MULTIDISCIPLINARY MODELING AND SIMULATION FRAMEWORK FOR REUSABLE LAUNCH VEHICLE SYSTEM DYNAMICS AND CONTROL

Lâle Evrim Briese, Paul Acquatella B., and Klaus Schnepper

DLR, German Aerospace Center
Institute of System Dynamics and Control
D-82234 Oberpfaffenhofen, Germany

ABSTRACT

Future concepts and key technologies for reusable launch vehicles are currently investigated by the DLR project AKIRA, focusing on vertical takeoff and horizontal landing (VTHL), as well as horizontal takeoff and horizontal landing (HTHL) concepts.

Dedicated developments of multidisciplinary frameworks for launch vehicle modeling and preliminary design optimization have been presented in the relevant literature. These activities are often performed by several independent and discipline-specific tools; such an approach can only account for limited interactions of the involved disciplines with the overall system dynamics.

Therefore, it is the objective of this paper to focus on a multidisciplinary launch vehicle dynamics modeling, guidance, and control framework to support reusable launch vehicle design activities at DLR while taking into account the highly interconnected disciplines involved and the changing environmental conditions. The modeling framework is based on the object-oriented, multidisciplinary, and equation-based modeling language MODELICA. Dedicated 3-DOF and 6-DOF model implementations, covering the kinematics and dynamics formulation, environmental effects, aerodynamics, and propulsion models are presented.

Within this framework, a method to obtain a direct connection between 3-DOF and 6-DOF models is shown. This is done by considering results from the trajectory optimization package '*trajOpt*' in combination with nonlinear 6-DOF inverse models obtained automatically by MODELICA. Angular rates and the resulting moments can be obtained by this intermediate 6-DOF modeling approach for subsequent controllability studies.

We discuss some of these benefits in terms on nonlinear flight control simulations for an HTHL reusable launch vehicle concept.

Index Terms— Launch Vehicle Modeling, Object-Oriented Modeling, Nonlinear Inverse Models, Flight Dynamics

1. INTRODUCTION

Future launch vehicle concepts and technologies for expendable and reusable launch vehicles are currently investigated by the DLR research project AKIRA, focusing on vertical takeoff and horizontal landing (VTHL), as well as horizontal takeoff and horizontal landing (HTHL) launch vehicle concepts [1, 2, 3, 4, 5].

Within this context, preliminary design studies have to be performed, where the overall launch vehicle system dynamics have to be investigated using a multibody approach for subsequent controllability and trimmability studies. The multibody modeling of reusable launch vehicles can be a challenging task due to multiple disciplines involved in the modeling and simulation process, such as environment, aerodynamics, propulsion, structural dynamics, separation dynamics, and Guidance, Navigation and Control (GNC). Depending on the launch vehicle design and its mission requirements, the launch vehicle configuration can experience significant changes in its system structure and parameter database which has to be accounted for within a modeling framework.

Dedicated developments of multidisciplinary frameworks for launch vehicle modeling and preliminary design optimization have been presented for instance in [6, 7, 8, 9, 10, 11]. Moreover, it is common that each discipline-specific analysis requires different types of models with appropriate level of detail and that these activities are often performed by several independent, discipline-specific tools. Depending on such an approach, the consistency between adequate models for each simulation workflow can be difficult to ensure and only a limited amount of interactions of the involved disciplines with the overall system dynamics can be accounted for.

For this purpose, a multidisciplinary launch vehicle dynamics modeling, guidance, and control framework has been developed at the German Aerospace Center (DLR), Institute of System Dynamics and Control in support of the above mentioned reusable launch vehicle design activities. We do this by taking into account highly interconnected disciplines, changing environmental conditions and the variable structure of the system due to time- and state-dependent separation events or engine cutoffs [12, 13, 14, 15, 16].

Dedicated three and six degrees of freedom (DOF) models for specific analyses are presented based on the object-oriented, equation-based and acausal modeling language MODELICA, including the kinematics and dynamics formulation, environmental effects, aerodynamics, and propulsion dynamics. These models support different levels of detail and can then be used consistently for multi-objective and multi-phase trajectory optimization with *Functional Mock-up Units* (FMU) [15, 17], for system dynamics analyses using nonlinear inverse models [18, 19], and for G&C design based on *Nonlinear Dynamic Inversion (NDI)* [20, 21, 22, 23].

A brief overview on the main modeling methods is given in Section 2. This includes an introduction into the modeling language MODELICA, the implementation of the multibody dynamics, and an overview on the nonlinear inverse modeling supported by MODELICA. In Section 3, the launch vehicle modeling framework including the implementation of multidisciplinary components is introduced. The consistent implementation of these components for different applications, such as trajectory optimization and nonlinear inverse modeling, is described in Section 4. Some of the capabilities of the modeling framework are demonstrated and discussed in Section 5 for a delta-winged reusable launch vehicle concept. The main benefits of the modeling framework are summarized in Section 6.

2. METHODS

The launch vehicle dynamics modeling, guidance, and control framework is based on the modeling language MODELICA. This modeling language and its advantages in comparison to other modeling tools will be introduced in the following sections. In particular, the modeling of multibody systems, as well as the nonlinear inverse modeling approach will be presented.

2.1. MODELICA

The object-oriented modeling language MODELICA, introduced in [24, 25, 26, 27], is well suited to model complex physical systems containing, e.g., mechanical, electrical, thermal, control, or process-oriented subsystems and components from multiple physical domains.

In MODELICA, models of complex physical systems are described using differential, algebraic, and discrete equations which are mapped into a mathematical description form called hybrid *Differential Algebraic Equations* (DAE). A DAE system in its implicit form can be expressed as:

$$\boldsymbol{F}\left(\dot{\boldsymbol{x}}(t),\boldsymbol{x}(t),\boldsymbol{y}(t),t\right)=0, \tag{1}$$

where $\dot{\boldsymbol{x}}$ contains the state derivatives, $\boldsymbol{x}$ the state variables, $\boldsymbol{y}$ the pure algebraic variables, and t the time variable [28]. These high-index systems of DAE are reduced to lower index using the *Pantelides* algorithm which identifies the equations to be differentiated and then solved directly by a DAE solver like the differential and algebraic system solver DASSL included in the MODELICA-based simulation environment DYMOLA [29]. Alternatively, the system can be mapped to an explicit *Ordinary Differential Equation* (ODE) form by reordering the derivatives and the algebraic variables, and then subsequently solved numerically by a dedicated ODE solver.

MODELICA is a ***declarative*** language, meaning that declarations are given through equations in contrast to imperative languages, in which statements and algorithms are assigned in explicit steps [13, 15, 23, 27]. These declarations most often describe the model's first-principles without explicit indication on how to compute them; therefore, MODELICA is an ***equation-based*** modeling language. These declarative models can then be translated into efficient code allowing for ***acausal*** modeling capabilities that give better reuse of classes since equations do not specify a certain data flow direction, which is one of the most important features of the language.

MODELICA has ***multi-domain*** modeling capability, meaning that model components corresponding to physical objects from several different domains can be described and connected to each other. The interaction between components is defined by means of physical ports, called connectors, and the interconnection is given accordingly to their physical meaning. This physical meaning is typically represented by *flow variables*, which describe quantities whose values add up to zero in a node connection (Kirchhoff's first rule), and by *non-flow (or potential) variables*, which in contrast remain equal (Kirchhoff's second rule).

Most important for the generation of a modeling framework is, that MODELICA is designed to be an ***object-oriented*** language. This helps to model complex systems and their physical meaning within an object-oriented structure, facilitating the modularization and reuse of component models and the evolution of the structure itself. Object-orientation is primarily used as a ***structuring*** concept which exploits the declarative feature of the language, the adaptiveness of models to changes within the framework structure, as well as the reusability of models in a ***modular*** fashion. Thus, the language is ideally suited as an ***architectural*** description language for complex physical systems.

The *Modelica Association* provides the *Modelica Standard Library* (MSL) [28] containing multi-physical models including multibody components which can be used as base components for individual modeling frameworks.

2.2. Multibody Modeling

Typically, a multibody system is described by a collection of so-called *bodies* and their interactions. Generic body components, as defined in the *Modelica Standard Library*, are represented by their physical properties such as constant mass and moments of inertia, as well as some geometric quantities like position of center of mass, shape, and density, for instance.

The translational and rotational dynamics of each body

are described depending on the physical nature of the system and their components. For rigid body models, the *Newton-Euler* equations of motion with respect to the body fixed coordinate system (B) are implemented by default:

$$\begin{bmatrix} m\boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_B \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_B \\ \boldsymbol{\alpha} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\omega} \times \boldsymbol{I}_B \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{M} \end{bmatrix}, \quad (2)$$

where $\boldsymbol{F}$ is the external force vector resolved in the body fixed coordinate system, m is the constant mass, $\boldsymbol{a}_B$ the translational acceleration, $\boldsymbol{M}$ the external moment vector, $\boldsymbol{I}_B$ the constant inertia matrix with respect to the center of mass of the body, $\boldsymbol{\omega}$ the angular velocity and $\boldsymbol{\alpha}$ the angular acceleration vector, respectively. It is important to notice, that the Equation (2) is based on a point mass which coincides with the center of mass of the multibody, such that any term related to a certain mass distribution with a distance from the center of mass are set to zero.

In MODELICA, the physical coupling between multibody components is described by rigid connections between bodies, defined by *frames* and *joints*. A frame connector is assigned to each multibody and contains information about its resulting cut-force $\boldsymbol{f}$ and cut-torque $\boldsymbol{t}$ as flow variables, as well as its position $\boldsymbol{r}_0$ and orientation $\boldsymbol{R}$ defined by a transformation matrix together with the corresponding angular velocity vector as non-flow variables [28], respectively. These variables are defined with respect to the global (world) coordinate system given in Cartesian coordinates as shown in Figure 1 and can be resolved in problem-specific coordinate systems for better usability or numerical efficiency. Joints are used to model motion constraints between multibody frames, where specific elements can be implemented to describe physical connections via automatic computation of joint loads, or to describe stage separation dynamics where some or all constraints are released, for instance. These aspects are discussed in more detail in [12, 13, 16].

2.3. Nonlinear Inverse Modeling

Forces and torques that are applied directly on a rigid body using the *frame* connector result in a change of its kinematic states. These kinematic states can be obtained as outputs of the overall system dynamics by dedicated measurements. In MODELICA, such a nonlinear direct model can be transformed automatically into a nonlinear inverse model by simply exchanging the meaning of the input and output variables given in Equation 1 as shown in Figure 2 and under some assumptions, described in detail below.

