerodynamic coefficients combined with linear interpolation ensures a good compromise between physical fidelity and numerical efficiency. All coefficient evaluations are performed using simple arithmetic operations, enabling their computation at every simulation time step with negligible computational overhead.

Moreover, the absence of analytical expressions for the coefficients avoids the introduction of artificial nonlinearities or discontinuities that may arise from overly simplified polynomial fits. The model behavior is therefore entirely driven by the user-provided aerodynamic data, ensuring consistency between offline aerodynamic analyses and online system-level simulations.

5.3.5 Interfaces with Other Models

The aerodynamics model is integrated into the launcher simulation framework as a load provider interfacing the atmospheric environment with the vehicle dynamics. Its role is to transform local flow conditions and vehicle reference geometry into aerodynamic forces and moments applied to the equations of motion. The following paragraphs describe the interfaces established with the dynamics and atmosphere models, as well as the associated data exchange within the simulation loop.

Interface with the Atmosphere Model

The aerodynamics model interfaces with the atmosphere model to retrieve the local flow conditions required for aerodynamic load computation. At each simulation step, the atmosphere model provides the air density ρ, the local speed of sound a, and the relative airflow velocity vector $\mathbf{v}_{\text{rel}}$ expressed in the launcher body frame.

All atmospheric effects, including reference frame transformations, Earth rotation, mean wind profiles, and optional turbulence contributions, are handled upstream by the atmosphere model. By relying exclusively on the quantities supplied through this interface, the aerodynamics model remains independent of the specific atmospheric representation and can be coupled with alternative atmosphere implementations without modification.

Interface with the Dynamics Model

The aerodynamics model interfaces with the dynamics model by providing the aerodynamic loads used in the equations of motion. The resulting aerodynamic forces and moments acting on the launcher are returned to the dynamics model and applied during state propagation.

The aerodynamic loads are expressed in the launcher body frame and are subsequently transformed into the inertial (ECI) frame within the dynamics model for use in the equations of motion.

Interface with the Vehicle Model

The vehicle model provides fixed geometric reference quantities used for aerodynamic scaling, including the reference length and reference area, which are defined once at initialization and remain constant throughout the simulation.

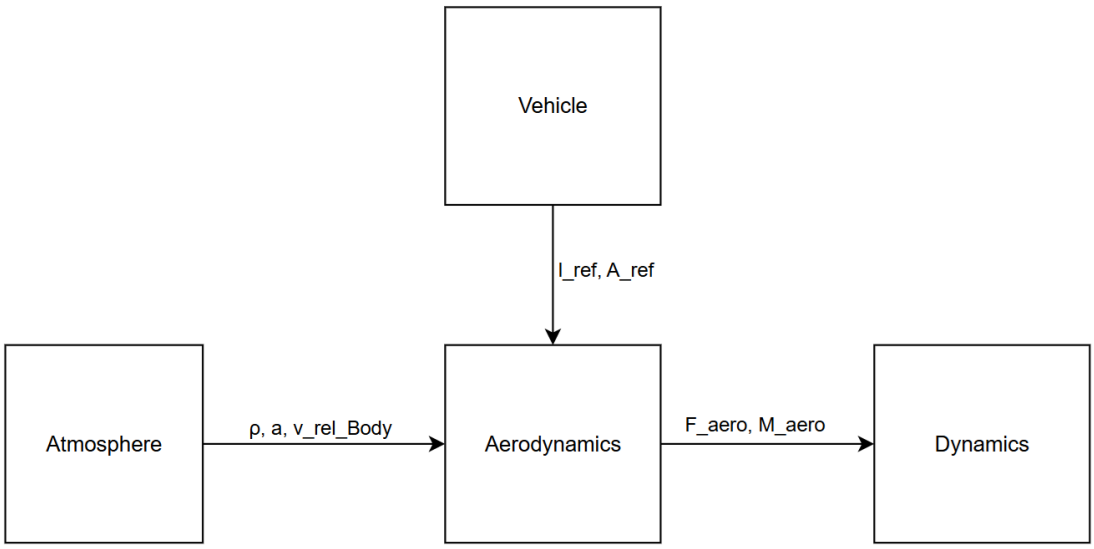


Figure 5.6: Block diagram illustrating the interfaces between the aerodynamics model and the other subsystems of the launcher simulation framework.

5.3.6 Verification and Validation

The aerodynamic model was verified through a combination of unit-level and integrated tests. Verification focused on the correct evaluation of aerodynamic quantities, the handling of tabulated coefficients, and the consistency of the forces and moments produced by the model when coupled to the rest of the simulation framework.

Unit-level verification

Unit tests were used to verify the correct evaluation of the main aerodynamic quantities derived from the relative airflow. These include the computation of aerodynamic angles, Mach number, and dynamic pressure, which are central to both coefficient selection and aerodynamic scaling. Each quantity was tested independently over representative ranges of input values to ensure numerical correctness and stability.

The handling of aerodynamic coefficient tables was also verified at unit level. In particular, the retrieval of coefficients and their interpolation as functions of Mach number and aerodynamic angles were compared against independent Python reference implementations. These tests ensured that the model correctly interprets user-provided tables and produces consistent coefficient values at both grid points and interpolated conditions.

Integrated verification within the simulation framework

Integrated tests were performed to verify the correct behavior of the aerodynamics model when connected to the atmosphere, vehicle, and dynamics components within the full simulation framework. For selected simulation states, the aerodynamic forces and moments computed by the C++ implementation were compared against independent Python reconstructions following the same physical formulation. All numerical comparisons were performed using a relative tolerance of 10^{-4}.

Additional integrated tests were conducted under time-varying flow conditions to assess the robustness of the model in realistic scenarios. In particular, simulations including atmospheric turbulence were used to verify that the aerodynamics model remains consistent when the relative airflow varies due to stochastic perturbations, and that the resulting forces and moments evolve smoothly without numerical instabilities.

Validation considerations

The aerodynamics model relies on user-provided aerodynamic coefficient tables and does not aim at high-fidelity aerodynamic prediction. As a result, validation is inherently limited by the assumptions and fidelity of the input data rather than by the numerical implementation itself.

Although telemetry data from some launcher flights were available, these data could not be used for direct aerodynamic validation. The recorded signals correspond to the combined effect of aerodynamic loads, thrust forces, control system actions, and other coupled dynamics, making the isolation of purely aerodynamic contributions impractical.

Nevertheless, the computed aerodynamic forces and moments were checked for physical plausibility in typical launcher ascent conditions. The obtained magnitudes and trends were found to be consistent with expected aerodynamic behavior, providing confidence that the model produces realistic outputs when used within the coupled simulation framework.

5.4 Gravity Model

5.4.1 Motivation and Role

Gravity constitutes a fundamental contribution to the motion of a launch vehicle from liftoff through ascent and into orbital flight. Unlike other external forces, gravitational acceleration acts continuously and remains significant over the entire trajectory, directly governing the translational dynamics and the overall energy balance of the vehicle. A consistent representation of gravity is therefore a prerequisite for any realistic launcher dynamics simulation.

In the initial configuration of the simulation framework, gravity was represented within the dynamics model as a constant acceleration, assumed uniform in magnitude and direction. While this approximation may be acceptable for simplified analyses over limited altitude ranges, it becomes increasingly inadequate for realistic ascent and early orbital simulations. Improving this representation was therefore necessary to ensure consistent trajectory propagation across the full flight envelope considered in this thesis.

The gravity model introduced in this work is designed to fulfill this role within the simulation framework by providing the gravitational acceleration acting on the vehicle as a function of its instantaneous position. This acceleration is combined with other force contributions, such as thrust and aerodynamic loads, within the equations of motion governing the launcher dynamics. In the present framework, gravity is treated as an equivalent acceleration applied at a single point, and does not contribute directly to the rotational equations of motion.

The dominant motivation for introducing a gravity model is the strong dependence of the gravitational acceleration on altitude, which leads to significant variations over the flight envelope of a launcher.

The objective of this gravity model is to provide an engineering-level representation of the Earth's gravitational field that is appropriate for launcher ascent and early orbital

flight analyses. The emphasis is placed on capturing the dominant effects governing vehicle motion while maintaining a level of complexity compatible with iterative simulation and design studies.

5.4.2 Modelling Scope and Assumptions

The gravity model implemented in this work is designed to capture the dominant components of the Earth's gravitational field that are relevant for launch vehicle trajectory simulation, while avoiding unnecessary complexity. In this context, the modelling scope is deliberately restricted to effects that have a first-order influence on the vehicle motion over the time scales and altitudes of interest.

The gravitational acceleration is modelled as the superposition of two contributions. The first contribution corresponds to the classical Newtonian central attraction of a spherically symmetric Earth. This term depends only on the distance between the vehicle and the Earth's center of mass and represents the dominant gravitational effect throughout the ascent and early orbital phases.

