Trajectory Modeling Methods for Supersonic Separation

H. AZEMA, A. BAS, A. ABAUZIT, A. BRISSAUD

Abstract

Hypersonic vehicles usually use initial boost systems leading to a high speed and dynamic pressure separation phase. Accurately replicating this separation phase through ground testing is very complex due to the significant mechanical and aerodynamic constraints. To minimize the number of real scale flight tests required to develop the separation solution, numerical simulations become essential.

Using the work of Tartabini (2011) [1], MBDA France developed a numerical simulation tool called SPLITS (*Simulation Physique de Largage InTer Solides*) able to compute the trajectories of an arbitrary number of mechanically linked objects. By combining a precise and dense mesh of CFD computations and analytical solving of the stress at mechanical joints using CFE methodology, it enables precise evaluation of all solids' relative kinematics during the separation phase. SPLITS was designed to be highly modular, with joints between solids potentially evolving during the simulation, reflecting loss of contact or breaking of an element. Moreover, the tool can monitor forces and moments at contact joints, helping engineers design a robust concept. From a more industrial point of view, modular-use in different projects is simplified with an unchanged simulation core. Only aerodynamics models and liaisons graph are to be updated using standard interfaces.

Once the design has converged, SPLITS can also be used in ground test, reducing or deleting the aerodynamic wrench and adding elements to match the test configuration. The ground tests results can then be compared to dedicated numerical simulations, which helps validate and adjust the models before flight tests. Finally, in-flight measurements can be used to correct any remaining discrepancies between simulation and reality.

Keywords: *Separation, Hypersonic, Model, SPLITS, LEA.*

Nomenclature

A, B – Rigid body A and rigid Body B
$\bar{A}, \bar{B}$ – Joint location in body A and body B
CFE – Constraint Force Equation
C_i – Aerodynamic coefficients: axial, lateral and normal forces, roll pitch and yaw moments
ΔC_i – Aerodynamic interactions coefficients, obtain by substract free moving body coefficients to total aerodynamic coefficients
$F_A^{(CON)}, F_B^{(CON)}$ – Joint constraint force vector of body A and body B
$F_A^{(EXT)}, F_B^{(EXT)}$ – External force vector applied to body A and body B
I_A, I_B – Inertia tensor of body A and body B
SPLITS – *Simulation Physique de Largage InTer Solides*
$T_A^{(CON)}, T_B^{(CON)}$ – Joint constraint torque vector of body A and body B
$T_A^{(EXT)}, T_B^{(EXT)}$ – External force vector applied to body A and body B
WBI – Weight, Balance and Inertia
e_A, e_B – Unit vector linked to body A and body B
q_A – Dynamic pressure of the body A
$\varphi_A, \theta_A, \psi_A$ – Euler angles of the body A
$Mach_A$- Mach number of the body A
m_A, m_B – Mass of body A and body B
r_A, r_B – Inertial position vector of $\bar{A}$ and $\bar{B}$
x_A, x_B – Inertial position vector to mass center of body A and body B
$\ddot{x_A}, \ddot{x_B}$ – Linear acceleration vector of body A and body B mass center
α_A – Angle attack of the body A with respect to the wind
β_A- Sideslip angle of the body A with respect to the wind
η – Baumgarte error control parameter
ρ_A, ρ_B – Position vector from body A mass center to point $\bar{A}$ and body B mass center to point $\bar{B}$

ω_A, ω_B – Angular velocity vector of body A and body B, relative to inertial reference frame

$\dot{\omega}_A, \dot{\omega}_B$ – Angular acceleration vector of body A and body B, relative to inertial reference frame

1. Introduction

Hypersonic vehicles are usually equipped with an initial acceleration system that allows them to reach a compatible flight point for the next part of their trajectory, at high supersonic or hypersonic speeds. The separation of this acceleration system is a crucial event that must be studied and characterized with precision to optimize the performance of the entire system.