In the direct multibody modeling approach, a torque $\boldsymbol{\tau}$ is applied to the multibody frame in order to exert a rotation with respect to the center of mass. The kinematic states including the attitude $\boldsymbol{\sigma}$ and the angular velocity $\boldsymbol{\omega}$ of the system are then calculated by solving the corresponding DAE. However, if we are interested in the torque required for a given reference angular velocity $\boldsymbol{\omega}_{ref}$, the nonlinear inverse modeling

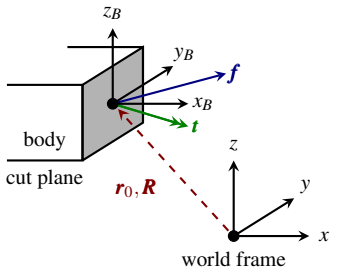


Fig. 1: Schematic Representation of Frames in MODELICA.

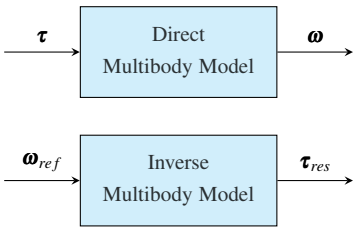


Fig. 2: Schematic Overview of Direct and Inverse Models.

approach can be used. For that, the reference angular velocity $\boldsymbol{\omega}_{ref}$ has to be provided as an input to the system and the required or resulting torque $\boldsymbol{\tau}_{res}$ is then internally derived automatically from the inverse multibody model.

Nonlinear inverse modeling can only be performed under certain assumptions. First of all, the direct multibody model itself must be invertible. Time delays and hard nonlinearities such as backlash, friction, or hysteresis may render a direct multibody model non-invertible, and these must be approximated or avoided for inverse modeling [18, 19]. Moreover, an inverse model must be a unique solution and it must be stable. Furthermore, the inverse model has to be continuously differentiable up to the necessary order of differentiation. If the inverse model is non-differentiable, for example due to externally provided tabular data, the required derivatives have to be provided either manually, approximated, or obtained analytically instead of using MODELICA's automatic symbolic differentiation [19]. Additionally, high-order filters can be used to smooth the inputs and to obtain the required derivatives provided by the filter states as described in [18].

Nonlinear inverse models have been applied for model-based control of industrial robots, aircraft, and satellites, where the inversion including flexible multibody dynamics are also considered [18, 19]. In this paper, nonlinear inverse models are used in the context of launch vehicle preliminary design studies and in terms of obtaining the required moments to follow a given reference trajectory. This step helps to subsequently evaluate the capabilities of the launch vehicle in terms of controllability, actuator sizing, and ΔV budgeting (e.g. for *Reaction Control Systems*).

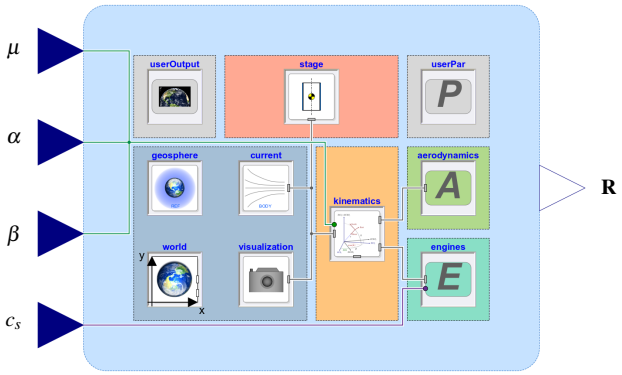
3. MODELING

When modeling an expendable or reusable launch vehicle system, several disciplines as well as design and mission aspects must be considered. Furthermore, launch vehicle models must provide the capabilities to fully operate under atmospheric and deep space conditions, respectively. For that reason, these aspects must be covered by a consistent and highly accurate definition of environmental conditions.

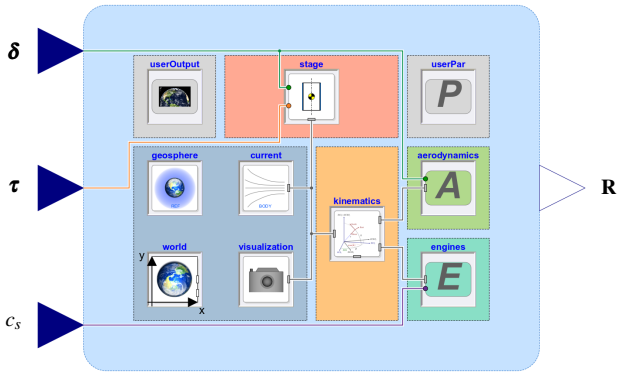
In addition to environmental conditions, the chosen structural design of the launch vehicle system can have significant influence in the modeling approach. For instance, stability and control aspects depend significantly on the chosen structural design. Typically, for conventional launch vehicles the control elements are limited to *Thrust Vector Control* (TVC) and *Reaction Control Systems* (RCS), whereas for winged launch vehicle systems aerodynamic control surfaces have to be also included into the modeling framework.

Based on requirements and constraints of the performed studies, like trajectory optimization, controllability studies, as well as G&C design, the launch vehicle modeling framework presented schematically for a 3-DOF model in Figure 3(a) and a 6-DOF model in Figure 3(b) has been developed using MODELICA. The following modularized, replaceable, extendable and object-oriented components shown in Figure 3 are used representing the main architecture of the launch vehicle modeling framework:

- **world**: The world component provides the global coordinate system and functions to calculate the gravity acceleration of a body for a given inertial position.
- **geosphere**: Within this component atmospheric parameters depending on the chosen atmospheric models for a given inertial position can be determined.
- **current**: In this component, wind forces are calculated with respect to the body frame based on wind profiles depending on the altitude of the body as well as atmospheric parameters.
- **stage**: The stage component is used for the calculation of the main stage dynamics due to gravity acceleration as provided by the *world* component. Whenever necessary, variable mass dynamics are considered by launcher-specific variable mass models as described by Eke [30].
- **kinematics**: Kinematic variables, states and transformation matrices are calculated in the *kinematics* component. This includes the definition of *frames* for any attached components, such as *aerodynamics* or *engines*.
- **aerodynamics**: This component provides aerodynamic forces and moments with respect to the kinematic or body fixed frame of the launch vehicle system.



(a) Schematic Representation of a 3-DOF Model.



(b) Schematic Representation of a 6-DOF Model.

Fig. 3: Overview of Launch Vehicle Models (3-/6-DOF).

- **engines**: Within this component, thrust forces are calculated and applied to the body frame with respect to its orientation or at a derived coordinate system for TVC.
- **userPar**: This component provides a parameter interface for consistent 3-DOF and extended 6-DOF model parameters. For instance, multi-dimensional tabular datasets for the interpolation of aerodynamic coefficients can be considered additionally.
- **userOutput**: This component provides launch vehicle specific parameters, which can be obtained by measurements and used as outputs for attached tools, like the trajectory optimization package '*trajOpt*'.

Except for the environment components *world* and *geosphere*, and the parameter datasets *userPar* and *userOutput*, all model components are connected with each other through 'acausal' physical frames, capitalizing on this modeling feature of MODELICA.

Furthermore, all model components are declared as replaceable models using MODELICA's object-oriented architecture. In that sense, the 3-DOF models can be replaced by 6-DOF models, according to the related application or to particular requirements. This is possible due to the fact that all models supporting several levels of detail have the same interfaces and identical parameter structure. Therefore, they can be replaced by corresponding models with different level of detail, assuming the interface definitions remain unchanged during the transition.

As shown in Figure 3, the connection interfaces between the frames of each model component remain unchanged for the 3-DOF and the 6-DOF cases. Only external inputs, such as aerodynamic angles $\{\mu, \alpha, \beta\}$ or torques $\boldsymbol{\tau}$ and aerodynamic control surface deflections $\boldsymbol{\delta}$, maintain a different connection with the corresponding model components. If other launch vehicle or analysis specific inputs are necessary, these can be extended accordingly. This leads to a highly modular, numerically accurate, flexible, and most importantly *consistent* model behavior.

3.1. Environment

All components related to environmental conditions such as *world*, *geosphere* and *current* are derived from the *DLR Environment Library* [14] and are defined as replaceable models.

As described in [15], the global and inertial coordinate system *Earth Centered Inertial* (ECI) denoted with I and the rotating *Earth Centered Earth Fixed* (ECEF) coordinate system denoted with E are provided by the *world* component. The ECEF coordinate system is derived from the ECI coordinate system by using the Earth's angular velocity ω_E, and the Julian Date to determine the Earth's current rotation angle α_E at a certain initialization time.

The *DLR Environment Library* allows for the modeling of spherical or ellipsoid planet shapes. Depending on the chosen planet shape, the position vector of the launch vehicle with respect to the center of the global coordinate system ECI can be significantly different for latitude, longitude and altitude formulations. These changes have to be considered not only in the calculation of the kinematic states of a launch vehicle, but also for the calculation of the gravity acceleration vector, the altitude-dependent atmospheric parameters, and most importantly, for the calculation of the orientation of the launch vehicle itself.

For example, if the planet is modeled as an ellipsoid, the altitude above the surface is determined by transforming the inertial position of the launch vehicle to its geodetic latitude λ^*, longitude ϕ, and geodetic altitude h^* representation, using the following equation as defined by the *World Geodetic System 1984* (WGS'84) [31]:

$$\boldsymbol{r}_E^* = \begin{bmatrix} (N+h^*)\ \cos\lambda^*\cos\phi \\ (N+h^*)\ \cos\lambda^*\sin\phi \\ (N(1+e^2)+h^*)\ \sin\lambda^* \end{bmatrix}. \tag{3}$$

In this case, the reference position $\boldsymbol{r}_E^*$ of the launch vehicle with respect to the ECEF coordinate system is calculated, where N represents the prime vertical radius of curvature and e the eccentricity of the planet's ellipse. Additionally, a more accurate altitude formulation considering the geoid undulation of the Earth based on the EGM geoid can be used. On the other hand, if the planet is assumed to be spherical, the following equation is used:

$$\boldsymbol{r}_E = \begin{bmatrix} (R_E+h)\cos\lambda\cos\phi \\ (R_E+h)\cos\lambda\sin\phi \\ (R_E+h)\sin\lambda \end{bmatrix}, \tag{4}$$

where R_E is the Earth's mean radius as defined in WGS'84, h the geocentric altitude and λ the geocentric latitude.

