ASTOS – The Complete Toolchain for Launch Vehicle Trajectory Simulation, GNC Design and V&V

Dimitrios Gkoutzos [(1)], David Levi [(1)], Jorge Valderrama [(1)]

[(1)] *Astos Solutions GmbH, Meitnerstrasse 8, 70563 Stuttgart Germany, phone: +49 711 892633 - 27, emails: dimitrios.gkoutzos@astos.de*, david.levi@astos.de, jorge.valderrama@astos.de,

ABSTRACT

There is tremendous benefit to be gained from a coherent and well-integrated toolchain within the GNC development cycle. The highly iterative nature of this development requires easy reconfigurable and adaptable tools to support the tasks ranging from trajectory optimization and simulation with varying levels of complexity for GNC design as well as Verification & Validation in Model/Software in the Loop (MIL/SIL) and Processor/Hardware in the loop (PIL/HIL). ASTOS offers a flexible configuration of the rocket scenario for all validation tasks. The user can move seamlessly step by step from trajectory optimization to 3-dof guidance and navigation and 6-dof closed-loop control simulations using MIL/SIL and PIL/HIL. Additionally, the GNC department of Astos Solutions has been focused on providing and developing dedicated algorithms and design and analyses tools for guidance, navigation and robust control to meet the increasing demand driven by the microlauncher competition. This paper presents how the well-known ASTOS software can be utilized to cover all the tasks within the GNC development cycle and describes the latest updates and current developments enabling efficient and effective GNC design for space missions.

1 INTRODUCTION

Astos Solutions has been a dominant player in the fields of vehicle design and trajectory optimizations with more than 43 Microlauncher companies, research organizations and space agencies as customers from all over the world. Over the last decades the multi-purpose ASTOS software has been used on a variety of space applications. It has been successfully applied in various industrial and ESA projects, including missions such as Ariane 6, Ariane 5, Vega, Next European Launch System, Automatic Transfer Vehicle, Hopper, Skylon, and Fly-back Booster.

The emergence of microlauncher companies face new kinds of challenges that come with the need of being fast and responsive while at the same time as cheap as possible. To overcome these and be competitive in today's landscape a departure from the classical industry approach exemplified by the established launch providers needs to be pursued. With only a few of the microlauncher companies surviving in the long run, first to market plays a significant role in this competition.

The GNC development remains one of the most time-consuming tasks for launch vehicle development. The typical highly iterative GNC development cycle requires a multitude of dedicated tools to perform each task. Considering the time and resources it takes to maintain all the necessary tools as well as the effort requires a high upfront investment before even starting to the GNC development. While this is feasible for the established larger launch providers, especially young microlauncher companies lack the basis of decades of R&D including the experience and development of tools and software that comes with it. With the cost being the main driver in the competition many companies are therefore outsourcing the GNC development to further reduce cost and save time. Astos Solutions has recently been hired to develop the GNC flight software for a microlauncher with more customers stating their interest to do so as well. This has led to many upgrades in the features and tools of the ASTOS toolchain for GNC.

The toolchain described hereafter, covers the complete GNC design cycle with dedicated tools for each task. The transition between each step is automatically handled by ASTOS requiring only minimal effort. This allows users to focus their valuable time and effort on the actual GNC design and V&V activities rather than maintaining and improving their tools in the first place.

2 Workflow

The ASTOS scenario forms the basis of the toolchain and contains all relevant information such as environmental models (e.g., gravity, atmosphere, wind), vehicle properties (e.g., aerodynamic coefficients), vehicle parts (e.g., actuators, sensors), and dynamics configurations (e.g., phases, equations of motion, numerical integration settings) that represent a specific launch vehicle case. The vehicle design optimization has been discussed numerous times in the past and will not be described in this paper. This freedom in the configuration and set up allows to perform any mission from the launch vehicle ascent to orbital scenario and even to interplanetary transfer missions. Although the ASTOS software is specialized in aerospace applications such as spacecrafts and launch vehicles it should be noted that ASTOS is not constrained to these and allows to model any dynamical system.

The extensive model library offers a wide range of options to setup the specific launch vehicle case. The atmosphere can, for instance, be modelled according to the U.S Standard atmosphere or the Global Reference Atmospheric Model (GRAM) and introduce uncertainties on desired parameters. Different models for vehicle parts allow the definition of a wide variety of rockets. Multi-stage vehicles with different combinations of solid propellant, liquid propellant and hybrid propulsion stages can be modelled.

The scenarios can be used – with minor modifications – to perform all the tasks required for launch vehicle trajectory simulation, GNC design and V&V; like trajectory optimization, 3DoF and 6DoF simulation including flexible and multibody dynamics simulations. This ensures that the information related to environmental models, vehicle parts, vehicle characteristics and the necessary dynamics configurations are handled in a correct and efficient manner across the different design and simulation activities and between different teams working together.

The tools presented in detail in the following sections are all seamlessly integrated with one another and together form a coherent toolchain, which aims to simplify the GNC design process. The workflow summarised in this section provides GNC engineers with modularity and flexibility and adapts to several launcher configurations.