However, the real conditions of separation are complex to reproduce in ground tests with the aim of risk reduction or validation. Different types of tests and their advantages/disadvantages can be distinguished:

- Wind tunnel aerodynamic test: allows for Mach number similarity and helps obtaining the interaction field, but cannot use the real mechanical systems at a reduced scale;
- Ground mechanical tests: can use and test real mechanical equipment, but external efforts are not representative of the actual flight point;
- Numerical simulations: are able to quickly and efficiently parameterize different trajectories and solutions (at low cost), but their representativeness is limited by the precision of the modelled phenomena.

The SPLITS software has been developed to simulate the separation between a vehicle and its acceleration system by calculating the kinematics of all concerned objects. It is built to support the design of a separation system throughout all industrial process steps, from the initial creation of the kinematics scheme to the flight tests.

It is able to:

- Consider the aerodynamic forces on each of the bodies, previously characterised through simulations or wind tunnel tests;
- Simulate the movements of the vehicles during separation, as well as calculating the forces at the mechanical joints between solid objects;
- Add external forces to simulate actuators (jacks, etc.).

This modelling requires a detailed understanding of the aerodynamics of each of the bodies, with particular attention given to the variations during the early phases of the separation.
The theoretical equations used to calculate the movements and forces transiting the different joints have been studied and explained, in particular in [1]. The goal of this article is to present a tool based on these equations, whose modularity allows it to adapt to initial pre-studies and the entire range of ground-tests conducted before flight.

In the initial design phases, the goal is to iterate quickly and easily through different types of joints to evaluate the strengths and weaknesses of each solution. As the industrial solution matures, each new constraint can be added quickly to the overall modelling, and its impact on the trajectory evaluated. During the ground-test phases, specific variations of the model can be developed (suppress aerodynamic forces, add elements dedicated to the test, etc.) in order to confirm the validity of the numeric modelling while providing design elements to the test teams.
Finally, during the flight tests, the predictability of the trajectory is evaluated and the models are recalibrated with the help of real flight conditions measures.

2. SPLITS Implementation Methodology

2.1. CFE methodology

This paragraph will present the equations used to compute the dynamics of an N-body system connected by L arbitrary joints.

The chosen method, called the Constraint Force Equation (CFE), computes the forces that all moving bodies will experience, their movements throughout the entire separation phase and the constraints through each joints. These results will depend on the external forces (aerodynamic, additional efforts), the properties of the two bodies (center of gravity, inertias) and their joints properties.

The CFE method thus sums several terms: the external forces, whose resultant force and moment will be reduced to the center of gravity of each solid, and the constraint forces, whose force and moment will be reduced to the point of contact of each joint. The action-reaction law imposes that these forces be of opposite signs and equal magnitude for the two solids to which they are applied.

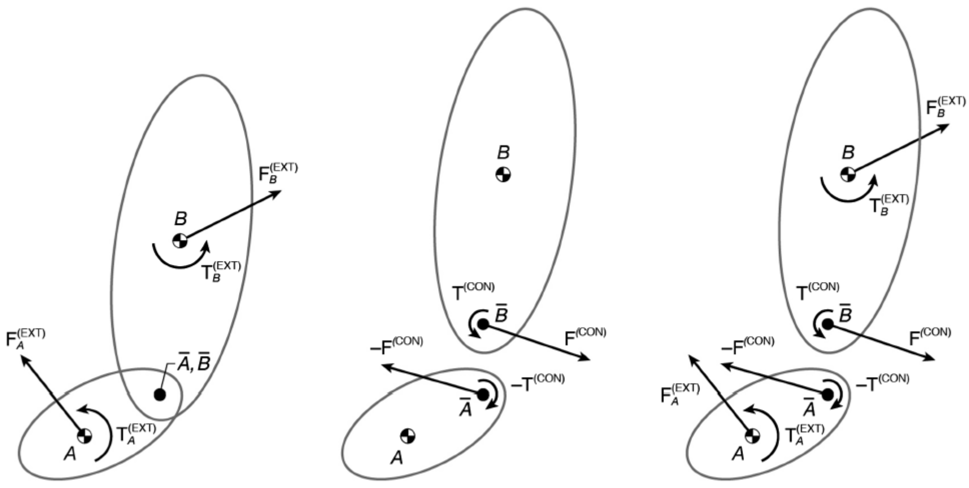

Fig. 1 : Illustration of the forces considered in the CFE method [1]