The second contribution accounts for the oblateness of the Earth through the inclusion of the J_2 zonal harmonic. This perturbation models the deviation of the Earth's mass distribution from perfect spherical symmetry due to its equatorial bulge. Although smaller in magnitude than the central term, the J_2 contribution introduces measurable deviations in the gravitational acceleration direction and magnitude, particularly in the vicinity of the Earth and for trajectories with significant vertical or out-of-plane components. Including this effect improves the realism of the simulated trajectory without a significant increase in computational cost.

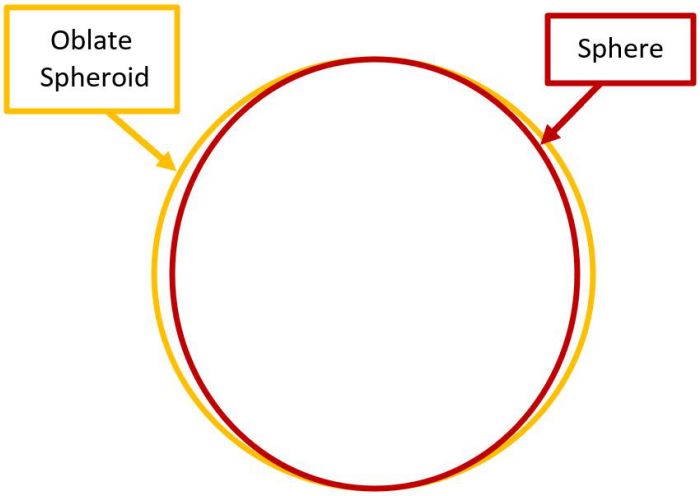


Figure 5.7: Illustration of the Earth's oblateness, highlighting the equatorial bulge resulting from planetary rotation. The deviation from spherical symmetry is primarily concentrated at the equator, while the polar flattening remains comparatively small [24].

Beyond the large-scale effects captured by the central term and the J_2 contribution, the Earth's gravitational field exhibits additional spatial variations associated with local mass heterogeneities. These higher-order gravity anomalies are several orders of magnitude smaller than the dominant contributions considered in this work and are not explicitly modeled. Figure 5.8 illustrates the spatial complexity of the terrestrial gravity field beyond the adopted modeling scope.

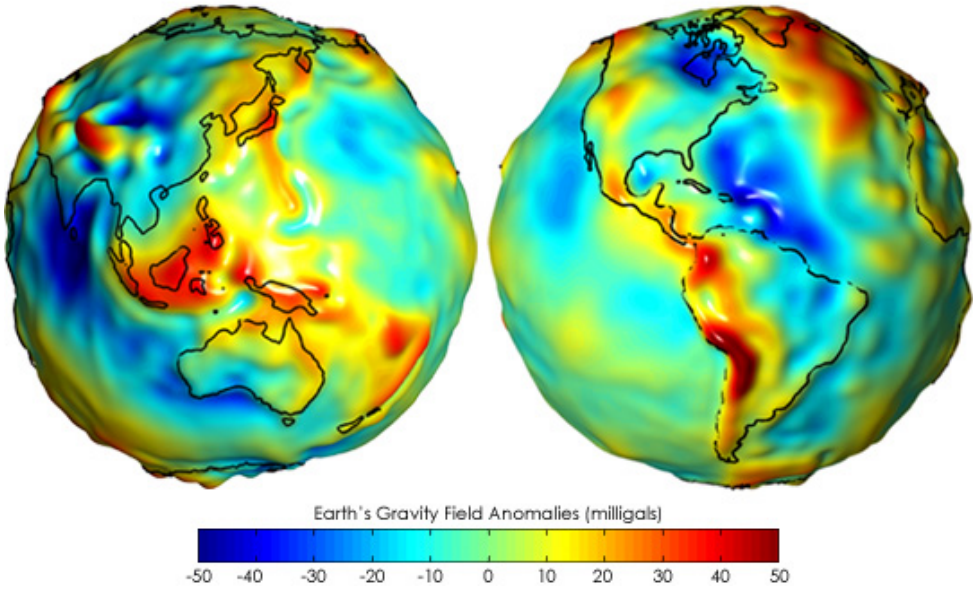


Figure 5.8: Global representation of Earth's gravity field anomalies relative to a geopotential reference model, expressed in milligals (1 mGal = 10^{-5} m/s^2) [25].

Several simplifying assumptions are made in the gravity model. The Earth is assumed to be rigid and time-invariant, with constant gravitational parameters. Temporal variations of the gravity field, such as tidal effects or mass redistribution, are neglected. Higher-order zonal, tesseral, and sectorial harmonics beyond J_2 are also omitted, as their influence is considered negligible within the scope of this thesis.

Furthermore, the gravitational acceleration is computed in an Earth-centered reference frame using a point-mass approximation for the vehicle. Relativistic effects and non-inertial corrections are not considered. These assumptions are consistent with the intended use of the model and with common practices in engineering-level launch vehicle simulations.

Overall, the adopted modelling scope strikes a balance between fidelity and simplicity. It provides a physically meaningful representation of the Earth's gravity field that is well suited for integration within a multi-physics simulation framework, while preserving numerical robustness and computational efficiency.

5.4.3 Mathematical Formulation of the Gravity Model

The gravity model implemented in this work is derived from a reduced-order representation of the Earth's gravitational field. Starting from the general spherical harmonic expansion, the formulation is truncated to retain only the dominant contributions relevant for launcher ascent and early orbital flight.

In its most general form, the Earth's gravitational field can be described by expanding the gravitational potential in spherical harmonics as

$$U(r,\theta,\lambda) = \frac{\mu_E}{r}\left[1 - \sum_{n=2}^{\infty}\left(\frac{R_E}{r}\right)^n \sum_{m=0}^{n} \bar{P}_{nm}(\sin\theta)\left(C_{nm}\cos m\lambda + S_{nm}\sin m\lambda\right)\right], \quad (5.60)$$

where r denotes the distance to the Earth's center, θ the geocentric latitude, λ the longitude, and $\bar{P}_{nm}$ the normalized associated Legendre functions, with C_{nm} and S_{nm} the corresponding gravitational coefficients [18].

For the flight phases considered in this work, the dominant non-spherical contribution arises from the Earth's oblateness and is captured by the second zonal harmonic J_2.

Truncating the spherical harmonic expansion accordingly leads to the following expression for the gravitational potential:

$$U = -\frac{\mu_E}{r}\left[1 - J_2\left(\frac{R_E}{r}\right)^2 \frac{1}{2}(3\sin^2\theta - 1)\right], \tag{5.61}$$

where $\mu_E = 3.986004418 \times 10^{14}\ \mathrm{m^3\,s^{-2}}$ denotes the Earth's gravitational parameter, R_E is the mean Earth radius, and θ is the geocentric latitude.

The gravitational acceleration acting on the vehicle is obtained as the negative gradient of the gravitational potential,

$$\mathbf{g} = -\nabla U, \tag{5.62}$$

and is expressed in Cartesian form to allow direct integration within the translational equations of motion.

The truncated gravitational potential can be decomposed into a central Newtonian term and a perturbative contribution associated with the J_2 harmonic. The corresponding gravitational acceleration components are obtained by taking the gradient of each term separately.

Central Newtonian Attraction

The leading contribution to the gravitational acceleration corresponds to the classical Newtonian attraction of a spherically symmetric Earth. Let $\mathbf{r} = [x\ y\ z]^\top$ denote the position vector of the vehicle with respect to the Earth's center, expressed in an Earth-centered reference frame whose z-axis is aligned with the Earth's rotation axis, and let $r = \|\mathbf{r}\|$ be its Euclidean norm. The central gravitational acceleration is given by

$$\mathbf{g}_{\mathrm{N}} = -\frac{\mu_E}{r^3}\,\mathbf{r}. \tag{5.63}$$

This term yields a purely radial acceleration directed toward the Earth's center and represents the dominant contribution to the vehicle's translational dynamics throughout the ascent and early orbital phases.

Oblateness Correction via the J_2 Harmonic

To account for the deviation of the Earth from perfect spherical symmetry, an additional perturbative acceleration associated with the second zonal harmonic J_2 is included. This term models the effect of the Earth's equatorial bulge and introduces latitude-dependent corrections to the gravitational field.

The perturbative acceleration associated with the J_2 term is obtained by differentiating the truncated gravitational potential and expressing the result in Cartesian coordinates. This leads to the following expression:

$$\mathbf{g}_{J_2} = \frac{3}{2} J_2 \mu_E \frac{R_E^2}{r^5}\begin{bmatrix} x\left(5\frac{z^2}{r^2} - 1\right) \\ y\left(5\frac{z^2}{r^2} - 1\right) \\ z\left(5\frac{z^2}{r^2} - 3\right) \end{bmatrix}. \tag{5.64}$$

This formulation captures the first-order effects of the Earth's oblateness, leading to a reduction of the gravitational acceleration near the equator and a corresponding enhancement along the polar direction.