The workflow presented in Figure 1 is based upon the ASTOS software which already provides several useful functionalities for GNC design, as described in section 3. The process starts - once the vehicle design has been completed - by defining the physical rocket and the launcher mission from lift-off to orbit insertion. Optimization of the trajectory can be directly performed with ASTOS, with the user defining some constraints and a cost function to optimize, e.g., time to orbit or payload carrying capacity, detailed in section 3.1. Once an optimized trajectory has been obtained, it can be simulated, either directly in the ASTOS software or in the Functional Engineering Simulator (FES). Simulation is possible both in a three-degree of freedom (3 DoF) mode, where the launcher vehicle is modelled as a point mass with rigid dynamics, described in section 3.2, or in a more complex six-degree of freedom (6 DoF) mode, expanded upon in section 3.3. In this latter mode, the flexibility of the launch vehicle, a factor greatly affecting the launcher's dynamics, is also well handled by ASTOS, as explained in section 3.4.

This allows for a step-by-step approach to the GNC development. Firstly, guidance and navigation algorithms are developed making use of the simplified points mass dynamics of the 3DoF scenario.

Extending the dynamics to include attitude dynamics, the 6DoF scenario allows to further develop the navigation algorithm and design the controller for thrust vector control (TVC) and the reaction control system (RCS).

Dedicated algorithms for guidance and navigation as well as tools for robust control design are available, minimizing the effort developing the GNC solution from scratch detailed in 4.3. The robust control design framework, described in 4.4 is currently being developed making use of state-of-the-art approaches to increase versatility, autonomy and safety in an effective and efficient development process. The novelty of these tools is the focus on linear time varying systems compared to the 'frozen time point' approach commonly used for launcher GNC.

The developed GNC software can then be validated in a model in the loop (MIL) simulation, software in the loop (SIL) where the GNC algorithms can be auto-coded in the context of rapid prototyping and be further validated in processor and hardware in the loop tests as described in 3.5.

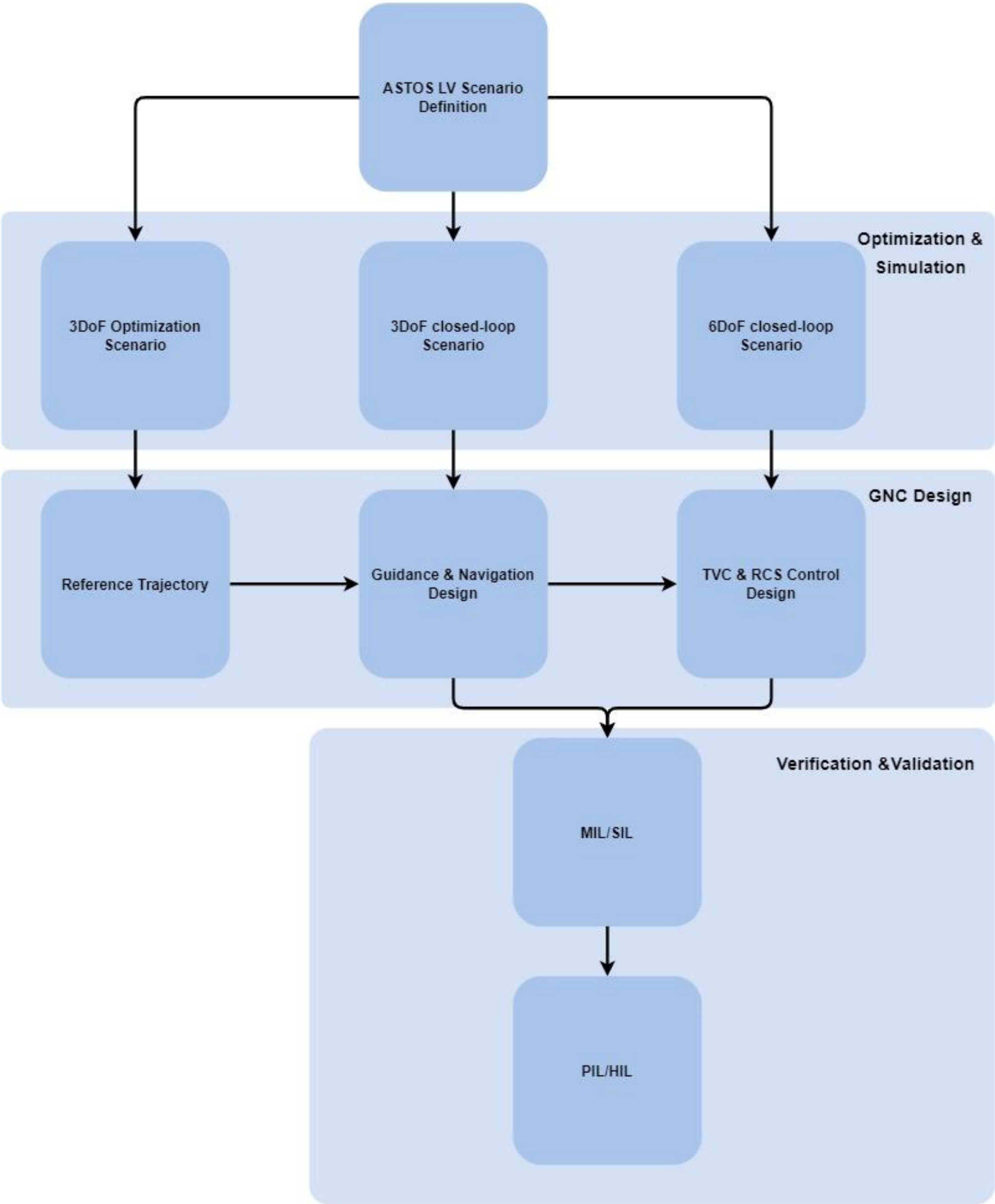

Figure 1 Workflow

3 TRAJECTORY OPTIMIZATION AND SIMULATION

3.1 Optimization

ASTOS is well known for the trajectory optimization as a dominant player in the industry. It has been successfully used in expendable rocket cases and reusable concepts are also strongly supported, such as first stage return flights making use of retro engines and aerodynamic control flaps, towing of a fly-back booster as studied in the FALCon project, air-launch concepts or hypersonic flight vehicles.