First, we can simply write the equations of a constrained motion of a body A subjected to an arbitrary number of joints from Newton's second law. All equations are written in the local reference frame, considered Galilean:

$$F_A^{(EXT)} + F_A^{(CON)} = m_A \ddot{x}_A \tag{1}$$

$$T_A^{(EXT)} + \sum \rho_A \times \mathrm{F}_A^{(CON)} + T_A^{(CON)} = I_A \cdot \dot{\omega}_A + \omega_A \times I_A \cdot \omega_A \tag{2}$$

where ρ_A is the position vector from the center of gravity of A to the point $\bar{A}$ where the constraint force is applied.

From these vector equations we obtain 6*N distinct scalar equations.

We will then derive the equations specific to the joints. Newton's third law immediately provides two action-reaction vector equations, which add 6*L scalar equations to the system.

$$F_A^{(CON)} + F_B^{(CON)} = 0 \tag{3}$$

$$T_A^{(CON)} + T_B^{(CON)} + (r_B - r_A) \times F_B^{(CON)} = 0 \tag{4}$$

The $(r_B - r_A)$ term, where r_a is the position vector of the local frame to contact point $\bar{A}$, represents additional torque only if translation is allowed.

Finally, the last equations concern the degrees of freedom of each joint and must therefore be adapted to each specific problem. They allow for the cancellation of internal constraints at the degrees of freedom of the joint if the axis is free in translation or rotation, or for preventing movement in the opposite case. Therefore, for each joint, 6 scalar equations need to be added. These are expressed as follows:

- For each free translation along an axis e, we can write :

$$F^{(CON)} \cdot e = 0 \tag{5}$$

- For each free rotation along an axis e, we can write :

$$T^{(CON)} \cdot e = 0 \tag{6}$$

- For each blocked translation along an axis e_A for 2 bodies A and B we can write :

$$\frac{d^2}{dt^2}[(r_B - r_A) \cdot e_A] = 0 \tag{7}$$

$$\Leftrightarrow$$

$$(\ddot{x_B} + \dot{\omega}_B \times \rho_B - \ddot{x_A} - \dot{\omega}_A \times \rho_A) \cdot e_A + (r_B - r_A) \cdot (\dot{\omega}_A \times e_A)$$
$$= 2(\dot{r}_B - \dot{r}_A) \cdot (e_A \times \omega_A) - (r_B - r_A) \cdot \big(\omega_A \times (\omega_A \times e_A)\big) + [(\omega_A \times (\omega_A \times \rho_A) - (\omega_B \times (\omega_B \times \rho_B)] \cdot e_A$$

- For each blocked rotation between two axis e_A and e_B for 2 bodies A and B we can write

$$\frac{d^2}{dt^2}[(e_A \cdot e_B)] = 0 \tag{8}$$

$$\Leftrightarrow$$

$$(\dot{\omega}_B - \dot{\omega}_A) \cdot (e_B \times e_A) = (\omega_B - \omega_A) \cdot [e_A \times (\omega_B \times e_B) - e_B \times (\omega_A \times e_A)]$$

Equation (7) et (8) can be rewritten as $\ddot{g} = 0$, the solution of which are known to induce drifts due to the accumulation of numerical errors. As a result, one may observe the displacement of points that are supposed to be stationary or misalignments that amplify over time.