Depending on the required accuracy, the gravity acceleration can be calculated assuming either a point gravity formulation, or applying more accurate gravity models such as the GPS-based gravitational models given in the *down*-direction of the *North-East-Down* coordinate system or the *Earth Gravitational Model 1996* (EGM96). However, it is important to use the appropriate definition of geodetic and geocentric parameters for the computation of the gravity acceleration vector to avoid significant deviations between trajectories [32].

For the calculation of atmospheric parameters, the *geosphere* component is used, providing several planet-dependent atmospheric models, like the *International Standard Atmosphere* (ISA) or the *NRL-MSISE-00* model based on experimental data. The *geosphere* model supplies (amongst others) the absolute pressure p, absolute temperature T, atmospheric density ρ and the corresponding speed of sound a from which the Mach number of the launch vehicle can be extracted using the current relative velocity of the vehicle.

The global wind vector as provided by the *geosphere* component, is computed by using a logarithmic boundary layer approach based on the current inertial position of each vehicle. Alternatively, the global wind vector can be substituted with vehicle-specific local wind profiles provided by the *current* component. The wind velocity calculated in this component is based on the *North-East-Down* coordinate system and resolved in the vehicle's body fixed coordinate system. The wind effects as described in [14] are especially important for 6-DOF launch vehicle models, since they are used for the calculation of the effective aerodynamic angles as well as the wind force $\boldsymbol{W}$ provided by the following equation:

$$\boldsymbol{W} = \frac{1}{2}\rho S_{ref}\boldsymbol{v}_W^2, \tag{5}$$

where ρ is the atmospheric density provided by the geosphere component, S_{ref} is the aerodynamic reference area and $\boldsymbol{v}_W$ is the wind velocity vector resolved in the body fixed coordinate system. Furthermore, the *current* component considers turbulence models such as the *Dryden Wind Turbulence Model* [33], accurate wind data based on experiments or common wind profiles superposed with white noise.

3.2. Kinematics & Transformations

Kinematic dependencies between flight coordinate systems used for launch vehicles are schematically shown in Figure 4. In general, the launch vehicle modeling framework has to provide functions and methods to represent the basic dependencies between these coordinate systems, especially with respect to the definition of appropriate model states.

The *North-East-Down* coordinate system, denoted as N, can be derived from the inertial coordinate system ECI by using the Earth's rotation angle α_E and appropriate geodetic or geocentric latitude, longitude, and altitude formulation, as described by the transformation matrices $\boldsymbol{T}_{EI}$ and $\boldsymbol{T}_{NE}$:

$$\boldsymbol{T}_{EI} = \begin{bmatrix} \cos\alpha_E & \sin\alpha_E & 0 \\ -\sin\alpha_E & \cos\alpha_E & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{6a}$$

$$\boldsymbol{T}_{NE} = \begin{bmatrix} -\sin\lambda\cos\phi & -\sin\lambda\sin\phi & \cos\lambda \\ -\sin\phi & \cos\phi & 0 \\ -\cos\lambda\cos\phi & -\cos\lambda\sin\phi & -\sin\lambda \end{bmatrix}. \tag{6b}$$

From this local horizontal and body centered coordinate system, the *kinematic* coordinate system, denoted as K, can be calculated using the flight path angle γ and the flight path azimuth angle χ in correlation with the relative velocity V of the launch vehicle by using the transformation matrix $\boldsymbol{T}_{KN}$:

$$\boldsymbol{T}_{KN} = \begin{bmatrix} \cos\gamma\cos\chi & \cos\gamma\sin\chi & -\sin\gamma \\ -\sin\chi & \cos\chi & 0 \\ \sin\gamma\cos\chi & \sin\gamma\sin\chi & \cos\gamma \end{bmatrix}. \tag{7}$$

In the next step, the launch vehicle is rotated around its longitudinal axis using the aerodynamic bank angle μ. This new coordinate system is prescribed as the *intermediate kinematic* coordinate system and denoted as $\bar{K}$. Subsequently, the transformation into the body fixed coordinate system B can be performed using the aerodynamic angle of attack α and the aerodynamic sideslip angle β. The corresponding transformation matrices are given as:

$$\boldsymbol{T}_{\bar{K}K} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\mu & \sin\mu \\ 0 & -\sin\mu & \cos\mu \end{bmatrix}, \tag{8a}$$

$$\boldsymbol{T}_{B\bar{K}} = \begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix}. \tag{8b}$$

For general 3-DOF models, these transformation matrices are sufficient for defining the overall launch vehicle orientation. The flight path parameters are directly derived from the velocity vector which describe the relationship between the *North-East-Down* coordinate system and the *kinematic* coordinate system. The aerodynamic angles $\{\mu, \beta, \alpha\}$ are retrieved directly from the trajectory optimization results, where

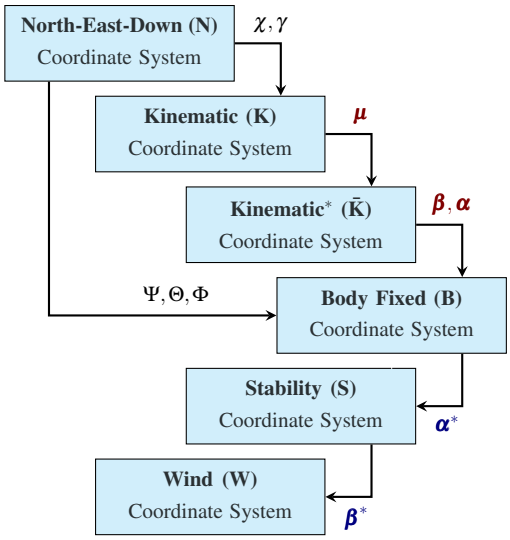


Fig. 4: Overview of Flight Coordinate Systems.

they are defined as the control inputs of the launch vehicle system. For instance, the thrust direction can then be defined with respect to the body fixed coordinate system.

The orientation of the launch vehicle with respect to the *North-East-Down* coordinate system can then be calculated using the *Euler* angle formulation as defined by the body rotation angles Ψ (yaw), Θ (pitch) and Φ (roll) as defined in [34]:

$$\boldsymbol{T}_{BN}[:,1] = \begin{bmatrix} \cos\Psi\cos\Theta \\ -\cos\Phi\sin\Psi + \sin\Phi\cos\Psi\sin\Theta \\ \sin\Phi\sin\Psi + \cos\Phi\cos\Psi\sin\Theta \end{bmatrix}, \tag{9a}$$

$$\boldsymbol{T}_{BN}[:,2] = \begin{bmatrix} \sin\Psi\cos\Theta \\ \cos\Psi\cos\Phi + \sin\Phi\sin\Psi\sin\Theta \\ -\cos\Psi\sin\Phi + \cos\Phi\sin\Psi\sin\Theta \end{bmatrix}, \tag{9b}$$

$$\boldsymbol{T}_{BN}[:,3] = \begin{bmatrix} -\sin\Theta \\ \sin\Phi\cos\Theta \\ \cos\Phi\cos\Theta \end{bmatrix}. \tag{9c}$$

Aerodynamic forces within 6-DOF models are typically applied with respect to the so-called *stability* frame defined by the rotation from the body fixed coordinate system using the effective angle of attack α^*. The effective angle of attack differs from the control inputs given by 3-DOF based trajectory optimization by also considering the influence of the wind velocity vector. This influence on the overall calculation of the aerodynamic angles is covered by subtracting the wind velocity vector $\boldsymbol{v}_W$ present at the vehicle's current location resolved in the body fixed coordinate system from the vehicle's relative velocity vector $\boldsymbol{v}_B$:

$$\boldsymbol{v}_{air} = \boldsymbol{v}_B - \boldsymbol{v}_W, \tag{10}$$

and by using the effective velocity vector $\boldsymbol{v}_{air}$, the effective aerodynamic angles can be calculated as [35, 36]:

$$\alpha^* = \arctan\left(\frac{\boldsymbol{v}_{air,3}}{\boldsymbol{v}_{air,1}}\right), \tag{11a}$$

$$\beta^* = \arcsin\left(\frac{\boldsymbol{v}_{air,2}}{V_{air}}\right), \tag{11b}$$

with $V_{air} = |\boldsymbol{v}_{air}|$ being the effective relative speed of the launch vehicle. Finally, by using the effective aerodynamic angles α^* and β^*, the *wind* coordinate system based on the effective velocity vector $\boldsymbol{v}_{air}$ can be determined.

3.3. State Selection

In MODELICA, state variables are derived from equations describing the system states as discussed in [37]. These state variables have to be differentiable at least once within the given model equations. Typically, the selection of state variables for multibody systems is done automatically by DYMOLA. For instance, the absolute position and velocity vectors are used as translational states, while the absolute angles and angular velocities with respect to the inertial coordinate system are typically used as rotational states.

However, in order to improve accuracy and user-friendliness of the model, these states can be reformulated to enable the calculation of states with respect to dedicated flight coordinate systems. The *DLR Environment Library* [14] provides modularized drag & drop models to enable individual state selection depending on the specific task.

3.3.1. Translational States

For a generic point mass model, these transformations and associated coordinate systems can be derived as follows. We begin with the definition of the position of the launch vehicle with respect to the inertial frame of the planet. Considering the transformation matrix $\boldsymbol{T}_{EI}$, we transform the inertial position $\boldsymbol{r}_I$ of the launch vehicle into its reference position $\boldsymbol{r}_E$ or $\boldsymbol{r}_E^*$, respectively. Consequently, the kinematic position states with respect to the ECEF frame can be denoted as (shown here for the geocentric case):

$$\boldsymbol{r}_G = \begin{bmatrix} \lambda \\ \phi \\ h \end{bmatrix}_G. \tag{12}$$

Denoting V as the relative velocity of the launch vehicle, χ as the flight path azimuth angle and γ as the flight path angle; we may consider the definition of the following velocity state vectors:

$$\boldsymbol{v}_K = \begin{bmatrix} V \\ \gamma \\ \chi \end{bmatrix}_K. \tag{13}$$

However, for the translational state definition, the launch vehicle modeling framework uses the relative velocity $\boldsymbol{v}_N$ of the point mass resolved in the *North-East-Down* coordinate system instead of velocity state vector defined by flight path parameters $\boldsymbol{v}_K$. The velocity state vector $\boldsymbol{v}_N$ is given as:

$$\boldsymbol{v}_N = \begin{bmatrix} v_{north} \\ v_{east} \\ v_{down} \end{bmatrix}_N. \tag{14}$$

This approach results from the constraint that conventional launch vehicles take off vertically. For vertical takeoff and/or landing, the flight path state definition $\boldsymbol{v}_K$ is singular, since the flight path angle points 90° upwards with respect to the *North-East-Down* coordinate system at a relative velocity of $0\ m/s$. Using these initialization parameters, the flight path azimuth angle χ remains undefined up until the start of the pitch over maneuver of the launch vehicle. More importantly, state variables have to be differentiable. The declaration of the vector $\boldsymbol{v}_K$ as states could lead to a potential division by zero in the derivatives.