The optimization typically involves 3Dof dynamics and the definition of attitude control angles (roll, pitch, and yaw) to be determined by the optimizer. An objective function (e.g., maximize payload, minimize time to orbit) is defined by the user and drives the optimization procedure. Final boundary constraints (e.g., apogee altitude, inertial speed) ensure that the vehicle reaches a desired orbit or that given criteria are met at the end of a phase (e.g., heat flux, dynamic pressure before stage separation and fairing jettison). Path constraints can also be defined to ensure that, for instance, the vehicle does not violate any range constraints.

The optimal trajectory that minimizes the cost function while satisfying the boundary and path constraints can be obtained using different transcriptions (e.g., multiple shooting, collocation) and optimizers (e.g., WORHP) available in ASTOS. ASTOS offers a seamless integration with different optimization transcription packages with friendly user interfaces that allow for a wider range of users compared to other packages that require advanced programming skills. The transcription of the optimization problem using the collocation method is based on the packages TROPIC, SOS, and CAMTOS. Multiple shooting transcription is possible with the interface to PROMIS and CAMTOS. The transcribed trajectory optimization problem can then be solved by choosing one of the solver interfaces available in ASTOS, like SNOPT [1] or WORHP [2]. The resulting optimal trajectory can be exported and used as reference in closed loop control simulation phases that consider open-loop guidance. It also allows to feed the linearized models necessary for control design and flexible dynamics.

3.2 3DoF Simulation

It is possible to set up the ASTOS scenario to perform 3DoF simulation based on a set of initial conditions and attitude. This is of special utility for the design of guidance and navigation algorithms, which benefit from the simplicity of a point mass representation where rotational dynamics of the vehicle are ignored. The user can perform 3 DoF simulations in two different ways. The first one corresponds to an open loop simulation using the ASTOS GUI, where the user can define values for the pitch and yaw control angles or load their optimal values after performing a trajectory optimization. The second way is to define the attitude control angles as run-time variables that enables the control via the ASTOS S-function block using the Matlab / Simulink interface. In Simulink, it suffices to introduce the ASTOS S-function block and associate it to the ASTOS scenario. Sensor error models are provided allowing for a more realistic navigation design to assess the impact of different sensors on the solution of navigation algorithms.

3.3 6DoF Simulation

As a next step, the ASTOS scenario can be set up to perform 6 DoF simulations. For this, the software computes the inertia tensor of the different components based on models that are specific to the stage type. The evolution of the inertia tensor on a solid rocket stage depends on the grain geometry burn direction while in liquid propellant stages would depend on the liquid propellant level and tank geometry. As an option, the user can also input values for the inertia tensor based on its own mass distribution analysis. The attitude angles must be defined as states to perform a 6 DoF simulation, which are computed based on Euler's equations using either Euler angles representation or quaternions. The deflection angles of the engines should then be defined as the controls, commonly referred to as Thrust Vector Control (TVC). Just as in the 3Dof case, control angles can be defined in the ASTOS GUI to perform open loop simulations, or as run time variables in closed-loop using the ASTOS s-function in Simulink. ASTOS allows to model detailed dynamic effects of the launch vehicle 6 DoF simulation, like jet damping, propellants sloshing, flexible structures, and engine dynamics. The user can activate them individually and specify the parameters that define such models increasing the fidelity of the models depending on the task at hand.

For the closed loop simulation of launch vehicle ascent trajectories, it is common to use an open-loop guidance for the atmospheric portion of the flight. The open loop guidance is modelled as look-up tables that can be scheduled simple in time, or more robust as non-gravitational velocity or altitude. The look-up tables for attitude control contain the reference pitch, yaw, and roll angles that the launch vehicle should follow. When position control is also used, it is necessary to have look-up table for the reference position too. The reference position can be expressed, for instance, as longitude, latitude, and altitude. The reference values are obtained in the form of a Matlab export from the 3 DoF ASTOS optimization scenario and can be easily loaded into Simulink.

3.4 Flexible Dynamics and Multibody Dynamics

The ASTOS software offers two different methods to simulate flexible launch vehicles, namely Linearized Flexible Dynamics (LFD) and nonlinear flexible multi-Body dynamics Simulation (MBS). In the LFD method, a structural modal analysis is performed at different time Points Of Interest (POI) of a reference trajectory to generate mode shapes, slopes, and frequencies of the flexible launch vehicle structure. One POI is automatically defined by ASTOS every 5 seconds of the trajectory. Later, this information is featured in the 6 DoF simulation to model the interactions between the flexible structures and the forces experienced by the vehicle.