The Baumgarte stabilization method involves replacing this type of equation with the modified system $\ddot{g} + 2\eta\dot{g} + \eta^2 g = 0$, which naturally returns the various points to their theoretical positions. The damping coefficient η must then be tailored to the specific problem.

However, given the short durations of the phenomena modeled in our software, the observed numerical drift remained negligible and did not justify the significant increase in both the complexity of the equations and the computation time that this modification would cause. It was included in the matrix system in case future studies deemed it necessary, but unless otherwise specified the cases presented were conducted with $\eta = 0$.

2.2. Matrix System

This set of constraints creates a system of 6*N+12*L equations for the same number of unknowns. The solutions are the accelerations and angular accelerations of each body, as well as each of the constraint forces and moments at the joints. The entire problem can be expressed in the form of a matrix equation Ax=B, for which modern softwares are easily able to find a numerical solution.

However, creating this matrix is a tedious operation and a potential source of errors. The standardization of its creation and the use of Matlab's symbolic computation modules allow a simplification of this procedure and minimize the risks of errors. It also helps achieving greater flexibility and makes the user able to modify quickly the degrees of freedom considered. Different routines are thus created, allowing from the number of solids, the joint matrix, and the characteristics of each joint, to quickly recreate the matrix corresponding to an arbitrary system, minimizing the risks of errors.

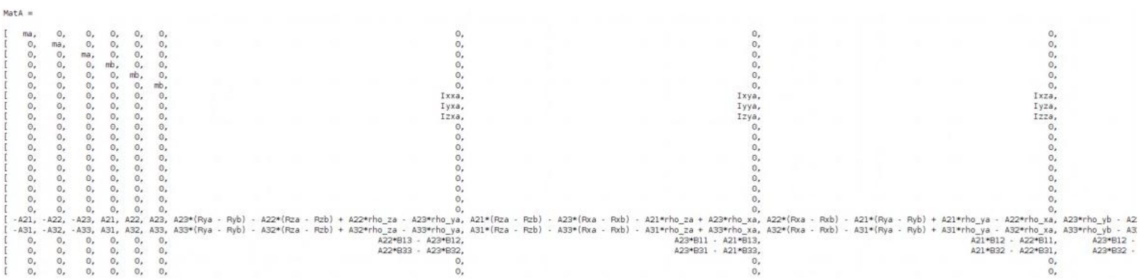

Fig. 2 : Excerpt of a constraint matrix for a four bodies problem

The external forces considered in the resolution of the equation system below include aerodynamic forces and gravity, but it is also possible to add different additional forces to model the effects of other systems (jacks, pushers, etc.) or recall/friction forces at the joints. An example will be presented in chapter 3.

It should be noted that the method in no way assumes the isostatism of the problem. With all joints being perfect and the solids non deformable, it falls to the user to wisely choose the graph to model and the approximations potentially necessary to allow movements that would only be possible by exploiting assembly clearances in reality.

If the modeled joints are mathematically perfect, it is also possible to approximate bending or failure behavior by adapting the orientation of the joints based on the forces taken up by the joint, or by removing the joint if these forces exceed a predefined threshold.

2.3. Aerodynamic Modeling

During the stage separation between the vehicle v (the carrier) and its acceleration system l (the left element), it can be required to model both aerodynamic tensors in order to get trajectories as precise as possible to mitigate risks during this stage.

Thus, the aerodynamic modelling can be defined as a decomposition of the total aerodynamic tensor T^{Total} of the moving body i such as:

$$\mathrm{T}_i^{Total} = \mathrm{T}_i^{Free\ moving\ body} + \mathrm{T}_i^{Interaction}$$

where $\mathrm{T}_i^{Interaction}$ represents the aerodynamic tensor of the interactions, and it is defined as:

$$\mathrm{T}^{Interaction} = \left(C_i^{Total} - C_i^{Free\ moving\ body}\right) = (\Delta C_i)$$

Applied to the acceleration system l for example:

- $\mathrm{T}_l^{Free\ moving\ body}$ the aerodynamic tensor of l without the vehicle v;
- $\mathrm{T}_l^{interaction}$ the aerodynamic tensor of interactions of v on l.

