

Sistemi di Lancio Innovativi e Navigazione

**PROF. PAOLO GASBARRI,
ING. ROBERTO CAPUA**



Outline of the presentation:

Part 1 Advanced Dynamic Modeling of Flexible Launch Vehicles: A Multibody Approach

Part 2 Launch Phase Modal and Stability Analysis Simulations



Introduction

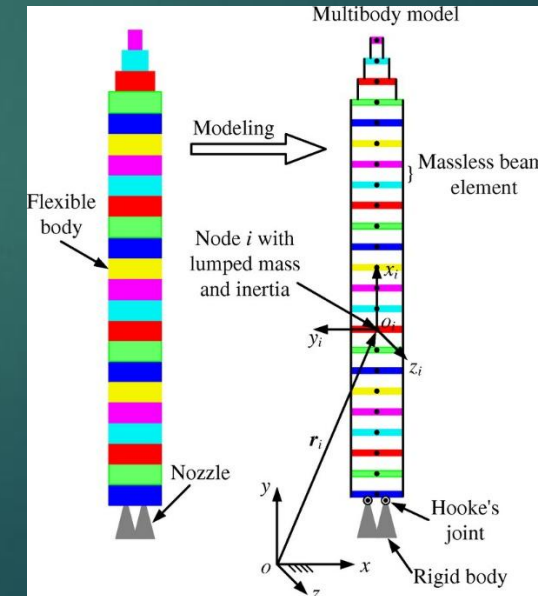
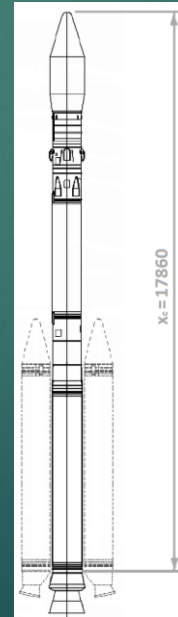
The Problem: Why is Flexibility Crucial?

Modern launch vehicles are designed to be extremely lightweight to maximize payload-to-orbit capability. This lightweight design inevitably leads to significant structural flexibility.

The launcher can no longer be modeled as a simple rigid body. Its elastic deformations (like bending and vibrations) profoundly affect its:

- Flight dynamics
- Stability
- Guidance and Control (GNC) system

The interaction between structural vibrations and the control system can cause instabilities, load amplification, or dangerous resonance phenomena (e.g., "pogo effect").



Introduction

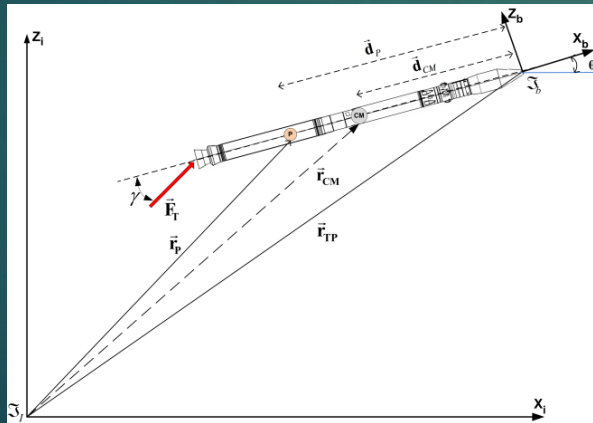
The Problem: Why is Flexibility Crucial?

A flexible launcher requires additional elastic degrees of freedom to describe how each segment can deform relative to the others. These deformations may include bending of stages, vibration of tanks, or oscillations of payload adapters.

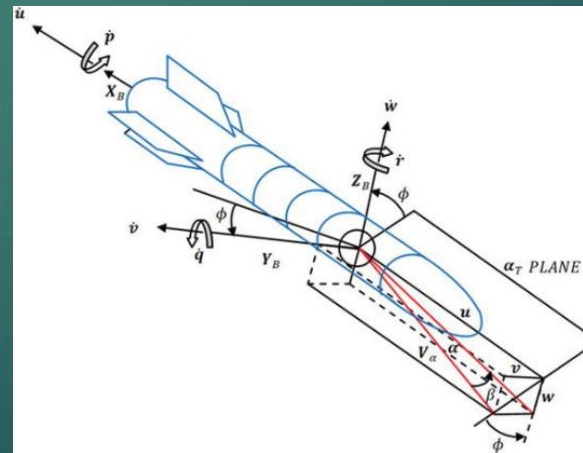
The total motion of the launcher is thus a combination of:

Total motion = Rigid-body motion + Elastic deformation

where both components evolve simultaneously and influence each other.



Frames definition for the Launcher: Inertial frame and body frame .



Six-degree-of-freedom motion of a rocket.

Introduction

Sources of Flexibility:

Flexibility in a launcher originates from several physical aspects:

- Slender geometry: long and thin stages have low bending stiffness.
- Structural joints: connections between stages, tanks, or boosters introduce local compliance.
- Payload adapters: often behave like flexible springs between payload and upper stage.
- Propellant dynamics: liquid sloshing adds effective flexibility and damping.

In addition, aerodynamic forces acting along the slender body amplify the effect of even small deformations.

Introduction

Propellant Sloshing Effects:

In liquid-propelled launch vehicles, propellant sloshing—the oscillatory motion of liquid inside partially filled tanks—introduces an additional dynamic coupling between the vehicle's structural and attitude dynamics.

The moving liquid mass generates time-varying forces and moments that can:

- shift the instantaneous center of mass,
- induce bending moments on the tank walls,
- interact with the attitude control system, possibly leading to limit-cycle oscillations.

The complexity of sloshing dynamics depends on the fill ratio, tank geometry, and excitation source (engine thrust oscillations, stage separation, or aerodynamic disturbances).

From a modeling point of view, different levels of fidelity can be used:

1. Equivalent Mechanical Models – represent the fluid as a pendulum or a spring–mass–damper system tuned to the fundamental sloshing frequency;
2. Modal Coupled Models – couple fluid oscillation modes with the structural vibration modes of the tank;
3. CFD-based Fluid–Structure Interaction – capture nonlinear surface motion and viscous effects, though at high computational cost.

Introduction

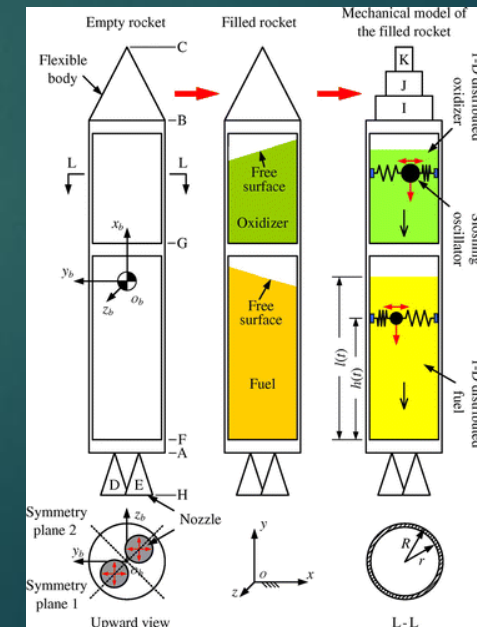
Propellant Sloshing Effects

From a modeling point of view, different levels of fidelity can be used:

1. Equivalent Mechanical Models – represent the fluid as a pendulum or a spring–mass–damper system tuned to the fundamental sloshing frequency;
2. Modal Coupled Models – couple fluid oscillation modes with the structural vibration modes of the tank;
3. CFD-based Fluid–Structure Interaction – capture nonlinear surface motion and viscous effects, though at high computational cost.