The modal analysis can be performed using the DCAP software that is part of the ASTOS toolchain. DCAP can model the different components as constant-cross section beams whose flexible response is described by analytic models. The modal analysis of the overall system is then obtained through the numerical linearization of a multi-body model that contains the flexible components. Additionally, a Finite Element Analysis (FEA) tool is currently being developed and integrated into ASTOS. The tool models the launch vehicle as a collection of beam or frame elements and uses spring-mass systems to represent the sloshing dynamics. Such a tool will allow a higher level of fidelity for the modal analyses and will reduce the computation time. A user can also provide the modal analysis results performed with external high fidelity Finite Element Analysis (FEA) tools (e.g., Ansys, NASTRAN) to be used for the ASTOS LFD simulation.

For MBS the DCAP software is used to generate multi-body equations of motion based on Kane's method, allowing to model the interactions between gimballed engines, stages, and spring-masses (used for modelling sloshing) as a multibody system. This method can consider the nonlinear flexibility of bodies via analytic beam representation or with a NASTRAN beam model. The MBS methodology can be used to model detailed stage separation events and payload deployment dynamics.

3.5 FES and SVF

For purpose of verification and validation of the GNC algorithms Functional Engineering Simulators (FES) and Software Validation Facilities (SVF) are used to perform model-, software-, processor- and hardware-in-the-loop simulations depending on the purpose of the validation task.

The ASTOS software supports those validation steps, where ASTOS provides all data-driven configuration capabilities and acts as dynamics, kinematics, and environment (DKE) simulator as described in the previous chapters.

Monte Carlo (MC) simulations are among the most important steps in the verification & validation of the GNC software. By utilizing MC simulations, engineers can effectively model and analyse a wide range of potential scenarios, considering uncertainties in parameters such as sensor noise, system dynamics, and environmental conditions. This comprehensive testing enables the assessment of GNC system robustness, identification of potential failure modes, and evaluation of mission success probabilities. MC simulations play a vital role in optimizing GNC design, providing insights into risk factors, and ensuring the spacecraft's overall reliability and performance.

ASTOS offers two ways to perform MC simulations. The first one can be performed directly from the ASTOS software using the batch mode inspector. This allows executing models externally in Simulink. For this all the perturbation parameters need to be defined as batch variables together with the associated uncertainty distribution and uncertainty range.

The second one utilizes the MC environment developed in MATLAB for model and software in the loop test campaigns. An Excel Sheet contains all the perturbation parameters, their associated probability distribution and uncertainty range. Once this is defined, the environment interfaces with the actual ASTOS scenario, varying the parameters of the DKE according to the parameter definition and performs the simulations. The number of simulations can be defined as well as the possibility to run a sub-set of the simulations, including to re-run a specific simulation for further investigation. This allows to perform the MC simulations in batches and in parallel and thereby reducing the time significantly. Post-processing includes plotting and reporting capabilities as well as checking against requirements. This approach allows also to link the uncertainties defined in the linearized models for the control design 4.1 to the ones in the FES, further reducing complexity and eliminating potential error sources.

For validation tasks under real-time conditions, the ASTOS DKE is executed on the dSPACE SCALEXIO system and the onboard software on a flight computer linked to the SCALEXIO to perform PIL simulations. For HIL simulations navigation sensors and actuators are linked to the frontend of the SCALEXIO and stimulated or emulated by the ASTOS software. Either ASTOS internal equipment models are used for this purpose or used defined models provided in Simulink.

4 GNC DESIGN, ANALYSIS AND VERIFICATION & VALIDATION

4.1 Launcher Dynamics Linearization

A representative model of the dynamics is essential and forms the basis of all model-based control design approaches. Once a scenario is defined and optimized in ASTOS, as described above, the trajectory can be exported to obtain the linearized model for control development. Note that the model obtained is analytic, building upon the heritage of similar launch vehicle dynamics models, as in [3, 4, 5]. More specifically, these analytic dynamics are put in state-space form and allow to design a controller to stabilise a launcher following a certain trajectory. These dynamics aim to describe how the launch vehicle's attitude and position will evolve away from a steady-state position when subject to perturbation.

The dynamics of the launch vehicle vary with time. The first step of the linearization is to obtain the launch vehicle's trajectory. Once a scenario has been simulated in 6 DoF, the user can export the whole trajectory with only a few clicks using the ASTOS software. The user can then specify the time grid of points of interests (POI) at which to obtain linearized dynamics. The interval between successive POIs is chosen according to a specific need: for example, the user may choose a small interval of 0.01 s for the generation of an LTV system, and then a larger one of 10 s for gain schedule control applications.

The strength of the tool is that a wide range of physical effects may be added or removed from the analytically derived linear dynamics. Besides rigid body dynamics, the user may choose to include any of the following physical effects: flexibility, sloshing, tail-wag-dog (TWD), varying aerodynamic properties along the launcher, jet damping, sensor position, pitch-yaw coupling and actuator dynamics. This modularity of the linearization tool is a considerable benefit in the GNC design process. Indeed, the user can easily proceed in a two-step manner: first, simplified linear dynamics models of the launch vehicle – including only the main physical effects – are generated. These allow to perform control design on a simplified system of the launch vehicle including only its main features. Second, the user can then check that the designed controller still achieves its objectives on a more detailed linear system before validating the design through an extensive MC campaign. The new physical effects included are then considered as perturbations that the controller should be robust against. The ease with which analytically derived linear systems of varying complexity can be obtained allows the process of GNC design to be performed in a streamlined fashion.