This decomposition allows handling interactions between the moving bodies separately. On the one hand the free moving body tensor of the vehicle is already available for other purposes (performance trajectories), and on the other hand the interactions modeling can be specifically adapted to the studied separation.

As an example, if the acceleration system has control surfaces on the rear of the body, the aerodynamic effects of these surfaces can only be modelled in the free moving body tensor of the body and not in the interactions tensor. This kind of hypothesis can lead to two separate models: free stream and interactions for the two bodies. Adapting the modeling to the separation problem lowers the cost of the models and provides higher fidelity of the results. Having two models also implies that both of them can be developed separately, and data could be obtained with different methods: wind tunnel tests can be performed specifically to get interactions tensors with captive trajectory system (CTS) while the free moving tensors can be obtained with adapted numerical campaigns (CFD) on the vehicle and its acceleration system.

For the two bodies, and a six degrees of freedom (6-DOF) problem, the proposed modeling can be:

$$C_i^{Free\ moving\ body} = f(Mach_i, \alpha_i, \beta_i, q_i, \dots)$$
$$\Delta C_i = f(Mach_v, \alpha_v, \beta_v, X_l, Y_l, Z_l, \varphi_l, \theta_l, \psi_l)$$

The interactions coefficients ΔC_i can only be dependent of the vehicle flight conditions $(Mach_v, \alpha_v, \beta_v)$, and the relative positions and orientations of the acceleration system l with respect to its original position from the vehicle v $(X_l, Y_l, Z_l, \varphi_l, \theta_l, \psi_l)$.

Linearization of some effects can be proposed, for example for a three dimensions' problem where the vehicle incidence is fixed to a reference incidence α_{ref}:

$$\Delta C_i(Mach_v, \alpha_v, X_l, Z_l, \theta_l) = \Delta C_i\left(Mach_v, \alpha_v = \alpha_{ref}, X_l, Z_l, \theta_l\right) + \Delta_\alpha C_i(Mach_v, \alpha_v, X_l, Z_l, \theta_l)$$

with, $\Delta_\alpha C_i$ the effect of the vehicle incidence on the interactions coefficients:

$$\Delta_\alpha C_i(Mach_v, \alpha_v, X_l, Z_l, \theta_l) = \Delta C_i\left(Mach_v, \boldsymbol{\alpha_v} \neq \boldsymbol{\alpha_{ref}}, X_l, Z_l, \theta_l\right) - \Delta C_i\left(Mach_v, \boldsymbol{\alpha_v} = \boldsymbol{\alpha_{ref}}, X_l, Z_l, \theta_l\right)$$

and where $\Delta C_i\left(Mach_v, \boldsymbol{\alpha_v} = \boldsymbol{\alpha_{ref}}, X_l, Z_l, \theta_l\right)$ can be seen as the main interaction coefficients.

Having linearized effects provides flexibility in the modeling. As such, more positions close to the vehicle can be calculated and be used for the main interaction coefficients, while fewer points can be used for the incidence effects: This approach provides higher fidelity in the area of interest while optimizing the cost of the models.