Therefore, the launch vehicle modeling framework uses the relative velocity vector of the point mass with respect to the coordinate system ECEF and transforms it into a velocity vector $\boldsymbol{v}_N$. Using this method, the singularity condition at initialization of vertical takeoff launch vehicles can be avoided and a user-friendly and easily understandable velocity representation is obtained. The flight path parameters can then be algorithmically calculated based on $\boldsymbol{v}_N$.

For a 3-DOF point mass model of a launch vehicle, we consider only the translational degrees of freedom, where the rotational degrees of freedom – in particular prescribed by the orientation object $\boldsymbol{R}$ – have to be constrained since multibody models within MODELICA are represented by 6-DOF models by default. The reduction from a 6-DOF to a 3-DOF model can be performed utilizing state constraint models provided by the *DLR Environment Library*. This is accomplished, by explicitly setting the angular velocity to zero and creating new unknown variables defined by a default orientation quaternion $\boldsymbol{Q}$. The model then interrupts the usual flow of the calculation – conceptually a double integration, first from torques to rates and then to attitude angles, as described in [14]. Consequently, any rotational movement of the point mass due to external forces is imposed immediately to the body dynamics without considering the influence of the moments of inertia, external torques, or changing angular velocities and accelerations.

3.3.2. Rotational States

In addition to the previously shown translational states, rotational states can be defined by models provided in the *DLR Environment Library*. For the attitude states $\boldsymbol{\sigma}_B$, we use *Euler* angles. Basically, we obtain the kinematic states by application of a chain of transformations from the *North-East-Down*

coordinate system to the body fixed coordinate system:

$$\boldsymbol{\sigma}_B = \begin{bmatrix} \Phi \\ \Theta \\ \Psi \end{bmatrix}_B . \tag{15}$$

Obviously, in the 6-DOF case, special care has to be taken in terms of singularities occurring at $\theta = 90°$, which cannot be avoided in the same way as described for the flight path parameters. For this reason, the quaternion approach can be preferable if the numerical integration can be guaranteed to be accurate using higher-order integration methods. The relationship between the attitude kinematics and the quaternion formulation is described in [34]:

$$\boldsymbol{q} = \pm \begin{bmatrix} \cos\frac{\Phi}{2}\cos\frac{\Theta}{2}\cos\frac{\Psi}{2} + \sin\frac{\Phi}{2}\sin\frac{\Theta}{2}\sin\frac{\Psi}{2} \\ \sin\frac{\Phi}{2}\cos\frac{\Theta}{2}\cos\frac{\Psi}{2} - \cos\frac{\Phi}{2}\sin\frac{\Theta}{2}\sin\frac{\Psi}{2} \\ \cos\frac{\Phi}{2}\sin\frac{\Theta}{2}\cos\frac{\Psi}{2} + \sin\frac{\Phi}{2}\cos\frac{\Theta}{2}\sin\frac{\Psi}{2} \\ \cos\frac{\Phi}{2}\cos\frac{\Theta}{2}\sin\frac{\Psi}{2} - \sin\frac{\Phi}{2}\sin\frac{\Theta}{2}\cos\frac{\Psi}{2} \end{bmatrix}, \tag{16}$$

where $\boldsymbol{q} = [q_0, q_x, q_y, q_z]^T$ is the *Euler-Rodrigues* quaternion, relating its four elements to the *Euler* angles, and which requires the quaternion $\boldsymbol{q}$ to be a unit quaternion as defined by

$$q_0^2 + q_x^2 + q_y^2 + q_z^2 = 1. \tag{17}$$

Using the quaternion $\boldsymbol{q}$ as rotational states, the singularity can be avoided and the *Euler* angles can then be retrieved algorithmically as described in [34] by

$$\boldsymbol{\sigma}_B = \begin{bmatrix} \operatorname{atan2}\left(2(q_0 q_x + q_y q_z), (q_0^2 + q_z^2 - q_x^2 q_y^2)\right) \\ \arcsin\left(2(q_0 q_y - q_x q_z)\right) \\ \operatorname{atan2}\left(2(q_0 q_z + q_x q_y), (q_0^2 + q_x^2 - q_y^2 q_z^2)\right) \end{bmatrix}. \tag{18}$$

Since the reusable launch vehicle presented in Section 5 is mainly analyzed in design regimes where the *Euler* angles are not expected to be singular, the quaternion modeling approach will be avoided and is therefore out of scope of this paper.

Complementing the attitude states, the angular velocity state vector is defined using the angular rates $\boldsymbol{\omega}_B$ resolved in the body fixed coordinate system (B):

$$\boldsymbol{\omega}_B = \begin{bmatrix} p \\ q \\ r \end{bmatrix}_B . \tag{19}$$

The kinematic relationship between the angular velocity vector $\boldsymbol{\omega}_B$ and the *Euler* angle rates are automatically derived using the kinematic equations defined in [34, 38]:

$$\begin{aligned} \boldsymbol{\omega}_B &= \begin{bmatrix} \dot{\Phi} \\ 0 \\ 0 \end{bmatrix} + \boldsymbol{T}_\Phi \left(\begin{bmatrix} 0 \\ \dot{\Theta} \\ 0 \end{bmatrix} + \boldsymbol{T}_\Theta \begin{bmatrix} 0 \\ 0 \\ \dot{\Psi} \end{bmatrix} \right) \\ &= \begin{bmatrix} 1 & 0 & -\sin\Theta \\ 0 & \cos\Phi & \sin\Phi\cos\Theta \\ 0 & -\sin\Phi & \cos\Phi\cos\Theta \end{bmatrix} \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix}. \end{aligned} \tag{20}$$

Alternatively, the reference angular velocity $\boldsymbol{\omega_B}$ can be obtained by resolving the derivatives of the aerodynamic angles and flight path parameters with respect to the body fixed coordinate system, following [22] for a flat Earth approach:

$$\begin{aligned} \boldsymbol{\omega_B} = &\begin{bmatrix} \cos\alpha\cos\beta & 0 & \sin\alpha \\ \sin\beta & 1 & 0 \\ \sin\alpha\cos\beta & 0 & -\cos\alpha \end{bmatrix} \begin{bmatrix} \dot{\mu} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} \\ &+ \boldsymbol{T}_{B\bar{K}} \boldsymbol{T}_{\bar{K}K} \begin{bmatrix} -\dot{\chi}\sin\gamma \\ \dot{\gamma} \\ \dot{\chi}\cos\gamma \end{bmatrix} \end{aligned} \tag{21}$$

In particular, the launch vehicle modeling framework can additionally consider the influence of a rotating spherical or ellipsoid Earth. The cross-couplings between translational and rotational degrees of freedom are contained in the derivatives of the flight path parameters.

3.4. Dynamics

For modeling of variable mass systems, the standard *Newton-Euler* equations of motion (2) are augmented with additional terms related to and described by Kane's equations as obtained in [30, 39]. Assuming that the particle movement and whirling motion within the body are neglected and that the mass distribution as well as mass loss is axis-symmetric, the forces and moments can be extended as follows:

$$m\boldsymbol{a}_B = \boldsymbol{F} + \boldsymbol{F}_{C2} + \boldsymbol{F}_T, \tag{22a}$$

$$\boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} = \boldsymbol{M} + \boldsymbol{M}_{C1} + \boldsymbol{M}_{C2}. \tag{22b}$$

In these equations, the force $\boldsymbol{F}_{C2}$ is referred to as the Coriolis force of the particles and $\boldsymbol{F}_T$ represents the thrust vector which is calculated using the rate at which the relative linear momentum is lost across the boundary of the body. Thus, the simplified thrust vector $\boldsymbol{F}_T$ is defined by the propellant mass flow rate which is the derivative of the time-dependent mass m:

$$\boldsymbol{F}_T = -\frac{\mathrm{d}m}{\mathrm{d}t}\boldsymbol{v}_r, \tag{23}$$

with $\boldsymbol{v}_r$ being the relative velocity vector. Additionally, $\boldsymbol{M}_{C2}$ is related to jet damping as part of the Coriolis Effect and $\boldsymbol{M}_{C1}$

contains the contribution of the time-dependent inertia matrix $\boldsymbol{I}_B$ defined in:

$$\boldsymbol{M}_{C1} = -\frac{\mathrm{d}\boldsymbol{I}_B}{\mathrm{d}t}\boldsymbol{\omega}. \tag{24}$$

If the mass of the system remains constant, the variable mass dynamics are neglected. Consequently, the additional forces and moments above are set to zero, and the *Newton-Euler* equations of motion for a rigid body are recovered. In general, depending on the nature of the propulsion system and its corresponding burn profiles, these terms can be further simplified and implemented in closed form as described in [13, 19, 30].

Regarding 3-DOF point mass models used for trajectory optimization, for instance, it can be sufficient to model variable mass systems based on the mass derivative as provided by the launch vehicle propulsion components' specifications [15]. Regarding 6-DOF models, the full influence of variable mass dynamics can be considered.

3.4.1. Translational Dynamics

For translational dynamics, the external forces consist of the gravity vector $\boldsymbol{F}_G$ applied on the bodies' center of mass, the aerodynamic forces $\boldsymbol{F}_A$ containing drag D, side force Y, and lift L, and the thrust $\boldsymbol{F}_T$:

$$\boldsymbol{F} = \boldsymbol{F}_G + \boldsymbol{F}_A + \boldsymbol{F}_T\ . \tag{25}$$

More explicitly, the forces resolved in the indicated coordinate systems are defined as:

$$\boldsymbol{F}_G = \begin{bmatrix} mg_x \\ mg_y \\ mg_z \end{bmatrix}_N, \boldsymbol{F}_A = \begin{bmatrix} -D \\ Y \\ -L \end{bmatrix}_S, \boldsymbol{F}_T = \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix}_B, \tag{26}$$

where $\boldsymbol{g} = [g_x, g_y, g_z]^T$ is the gravity acceleration vector provided by the *world* component. The gravity acceleration vector is defined by the gravity models described in Section 3.1. If the gravity acceleration is assumed to point downwards with respect to the *North-East-Down* coordinate system, then the elements in x and y direction are set to zero (see [15, 32]).