The analytically derived linear dynamics are obtained in the dynamics (D) frame. These dynamics aim to describe the motion of a launch vehicle that is in a state slightly perturbed from its steady state. The steady state has a particular meaning in this situation as it corresponds to a launch vehicle perfectly following its reference trajectory with the desired attitude. In that sense, the "steady state" is extremely dynamic, involving large velocities and at times angular rates for the body. This frame allows to compare the position and attitude of the actual launch vehicle to an ideal reference vehicle.

Figure 2 shows the actual launcher in the pitch plane of the dynamics frame (X_D, Z_D), with a deviation in attitude (θ) and position (z) with respect to the reference launcher vehicle. The advantage of the dynamics frame appears clearly: when the attitude and position of the actual LV are zero in the dynamics frame, it corresponds to tracking the reference launcher vehicle perfectly.

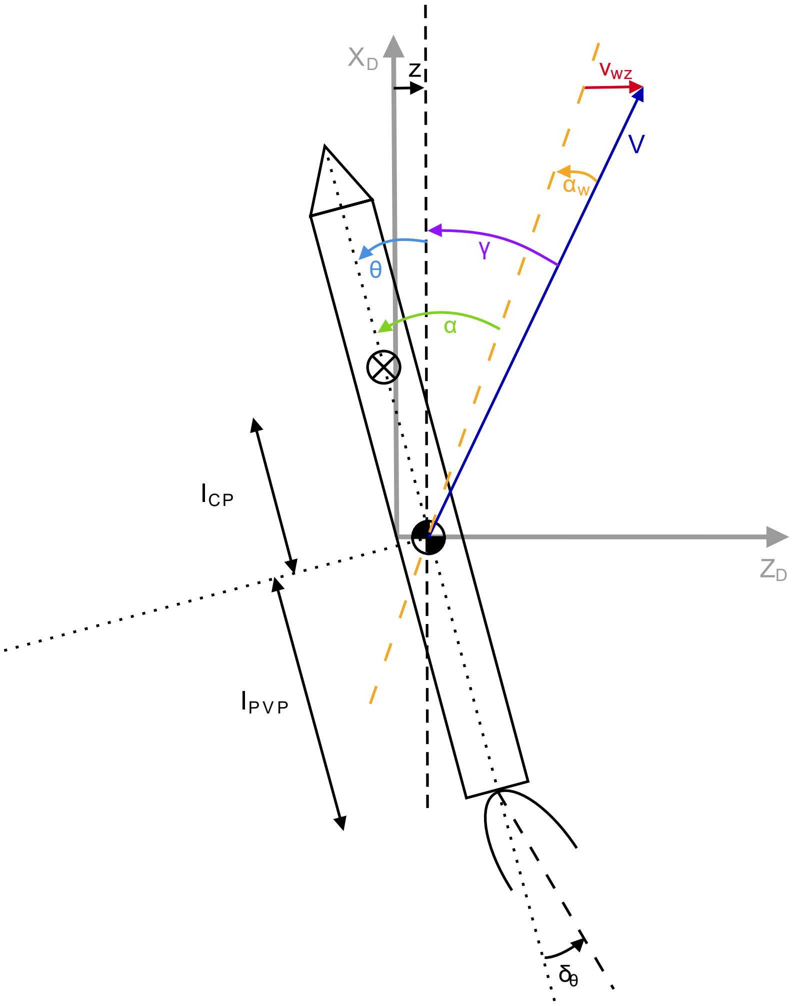

Figure 2. Actual launcher in the pitch plane of the dynamics frame

Starting from Newton's 2nd law of motion, it is possible to derive equations of motion for the rigid body dynamics of the actual launcher vehicle.

Along the Z_D axis:

$$M\ddot{z} = -(T_\theta - D)\theta - N_z\alpha - T_\theta\delta_\theta \tag{1}$$

About the Y_D axis (coming out of the page in Figure 2):

$$I_{yy}\,\ddot{\theta} = N_z\alpha\, l_{CP} - T_\theta\, l_{PVP}\,\delta_\theta \tag{2}$$

Note that in deriving both equations (1) and (2) all angles are assumed small and the corresponding approximations are used to obtain linear equations of motion.

The angle of attack (α) is an important component of both equations as the aerodynamic forces on the launcher vehicle are directly proportional to it.

It is made up of several components as can be seen in Figure 2 and can be approximated as

$$\alpha \approx \theta + \frac{\dot{z}}{V} - \frac{v_{wz}}{V} - \frac{l_{CP}\dot{\theta}}{V} \tag{3}$$

Note the last term in this expression which represents a contribution to the angle of attack from to the lateral velocity of the centre of pressure due to the angular velocity of the launcher about its CG, as detailed in [3, 6].