Total Gravitational Acceleration

The total gravitational acceleration applied to the vehicle is obtained by superposition of the central and oblateness contributions,

$$\mathbf{g} = \mathbf{g}_{\mathrm{N}} + \mathbf{g}_{J_2}. \tag{5.65}$$

This acceleration vector is evaluated at each simulation step using the instantaneous vehicle position provided by the dynamics model and is applied as a translational acceleration at a single point. By retaining only the central term and the leading zonal harmonic, the adopted formulation provides a physically meaningful and computationally efficient gravity model that is well suited for launcher ascent and early orbital flight simulations.

5.4.4 Numerical Implementation

The gravity model is implemented as a standalone component within the simulation framework and is designed to interact directly with the dynamics model through a well-defined interface. Its primary responsibility is to compute the gravitational acceleration vector at the current vehicle position and to make this information available to the rest of the simulation.

At each evaluation step, the gravity model retrieves the launcher position expressed in an Earth-centered reference frame from the dynamics model. This position vector $\mathbf{r} = [x\ y\ z]^\top$ is used as the sole input for the gravity computation. No other informations are required.

The Euclidean norm of the position vector is first computed to obtain the distance from the Earth's center,

$$r = \sqrt{x^2 + y^2 + z^2}. \tag{5.66}$$

This scalar quantity is then used to evaluate the central Newtonian contribution as well as the J_2 perturbation term. To maintain numerical efficiency, intermediate quantities such as r^3 and r^5 are computed explicitly and reused in subsequent calculations.

The Newtonian acceleration is evaluated as a scaled version of the position vector, while the J_2 correction is applied component-wise using precomputed factors involving the Earth's gravitational parameter, equatorial radius, and zonal harmonic coefficient. The resulting acceleration components are accumulated to form the total gravitational acceleration vector

$$\mathbf{g} = [g_x\ g_y\ g_z]^\top.$$

The computed acceleration is stored internally within the gravity model and can be accessed by other simulation components through a dedicated getter function. This design ensures that the gravity computation remains encapsulated and that the internal implementation can be modified or extended without impacting the external interfaces.

From a numerical standpoint, the gravity model is lightweight and robust. It relies only on basic floating-point operations and avoids conditional branching or iterative procedures. This makes it well suited for repeated evaluation at high simulation rates, as required during launch vehicle ascent and early trajectory propagation.

5.4.4.1 Interfaces with Other Models

The gravity model is interacts exclusively with the dynamics model, following a simple and bidirectional data exchange pattern.

At each simulation step, the gravity model receives the instantaneous vehicle position from the dynamics model. The position vector is expressed the ECI and constitutes the only input required to evaluate the gravitational acceleration. This limited interface ensures that the gravity computation remains independent of the internal structure of the dynamics model and avoids unnecessary coupling between subsystems.

The gravity model returns the gravitational acceleration vector expressed in the same frame. This acceleration is subsequently combined with other force contributions, such as aerodynamic forces and thrust, within the dynamics equations of motion.

The gravity model does not modify the vehicle state directly and does not maintain any internal dynamic state. It behaves as a purely algebraic component whose output depends solely on the current vehicle position and on constant Earth parameters. This design preserves a clear separation between force modelling and state propagation and facilitates verification and future extensions of the gravity formulation.

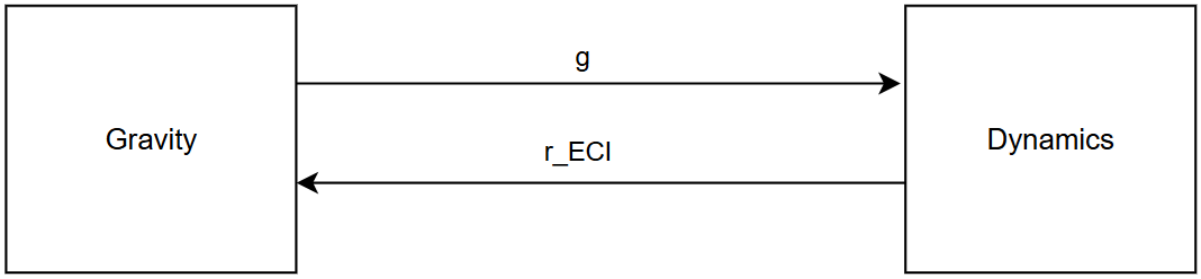


Figure 5.9: Block diagram illustrating the interfaces between the gravity model and the other dynamics model.

5.4.5 Verification and Validation

The gravity model was verified within the integrated simulation environment in order to ensure the correctness of its numerical implementation and its proper interaction with the dynamics model. The gravity model cannot be exercised independently of the dynamics layer, as it relies directly on the vehicle position provided by the dynamics model and delivers its output in a form directly consumed by the equations of motion. For this reason, the verification strategy is based on integrated testing rather than on isolated unit-level tests.

Integrated verification within the simulation framework

The gravity model was verified through an integrated test in which the gravitational acceleration computed by the C++ implementation is compared against an independent Python reference computation. The reference implementation reproduces the same analytical expressions for the central Newtonian attraction and the J_2 perturbation and uses identical physical constants.

In the test scenario, the vehicle state is initialized at a prescribed position $\mathbf{r} = [x\ y\ z]^\top$ expressed in an Earth-centered inertial reference frame. The dynamics model provides this position to the gravity component, which computes the corresponding gravitational acceleration. The resulting acceleration vector is then compared component-wise to the Python reference result. All other force contributions, including aerodynamics and thrust, were disabled in order to isolate the gravity effect within the integrated simulation loop.

The numerical agreement is assessed by verifying that the absolute difference between

the computed and reference acceleration components satisfies

$$|g_{\mathrm{cpp},i} - g_{\mathrm{ref},i}| \leq \mathrm{TOL}, \qquad i \in \{x, y, z\}, \tag{5.67}$$

with $\mathrm{TOL} = 10^{-4}$, where the acceleration components are expressed in $\mathrm{m\,s^{-2}}$.

Remarks on verification strategy

The use of an integrated verification test allows both the numerical correctness of the gravity formulation and the consistency of the data exchange between the gravity and dynamics models to be verified simultaneously. This approach ensures that the gravity model behaves as expected when embedded in the full simulation framework.

Within the scope of this thesis, no direct validation against experimental flight data was performed. In practice, isolating gravitational effects from other force contributions in launcher telemetry is particularly difficult during atmospheric ascent. Nevertheless, the performed integrated verification against an independent reference implementation provides a high level of confidence in the correctness and robustness of the implemented gravity model.

Chapter 6

Monte Carlo and Sensitivity Analyses

6.1 Motivation and Objectives

Atmospheric effects constitute one of the primary sources of uncertainty in launch vehicle trajectory simulations. Unlike deterministic forces such as gravity or thrust, atmospheric phenomena exhibit both spatial variability and stochastic behavior, particularly in the presence of winds and turbulence. As a result, even when the vehicle dynamics and control laws are perfectly defined, the aerodynamic loads experienced during ascent may vary significantly from one realization to another.

In the present work, atmospheric uncertainties arise mainly from two distinct sources. First, the mean wind profiles, although typically derived from pre-flight measurements or climatological data, remain subject to limited accuracy and temporal variability. Second, the turbulent component of the atmosphere is inherently stochastic and cannot be predicted deterministically. The Dryden turbulence model introduced in the previous chapter provides a statistically consistent representation of this phenomenon, but individual realizations depend on random seeds and model parameters.

From an engineering perspective, it is therefore insufficient to analyze a single nominal trajectory. Instead, it is necessary to assess how variations in atmospheric inputs propagate to key aerodynamic quantities such as aerodynamic angles and loads. These quantities directly influence structural sizing, control authority, and overall flight robustness.

The objective of this chapter is twofold. First, sensitivity analyses are performed to evaluate the influence of selected atmospheric parameters on the vehicle response. In particular, the impact of the turbulence intensity parameter and of the mean wind magnitude is investigated using controlled parameter variations around a nominal configuration. These analyses aim to provide first-order insights into which parameters have the strongest effect on aerodynamic loads and angles.

Second, a Monte Carlo analysis is conducted to characterize the statistical dispersion induced by atmospheric turbulence. By running multiple simulations with different stochastic realizations of the turbulence model, the variability of aerodynamic quantities can be quantified in a probabilistic sense. This approach allows the identification of typical ranges, extreme values, and dispersion levels associated with the stochastic nature of the atmosphere.

It is important to note that the analyses presented in this chapter are not intended to constitute a full certification-level probabilistic assessment. Rather, they are designed to support model validation and to provide qualitative and quantitative insight into the

sensitivity and robustness of the coupled atmosphere–aerodynamics framework developed in this thesis.