For launcher-level simulations, the equivalent mechanical or modal approaches are preferred, as they provide accurate dynamic coupling with minimal computational effort and integrate seamlessly within a multibody formulation.

Diagram of a typical liquid rocket



Introduction

Coupling Between Structural and Flight Dynamics

The interaction between structural flexibility and rigid-body motion is known as aeroelastic or flight–structure coupling. This coupling affects both the forces acting on the launcher and the response of the control system.

Examples include:

- Structural vibrations influencing sensor readings (e.g., accelerometers, gyros).
- Thrust vector misalignment due to bending of engine mounts.
- Control-induced oscillations (the “pogo effect”) resulting from feedback between thrust, structure, and propellant flow.

The correct modeling of this coupling is therefore critical to achieve realistic simulation and reliable control design.

Modeling Strategies: A Comparison

Two main approaches exist for modeling launcher dynamics:

1. Finite Element Method (FEM).

How it works: Discretizes the entire structure into a mesh of elements.

Pros: Provides highly accurate local stress and strain data.

Cons: Produces enormous models (millions of Degrees of Freedom - DOF), making dynamic simulation and control integration computationally prohibitive.

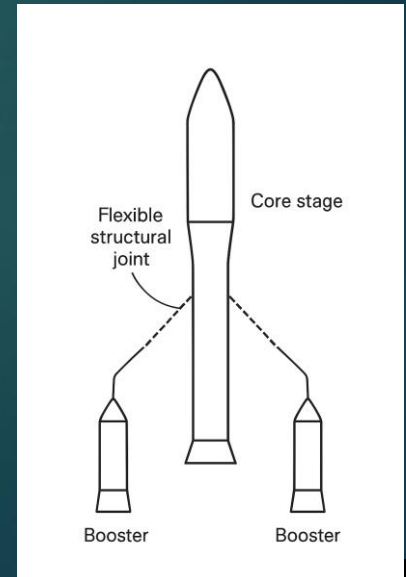
2. Multibody Approach

How it works: Models the launcher as a system of interconnected bodies (stages, adapters, payload).

Pros:

- Each body can be rigid or flexible.
- Captures the essential dynamics with far fewer DOFs.
- Natively integrates flight mechanics and structural deformation.

Conclusion: This is the ideal approach for dynamic analysis and control design.



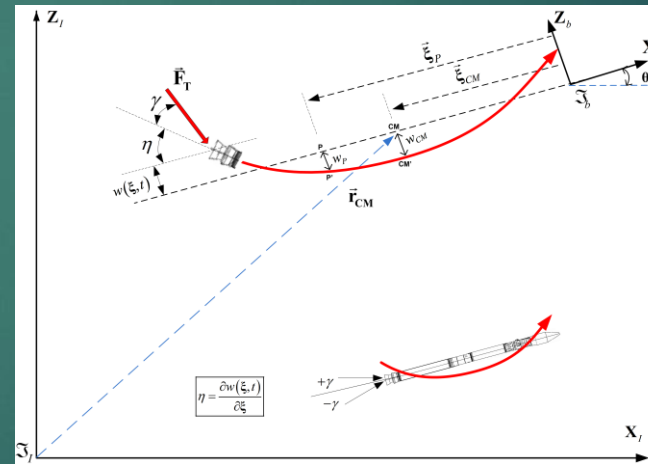
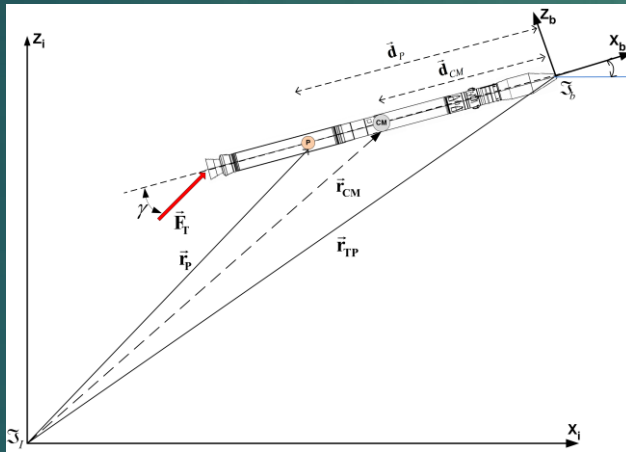
Key Technique (1): The Floating Frame of Reference (FFRF)

How do we describe a *single flexible body* (like a stage) in the multibody approach? We use the Floating Frame of Reference Formulation (FFRF).

Key Concept: The total motion of any point on the body is decomposed into two parts:

Rigid Motion: The motion of a "floating frame" attached to the body (translation + rotation).

Elastic Motion: The deformation of point P relative to this floating frame.



Frames definition for the Launcher: Inertial frame ($\mathcal{S}_I$) and body frame ($\mathcal{S}_b$).

The system's coordinates are thus separated into: Rigid Coordinates 6 DOFs describing the frame's motion. Flexible Coordinates describing the deformation.

Key Technique (2): Modal Reduction

How do we describe the elastic deformation efficiently?

We don't use all the FEM nodes. We apply Modal Reduction.

- The entire deformation field $\mathbf{w}_f$ is approximated as a superposition (sum) of a small number of "**modal shapes**" (the structure's natural vibration modes).
- These "modal shapes" $\mathbf{U}_n$ are pre-calculated (e.g., from an FEM analysis).
- The only flexible unknowns (q_f) become the **modal amplitudes** $A_n(t)$, i.e., "how much" each mode is vibrating over time.

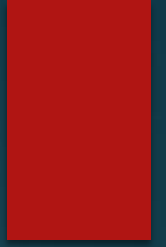
$$\mathbf{w}_f(\xi, t) = \sum_{n=1}^N \mathbf{U}_n(\xi) A_n(t)$$

- Typically, for a launcher, the first few **bending** and **torsion** modes are sufficient.
- This approach provides an **optimal balance** between accuracy and computational cost.

For a slender launcher stage, the most relevant modes are typically:

- **First and second bending modes. Torsional mode. Occasionally, axial or shear modes if mass coupling is strong.**

Dynamics of single body



Different formulations can be used to derive the dynamic equations of a single body (element) of the system.