2.4. Global Simulink Architecture

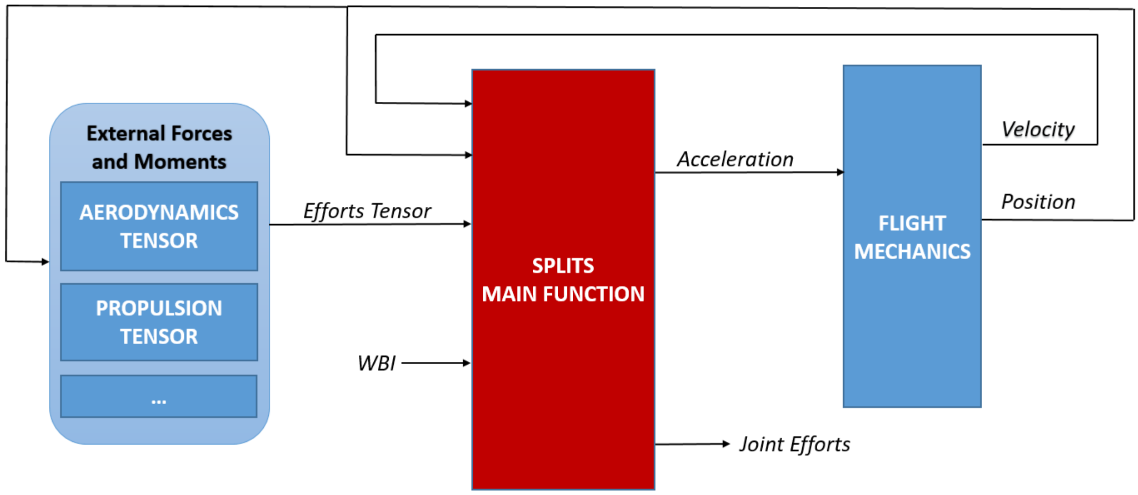

Fig. 3 : SPLITS Matlab-Simulink Architecture

External forces and moments are calculated with dedicated blocks. It includes the resultant of aerodynamic, propulsion and each contributor to external forces and moments. The output wrench mainly depends on the solid's relative position and attitude.

SPLITS main function takes weight, balance and inertia (WBI) as input which can be constant or a function of another parameter such as propulsion or flight phase. The other input are the angular and linear velocities and positions. The function calculates the linear and angular accelerations which are sent to flight mechanics block. The function which is a matrix system solved each step time also calculates internal constrained joints efforts and moments. These wrenches are part of the model outputs and allows a fine dimensioning of the system.

Flights mechanics block has two main purpose:

- Calculate the velocity of each solid with a first integration;
- Calculate the position of each solid with a second integration.

These outputs are re-injected in the main function and close the loop.

This Matlab-Simulink architecture is highly modular and fast to implement. Two study cases of the same system will be exposed in the next part and will show the whole range of applications of this model, from a 150 milliseconds stage separation to a complete 50 seconds trajectory.

3. LEA hypersonic demonstrator study case

LEA is a vehicle built to demonstrate hypersonic scramjet capabilities.

The vehicle has been designed to reach hypersonic speeds thanks to a 3-stage boost phase, perform a hypersonic separation, ignite its scramjet and demonstrate a positive aeropropulsive balance.

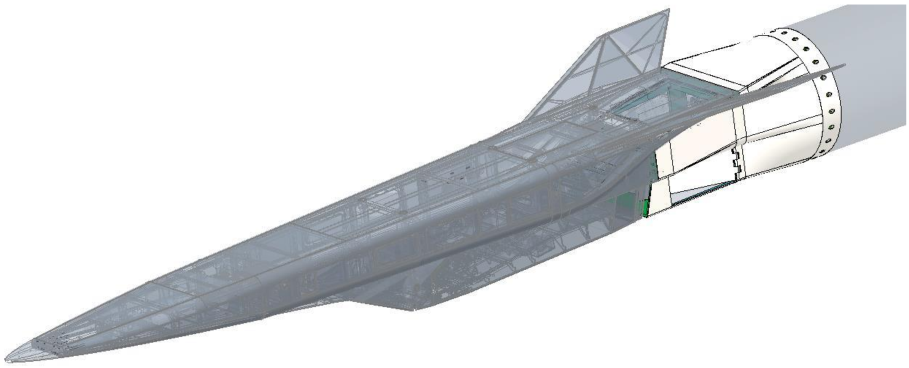

Fig. 4 : LEA and ISS illustration

Fig. 4 shows LEA in grey on the left, the interstage (ISS) in white and the tip of the 3rd booster in grey on the right.