6.2 Quantities of Interest and Evaluation Metrics

In order to assess the impact of atmospheric uncertainties on the vehicle behavior, a set of representative aerodynamic quantities is selected and monitored throughout the simulations. These quantities are chosen based on their physical relevance, their role in aerodynamic loading, and their sensitivity to wind and turbulence effects.

The primary quantities of interest considered in this chapter are the aerodynamic angles, the dynamic pressure, and selected aerodynamic force components. Together, they provide a comprehensive view of both the kinematic and load-related consequences of atmospheric variability.

The angle of attack α is monitored as it directly influences the aerodynamic force and moment coefficients and plays a central role in vehicle stability and control. In the context of atmospheric disturbances, fluctuations in α reflect the sensitivity of the vehicle longitudinal response to variations in the relative airflow.

The sideslip angle β is considered as a key indicator of lateral aerodynamic effects. Mean wind variations and turbulence components in the horizontal plane can induce non-negligible sideslip angles, which in turn generate lateral forces and yawing moments. Monitoring β is therefore essential for evaluating lateral load excursions and potential control challenges.

The dynamic pressure q_{dyn} is used as a global measure of aerodynamic loading intensity. Since it scales with both air density and the square of the relative airspeed, it provides a direct indication of the severity of aerodynamic loads acting on the vehicle. Variations in q_{dyn} are particularly relevant for structural sizing and for identifying critical flight phases such as the maximum dynamic pressure region.

In addition to these scalar quantities, the lateral aerodynamic force component F_y is explicitly analyzed in selected cases. This force component is directly associated with sideslip effects and is of particular interest when assessing the impact of crosswinds and lateral turbulence on vehicle loads.

For each quantity of interest, two complementary evaluation metrics are employed. The maximum value observed over the simulation time is used to characterize peak loads or extreme excursions, which are often critical for design considerations. In parallel, the root-mean-square (RMS) value is computed to quantify the overall level of fluctuation and the sustained intensity of the response over time.

The combined use of maximum and RMS metrics allows distinguishing between isolated extreme events and persistent oscillatory behavior. This distinction is particularly relevant when analyzing stochastic turbulence effects, where large instantaneous peaks and long-term variability may have different engineering implications.

All quantities are extracted in the launcher body frame and are evaluated over identical time windows to ensure consistent comparison between different simulation cases. This common evaluation framework enables meaningful interpretation of sensitivity trends and statistical dispersion in the subsequent analyses.

6.3 Common Simulation Setup

All sensitivity and Monte Carlo analyses presented in this chapter are performed using identical initial vehicle conditions and simulation settings within each study, in order to

ensure consistent and meaningful comparisons between different cases.

For the sensitivity analyses, the vehicle is initialized at rest, corresponding to a ground-level launch configuration. The initial position is defined on the Earth surface, and the initial translational velocity is set to zero. The vehicle attitude is initialized using a unit quaternion corresponding to a nominal orientation, with the vehicle longitudinal axis aligned with the local vertical at the launch site.

For the Monte Carlo simulations, the same initial position and attitude are used, but a small initial forward velocity is prescribed to ensure a non-zero relative airflow. The initial translational velocity is set to [50,0,0],m/s in the body frame. This choice avoids ill-defined aerodynamic angles and ensures that turbulence effects are well defined from the start of the simulation.

Atmospheric turbulence is enabled for all simulations, and wind effects are applied using a consistent wind reference frame and model configuration for all runs. The nominal turbulence intensity provided by the Dryden model is used as a baseline and may be scaled depending on the specific sensitivity or Monte Carlo case.

The total simulation duration is set to 120 seconds for the sensitivity analyses and to 30 seconds for the Monte Carlo simulations. The shorter duration is sufficient to capture the short-term effects of atmospheric turbulence while keeping the computational cost of the Monte Carlo campaign reasonable. All vehicle, propulsion, and integration parameters are kept identical across runs within each study, ensuring that observed variations in aerodynamic quantities are solely attributable to atmospheric effects.

6.4 Sensitivity Analysis

Prior to the Monte Carlo campaign, a series of sensitivity analyses is conducted to assess the influence of selected atmospheric parameters on the aerodynamic response of the vehicle. The objective of these analyses is not to provide a full statistical characterization, but rather to identify first-order trends and to determine which parameters have a dominant or secondary impact on key aerodynamic quantities.

To this end, controlled parameter variations are performed around a nominal configuration while keeping all other inputs, including the stochastic realization of turbulence, fixed. This approach allows isolating the effect of individual parameters and provides qualitative insight into their relative importance. Two sources of atmospheric variability are considered: the turbulence intensity scaling factor and the mean wind magnitude and direction.

6.4.1 Sensitivity to the Turbulence Intensity Scaling Factor

A sensitivity analysis is performed to assess the influence of turbulence intensity on the aerodynamic response of the vehicle. In the Dryden turbulence model, the standard deviation of the stochastic wind components depends on altitude and defines a nominal level of turbulence intensity. At ground level, which corresponds to the initial conditions of the present simulations, the model yields a nominal turbulence intensity of $\sigma_{\text{nominal}} = 4$.

In order to investigate sensitivity effects without modifying the structure of the turbulence model itself, a dimensionless scaling factor k_σ is introduced and applied to the nominal turbulence intensity. The effective turbulence level used in the simulations is therefore defined as

$$\sigma_{\text{eff}} = k_\sigma \, \sigma_{\text{nominal}}.$$

Three representative values of the scaling factor are considered:

$$k_\sigma \in \{0.7,\ 1.0,\ 1.3\},$$

corresponding respectively to reduced, nominal, and increased turbulence levels with respect to the reference Dryden model.

For all cases, identical initial conditions and the same turbulence realization are used. The turbulence state is reset prior to each simulation to ensure strict reproducibility and to isolate the effect of the turbulence intensity scaling factor from realization-to- realization stochastic variability.

The analysis focuses on the angle of attack α and the dynamic pressure q, which are key indicators of aerodynamic loading. For each simulation, both the maximum value (defined as the maximum absolute value over time) and the root-mean-square (RMS) value of these quantities are extracted over the full simulation duration. The resulting sensitivity metrics are summarized in Table 6.1.

k_σ	$\alpha_{\max}$ [deg]	α_{RMS} [deg]	$q_{dyn,\max}$ [Pa]	$q_{dyn,\mathrm{RMS}}$ [Pa]
0.7	3.104	1.769	32346	9691
1.0	3.129	1.714	32346	9162
1.3	3.115	1.673	32346	9676

Table 6.1: Sensitivity of aerodynamic quantities to the turbulence intensity scaling factor k_σ.

The results indicate that the maximum angle of attack exhibits a very weak sensitivity to the turbulence intensity scaling factor. Variations remain below 1% across the considered range of k_σ, and no monotonic trend is observed. This behavior reflects the fact that peak angle-of-attack excursions are primarily governed by the nominal trajectory and vehicle dynamics, rather than by the amplitude of stochastic turbulence perturbations in a single realization.

In contrast, the RMS value of the angle of attack shows a moderate sensitivity to k_σ, with variations on the order of a few percent. This behavior reflects changes in the overall level of stochastic longitudinal excitation rather than in extreme events, highlighting the distinction between peak-driven and variability-driven aerodynamic metrics.

For the dynamic pressure, the maximum value is found to be insensitive of the turbulence intensity scaling factor, confirming that peak aerodynamic loading is dominated by the nominal ascent profile and atmospheric density distribution. The RMS dynamic pressure exhibits moderate but non-monotonic variations with k_σ, indicating a secondary influence of turbulence-induced velocity fluctuations without affecting global load maxima.

It is important to emphasize that this sensitivity study is based on a limited number of discrete scaling values and a single turbulence realization. The objective is therefore not to establish a quantitative or statistical relationship, but rather to identify first-order trends and to assess the relative importance of turbulence intensity effects. The influence of realization-to-realization variability and the statistical dispersion induced by atmospheric turbulence are addressed separately through the Monte Carlo analysis presented later in this chapter.

6.4.2 Sensitivity Analysis to Mean Wind

In this part, a sensitivity analysis is conducted to evaluate the influence of the mean wind on the lateral aerodynamic response of the vehicle. Unlike atmospheric turbulence, which

introduces stochastic fluctuations, the mean wind is treated here as a deterministic input parameter whose magnitude and direction can be explicitly controlled.

The analysis is performed by independently scaling the East and North components of the mean wind around a nominal configuration, while keeping all other simulation parameters unchanged. For each wind direction, representative scaling factors are applied in order to assess the resulting variations in sideslip angle, lateral aerodynamic force, and dynamic pressure.

This approach allows isolating the directional effects of the mean wind and identifying potential asymmetries in the vehicle response associated with wind orientation. The results are analyzed using peak and RMS metrics, providing complementary insight into extreme load excursions and sustained aerodynamic effects.

Only the horizontal components of the mean wind are considered in this sensitivity analysis. These components primarily drive lateral aerodynamic phenomena and directly affect the quantities of interest investigated in this study. The vertical wind component is typically of smaller magnitude compared to the horizontal components and is therefore not considered in this analysis.