Equation (27) shows the simplified equations of motion for the translational dynamics of a 3-DOF point mass defined for the derivatives of the flight path parameters without considering the effect of side forces and aerodynamic sideslip angle [40] :

$$\dot{V} = \frac{1}{m}\Big[T\cos\alpha - D - G\sin\gamma\Big], \tag{27a}$$

$$\dot{\gamma} = \frac{1}{mV}\Big[(T\sin\alpha + L)\cos\mu - G\cos\gamma\Big], \tag{27b}$$

$$\dot{\chi} = \frac{1}{mV\cos\gamma}\Big[(T\sin\alpha + L)\sin\mu\Big]. \tag{27c}$$

3.4.2. Rotational Dynamics

For rotational dynamics, we consider the aerodynamic moments $\boldsymbol{M}_A$ applied on the bodies' center of mass as external moments including the roll moment l, the pitch moment m, the yaw moment n, and additional moments $\boldsymbol{M}_C$ provided for example by the *Reaction Control System*. These generic additional moments are applied directly to the bodies' center of mass. Obviously, if the thrust vector is not aligned with the x-axis of the body fixed coordinate system (B), then the external moments also considers a thrust moment $\boldsymbol{M}_T$ which corresponds to the cross product of the thrust force with its lever arm.

$$\boldsymbol{M} = \boldsymbol{M}_A + \boldsymbol{M}_C + \boldsymbol{M}_T \tag{28}$$

Consequently, the simplified rotational equations of motion are given by an extended version of [38] without considering the Coriolis effect, as mentioned in Section 2.2:

$$\begin{aligned} d_x\dot{p} =& (\dot{I}_{xz}I_{xz} - \dot{I}_{xx}I_{zz})p + (\dot{I}_{zz}I_{xz} - \dot{I}_{xz}I_{zz})r + \\ & (-I_{xx}I_{xz} + I_{yy}I_{xz} - I_{zz}I_{xz})pq + (I_{yy}I_{zz} - I_{zz}^2 - I_{xz}^2)qr + \\ & (l + M_{C,x})I_{zz} - (n + M_{C,z})I_{xz}, \end{aligned} \tag{29a}$$

$$d_y\dot{q} = (m + M_{C,y}) - \dot{I}_{yy}q + (I_{zz} - I_{xx})pr + (p^2 - r^2)I_{xz}, \tag{29b}$$

$$\begin{aligned} d_x\dot{r} =& (\dot{I}_{xx}I_{xz} - \dot{I}_{xz}I_{xx})p + (\dot{I}_{xz}I_{xz} - \dot{I}_{zz}I_{xx})r + \\ & (I_{xz}I_{zz} + I_{xz}I_{xx} - I_{yy}I_{xz})qr + (I_{xz}^2 + I_{xx}^2 - I_{xx}I_{yy})pq + \\ & (n + M_{C,z})I_{xx} - (l + M_{C,x})I_{xz}, \end{aligned} \tag{29c}$$

with $\boldsymbol{I}_B$ being the inertia matrix with respect to the body fixed coordinate system assuming that $I_{xy} = I_{yz} = 0$ and $I_{xz} = I_{zx}$:

$$\boldsymbol{I}_B = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{zx} & 0 & I_{zz} \end{bmatrix}, \tag{30}$$

and the coefficients of the determinant of the inertia matrix defined as:

$$d_x = I_{xx}I_{zz} - I_{xz}^2, \tag{31a}$$

$$d_y = I_{yy}. \tag{31b}$$

3.4.3. Aerodynamics

The aerodynamic forces and moments acting on a launch vehicle during flight are considered as external forces and moments, which are calculated using the following equations [35]:

$$\boldsymbol{F}_A = \begin{bmatrix} -D \\ Y \\ -L \end{bmatrix} = \frac{1}{2}\rho V^2 S_{ref} \begin{bmatrix} -c_D \\ c_Y \\ -c_L \end{bmatrix}, \tag{32}$$

and

$$\boldsymbol{M}_A = \begin{bmatrix} l \\ m \\ n \end{bmatrix} = \frac{1}{2}\rho V^2 S_{ref} \begin{bmatrix} l_{ref}c_l \\ b_{ref}c_m \\ l_{ref}c_n \end{bmatrix}. \tag{33}$$

Here, ρ is the atmospheric density provided by the *geosphere* component, V the relative velocity of the vehicle, S_{ref} the aerodynamic reference area, l_{ref} the aerodynamic reference length, and b_{ref} the aerodynamic reference width or chord length. If wind effects are considered, the relative velocity is substituted by the effective relative velocity.

The aerodynamic coefficients denoted by $c_{(\cdot)}$ can be calculated based on either multi-dimensional look-up tables or approximate analytical expressions. The aerodynamic coefficients used for the studies presented in this paper were provided by the *Space Launcher System Analysis* (SART) department of DLR Institute of Space Systems (DLR-RY) using internal tools such as CAC [41] and HOTSOSE [42], which share similarities to the tool DATCOM as described in [43]. These coefficients are obtained for the overall launch vehicle operating regime depending on the current Mach number, aerodynamic angle of attack α and the aerodynamic control surface deflections $\boldsymbol{\delta}$:

$$\boldsymbol{\delta} = \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix}, \tag{34}$$

where δ_a is the aileron deflection angle, δ_e the elevator deflection angle, and δ_r the rudder deflection angle.

Usually, the full aerodynamic coefficient matrix covering all dependencies between the aerodynamic angles, angular velocities and aerodynamic control surface deflections should be provided for accurate 6-DOF analyses. However, in preliminary design studies, this is not possible due to limitations of the tools used or simplifying assumptions, so that the available coefficient matrix in this paper reduces to:

$$\boldsymbol{c}_A = \begin{bmatrix} c_D \\ c_Y \\ c_L \\ c_l \\ c_m \\ c_n \end{bmatrix} = \begin{bmatrix} c_{D,\alpha} + c_{D,\delta_a} + c_{D,\delta_e} + c_{D,\delta_r} \\ c_{Y,\delta_r} \\ c_{L,\alpha} + c_{L,\delta_a} + c_{L,\delta_e} \\ c_{l,\delta_a} \\ c_{m,\alpha} + c_{m,\delta_a} + c_{m,\delta_e} \\ c_{n,\delta_r} \end{bmatrix}, \tag{35}$$

where the coefficients $c_{D,\alpha}, c_{L,\alpha}$, and $c_{m,\alpha}$ are given for the whole launch vehicle and depend solely on the angle of attack α and the Mach number. The other coefficients are dependent on the angle of attack α, the Mach number, and the corresponding aerodynamic control surface deflection angles.

For the winged reusable launch vehicle in Section 5 only two types of aerodynamic control surfaces are considered: wing flaps for pitch and roll control with a limited deflection angle range and winglet-type fins for yaw control. Within the launch vehicle modeling framework, the direction of these deflection angles are designed using common sign conventions.

Since the wing flaps are considered as ailerons and elevators simultaneously, the corresponding deflections have to be superimposed. Thus, two additional parameters are defined, representing the resulting deflections at the left flap δ_{fL} and the right flap δ_{fR}:

$$\delta_{fL} = \delta_e + \delta_a\,, \tag{36a}$$

$$\delta_{fR} = \delta_e - \delta_a\,. \tag{36b}$$

For each flap position the total aerodynamic coefficients can be retrieved separately so that the superimposed aerodynamic coefficients can be defined using their average values:

$$c_L = \frac{1}{2}\left(c_{L,\delta_{fL}} + c_{L,\delta_{fR}}\right), \tag{37}$$

where the coefficients $c_{L,\delta_{fL}}$ and $c_{L,\delta_{fR}}$ are interpolated using multi-dimensional tables considering angle of attack, Mach number and the corresponding deflection angles.

The roll moment coefficient depending on the aileron deflection angles have to be calculated using the resulting lift coefficients for each flap deflection because they cannot be obtained directly by the available preliminary design aerodynamic tools. Since lift coefficients are defined with respect to the aerodynamic reference length of the vehicle, the resulting roll moment coefficient has to be divided by the aerodynamic reference length l_{ref} and multiplicated with its aerodynamic reference width b_{ref} defined by the distance from the center of the vehicle to the position of the flaps. The resulting equation is:

$$c_{l,\delta_a} = \frac{1}{2}\frac{b_{ref}}{l_{ref}}\left((c_{L,\delta_{fL}} - c_{L,\delta_e}) + (c_{L,\delta_{fR}} - c_{L,\delta_e})\right). \tag{38}$$

The aerodynamic forces are applied with respect to the stability coordinate system at the center of pressure. The aerodynamic moments are applied at the center of gravity of the body with respect to the body fixed coordinate system.

4. IMPLEMENTATION

This section presents the launch vehicle modeling and simulation framework. The framework itself covers several types of modeling methods and levels of detail, in particular for the usage within preliminary design studies of expendable or reusable launch vehicles as shown in Figure 3.

4.1. Trajectory Optimization

The 3-DOF launch vehicle model shown in Figure 3(a) can be translated into a so-called *Functional Mock-up Unit* (FMU), which can then be imported into DLR-SR's Trajectory Optimization Package '*trajOpt*' [15, 17, 44, 45]).

As results of the trajectory optimization, the translational states, flight path variables and aerodynamic control parameters can be obtained directly. The results depend highly on problem-specific optimization goals, requirements, and constraints. These goals can include maximizing the payload to a desired orbit or maximizing the downrange for the

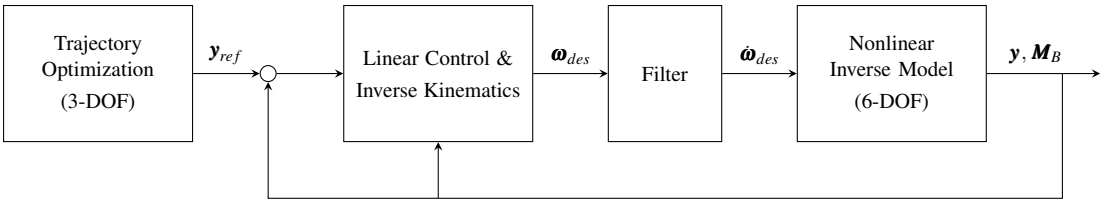


Fig. 5: Intermediate 6-DOF Modeling Approach with Nonlinear Inverse Models.

descent vehicle while minimizing accelerations and dynamic pressure, and thus mechanical and thermal loads, for instance.