The most relevant are:

- The Lagrangian Formulation
 - The dynamics is derived by minimizing the total energy (kinetic, elastic, work performed by loads)
- The Newtonian Formulation
 - The dynamics is derived by writing the equilibrium equation (translation, rotation) for a generic element volume of the body and then integrating on the entire body.
 - The generic mode equation is obtained using by projecting the local translational equilibrium equation on the generic modal shape and then integrating on the volum (Galerkin approach)

Equations of Motion of a Flexible Launcher

Once each launcher component has been modeled — either as a rigid or flexible body — the next step is to assemble all these elements into a single dynamic system.

This is the essence of the multibody formulation: representing the launcher as an interconnected system of subsystems, each governed by its own dynamics but linked through physical constraints.

The resulting equations describe the complete motion of the launcher, including:

- Rigid-body flight dynamics (translation and rotation),
- Structural flexibility and deformation,
- Interaction forces at joints and interfaces,
- External loads such as thrust, gravity, and aerodynamic pressure.

Equations of Motion of a Flexible Launcher

System Coordinates and Constraints.

Let the global configuration of the launcher be represented by the vector of generalized coordinates:

$$\mathbf{q} = [\mathbf{q}_1 \ \mathbf{q}_2 \ \cdots \ \mathbf{q}_N]$$

where $(\mathbf{q}_i)$ includes both the rigid and flexible coordinates of body (i) , and (N) is the total number of bodies.

Each connection between bodies introduces constraint equations:

$$\Phi(\mathbf{q}, t) = \mathbf{0}$$

These equations ensure the kinematic compatibility between connected components (for example, continuity at stage interfaces).

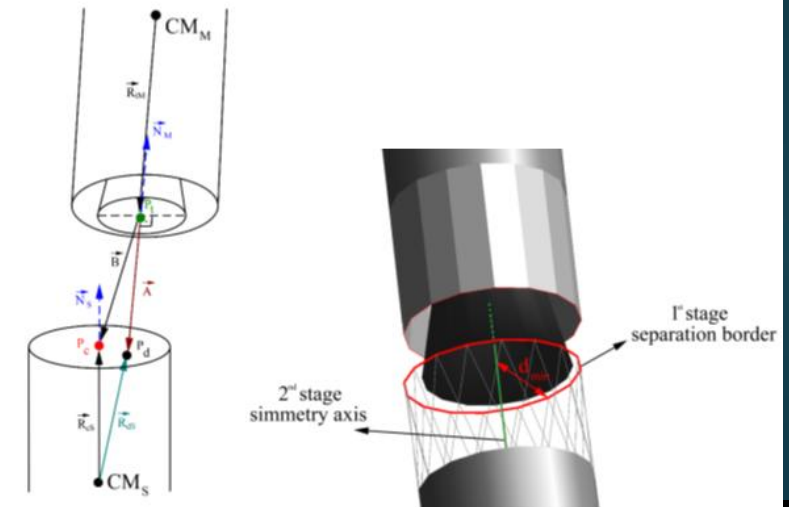
Equations of Motion of a Flexible Launcher

System Coordinates and Constraints.

By differentiating the constraints, we obtain velocity and acceleration relations:

$$\Phi_q \dot{\mathbf{q}} = \mathbf{0}, \quad \Phi_q \ddot{\mathbf{q}} + \dot{\Phi}_q \dot{\mathbf{q}} = \mathbf{0}$$

where (Φ_q) is the Jacobian matrix of the constraints.



Two bodies and one interstage example

Equations of Motion of a Flexible Launcher

Lagrange's Equations with Constraints.

The Lagrange equations of motion for the entire system, including constraints, are:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t) + \Phi_{\mathbf{q}}^T \boldsymbol{\lambda}$$

- $\mathbf{M}(\mathbf{q})$ is the **mass matrix**,
- $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$ represents **nonlinear inertial effects** (centrifugal and Coriolis terms),
- $\mathbf{Q}$ contains **external generalized forces** (gravity, thrust, aerodynamic loads, ...),
- $(\boldsymbol{\lambda})$ are the **Lagrange multipliers** enforcing the constraints.

This set of differential–algebraic equations (DAEs) describe the full coupled motion of the multibody system.

Equations of Motion of a Flexible Launcher

Structure of the Mass Matrix.

The mass matrix $(\mathbf{M})$ is block-structured, reflecting the division between rigid and flexible coordinates:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{rr} & \mathbf{M}_{rf} \\ \mathbf{M}_{fr} & \mathbf{M}_{ff} \end{bmatrix}.$$

where:

- $(\mathbf{M}_{rr})$: rigid-body inertia terms,
- $(\mathbf{M}_{ff})$: elastic inertia terms,
- $(\mathbf{M}_{rf} = \mathbf{M}_{fr}^T)$: **coupling** between rigid and flexible motion.

The presence of these coupling terms means that translational or rotational accelerations of the launcher can excite elastic vibrations — and vice versa.

Equations of Motion of a Flexible Launcher

External forces acting on the launcher.

The vector of generalized forces ($\mathbf{Q}$) includes several physical contributions:

1. Thrust Force

Applied at the engine nozzle and directed along the thrust axis.

If the launcher bends, the thrust line may deviate from the center of mass, creating a moment that couples with flexibility.

2. Gravitational Force

Acts on each body's mass distribution.

In multibody form, it is represented as the projection of gravity into generalized coordinates.

3. Aerodynamic Loads

Depend on vehicle velocity, angle of attack, and local deformation.

Aerodynamic pressure along the body can amplify bending and torsion.

4. Damping and Internal Forces

Include structural damping, elastic restoring forces, and constraint reaction forces at joints.

These generalized forces can be computed by virtual work principles or by direct projection of physical forces onto the coordinate space.

Equations of Motion of a Flexible Launcher

Constraint Forces and Lagrange Multipliers.

^(λ) The lagrangian multipliers (**λ**) represent the constraint reaction forces.

They ensure that the constraints are exactly satisfied at all times :

In matrix form, the system of equations can be written as:

$$\begin{bmatrix} \mathbf{M} & \Phi_q^T \\ \Phi_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{Q} - \mathbf{h} \\ -\dot{\Phi}_q \dot{\mathbf{q}} \end{bmatrix}$$

This system couples differential equations (motion) and algebraic equations (constraints).

Linear model

In many cases — such as the analysis of small elastic vibrations around a nominal flight trajectory — the equations can be linearized.

Assuming small elastic deformations and small angular perturbations, we can write:

$$\mathbf{M}_0 \ddot{\mathbf{q}} + \mathbf{C}_0 \dot{\mathbf{q}} + \mathbf{K}_0 \mathbf{q} = \mathbf{Q}(t)$$

Here:

- $(\mathbf{M}_0, \mathbf{C}_0, \mathbf{K}_0)$ are constant or slowly varying matrices (mass, damping, stiffness),
- Nonlinear inertial coupling terms are neglected or simplified.

This linearized model is extremely useful for:

- **Modal analysis** of the launcher's natural frequencies,
- **Control–structure interaction studies**,
- **Preliminary design** and stability assessments.

Coupling Between Elastic and Rigid Motion

The coupling between flexibility and rigid motion manifests in several ways:

- Attitude coupling: bending causes small angular deflections that affect guidance sensors.
- Load redistribution: flexibility changes the local load paths during dynamic maneuvers.
- Control coupling: actuators (e.g., thrust vector control) excite structural modes.