The definition of the angle of attack, can be included in equations (1) and (2), giving

$$\ddot{z} = -\frac{N_z}{MV}\dot{z} - \frac{T_\theta + N_z - D}{M}\theta + \frac{N_z l_{CP}}{MV}\dot{\theta} - \frac{T_\theta}{M}\delta_\theta + \frac{N_z}{MV} v_{wz} \tag{4}$$

$$\ddot{\theta} = \frac{N_z\, l_{CP}}{I_{yy}\, V}\dot{z} + \frac{N_z\, l_{CP}}{I_{yy}}\theta - \frac{N_z\, (l_{CP})^2}{I_{yy}\, V}\dot{\theta} - \frac{T_\theta l_{PVP}}{I_{yy}}\delta_\theta - \frac{N_z\, l_{CP}}{I_{yy}\, V} v_{wz} \tag{5}$$

Using z, $\dot{z}$, θ and $\dot{\theta}$ as states, these equations may be put into standard state-space form to obtain

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{N_z}{MV} & -\frac{T_\theta + N_z - D}{M} & \frac{N_z l_{CP}}{MV} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{N_z\, l_{CP}}{I_{yy}\, V} & \frac{N_z\, l_{CP}}{I_{yy}} & \frac{N_z (l_{CP})^2}{I_{yy} V} \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\frac{T_\theta}{M} & \frac{N_z}{MV} \\ 0 & 0 \\ -\frac{T_\theta l_{PVP}}{I_{yy}} & -\frac{N_z\, l_{CP}}{I_{yy}\, V} \end{bmatrix} \begin{bmatrix} \delta_\theta \\ v_{wz} \end{bmatrix} \tag{6}$$

$$\Leftrightarrow \dot{\boldsymbol{x}}_{\boldsymbol{\theta}}^{\boldsymbol{rig}} = \mathbf{A}_\theta^{rig} \boldsymbol{x}_{\boldsymbol{\theta}}^{\boldsymbol{rig}} + \mathbf{B}_\theta^{rig} \boldsymbol{u}_{\boldsymbol{\theta}}^{\boldsymbol{rig}} \tag{7}$$

Besides the four rigid states of the system, $Q\alpha$ is also chosen as an output. This is the product of the dynamic pressure and the angle of attack. It is an important quantity, corresponding to the aerodynamic load on the launcher, and is often subject to limits in rocket design.

Our output equation is then

$$\begin{bmatrix} Q\alpha \\ \dot{z} \\ \ddot{z} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & \frac{Q}{V} & Q & -\frac{Q\, l_{CP}}{V} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & -\frac{Q}{V} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_\theta \\ v_{wz} \end{bmatrix} \tag{8}$$

$$\Leftrightarrow \boldsymbol{y}_{\boldsymbol{\theta}} = \mathbf{C}_\theta^{rig} \boldsymbol{x}_{\boldsymbol{\theta}}^{\boldsymbol{rig}} + \mathbf{D}_\theta^{rig} \boldsymbol{u}_{\boldsymbol{\theta}}^{\boldsymbol{rig}} \tag{9}$$

One of the strengths of the linearization tool is its modularity. As the user choses the effects that they wish to include, the state-space system is modified as described in Table 1.

Table 1: Necessary modifications to the state-space system for each physical effect included

Physical effect added	**Modification to the state space**
Flexibility	To add k flexible modes to the state-space model, the state vector $\boldsymbol{x_\theta}$ is augmented with $2k$ states for the modal coordinate of each flexible mode and its first-time derivative.
Sloshing	To add S sloshing masses (usually one per liquid propellant tank) to the state-space model, the state vector $\boldsymbol{x_\theta}$ is augmented with $2S$ states for the angle describing the position of each sloshing mass and its first-time derivative.
Tail-wag-dog	To account for the tail-wag dog effect, the input vector $\boldsymbol{u_\theta}$ is augmented with the angular acceleration of the nozzle $\ddot{\delta}_\theta$. Note that this is obviously not strictly independent from the angular deflection of the nozzle δ_θ and is therefore an improper way of handling this effect, but a sufficient approximation. This can be fixed with the inclusion of the actuator dynamics (see below).
Distributed aerodynamics	To account for varying aerodynamic properties (force and moment coefficients) along the length of the launcher vehicle, the coefficients of the $\mathbf{A}_\theta$ and $\mathbf{B}_\theta$ matrices are changed to reflect the variation in angle of attack due to the launcher's flexibility along its length.
Jet damping	To account for the loss of momentum due to the expulsion of propellant, the terms in the $\mathbf{A}_\theta^{rig}$ matrix are modified.
Sensor position	The output vector $\boldsymbol{y_\theta}$ can be modified to include quantities that are normally obtained by the launcher vehicle's sensors (at a fixed position in the LV) rather than at the centre of gravity. The position and attitude output by these sensors will be different to the states of state-space model, due to flexibility and position offset of the sensor with respect to the centre of gravity. These effects can be modelled by modifying the terms in the $\mathbf{C}_\theta^{rig}$ matrix.
Actuator dynamics	This effect is about recognizing that the commanded TVC deflection differs from the actual obtained one due to dynamics in the actuator commanding the orientation of the engine. This is modelled by making the input deflection angle be the commanded deflection angle δ_θ^{com} and augmenting the state vector $\boldsymbol{x_\theta}$ with the actual TVC

	deflection angle δ_θ^{act} and its first derivative $\dot{\delta}_\theta^{act}$. These three quantities are then linked through a second order model. This change also allows to calculate the TWD effect in a more realistic fashion, as the actual angular acceleration of the TVC engine can be calculated.
Pitch-yaw coupling	In the presence of a roll rate, pitch and yaw dynamics become coupled. This can be modelled by extending the state vector $\boldsymbol{x_\theta}$ to $\boldsymbol{x}$, including both pitch-plane and yaw-plane states. This change in turn dictates an extension of the state-space model's four matrices, with the addition of terms modelling this cross-coupling.