In that sense, the reference trajectory provides the guidance commands in terms of position and flight path variables, that have to be tracked by the attitude control subsystem. Some aspects related to G&C in terms of guidance command generation and nonlinear dynamic inversion attitude control are discussed further in [23]. This paper focuses more on an intermediate modeling and simulation approach for actuator budgeting and sizing within preliminary design studies, by providing knowledge about the moments $\boldsymbol{M}_B$ which have to be generated in order to follow the reference attitude $\boldsymbol{y}_{ref}$.

4.2. Nonlinear Inverse Model

In preliminary design studies, control efforts are of major importance for the sizing of actuators. This applies particularly to the moments that have to be delivered by the actuators like the *Reaction Control System* (RCS), *Thrust Vector Control* (TVC), and the moments generated by the aerodynamic control surfaces. To study the required moments, we propose an intermediate 6-DOF modeling approach based on nonlinear inverse models, which uses the reference attitude commands obtained by the trajectory optimization based on a simplified 3-DOF model considering only thrust, lift and drag forces. From a consistent 6-DOF model a nonlinear inverse model is derived and used to identify the moments required by the launch vehicle system to follow the requested reference attitude commands. It is important to notice, that this approach does not consider any control allocation algorithm to assign the moments to appropriate actuators. The simplified workflow for such an approach is displayed in Figure 5.

First, a trajectory optimization as described in the previous section is performed offline, resulting in optimal guidance commands represented by the aerodynamic angle of attack α, the sideslip angle β and the bank angle μ. These angles are used as reference values for the subsequent computations. The thrust throttle factor c_S is not included in the control structure and thus, is provided to the 6-DOF inverse model directly without any modification.

Subsequently, the desired angular velocity with respect to the body fixed coordinate system is calculated using a feedback control loop (prescribing desired dynamics) together with a kinematic inversion of Equation (21) as shown in [22, 23]. The inverse kinematics requires the aerodynamic angles and accelerations resulting from the nonlinear inverse model in the feedback loop.

Finally, a low-pass second-order filter is used to smooth the angular velocity vector provided by the inverse kinematics and to obtain the derivative (angular acceleration) that is required by the nonlinear 6-DOF inverse model. Using such a filter implies imposing a certain cut-off frequency or bandwith which can be related to the limits of the actuators or to some performance requirements.

5. RESULTS

The reusable launch vehicle concept AURORA shown in Figure 6 has been investigated at DLR-RY [4, 5], including iterative studies regarding mass budget, propulsion, aerodynamics and structural optimization. It has been further studied at DLR-SR [15, 23] regarding trajectory optimization and G&C design. AURORA is based on a delta-winged, two stage to orbit (TSTO) concept providing a high lift-to-drag ratio for horizontal takeoff and horizontal landing maneuvers (HTHL). The proposed propellant combination is LOX/Kerosene allowing placement of the kerosene tanks in the wing structure. The upper stage (US) is located inside the main stage (MS) and is released at separation time for further ascent to the target orbit.

The mission phases are defined in Table 1, where Phases 1 to 3 represent the ascent phase of the overall launch vehicle, Phase 4 the ascent of the upper stage including payload, and finally Phase 5 the unpowered return maneuver and subsequent flight of the main stage to the landing site. Phase 5 can further be subdivided into three characteristic flight phases based on the overall control strategy:

- Phase 5a: Reaction Control System (RCS)
- Phase 5b: Combined Control (RCS + ACS)
- Phase 5c: Aerodynamic Control Surfaces (ACS)

This paper focuses mainly on Phase 5a and 5c for the single actuator type control strategies mainly to avoid control allocation issues not considered in the intermediate modeling approach using nonlinear inverse models. Since the flight in Phase 5 is completely unpowered, the mass and inertia of the overall system are both assumed to be constant, so that the variable mass dynamics as defined in the Equations 22a and 22b can be neglected.

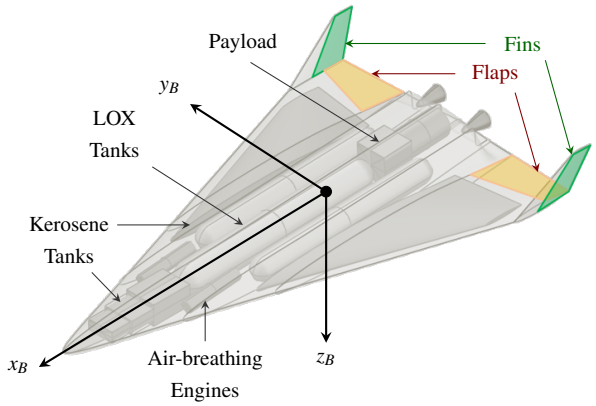


Fig. 6: Overview of the Reusable Launch Vehicle AURORA.

Table 1: Overview of the Mission Phases of AURORA.

Phases	Stages	Description
Phase 1	US+MS	Horizontal liftoff
Phase 2	US+MS	Ascent phase (rocket engines)
Phase 3	US+MS	Ballistic phase & separation
Phase 4	US	Ascent of the upper stage
Phase 5	MS	Unpowered flight to landing site

The mass and inertia matrix, propulsion specifications, and the aerodynamic database used within our studies were provided by DLR-RY using internal tools, like CAC and HOTSOSE. The reference trajectory was obtained by the trajectory optimization package '*trajOpt*' using 3-DOF models.

5.1. Discussion

In this paper, the required moments needed for subsequent controllability studies are presented in terms of 3-DOF and 6-DOF inverse models. The intermediate modeling approach using nonlinear inverse modeling described in Section 4 is discussed to obtain the required moments for the *Reaction Control System* used in Phase 5a. The same modeling approach is applied to Phase 5c in order to estimate the required moments while assuming the aerodynamic surface deflection angles to be zero. The internally calculated aerodynamic moments and the additional moments obtained by the nonlinear inverse model are compared with the ones obtained by the trajectory tracking using nonlinear dynamic inversion control as described in [23]. Finally, the influence of wind effects on the system dynamics of the winged reusable launch vehicle AURORA are shown for Phase 5c.

Even though AURORA is located within the atmospheric design regime at the end of Phase 5a, it is assumed for this paper, that the re-entry maneuver is completely performed using only its *Reaction Control System*. In Figure 7, the results for Phase 5a obtained by the nonlinear inverse modeling approach presented in Figure 5 are shown. The reference trajectory is depicted by the black dotted lines, while the output of the nonlinear inverse model is given by the colored solid lines. As can be seen for the geodetic altitude, flight path velocity, flight path angle, accelerations, and aerodynamic angles; the reference trajectory can be tracked by the inverse model accurately. The angular velocities with respect to the body fixed coordinate system are calculated by the inverse kinematics and applied directly to the nonlinear inverse model. The state derivatives required for the internal DAE index reduction used in the inverse model are taken from the filter.

The required moments needed to follow the reference trajectory provided by the offline trajectory optimization results are obtained by the nonlinear inverse model. These moments can be further considered for sizing and positioning of the *Reaction Control System*. For instance, the required propellant mass of the *Reaction Control System* can be calculated, which in turn can be used in subsequent trajectory optimization within iterative mission analyses.

In Phase 5c, the winged launch vehicle AURORA is controlled by its aerodynamic control surfaces per definition. However, in the nonlinear inverse modeling approach, we assume that the aerodynamic control surfaces are inactive – and thus, the aerodynamic control surface deflections remain at zero. In that sense, we basically obtain the required moments for the overall launch vehicle without aerodynamic control surfaces while still considering the influence of the full 6-DOF aerodynamic coefficient matrix.

For the control allocation approach presented in [23], the externally required moments in Phase 5c are set to zero and instead the aerodynamic control surface deflections are controlled in order to generate the corresponding aerodynamic moments to track the reference trajectory. In that sense, the nonlinear inverse model provides information about the aerodynamic moments ($\boldsymbol{\delta} = \mathbf{0}$) and the required moments assumed to be provided by additional actuators. If both of these values are superimposed as depicted by the diamond shaped green line in Figure 8 (ΔM_y), these values result in the aerodynamic moments realized by the controller in [23].

Even though the aerodynamic multi-dimensional look-up tables might not be invertible per definition, they are still considered in the computation of the aerodynamically required moments using the nonlinear inverse modeling approach. Consequently, the actuators can be sized to fulfil the additional aerodynamic moments using a reverse engineering approach based on the aerodynamic coefficients given by the multi-dimensional look-up tables. The resulting aerodynamic deflections to generate these aerodynamic moments are shown in Figure 8 based on [23].

Figure 9 presents the results obtained by the nonlinear inverse models considering wind effects. The wind velocity is given by using standard wind profiles with respect to the *North-East-Down* frame in East-West direction. Additionally,