This coupling can lead to unwanted oscillations or instabilities if not properly modeled and controlled. For this reason, flexible–rigid coupling terms must be retained at least up to first order in accurate simulations.

Numerical Implementation

The equations of motion previously shown describe the complete dynamics of a flexible launcher: a coupled system of differential and algebraic equations.

However, obtaining accurate results requires robust numerical implementation.

In practice, simulation involves:

- Assembling the global mass, stiffness, and damping matrices,
- Evaluating external forces and constraints at each time step,
- Integrating the equations of motion in time,
- Stabilizing the constraints to prevent numerical drift.

System of Equations to be Solved

$$\mathbf{M}(\mathbf{q}), \ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t) + \Phi_{\mathbf{q}}^T \boldsymbol{\lambda}$$

$$\Phi(\mathbf{q}, t) = \mathbf{0}$$

This is a differential–algebraic system (DAE) of index 3, since the constraint equations must be satisfied at the position, velocity, and acceleration levels

Numerical Implementation

To integrate the system numerically

$$\mathbf{M}(\mathbf{q}), \ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t) + \Phi_{\mathbf{q}}^T \boldsymbol{\lambda}$$

$$\Phi(\mathbf{q}, t) = \mathbf{0}$$

We can either:

1. Solve it directly as a DAE, using specialized solvers;
2. Convert it into an ordinary differential equation (ODE) by eliminating or stabilizing the constraints.

Numerical Implementation

Constraint Handling Strategies

Several approaches exist for managing constraints in multibody simulations:

(a) Lagrange Multiplier Method

The most exact method. Constraints are included explicitly in the equations through multipliers

Advantages: mathematically precise and preserves constraint satisfaction.

Disadvantages: increases system size and can lead to stiff equations.

(b) Penalty Method

Instead of enforcing constraints exactly, *penalty forces* are introduced:

$$\mathbf{Q}_p = -\mathbf{K}_p \Phi - \mathbf{C}_p \dot{\Phi}$$

where $(\mathbf{K}_p)$ and $(\mathbf{C}_p)$ are stiffness and damping penalty matrices.

Advantages: simpler and purely differential form.

Disadvantages: approximate, may introduce artificial stiffness.

(c) Baumgarte Stabilization (preferred)

A hybrid approach that adds correction terms to constraint acceleration equations:

$$\ddot{\Phi} + 2\alpha \dot{\Phi} + \beta^2 \Phi = 0$$

Numerical Implementation

Time Integration Schemes

The choice of integrator strongly affects accuracy, stability, and computational efficiency.

(a) Newmark Family

Widely used in structural dynamics. Provides good accuracy for linear and mildly nonlinear systems.

(b) Generalized- α Method

An extension of Newmark with improved numerical damping of high-frequency modes. Particularly suited for flexible structures where unimportant high-frequency vibrations should be suppressed.

(c) Implicit vs. Explicit Methods.

- Implicit integrators (e.g., Newmark, generalized- α) are stable for stiff systems but require iterative solutions.
- Explicit integrators (e.g., Runge-Kutta) are simpler but limited by small time steps due to stability constraints.

For flexible launcher dynamics, implicit methods are generally preferred, as they can handle the stiffness introduced by structural flexibility.

Numerical Implementation

Large Rotations and Small Deformations

A particular challenge in flexible multibody dynamics is handling large overall motion (e.g., vehicle rotation) together with small elastic deformations.

In the Floating Frame of Reference approach:

- The reference frame undergoes large rotations and translations,
- The elastic deformation is assumed to remain small, so linear elasticity still applies.

This separation allows the use of linear modal shapes (from FEM) while still capturing the nonlinear rigid-body motion of the launcher.

To handle rotations correctly, rotation parameterizations such as quaternions, Euler angles, or rotation vectors can be used.

Quaternions are often preferred for numerical stability and the avoidance of singularities.

Numerical Implementation

Verification and Validation (V&V)

Simulation results should always be verified and validated:

- Verification: check that the numerical implementation correctly solves the equations (e.g., energy conservation, constraint satisfaction).
- Validation: compare with experimental data or high-fidelity FEM results to ensure physical accuracy.

Typical checks include:

- Constraint violation norms (should remain near zero),
- Total energy evolution (should be bounded),
- Mode frequencies and damping ratios (should match analytical or test data).

Numerical Implementation

Typical Outputs and Analysis

From a flexible launcher simulation, typical quantities of interest are:

- Time histories of bending deformation and angular attitude,
- Stress and strain distributions at critical sections,
- Thrust vector deviation due to structural bending,
- Control system response under flexibility effects,
- Modal participation factors showing which modes are most excited.

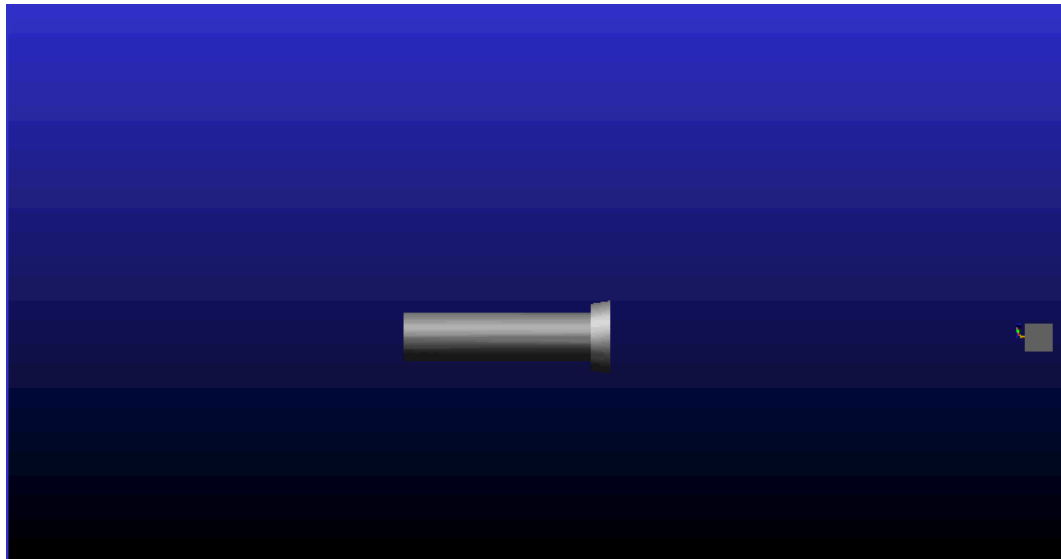
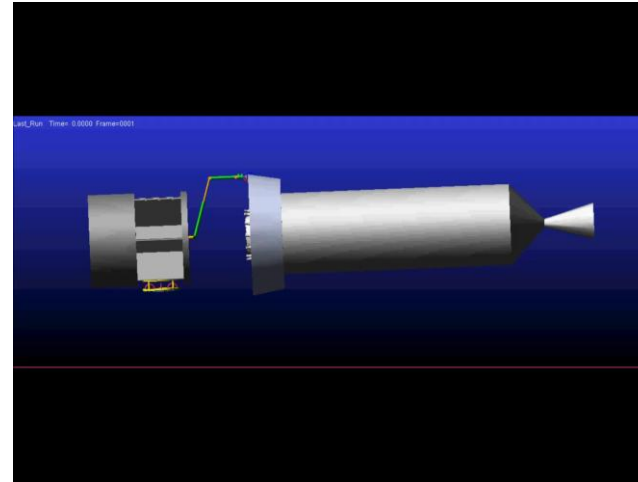
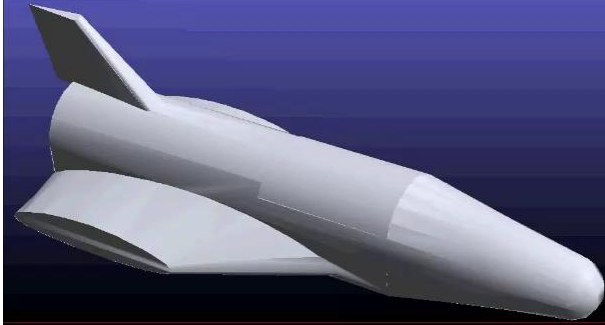
These outputs help engineers assess both structural and dynamic performance and optimize the design for stiffness, damping, and control authority.