Sensitivity to the East Wind Component

The influence of the East component of the mean wind is investigated by applying a scaling factor $k_E \in \{0.7,\ 1.0,\ 1.3\}$ to the nominal wind profile, while keeping all other simulation parameters unchanged. This configuration allows isolating the effect of lateral wind variations aligned with the East direction on the aerodynamic response of the vehicle.

The sensitivity analysis focuses on the sideslip angle β, the lateral aerodynamic force F_y, and the dynamic pressure q_{dyn}, using both maximum absolute values and root-mean-square (RMS) metrics. The extracted quantities for each scaling factor are summarized in Table 6.2.

k_E	$\beta_{\max}$ [deg]	β_{RMS} [deg]	$F_{y,\max}$ [N]	$F_{y,\mathrm{RMS}}$ [N]	$q_{dyn,\max}$ [Pa]	$q_{dyn,\mathrm{RMS}}$ [Pa]
0.7	1.368	0.706	629199	178608	32318	9604
1.0	1.424	0.693	629761	179717	32346	9703
1.3	1.463	0.662	532661	165084	27443	9015

Table 6.2: Sensitivity of aerodynamic quantities to the East wind scaling factor k_E.

The results indicate that variations in the East wind scaling factor primarily affect lateral aerodynamic quantities. An increase in k_E leads to a gradual increase in the maximum sideslip angle, reflecting the direct influence of the East wind component on the relative airflow orientation. In contrast, the RMS value of the sideslip angle shows a slight decreasing trend as k_E increases, suggesting a change in the distribution of lateral fluctuations rather than a uniform amplification.

The lateral aerodynamic force exhibits a non-monotonic response to variations in the wind scaling factor. Both the maximum and RMS values of F_y exhibit a marked non-monotonic response, with a significant reduction at $k_E = 1.3$, indicating a trajectory shaping induced by strong crosswind conditions.

In contrast to the sensitivity study on the turbulence intensity scaling factor k_σ, the dynamic pressure shows a much stronger sensitivity to k_E, with a significant reduction of both maximum and RMS values at $k_E = 1.3$. This indicates that strong crosswind conditions can indirectly alter the longitudinal ascent profile and the flight condition at which peak aerodynamic loading occurs, rather than merely introducing lateral perturbations around an unchanged nominal trajectory.

Sensitivity to the North Wind Component

A sensitivity analysis is conducted to assess the influence of the North component of the mean wind on the aerodynamic response of the vehicle. A scaling factor $k_N \in \{0.7,\ 1.0,\ 1.3\}$ is applied to the nominal North wind profile, while all other simulation parameters are kept unchanged. This approach allows isolating the effect of variations in the North wind intensity on lateral aerodynamic behavior.

For each simulation, maximum values are defined as the maximum absolute value observed over the simulation time, and RMS values are computed over the full duration. The resulting sensitivity metrics are summarized in Table 6.3.

k_N	$\beta_{\max}$ [deg]	β_{RMS} [deg]	$F_{y,\max}$ [N]	$F_{y,\mathrm{RMS}}$ [N]	$q_{dyn,\max}$ [Pa]	$q_{dyn,\mathrm{RMS}}$ [Pa]
0.7	1.332	0.709	705010	168452	36130	9331
1.0	1.339	0.585	629761	158071	32346	8983
1.3	1.381	0.671	618668	175419	31787	10196

Table 6.3: Sensitivity of aerodynamic quantities to the North wind scaling factor k_N.

The results show that variations in the North wind scaling factor have a noticeable impact on lateral aerodynamic quantities. An increase in k_N leads to a slight but consistent increase in the maximum sideslip angle, indicating a stronger coupling between the North wind component and the vehicle attitude response. In contrast, the RMS value of the sideslip angle exhibits a marked and non-monotonic variation, with a significant reduction at $k_N = 1.0$ and a subsequent increase at $k_N = 1.3$, indicating that the overall level of lateral fluctuations is strongly affected by changes in the mean North wind intensity rather than remaining constant.

The lateral aerodynamic force exhibits a clearly non-monotonic response to variations in k_N. The maximum value decreases by more than 10% as k_N increases, while the RMS value first decreases and then increases significantly at $k_N = 1.3$, highlighting the nonlinear interaction between wind-induced attitude changes, trajectory evolution, and aerodynamic force generation.

The dynamic pressure shows a moderate but non-negligible sensitivity to k_N, with a reduction of more than 10% in its maximum value as the North wind scaling factor increases. Although this effect is weaker than in the k_E case, it remains significantly stronger than in the turbulence intensity scaling study, confirming that steady wind components can indirectly alter the longitudinal ascent profile and the flight condition at which peak aerodynamic loading occurs.

6.5 Monte Carlo Analysis

While the previous sections focused on deterministic sensitivity analyses to specific parameters, such an approach cannot fully capture the intrinsic variability introduced by stochastic atmospheric disturbances. In particular, the Dryden turbulence model relies on random processes whose realizations may significantly affect the vehicle response, even when all other parameters are kept identical.

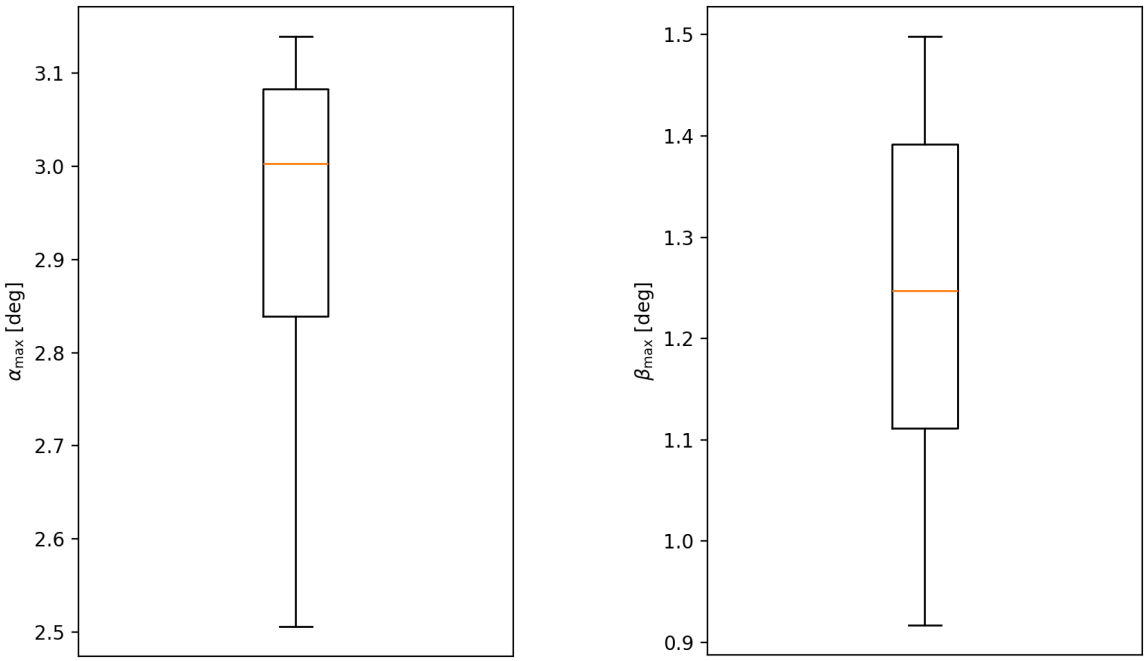
For this reason, a Monte Carlo approach is adopted in this section in order to assess the statistical dispersion of the main aerodynamic quantities of interest. By repeating the same simulation multiple times with different turbulence realizations, it becomes possible to characterize not only nominal trends, but also the variability, spread, and extreme values induced by atmospheric uncertainty.

Two complementary Monte Carlo campaigns are conducted. In the first one, only the turbulence seed is varied, while all physical parameters are kept identical. In the second one, both the turbulence seed and the turbulence intensity scaling factor are randomized, allowing a broader exploration of the uncertainty space. The results are analyzed using statistical indicators and distribution plots in order to quantify the robustness of the launcher response to atmospheric disturbances.

6.5.1 Monte Carlo with turbulence seed variation only

A first Monte Carlo campaign is performed in order to quantify the intrinsic variability of the launcher response induced solely by the stochastic nature of the turbulence model. In this first study, all simulation parameters are kept strictly identical from one run to another, and only the random seed of the Dryden turbulence generator is modified. This allows isolating the effect of the stochastic realization of the atmospheric disturbance, without introducing any additional parametric uncertainty.

The dispersion of these peak values over the Monte Carlo ensemble is visualized using boxplots, shown in Figures 6.1 and 6.2. These plots provide a compact statistical representation of the median, interquartile range, and extreme values observed across the 50 runs.