The modularity and interconnexions of the various physical effects detailed above are built in the $\mathbf{A}_\theta$, $\mathbf{B}_\theta$, $\mathbf{C}_\theta$ and $\mathbf{D}_\theta$ matrices of the state-space system. For example, the $\mathbf{A}_\theta$ matrix may be considered as composed of up to 16 smaller sub-matrices, depending on the physical effects considered, as shown in Equation (10). Each of these sub-matrices describes the influence of a group of state on the first derivative of another (off-diagonal sub-matrices) or of the same (diagonal sub-matrices) group of states.

$$\mathbf{A}_\theta = \begin{bmatrix} \mathbf{A}_\theta^{TVC} & \mathbf{A}_\theta^{rig\to TVC} & \mathbf{A}_\theta^{flex\to TVC} & \mathbf{A}_\theta^{slosh\to TVC} \\ \mathbf{A}_\theta^{TVC\to rig} & \mathbf{A}_\theta^{rig} & \mathbf{A}_\theta^{flex\to rig} & \mathbf{A}_\theta^{slosh\to rig} \\ \mathbf{A}_\theta^{TVC\to flex} & \mathbf{A}_\theta^{rig\to flex} & \mathbf{A}_\theta^{flex} & \mathbf{A}_\theta^{slosh\to flex} \\ \mathbf{A}_\theta^{TVC\to slosh} & \mathbf{A}_\theta^{rig\to slosh} & \mathbf{A}_\theta^{flex\to slosh} & \mathbf{A}_\theta^{slosh} \end{bmatrix} \tag{10}$$

For instance, $\mathbf{A}_\theta^{flex\to rig}$ describes the influence of the $2k$ flexible states on rigid body dynamics. Depending on the physical effects included in the analytic derived model, only a subset of these matrices may be included in the launcher linearization tool output. The three other matrices can similarly be broken down, providing the user with systems of the desired complexity.

Lastly, a great advantage in describing the launcher dynamics in a linear state-space form is the possibility to include uncertainties in the analytic linear models. This is done using the Linear Fractional Transformation (LFT) approach, which consists in placing the uncertainties on all parameters in a separate matrix connected to the state-space system as in [3]. The inclusion of uncertainties through this formulation in turn allows for more sophisticated robust control approaches.

4.2 Launcher Linearization Verification Tool

The Launcher Linearization Verification Tool (LL-VT) allows the user to verify that the linearized dynamics capture the behaviour of the LV with sufficient precision in the vicinity of the linearization point, using the 6DoF high-fidelity FES as a reference.

Two approaches are considered within this tool. The first one compares the analytical linear model along the reference trajectory computed by the Laucher linearization tool for a set of frozen time points to the 6DoF high-fidelity nonlinear FES, which will serve as the ground truth.

The coefficients that compose the state-space matrices of the linearized dynamics depend on the value of the state vector, $\boldsymbol{x}_{RT}(t_i)$, of a reference trajectory at a given time of interest, t_i (the subscript "RT" stands for the reference trajectory). The state-space matrices A_i, B_i, C_i, and D_i are labelled with the subscript "i" to associate them with their corresponding time of interest. Given a state vector $\boldsymbol{x}_{\theta_i}$ used as initial condition, a control input history $\boldsymbol{u}_{\boldsymbol{\theta}}(t)$ defined in the time interval $[t_i, (t_i + \Delta t)]$, and a desired simulation duration Δt, it is possible to perform a linear simulation to obtain the state-space output history $\boldsymbol{y}_{\boldsymbol{\theta}}(t)$. This output can be compared against the results of a simulation of the 6DoF FES that uses $\boldsymbol{x}_{RT}(t_i)$ as initial conditions and $\boldsymbol{u}_{\boldsymbol{\theta}}(t)$ as input. Following this methodology, it allows to test different control input at different times of interest in the reference trajectory to assess the precision of the analytically derived linear dynamics against the high-fidelity non-linear 6Dof FES. The outputs from the linear simulation are expected to be close to those of the 6DoF FES for low values of Δt, and are expected to diverge as Δt grows, given the time variant nature and the linearization procedure of the state-space representation.

The second approach for the LL-VT is based on formulating a Linear Time Variant (LTV) system based on a set of state-space matrices that are generated over a fine grid (LTI) on the reference trajectory using the LLT. Such an LTV system can be simulated in closed loop with a controller to generate the outputs $\boldsymbol{y}_{\boldsymbol{\theta}}(t)$ that capture the time-varying behaviour of the launch vehicle. Similarly, the 6DoF FES can be run using the same controller and initial conditions to assess the accuracy of the LTV system. This approach provides even more confidence to the control design engineer that not only a set of arbitrary time points are valid but the dynamics throughout the whole flight capture the behaviour of the launcher dynamics and its evolution in time.