Comparisono FEM and Multibody Approaches

Aspect	Full FEM	Multibody
Degrees of Freedom	Thousands–Millions	Tens–Hundreds
Accuracy	Very high (local stresses)	High for global dynamics
Computational Cost	High	Moderate
Suitable For	Structural design	Dynamic simulation, control integration
Model Input	Mesh, material properties	Modal data, mass and stiffness matrices

The multibody model provides an optimal balance between accuracy and computational efficiency, particularly suitable for system-level simulations of launch vehicles.

Uniroma1 - Esempio Deployment



Advanced
application
examples

Part 2 Launch Phase Modal and Stability Analysis Simulations

Motivations

- The impact of Aerodynamics and Sloshing on the Structural Stability cannot be neglected for the AOCS Determination in the LEOP Phase
- Simulations are carried out on a VEGA-like Launcher in the initial Launching Phase for evaluating such impacts

Stability of an Elastic Launcher

Rotation Equilibrium Equation

$$\mathbf{J}\ddot{\Theta} = M^{AER} + M^{SLOSH} + T\delta d_1 = J\mu_a\theta + \sum_k (M_i^L + M_i^M)\theta + T\delta d_1$$

where:

M^{AER} is the Moment of the Aerodynamic forces

M^{SLOSH} is the Moment due to the sloshing elements

M_i^L is the Moment due Liquid Oxygen Propellant

M_i^M is the Moment due to Methane

Actuator Equation: $\frac{\dot{\delta}}{K_A} + \delta = \epsilon = -k_p(\Theta - \Theta_0) - k_R\dot{\Theta}$

Stability of an Elastic Launcher

Modal-basis Elastic Equation

$$\ddot{A}_k + \omega_k^2 A_k + 2\zeta \omega_k \dot{A}_k = f_k = T \delta \phi(L)$$

Simulation Details

- Launcher Specificities:
 - Vega-C, like Launcher
 - 11 launcher element structure
 - Dual tank (Oxygen and Methane)
- Assumptions:
 - Masses at launch time: 210000 Kg
 - First stage fuel mass (P120C): 142000 kg.
 - First Stage empty mass: Circa 11000 kg.
 - Second stage (Zefiro 40) full mass: 36000 kg.
 - Second stage empty mass: 2500 kg

Digitare l'equazione qui.

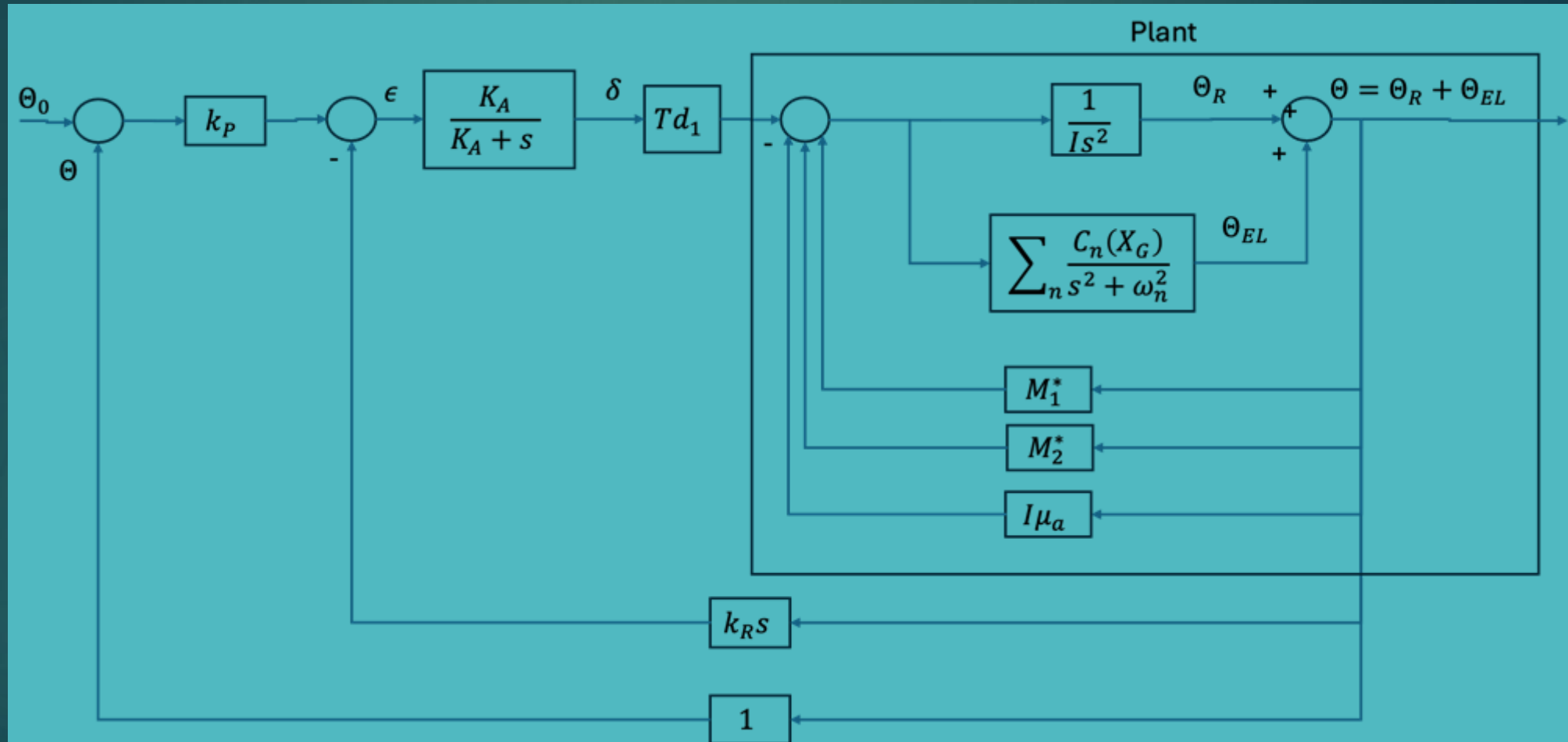


- Launch Phases:
 - $t=0$: launcher at the ground with full tanks
 - $t=145$ s: launcher after the first stage burnout