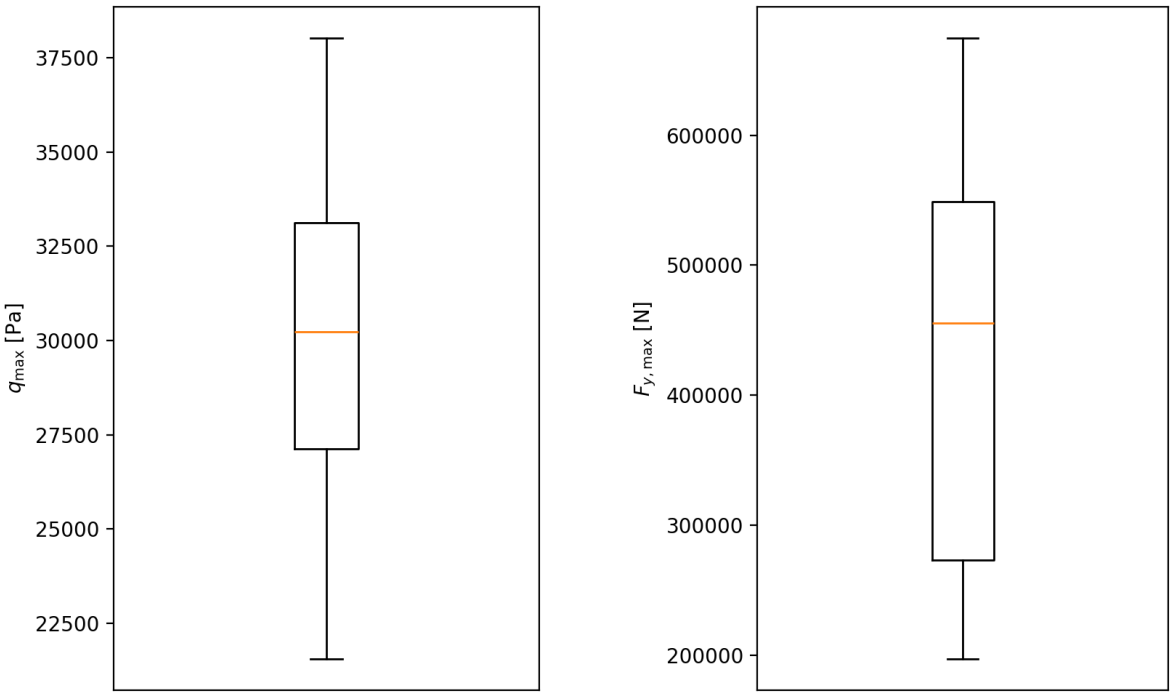


(a) $\alpha_{\max}$ (b) $\beta_{\max}$

Figure 6.1: Monte Carlo dispersion of peak aerodynamic angles (seed-only case).

The results show that, even with a fixed turbulence intensity and identical mean wind conditions, the stochastic nature of the turbulence realization induces a non-negligible dispersion in all considered aerodynamic quantities. In particular, the maximum lateral force $F_{y,max}$ exhibits a significant spread, reflecting the strong sensitivity of lateral loads to the instantaneous structure of the wind field encountered by the vehicle. The other variables also show noticeable variability.

A summary of the Monte Carlo statistics for the peak values of each quantity (minimum, maximum, mean and standard deviation over the ensemble) is provided in Table 6.4.



(a) $q_{dyn,\max}$ (b) $F_{y,\max}$

Figure 6.2: Monte Carlo dispersion of peak aerodynamic loads (seed-only case).

Quantity	Mean	Std. dev.	Min	Max
$\alpha_{\max}$ [deg]	2.95	0.17	2.51	3.14
$\beta_{\max}$ [deg]	1.23	0.17	0.92	1.5
$q_{\mathrm{dyn,max}}$ [Pa]	30057	4235	21546	38028
$F_{y,\max}$ [N]	424880	149182	196972	674939

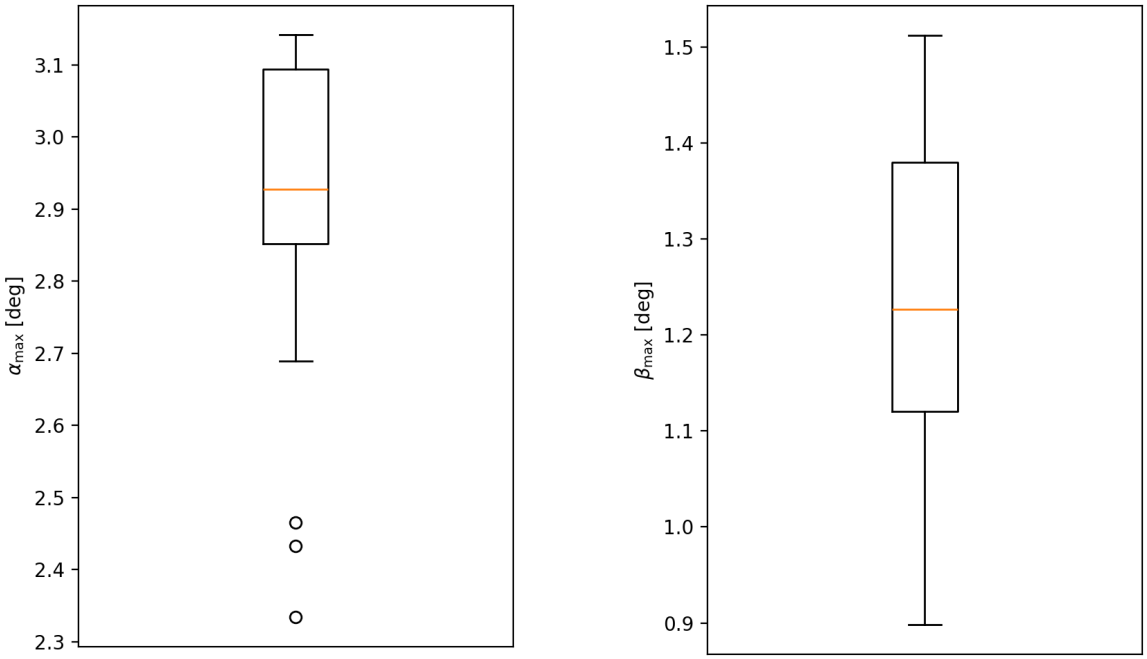
Table 6.4: Monte Carlo statistics of peak aerodynamic quantities (seed-only case).

6.5.2 Monte Carlo with turbulence seed and turbulence intensity variation

A second Monte Carlo campaign is performed in order to account not only for the stochastic realization of the turbulence, but also for uncertainty on its overall intensity. In this study, both the turbulence seed and the turbulence intensity scaling factor σ are varied simultaneously from one run to another. More precisely, the seed is changed at each run, while the factor σ is sampled from a normal distribution with mean $\mu = 1.0$ and standard deviation $\sigma_{\mathrm{std}} = 0.15$, and clipped to the interval [0.7, 1.3] in order to remain consistent with the sensitivity analysis bounds.

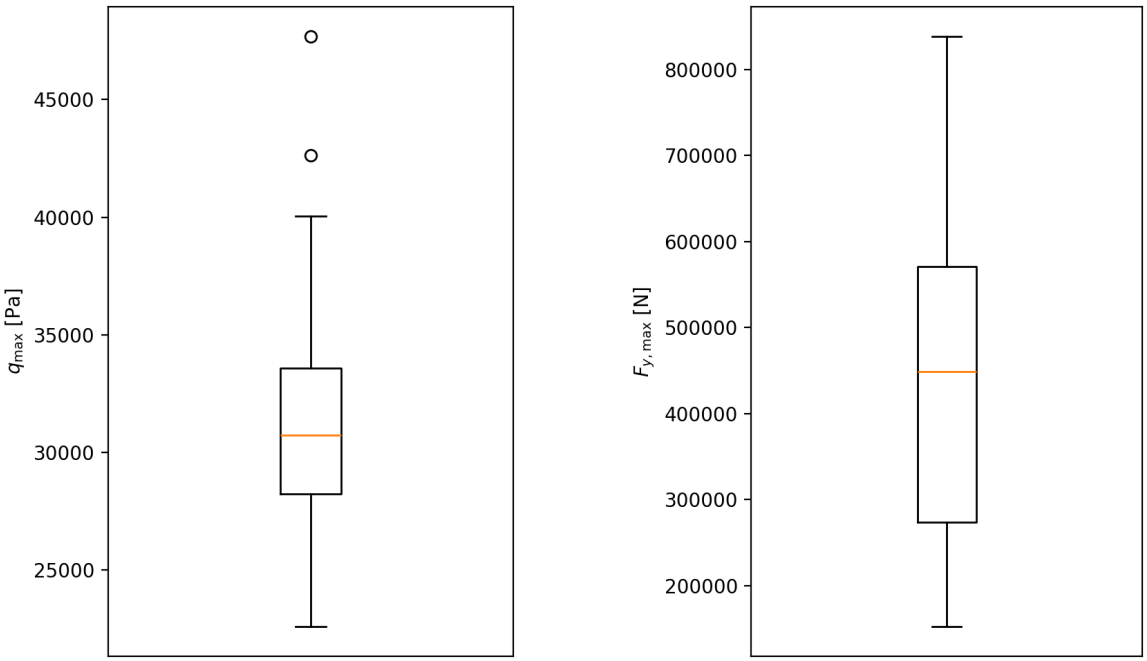
This approach makes it possible to combine two sources of variability: the intrinsic stochasticity of the turbulence realization and the uncertainty on its global intensity level. As in the previous case, the analysis focuses on the peak values reached during each simulation for the angle of attack α, the sideslip angle β, the dynamic pressure q, and the lateral aerodynamic force F_y.