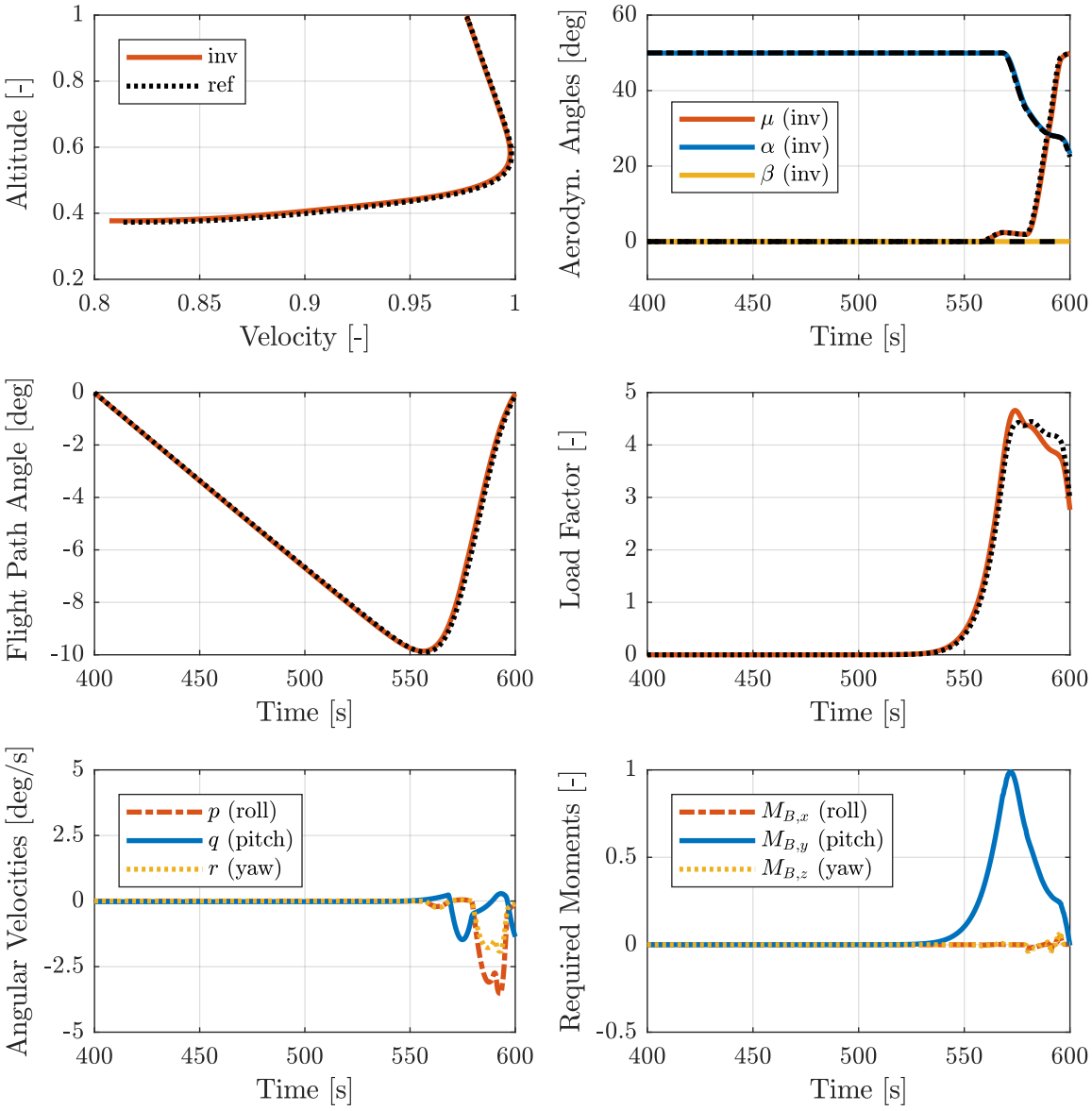


Fig. 7: Phase 5a: Flight Parameters based on the Nonlinear Inverse Modeling Approach (inv) and its Reference Trajectory (ref).

turbulence models (e.g. white noise models) can be superimposed with these wind profiles. Here, turbulence effects are neglected because the focus is set on the evaluation of wind velocities on the overall flight system dynamics as a perturbation factor.

For this purpose, the wind profiles are multiplicated by scaling factors in the range of $[-5,5]$ to provide a wide range of wind profiles with changing wind directions (from East to West, or from West to East). As shown in Figure 9, the perturbation on the system caused by wind velocities, have a major impact on the calculation of the effective aerodynamic angles. Since the aerodynamic angles are used in the feedback control loop based on Figure 5, and subsequently in the inverse kinematics, the required pitch rates differ significantly. Figure 9 shows, that wind disturbance not only influences the effective aerodynamic angles, but also has major effect for the altitude vs. velocity behavior of the launch vehicle, and the moments required for tracking the attitude reference trajectory.

6. CONCLUSION

The objective of this paper was to present an integrated and multidisciplinary modeling and simulation framework for flight dynamics, guidance, and control activities in support of preliminary design efforts of reusable launch vehicles studied at DLR.

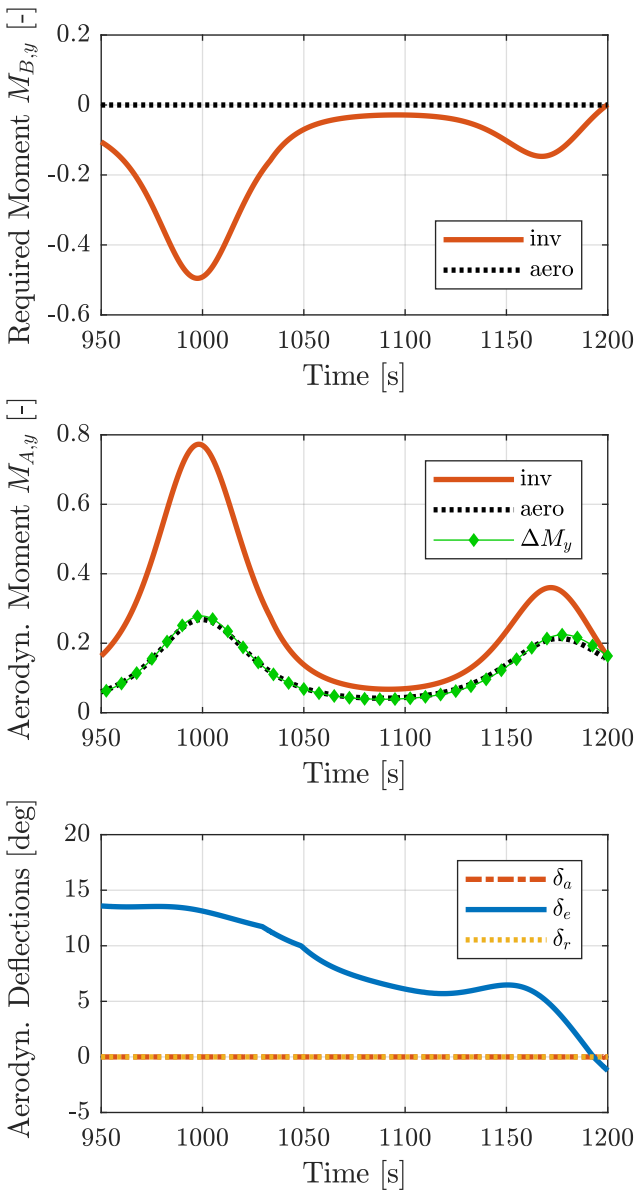


Fig. 8: Phase 5c: Comparison between Moments obtained by the Nonlinear Inverse Modeling Approach (inv) and Nonlinear Dynamic Inversion Control (aero).

The modeling framework, based on the object-oriented and equation-based modeling language MODELICA, takes into account highly interconnected disciplines, atmospheric and space environmental conditions, and wide flying envelopes covering ascent and descent phases.

To demonstrate the benefits of this approach, the AURORA reusable launch vehicle concept was investigated in the context of the methods presented here. Dedicated 3-DOF and 6-DOF vehicle dynamic models are described and implemented covering the kinematics and dynamics formulation, environmental effects, aerodynamics, and propulsion models for system dynamics analyses, trajectory simulations, and preliminary G&C studies in terms of vehicle controllability. The presented framework establishes a connection between 3-DOF and 6-DOF analyses allowing the generation of 6-DOF model control inputs for accurately tracking 3-DOF trajectories. These results can then be used in further studies regarding controllability and stability issues.

Flight simulations show that this nonlinear inverse model approach can account for the required moments to accurately track a reference trajectory, while providing the required aerodynamic moments to be produced by the aerodynamic control surfaces even if an invertible aerodynamic model is not yet available. Other external perturbations and the effect of parametric and modeling uncertainties are the subject of future work.

Acknowledgements

This study was performed within the DLR projects AKIRA and X-TRAS regarding preliminary system studies and evaluation of key technologies for future reusable launch vehicles. The authors would like to thank DLR colleagues from these projects, in particular colleagues from the *Space Launcher System Analysis* (SART) department of the DLR Institute of Space Systems (DLR-RY) in Bremen, for providing data for the considered reusable launch vehicle configuration.

7. REFERENCES

[1] M. Sippel, E. Dumont, and I. Dietlein, "Investigations of Future Expendable Launcher Options," in *Proceedings of the 63rd International Astronautical Congress (IAC)*, 2011.

[2] M. Sippel, S. Stappert, and L. Bussler, "Systematic Assessment of a Reusable First-stage Return Options," in *Proceedings of the 68th International Astronautical Congress (IAC)*, 2017.

[3] L. Bussler and M. Sippel, "Comparison of Return Options for Reusable First Stages," in *21st AIAA International Space Planes and Hypersonics Technologies Conference*, 2017.

[4] A. Kopp, M. Sippel, S. Stappert, N. Darkow, J. Gerstmann, S. Krause, D. Stefaniak, M. Beerhorst, T. Thiele, A. Gülhan, R. Kronen, K. Schnepper, L. E. Briese, and J. Riccius, "Forschung an Systemen und Technologien für wiederverwendbare Raumtransportsysteme im DLR-Projekt AKIRA," *Deutscher Luft- und Raumfahrtkongress*, 2017.

[5] A. Kopp, "Das Aurora-R2 RLV-Konzept," *Deutscher Luft- und Raumfahrtkongress*, 2017.