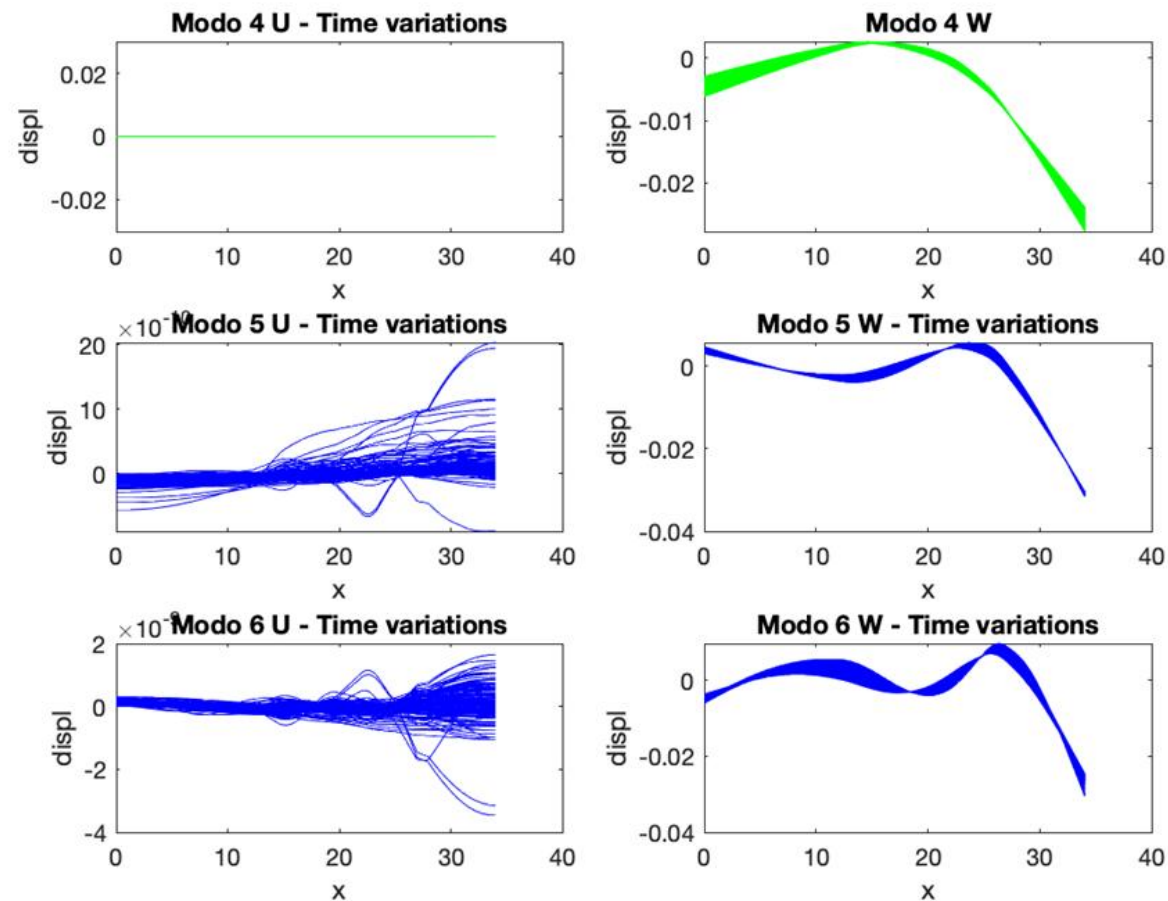
Simulation Details

- Simulation Steps:
 1. Sloshing analysis
 2. FEM Analysis to derive modes and natural frequencies
 3. Curve fitting of the first three modes
 4. Stability analysis for deriving the best Gyroscope position
- Testing Modes:
 - Simplified Model: Aereodynamic and Sloshing Moments are considered only on the measurements, but neglected into the rotation equation
 - Complete Model, where Aereodynamic and Sloshing Moments are fully taken into the rotation equation

Control System



FEM Analysis [0s, 140s]



Sloshing Analysis [0s, 140s]

Oxygen (Full Tank)

Masses	Heights
$m_0 = 703 \text{ Kg}$	$h_0 = 0.51 \text{ m}$
$m_1 = 207 \text{ Kg}$	$h_1 = 0.5 \text{ m}$
$m_2 = 6.23 \text{ Kg}$	$h_2 = 0.82 \text{ m}$
$m_3 = 1.48 \text{ Kg}$	$h_3 = 0.89 \text{ m}$

$$\omega_1 = 6 \text{ rad/s}$$

$$\omega_2 = 10.2 \text{ rad/s}$$

$$\omega_3 = 12.9 \text{ rad/s}$$

Methane (Full Tank)