The dispersion of these peak values over the Monte Carlo ensemble is shown using boxplots in Figures 6.3 and 6.4. Compared to the seed-only case, a slightly larger spread can be observed for most quantities, reflecting the combined effect of random turbulence realizations and variations in turbulence intensity.



(a) $\alpha_{\max}$ (b) $\beta_{\max}$

Figure 6.3: Monte Carlo dispersion of peak aerodynamic angles (seed and turbulence intensity variation).



(a) $q_{dyn,\max}$ (b) $F_{y,\max}$

Figure 6.4: Monte Carlo dispersion of peak aerodynamic loads (seed and turbulence intensity variation).

As expected, the lateral aerodynamic force F_y remains the most sensitive quantity, with a very large dispersion across the ensemble.

It can be noted that several outliers are present in the boxplots, in particular for the angle of attack α and the dynamic pressure q_{dyn}. These points correspond to specific combinations of turbulence realization and turbulence intensity factor that lead to more extreme, but still physically plausible, responses of the vehicle. Their presence highlights the strongly nonlinear nature of the launcher dynamics with respect to atmospheric disturbances: small changes in the turbulence characteristics can occasionally trigger significantly different transient responses.

From a robustness and load envelope perspective, these rare but more severe cases are of particular interest, since they may drive the sizing loads or the definition of operational margins. This further justifies the use of Monte Carlo approaches rather than relying solely on mean or nominal trajectories.

A numerical summary of the Monte Carlo statistics for the peak values of each quantity (minimum, maximum, mean and standard deviation over the ensemble) is provided in Table 6.5.

Quantity	Mean	Std. dev.	Min	Max
$\alpha_{\max}$ [deg]	2.97	0.12	2.33	3.14
$\beta_{\max}$ [deg]	1.24	0.13	0.9	1.51
$q_{\max}$ [Pa]	31157	4888	22591	47695
$F_{y,\max}$ [N]	455304	224223	164873	847426

Table 6.5: Monte Carlo statistics of peak aerodynamic quantities (seed and turbulence intensity variation).

6.6 Conclusion

This chapter investigated the propagation of atmospheric uncertainties to key aerodynamic quantities using a combination of deterministic sensitivity analyses and Monte Carlo simulations.

The sensitivity analyses provided first-order insights into the relative influence of the selected parameters. Scaling the turbulence intensity mainly affected fluctuation-related metrics (RMS values), while peak quantities such as $q_{dyn,\max}$ and $\alpha_{\max}$ remained essentially unchanged over the tested range, indicating a weak influence of turbulence intensity on the peak longitudinal loading in this configuration. In contrast, variations of the mean wind components had a clear impact on lateral quantities, with noticeable changes in β and F_y. Moreover, for sufficiently strong wind levels, especially in the East wind case, the mean wind was also observed to indirectly affect the longitudinal loading by modifying the ascent trajectory and the flight condition at which the maximum dynamic pressure occurs. This confirms the existence of a nonlinear coupling between lateral disturbances, guidance response, and longitudinal aerodynamics.

The Monte Carlo campaigns complemented these deterministic results by quantifying the run-to-run dispersion induced by the stochastic nature of the turbulence. Even in the seed-only case, all peak quantities exhibited a significant spread, with the lateral force $F_{y,\max}$ showing the largest variability. When uncertainty on the turbulence intensity was added, the dispersion increased further, leading to a wider envelope of possible peak responses. The presence of outliers in $\alpha_{\max}$, $q_{dyn,\max}$ and $F_{y,\max}$ confirms the strongly nonlinear character of the coupled atmosphere–aerodynamics–dynamics system, in which small changes in the disturbance realization can occasionally produce markedly different transient responses.

From an engineering standpoint, these results indicate that deterministic single-run simulations and local sensitivity studies are insufficient to characterize the envelope of aerodynamic loads. While they are useful to identify trends and physical mechanisms, the Monte Carlo results show that rare but severe cases dominate the load envelope, in particular for lateral loads. Consequently, probabilistic approaches are required to properly assess robustness and to identify potentially sizing-relevant conditions.

Finally, it must be emphasized that the present analyses remain limited in scope. The Monte Carlo campaigns are restricted in sample size, and the uncertainty modeling is limited to a Dryden turbulence representation and simplified wind scaling. Nevertheless, the results provide a consistent validation-oriented assessment of the implemented atmospheric and aerodynamic modules, and they demonstrate the capability of the simulation framework to support more comprehensive robustness and uncertainty analyses in future studies.

Chapter 7

Perspectives

The modelling and simulation framework developed in this work constitutes a solid basis for the analysis of launcher dynamics under atmospheric disturbances. While the implemented models already provide a good compromise between fidelity and computational efficiency, several perspectives can be identified to further enhance the capabilities and applicability of the tool.

7.1 Aerodynamic model extensions

A first major axis of improvement concerns the aerodynamic modelling. In the current implementation, aerodynamic coefficients are provided through a predefined formulation, which is well suited for preliminary trajectory analyses. However, future developments could offer the user increased flexibility in the definition of these coefficients.

In particular, the framework could be extended to support alternative aerodynamic models, such as two-dimensional or three-dimensional coefficient databases. In the latter case, multi-dimensional interpolation techniques would allow the aerodynamic forces and moments to be evaluated as functions of multiple state variables, enabling a more accurate representation of complex aerodynamic effects encountered during ascent.

Additionally, the aerodynamic coefficients could be made dependent on the angular velocity components p, q, and r, allowing rotational effects to be taken into account. Such extensions would be especially relevant for the analysis of highly dynamic phases or for vehicles exhibiting strong coupling between translational and rotational motion.

The introduction of aerodynamic control surfaces would also motivate a dependency of the coefficients on actuator deflections. This would naturally extend the scope of the model toward guidance and control studies, enabling closed-loop simulations under realistic aerodynamic conditions.

Finally, geometric refinements could be considered, such as accounting for thrusters located outside the main cylindrical body when computing aerodynamic drag contributions. This would improve the physical consistency of the model, particularly during low-altitude flight where aerodynamic effects are dominant.

7.2 Atmospheric disturbance modelling

With respect to atmospheric disturbances, the present work relies on a Dryden turbulence model to represent stochastic wind fluctuations. While this model is widely used and well suited for engineering-level analyses, alternative turbulence formulations could be investigated in future work.

The integration of additional turbulence models would allow a systematic comparison of their impact on vehicle loads, attitude excursions, and dynamic pressure. Such studies would be particularly relevant in the context of Monte Carlo analyses, where the sensitivity of the launcher to different representations of atmospheric uncertainty could be assessed more comprehensively.

7.3 Vehicle and mission modelling enhancements

Several perspectives can also be identified at the vehicle and mission modelling level. A more detailed fuel consumption model could be introduced, allowing the mass and inertia properties of the launcher to evolve dynamically throughout the simulation. This would improve the consistency between propulsion, dynamics, and guidance models.

Furthermore, the explicit modelling of aerodynamic fins would extend the applicability of the framework to configurations relying on aerodynamic stabilization or control. This would enable more advanced studies of lateral dynamics and control authority in the presence of atmospheric disturbances.

For multi-stage launch vehicles, the consideration of stage separation events represents an important extension. Modelling such events would allow the simulation of complete ascent trajectories and would further increase the realism and industrial relevance of the framework.

7.4 Robustness and simulation safeguards

From a numerical and software perspective, additional safeguards could be implemented to improve the robustness of the simulation framework. For instance, specific handling of ground contact situations could be introduced in order to prevent non-physical states when the vehicle intersects the Earth surface.

Such mechanisms would be particularly useful when performing large-scale parametric or Monte Carlo analyses, where extreme cases may arise. Overall, these improvements would contribute to making the framework more reliable, extensible, and suitable for long-term use in advanced launcher studies.

7.5 Uncertainty quantification and statistical analysis

A natural continuation of the present work concerns a more systematic and formal treatment of uncertainty quantification and statistical post-processing. In the current study, Monte Carlo simulations were mainly used to illustrate dispersion effects and to identify potentially critical load cases. However, more advanced statistical analyses could be performed in future work.