This high speed and high altitude separation between ISS and LEA is a critical flight phase. To perform this separation, a pusher system has been conceived. This pusher is an air cylinder located in the ISS. At the tip of the 3rd stage phase, the separation sequence starts. The cylinder expands up to 300mm and pushes on LEA's nozzle as shown in Fig. 5.

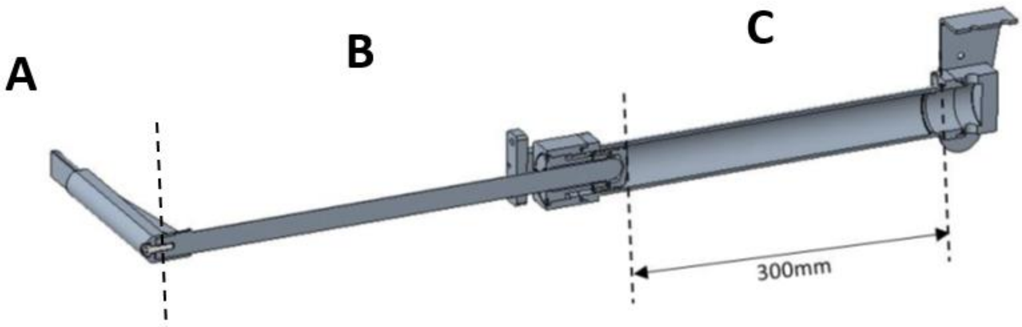

Fig. 5 : Pusher air cylinder in full expansion configuration, cross-sectional view

The sequence has to be precisely adjusted for several reasons:

- Masses of the two splitting bodies are around the same. Action and reaction forces and torques have a huge impact on the final LEA trajectory;
- LEA has no autopilot; all its control surfaces are static. We do not want to introduce strong oscillations in LEA attitude after separation;
- LEA is unstable with scramjet off. The scramjet ignite sequence must be precisely timed.

Modeling separation phase is mandatory to reach the scramjet objectives. This is where SPLITS model brings confidence and helps to choose the right system parameters such as the pusher orientation, pusher force, incidence targeted for separation, scramjet fuel injection profile, etc.

The first step was to determine the kinematic screw between LEA and the pusher. On the real equipment Fig. 5, two tips of the part A in contact with LEA can rotate around their axis. The inner tube B can also rotate and translate in the outer tube C. While the AB-C interface is a universal cylindrical joint, the A-LEA interface has to be simplified in a simple isostatic revolute joint. The A-LEA joint initial position is placed in the middle of the A tube as shown in Fig. 6.

Next figure shows the simplified isostatic kinematics implemented in SPLITS model.

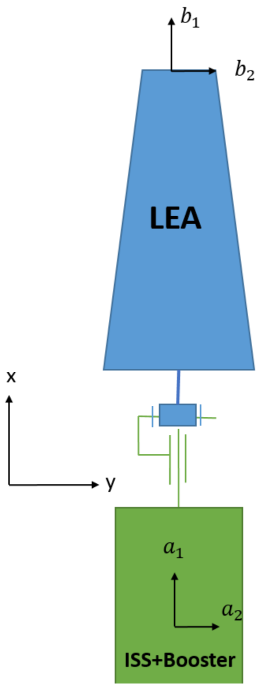

Fig. 6 : Kinematics scheme of the LEA separation configuration

The other part of global interactions are the aerodynamic interactions. The two splitting vehicles generate numerous complex oblique shocks, as shown in Fig. 7.

Fig. 7 : aerodynamic interactions illustration between LEA (left) and ISS (right)

With the methodology described in chapter 2.3, we are able to finely capture these phenomena. The final LEA final trajectory will depend on both perturbation contributors: mechanical and aerodynamic.

Ground test validation:

The separation system has been tested on the ground. This test was performed with the 3rd LEA's section and the ISS – which are linked with the pusher joint. LEA is vertically set on the test rig, the pusher expands and pushes the ISS down on the ground (Fig. 8).