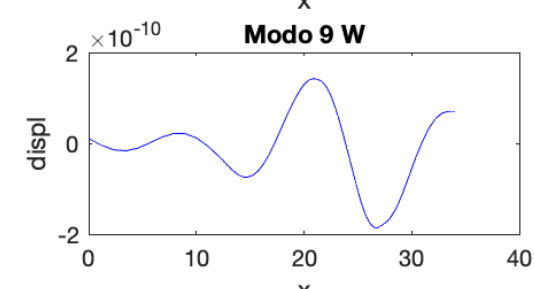
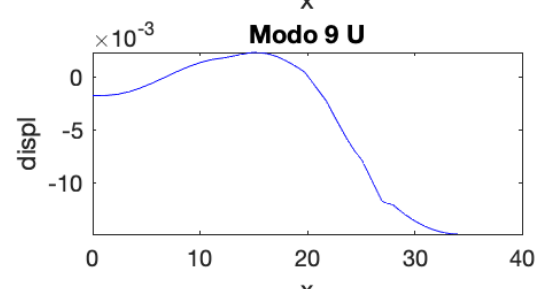
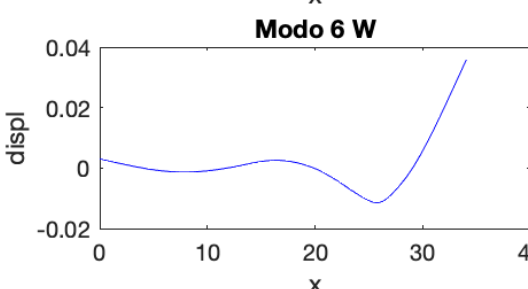
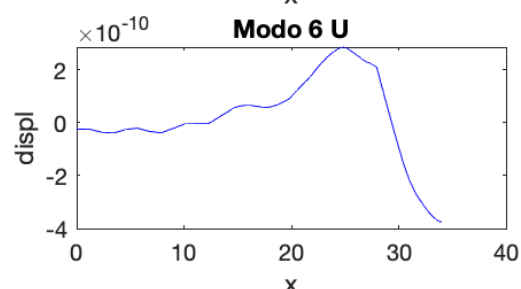
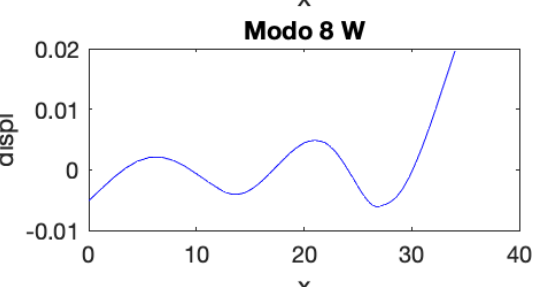
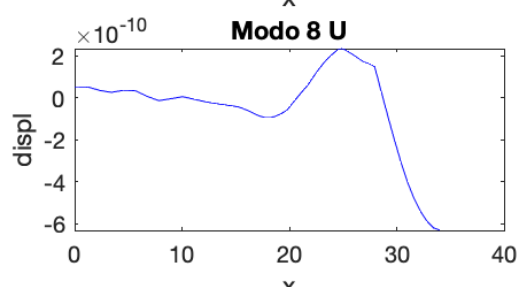
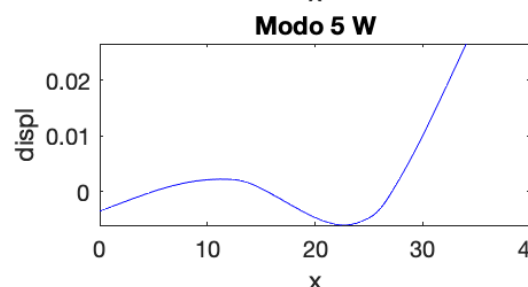
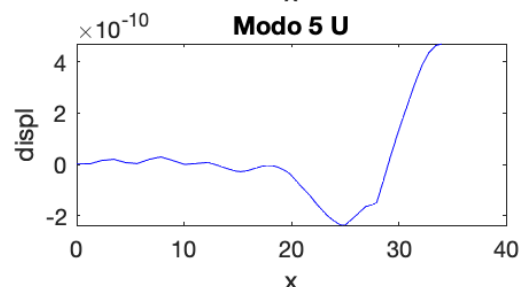
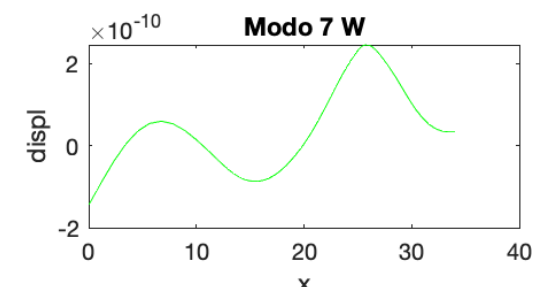
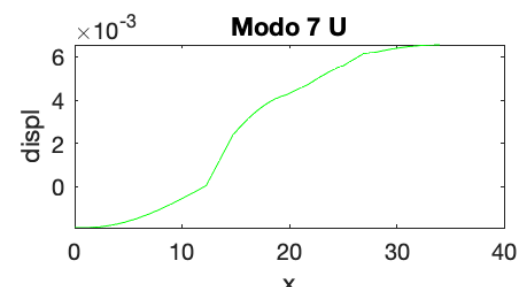
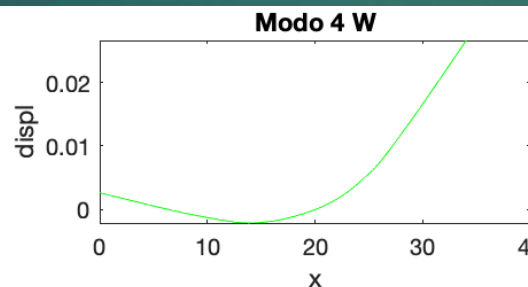
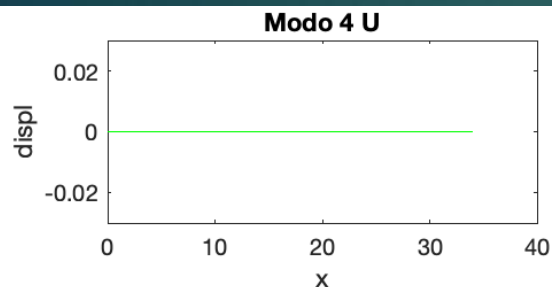
Masses	Heights
$m_0 = 0.4 \text{ Kg}$	$h_0 = 0.5 \text{ m}$
$m_1 = 0.11 \text{ Kg}$	$h_1 = 0.78 \text{ m}$
$m_2 = 0.0036 \text{ Kg}$	$h_2 = 0.93 \text{ m}$
$m_3 = 0.0009 \text{ Kg}$	$h_3 = 0.96 \text{ m}$