4.3 Guidance & Navigation

Within the 'Off-The-Shelf Guidance & Navigation for Microlauncher' (MLGN) activity, funded through the Future Launchers Preparatory Programme by the European Space Agency, reconfigurable guidance and navigation algorithms have been developed. Reconfigurable to different launch vehicles and missions and tested up to PIL, these algorithms allow to easily set up closed-loop simulations for preliminary and detailed analyses as well as integrate them into GNC Flight software.
The guidance software makes use of an open-loop reference profile for the endo-atmospheric ascent obtained through offline optimization using ASTOS. The closed-loop online optimization for the exo-atmospheric ascent - including orbital manoeuvres such as the circularization burn - is based on the Powered Explicit Guidance (PEG) algorithm developed by NASA. The modularity of the software allows to replace individual modules such as the closed loop optimization with other approaches. More advanced convex as well as nonlinear optimization approaches are being developed and integrated allowing to choose the strategy based on the application.
A hybrid (IMU+GNSS) extended Kalman Filter in an error state formulation has been developed for the navigation.
In addition, an inhouse development of a federated Kalman Filter (FKF) scheme as error state Schmidt-Kalman filter to account for multiple (low-cost) IMUs is also available. Allowing to reduce the cost while still meeting the requirements.
Both algorithms are integrated into the ASTOS toolchain. The architecture of both algorithms simplifies the initialization and configuration to specific launch vehicles and missions making use of automated processes and exports from the ASTOS scenarios, requiring only minimal user input. The

library of available guidance and navigation algorithms will be extended in the future with current work focusing on convex optimization as well as model predictive control (MPC).

4.4 Robust Control Design Framework

All previously described tools are currently integrated and extended by state-of-the-art robust control approaches in one common framework. The development of a robust control design framework is carried out under a programme of and funded by the European Space Agency - through the Future Launchers Preparatory Programme. This framework will comprise a set of tools, starting with the described LLT 4.1 and LL-VT 4.2, and include different tools for the analysis and design of robust controller. Making use of state-of-the-art techniques such as Integral Quadratic Contraint (IQC) theory for linear time varying worst case analyses it will serve as the foundation for tomorrows GNC development. This framework will unite all the aforementioned capabilities of the ASTOS toolchain and make state-of-the-art techniques accessible to the industry.

5 SUMMARY

This paper presented an overview of the capabilities of the ASTOS toolchain for launch vehicle trajectory optimization and simulation, GNC design and Verification and Validation. Due to the rising demand in tools and algorithms of the microlauncher customers, Astos Solutions has been focused on developing tools and algorithms extending the ASTOS toolchain to cover the full GNC development cycle.

The trajectory optimization can be performed in 3DoF with numerous techniques and solvers all integrated into the ASTOS software. The simulations of said trajectory can be performed in 3DoF, where the launcher is modelled as a point mass for the design and V&V of guidance and navigation algorithms or 6DoF including attitude dynamics, flexibility and many more detailed physical effects increasing the fidelity of the models as required for the full GNC design and V&V. The reconfiguration of the simulator is handled by ASTOS automatically just by loading the corresponding scenario.

Real-time tested algorithms for guidance and navigation are available as a library for the user to choose from, well integrated into the ASTOS toolchain thereby simplifying the reconfiguration and initialization process through automated exports.
A linearization tool allows to obtain linear algebraic derived models of the launcher along the optimized reference trajectory. The modularity enables to include many different physical effects depending on the required fidelity of the models. A verification tool allows to validate the obtained linear models against the nonlinear FES providing invaluable feedback to the control design engineer.
Based on the derived models additional advanced robust control tools are currently being developed such as for the interpretable design of weighting functions based on the requirements or a robust control synthesis tool that offers a library of implemented controllers and architectures such as $H\infty$, LPV, LTV control in an observer-based structure. A worst-case analysis tool based on the integral quadratic constraint theory for linear time varying systems provides quick feedback on the design much more reliable than with current LTI approaches, typically used for launcher GNC.

Easily set up Monte Carlo campaigns in MIL/SIL validate the designed GNC software. This setup can also be used for real-time verification & validation in PIL/HIL as demonstrated in various activities.

The presented toolchain is bridging the gap between state-of-the-art research and industry application making advanced techniques accessible to the industry. Overall, the toolchain based on ASTOS simplifies the GNC design process tremendously, provides modularity and flexibility, and is applicable to all kinds of launcher configurations. In fact, this allows a straightforward transition between the different steps with minimal effort as the reconfiguration is automatically handled by ASTOS and all data are contained within the scenario. With more than 43 microlauncher companies, it's safe to say that Astos' unique position will influence the launcher GNC design of the future.

REFERENCES

[1] P. E. Gill, W. Murray und M. A. Saunders, „SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Journal on optimization*, Bd. 12, Nr. 4, pp. 979-1006, 2002.

[2] C. Büskens und D. Wassel, „The ESA NLP Solver WORHP," *Modeling and Optimization in Space Engineering. Springer Optimization and Its Applications.*, Bd. 73, 2012.

[3] D. Navarro-Tapia, „Robust and Adaptative TVC Control Design Approaches for the VEGA Launcher," 2019.

[4] A. L. Greensite, „Analysis and Design of Space Vehicle Flight Control Systems," General Dynamics Corporation, San Diego. Calif. , 1967.

[5] T. M. Barrows und J. S. Orr, Dynamics and Simulation of Flexible Rockets, Elsevier, 2020.

[6] A. Marcos, S. Bennani, C. Roux und M. Valli, „LPV modeling and LFT Uncertainty Identification for the Robust Analysis: application to the VEGA Launcher during Atmospheric Phase," in *IFAC (International Federation of Automatic Control)*, Grenoble, France, 2015.

[7] „ASTOS: Analysis, Simulation and Trajectory Optimization Software for Space Applications.," Astos Solutions, [Online]. Available: www.astos.de.