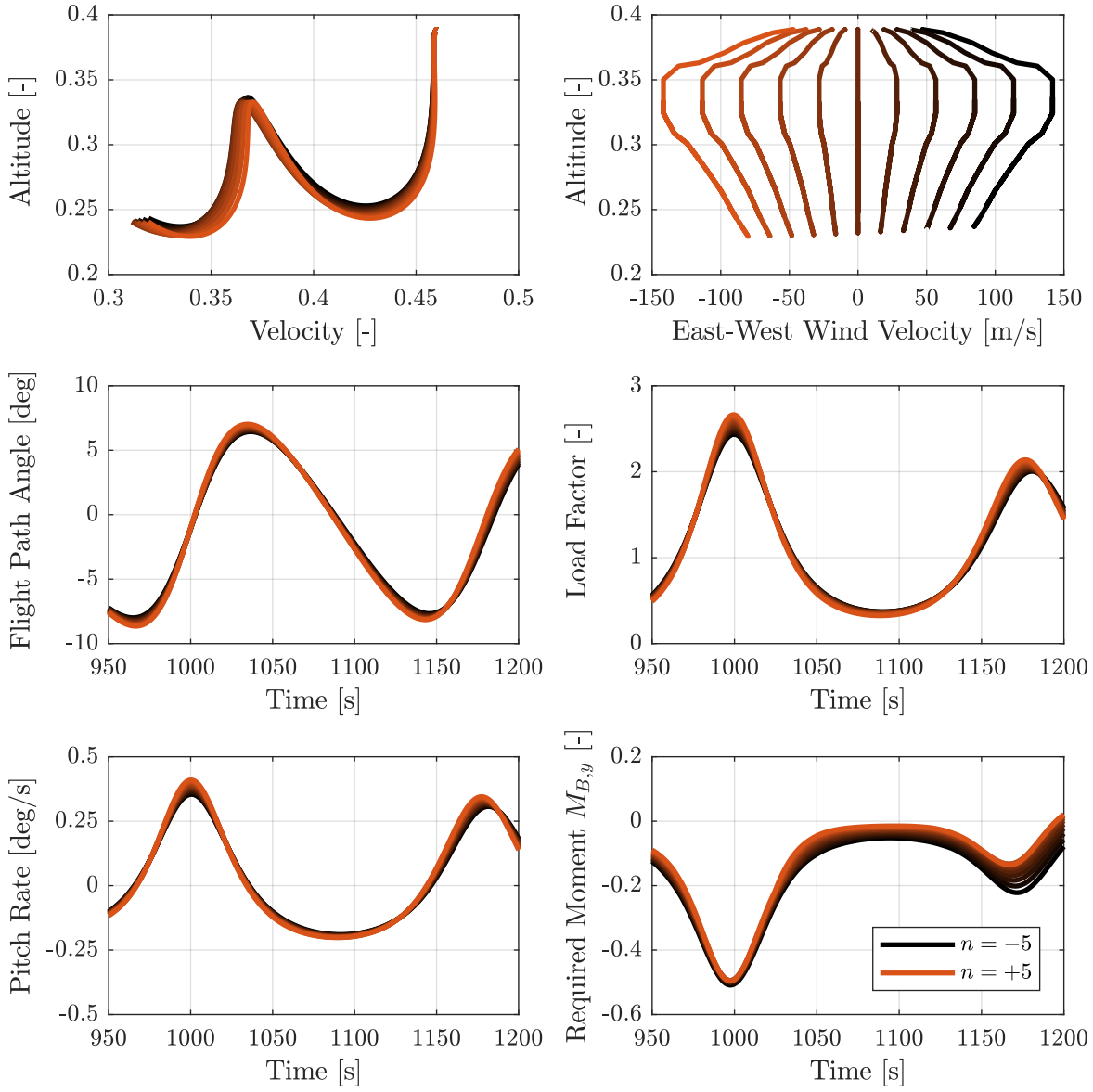


Fig. 9: Phase 5c: Wind-Dependent Flight Parameters based on the Nonlinear Inverse Modeling Approach (inv).

[6] S. A. Striepe, R. W. Powell, P. N. Desai, E. M. Queen, G. L. Brauer, D. E. Cornick, D. W. Olson, F. M. Peterson, R. Stevenson, M. C. Engel, S. M. Marsh, and A. M. Gromko, "Program to Optimize Simulated Trajectories (POST II), Volume 2: Utilization Manual," Tech. Rep., NASA, 2002.

[7] G. Baldesi and M. Toso, "ESA Launcher Flight Dynamics Simulator used for System and Subsystem Level Analyses," in *Proceedings of the 11th International Workshop on Simulation & EGSE facilities for Space Programmes (SESP 2010)*, 2010.

[8] G. Baldesi and M. Toso, "European Space Agency's Launcher Multibody Dynamics Simulator used for System and Subsystem Level Analyses," in *CEAS Space Journal*, 2012.

[9] Jebb S. Orr, John H. Wall, Tannen van Zwieten, and Charles E. S. Hall, "Space Launch System Ascent Flight Control Design," *2014 American Astronautical Society (AAS) Guidance, Navigation, and Control Conference*, January 2014.

[10] A. L. Bowes, J. L. Davis, S. Dutta, S. A. Striepe, M. C. Ivanov, R. W. Powell, and J. White, "LDSD POST2 Simulation and SFDT-1 Pre-flight Launch Operations

Analyses," *Advances in the Astronautical Sciences Spaceflight Mechanics*, vol. 155, 2015.

[11] M. Toso and V. Rossi, "ESA multibody tool for launchers and spacecrafts: lesson learnt and future challenges," *The 5th Joint International Conference on Multibody System Dynamics*, 2018.

[12] P. Acquatella B. and M. J. Reiner, "Modelica Stage Separation Dynamics Modeling for End-to-End Launch Vehicle Trajectory Simulations," in *Proceedings of the 10th International Modelica Conference*, 2014.

[13] P. Acquatella B., "Launch Vehicle Multibody Dynamics Modeling Framework for Preliminary Design Studies," *6th International Conference on Astrodynamics Tools and Techniques (ICATT)*, 2016.

[14] L. E. Briese, A. Klöckner, and M. Reiner, "The DLR Environment Library for Multi-Disciplinary Aerospace Applications," in *Proceedings of the 12th International Modelica Conference*, 2017.

[15] L. E. Briese, K. Schnepper, and P. Acquatella B., "Advanced Modeling and Trajectory Optimization Framework for Reusable Launch Vehicles," in *Proceedings of the IEEE Aerospace Conference*, 2018.

[16] L. E. Briese, "Mission-Dependent Sequential Simulation for Modeling and Trajectory Visualization of Reusable Launch Vehicles," in *Proceedings of the 2nd Japanese Modelica Conference*, 2018.

[17] K. Schnepper, "Trajektorienoptimierung in MOPS - Das Paket trajOpt Version 1.0," Tech. Rep., DLR German Aerospace Center, 2014.

[18] G. Looye, M. Thümmel, M. Kurze, M. Otter, and J. Bals, "Nonlinear Inverse Models for Control," in *Proceedings of the 4th International Modelica Conference*, 2005.

[19] M. J. Reiner and J. Bals, "Nonlinear Inverse Models for the Control of Satellites with Flexible Structures," in *Proceedings of the 10th International Modelica Conference*, 2014.

[20] G. Looye, "Design of Robust Autopilot Control Laws with Nonlinear Dynamic Inversion," *Automatisierungstechnik*, vol. 49, no. 12, 2001.

[21] G. Looye, *An Integrated Approach to Aircraft Modelling and Flight Control Law Design*, Ph.D. thesis, Delft University of Technology, Faculty of Aerospace Engineering, Switzerland, 2008.

[22] P. Lu, E.-J. van Kampen, C. de Visser, and Q. Chu, "Aircraft fault-tolerant trajectory control using Incremental Nonlinear Dynamic Inversion," in *Journal of Control Engineering Practice*, 2016, number 57.

[23] P. Acquatella B., L. E. Briese, and K. Schnepper, "Guidance Command Generation and Nonlinear Dynamic Inversion Control for Reusable Launch Vehicles," *Proceedings of the 69th International Astronautical Congress (IAC)*, 2018.

[24] S. E. Mattsson, H. Elmqvist, and M. Otter, "Physical System Modeling with Modelica," in *Control Engineering Practice*, 1998.

[25] M. Tiller, *Introduction to Physical Modeling with Modelica*, The Springer International Series in Engineering and Computer Science, 2001.

[26] H. Elmqvist, S. E. Mattsson, and M. Otter, "Modelica - An International Effort to Design an Object-Oriented Modeling Language," in *Summer Computer Simulation Conference*, 2003.

[27] P. Fritzson, *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1.*, John Wiley & Sons, 2004.

[28] M. Otter, H. Elmqvist, and S. Mattsson, "The New Modelica MultiBody Library," in *Proceedings of the 3rd International Modelica Conference*, 2003.

[29] DYMOLA, *Version 2018*, Vélizy-Villacoublay, France, Dassault Systèmes, 2018.

[30] F. O. Eke, "Dynamics of Variable Mass Systems," Tech. Rep. CR-1998-208246, NASA, 1999.

[31] NIMA, "World Geodetic System 1984 - Its Definition and Relationships with Local Geodetic Systems," Tech. Rep., National Imagery and Mapping Agency, 2000.

[32] J. Wendel, *Integrierte Navigationssysteme: Sensordatenfusion, GPS und Inertiale Navigation*, Number 2. De Gruyter Oldenbourg, 2011, ISBN: 978-3486704396.

[33] MATLAB, *Dryden Wind Turbulence Model*, Natick, Massachusetts, United States, The Mathworks, Inc., 2016.

[34] W. F. Phillips, C. E. Hailey, and G. A. Gebert, "Review of Attitude Representations Used for Aircraft Kinematics," *Journal of Aircraft*, vol. 38, no. 4, 2001.

[35] E. Mooij, "The Motion of a Vehicle in a Planetary Atmosphere," Tech. Rep. LR-768, Delft University of Technology, Faculty of Aerospace Engineering, 1997.

[36] T. Johansen, A. Cristofaro, K. Sorensen, J. Hansen, and T. Fossen, "On estimation of wind velocity, angle-of-attack and sideslip angle of small UAVs using standard sensors," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015.

[37] S. E. Mattsson, H. Olsson, and H. Elmqvist, "Dynamic Selection of States in Dymola," in *Modelica Workshop 2000*, 2000.

[38] P. H. Zipfel, *Modeling and Simulation of Aerospace Vehicle Dynamics*, Number 2. American Institute of Aeronautics and Astronautics, Inc. (AIAA), 2007, ISBN: 978-1-56347-875-8.

[39] A. K. Banerjee, "Dynamics of a Variable-Mass, Flexible Body System," in *Journal of Guidance, Control, and Dynamics*, 2000, vol. 23.

[40] D. G. Hull, *Fundamentals of Airplane Flight Mechanics*, Springer, 2007.

[41] J. Klevanski and M. Sippel, "Beschreibung des Programms zur aerodynamischen Voranalyse CAC Version 2," Tech. Rep. DLR IB 647-2003/04, DLR German Aerospace Center, 2003.

[42] U. Reisch and Y. Anseaume, "Validation of the Approximate Calculation Procedure HOTSOSE for Aerodynamic and Thermal Loads in Hypersonic Flow with Existing Experimental and Numerical Results," Tech. Rep., DLR German Aerospace Center, 1998.

[43] C. Rosema, J. Doyle, L. Auman, M. Underwood, and W. B. Blake, *Missile DATCOM, User's Manual - 2011 Revision*, AFRL, 2011.

[44] Modelica Association, *Functional Mock-Up Interface for Model Exchange and Co-Simulation*, 2015.

[45] H.-D. Joos, "MOPS - Multi-Objective Parameter Synthesis," Tech. Rep. DLR-IB-SR-OP-2016-128, DLR German Aerospace Center, 2016.