$$\omega_1 = 6 \text{ rad/s}$$

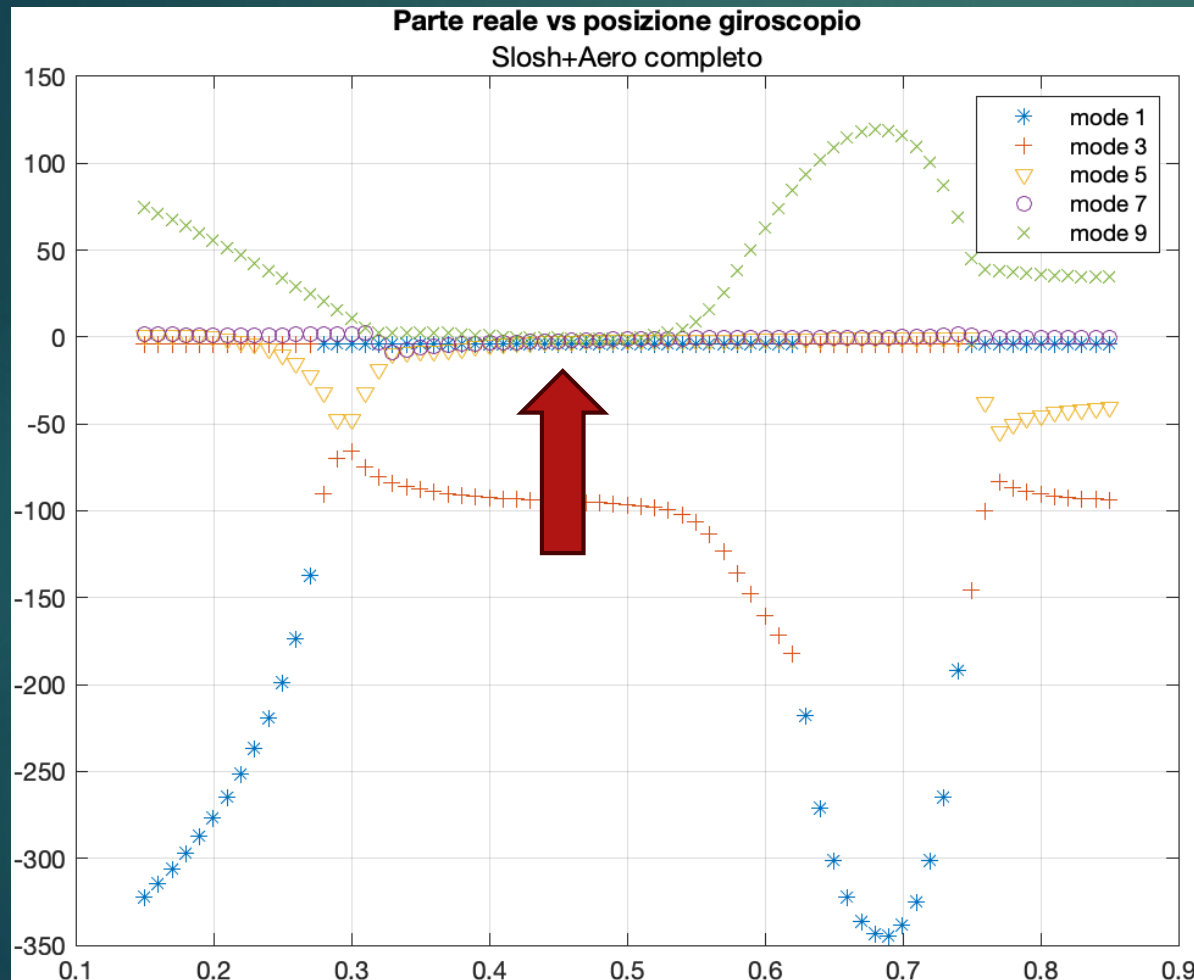
$$\omega_2 = 10.2 \text{ rad/s}$$

$$\omega_3 = 12.9 \text{ rad/s}$$

Mode Analysis (t=0s)



Stability Analysis



- $K_a = 20$
- $K_p = 200$
- $K_r = 50$
- Gyroscope position between in the 40%-50% range

Sloshing Analysis [145s]

Oxygen (75% Tank)

Masses	Heights
$m_0 = 510 \text{ Kg}$	$h_0 = 0.43 \text{ m}$
$m_1 = 206 \text{ Kg}$	$h_1 = 0.38 \text{ m}$
$m_2 = 6 \text{ Kg}$	$h_2 = 0.69 \text{ m}$
$m_3 = 1.5 \text{ Kg}$	$h_3 = 0.68 \text{ m}$

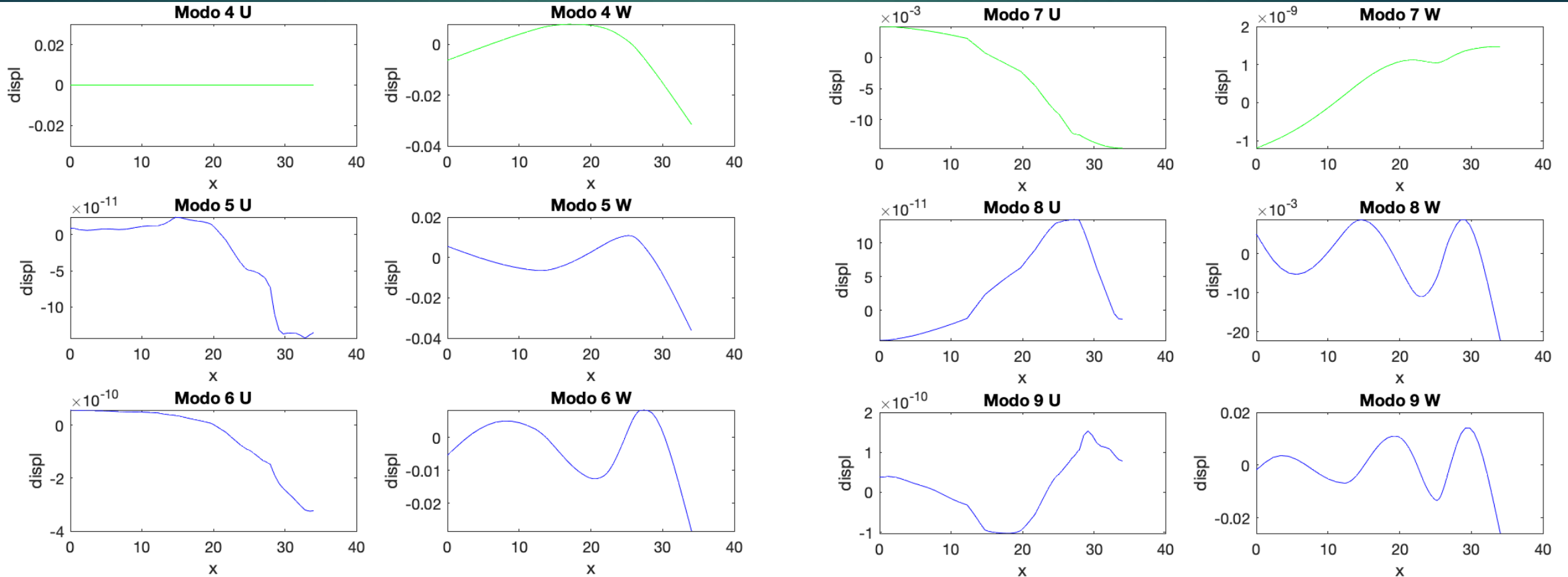
$$\omega_1 = 11.6 \text{ rad/s}$$
$$\omega_2 = 19.86 \text{ rad/s}$$
$$\omega_3 = 25.13 \text{ rad/s}$$

Methane (75% Tank)

Masses	Heights
$m_0 = 0.3 \text{ Kg}$	$h_0 = 0.42 \text{ m}$
$m_1 = 0.11 \text{ Kg}$	$h_1 = 0.32 \text{ m}$
$m_2 = 0.0036 \text{ Kg}$	$h_2 = 0.61 \text{ m}$
$m_3 = 0.001 \text{ Kg}$	$h_3 = 0.68 \text{ m}$

$$\omega_1 = 11.6 \text{ rad/s}$$
$$\omega_2 = 19.85 \text{ rad/s}$$
$$\omega_3 = 25.12 \text{ rad/s}$$

Mode Analysis (t=145 s)



Conclusions

- Mode Analysis and Stability Analysis on an 11 elements Vega-C have been carried out
- Simplified Models and Complete Model, with Aerodynamic and Sloshing Moments fully taken into account, are analysed
- The best solution is obtained $K_a=20$, $K_p=200$ and $K_r=20$, with position of the attitude sensor between 40% and 50% of the launcher
- After the burnout, the reduction of the impact of external forces and the reduction of propellant leads to reduced control gains
- The best results are obtained for $K_a=20$, $K_p=50$ and $K_r=50/4$, with position of attitude sensor, after 80% of the launcher
- Impacts on AOCS are not negligible and have to be considered in the LEOP Phase