In particular, larger Monte Carlo campaigns would allow more reliable estimation of tail quantities, such as high-percentile load levels (e.g. 99.9% envelopes), which are typically of interest for sizing and verification purposes. In addition, variance-based sensitivity analysis techniques could be introduced in order to quantify the relative contribution of each uncertain parameter (e.g. wind components, turbulence intensity, guidance parameters) to the overall dispersion of the response.

Beyond classical Monte Carlo sampling, more advanced uncertainty propagation methods could also be considered, such as Latin Hypercube Sampling or polynomial chaos expansions, in order to reduce the computational cost while preserving statistical representativeness.

Finally, the integration of these techniques within the simulation framework would enable the development of a more systematic robustness and margins management approach, fully aligned with modern launcher design and verification practices.

Chapter 8

Conclusion

The objective of this thesis was to develop and integrate a set of physical models enabling the simulation of launcher dynamics under realistic atmospheric and environmental conditions, with a particular focus on the influence of atmospheric disturbances on aerodynamic loads and vehicle response. To this end, dedicated models for the atmosphere, aerodynamics, gravity, vehicle attitude dynamics, and stochastic disturbances were implemented, validated, and coupled within a unified and modular simulation framework.

A physically consistent atmospheric model was first introduced, providing the thermodynamic properties required for aerodynamic load computation. Particular attention was devoted to the modelling of wind effects, including both deterministic wind profiles and stochastic turbulence through a Dryden-based formulation. The resulting relative wind velocity model offers a level of fidelity well suited for engineering-level trajectory and load analyses, while remaining numerically robust and computationally efficient.

Building upon this foundation, an aerodynamic model was developed to compute the forces and moments acting on the launcher as functions of the relative velocity, atmospheric properties, and user-defined aerodynamic coefficients. By allowing these coefficients to be provided through external data tables and interpolated with respect to Mach number and relevant angular parameters, the framework achieves a high level of flexibility and adaptability to different vehicle configurations. Extensive unit and integrated testing demonstrated the correctness, consistency, and good numerical behaviour of the implemented formulations.

In parallel, a gravity model incorporating both the central Newtonian attraction and the J_2 perturbation was implemented. This approach captures the dominant characteristics of the Earth's gravitational field relevant for launcher ascent trajectories while maintaining a limited computational overhead. Its integration within the overall framework ensures a coherent and physically consistent representation of the main external forces acting on the vehicle.

In addition, the attitude and rigid-body dynamics modelling was refined and integrated consistently within the simulation loop, ensuring a coherent coupling between translational motion, rotational motion, and aerodynamic loading throughout the ascent.

Beyond the development of the models themselves, a major contribution of this work lies in the systematic analysis of the impact of atmospheric disturbances on launcher dynamics and aerodynamic loads. A set of deterministic sensitivity analyses and Monte Carlo campaigns was performed in order to quantify both parametric effects and stochastic variability.

The sensitivity studies made it possible to identify clear trends and physical mechanisms. In particular, the turbulence intensity was shown to mainly affect fluctuation-

related metrics, while having only a limited influence on peak longitudinal loading in the tested configuration. In contrast, steady wind components, especially the crosswind component, were shown to strongly influence lateral quantities such as the sideslip angle and lateral aerodynamic force. Moreover, for sufficiently strong wind levels, these steady wind components were also observed to indirectly affect the longitudinal loading by modifying the ascent trajectory and the flight condition at which the maximum dynamic pressure occurs. This highlights the strongly nonlinear coupling between atmospheric disturbances, guidance response, and vehicle aerodynamics.

The Monte Carlo simulations complemented these deterministic analyses by revealing the magnitude of run-to-run dispersion induced by the stochastic nature of atmospheric turbulence. Even for fixed mean conditions, a significant variability was observed in all peak quantities, with the lateral aerodynamic force exhibiting particularly large dispersion. When uncertainty on the turbulence intensity was added, the envelope of possible responses was further enlarged. The presence of rare but severe outliers in several quantities confirms the highly nonlinear character of the coupled atmosphere–aerodynamics–dynamics system, in which small changes in the disturbance realization can occasionally trigger markedly different transient responses.

From an engineering perspective, these results lead to several important conclusions. First, they show that deterministic single-run simulations and local sensitivity analyses, while useful to understand trends and physical mechanisms, are not sufficient to characterize the envelope of aerodynamic loads. Second, they demonstrate that lateral loads are particularly sensitive to atmospheric disturbances and are likely to drive sizing and verification considerations. More generally, the Monte Carlo results confirm that rare but severe cases, rather than nominal or mean trajectories, dominate the load envelope and must therefore be captured through probabilistic approaches.

Overall, the developed framework offers a balanced compromise between physical fidelity, modularity, and numerical efficiency. It provides a solid basis for further investigations, including higher-fidelity aerodynamic models, more advanced disturbance representations, and extended mission scenarios. Although the present work remains limited in scope in terms of uncertainty modelling and statistical sampling, it nevertheless demonstrates the relevance of the proposed approach and confirms the suitability of the framework for future launcher simulation, robustness assessment, and uncertainty-aware design studies.

Bibliography

[1] Julian Gaßler, Tobias Lindemann, Rene Arendsen, and David de Groote. R2M2: A Modular Multi-Physics Simulation Framework for Reusable Launch Vehicles. *Journal of Aerospace Information Systems*, 2023. Used as a conceptual reference for modular multi-physics simulation frameworks.

[2] U.S. Standard Atmosphere 1976. Technical report, NOAA, NASA, and USAF, Washington, D.C., 1976.

[3] J. M. Picone, A. E. Hedin, D. P. Drob, and A. C. Aikin. NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues. *Journal of Geophysical Research: Space Physics*, 2002.

[4] I. Harris and H. Priester. Theoretical Models for the Aeronomy of the Upper Atmosphere. Technical report, NASA, Washington, D.C., 1962.

[5] L. G. Jacchia. Revised Static Models of the Thermosphere and Exosphere. *Smithsonian Contributions to Astrophysics*, 1970.

[6] Radiosonde observations and their role in atmospheric profiling. Technical report, National Oceanic and Atmospheric Administration (NOAA), 2020.

[7] H. L. Dryden. A review of the statistical theory of turbulence. Technical report, National Advisory Committee for Aeronautics (NACA), 1952.

[8] Theodore von Kármán. Progress in the statistical theory of turbulence. *Proceedings of the National Academy of Sciences*, 1948.

[9] J. C. Kaimal, J. C. Wyngaard, Y. Izumi, and O. R. Coté. Spectral characteristics of surface-layer turbulence. *Quarterly Journal of the Royal Meteorological Society*, 1972.

[10] A. G. Davenport. The spectrum of horizontal gustiness near the ground in high winds. *Quarterly Journal of the Royal Meteorological Society*, 1961.

[11] NASA Glenn Research Center. Rocket Rotations – Body Axes Diagram. Beginners Guide to Aeronautics web page. Consulted on 2025-12-10.

[12] Ames Research Center. Aerodynamic Data for Space Vehicles. Technical report, NASA, Washington, D.C., 1975.

[13] NASA. Wind Tunnel Testing Techniques. Technical report, NASA, 1990.

[14] John D. Anderson. *Computational Fluid Dynamics: The Basics with Applications*. McGraw-Hill, 1995.

[15] U.S. Air Force. Digital DATCOM User's Manual. Technical report, USAF, 1976.

[16] NASA. Aerodynamic modeling and simulation for launch vehicles. Technical report, NASA, 2010.

[17] NGA. Department of defense world geodetic system 1984. Technical report, National Geospatial-Intelligence Agency, 2014.

[18] David A. Vallado. *Fundamentals of Astrodynamics and Applications*. Microcosm Press, 2013.

[19] F. G. et al. Lemoine. The development of the joint NASA GSFC and NIMA geopotential model EGM96. Technical report, NASA Goddard Space Flight Center, 1998.

[20] European Cooperation for Space Standardization. ECSS-E-ST-40-07C: Simulation Model Portability. Technical report, ECSS, Noordwijk, The Netherlands, 2020.

[21] Wei Du, Bong Wie, and Mark Whorton. Dynamic Modeling and Flight Control Simulation of a Large Flexible Launch Vehicle. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, aug 2008.

[22] R. T. Austin. Local north-east-down reference frame illustration. Online illustration, accessed via Wikimedia Commons, 2007.

[23] Thanh Vu, Nicolas Rongione, and Amir Rahmani. Minimalist control design of canard-actuated micro-projectiles. In *Proceedings of the IEEE Conference on Decision and Control*, April 2015.

[24] Metathrope. Oblate spheroid and spherical earth comparison, 2022.

[25] NASA Jet Propulsion Laboratory. Grace — gravity recovery and climate experiment. `https://grace.jpl.nasa.gov`, 2017.