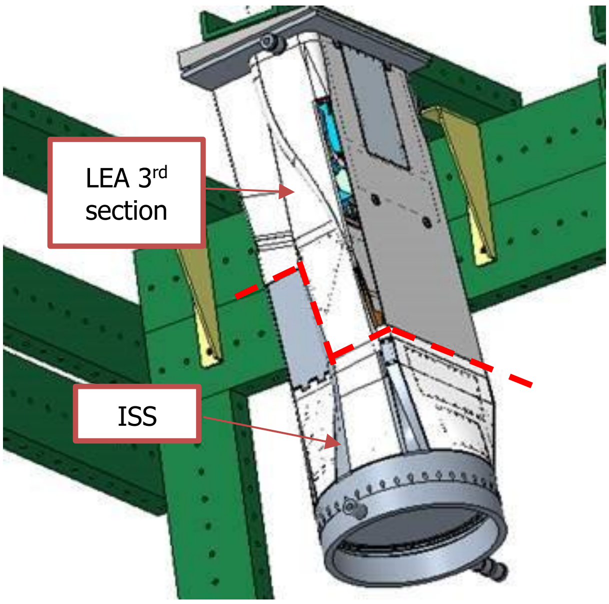

Fig. 8 : Separation vertical test rig

The objective was to validate functional capabilities of the system. We had to choose test parameters which matched the inflight relative position and attitude of the two solids. The only parameters we could play on were:

- Field of gravity orientation;
- Pusher orientation;
- Pusher force.

To choose these parameters, SPLITS software has been modified to match the test configuration.

On top of that, the test's simulation will validate SPLITS software.

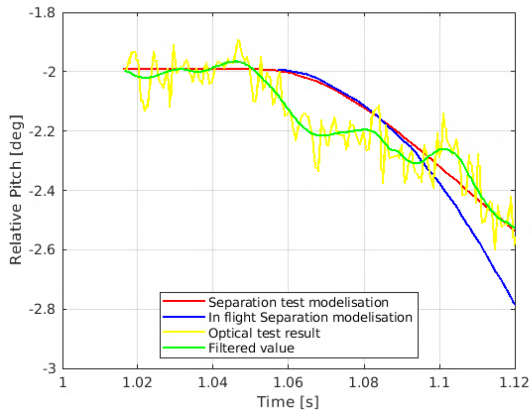

Fig. 9 : Separation Test Results: LEA-ISS relative pitch

Fig. 9 shows three different curves. Blue line shows inflight simulated relative LEA-ISS pitch. Red line is the simulated test which matches the blue line in the first milliseconds. Yellow line is the ground test result.

The angle measurement is made with a camera and leads to a medium resolution. Except for the 0.2° delta at 1.06 s, the test result line (yellow) mostly matches simulation line (red).

Three other tests have been performed: the simulation matched the test results too.

This test allowed us to validate SPLITS dynamic joint implementation.

Flight simulation example:

This section shows an example of LEA flight simulation from ISS separation to engine cut-off.

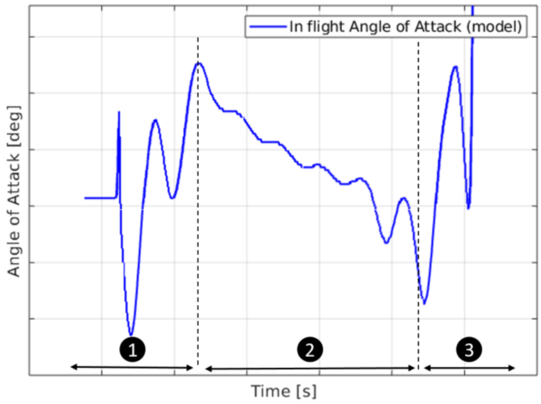

Fig. 10 : Angle of Attack function of flight time

Fig. 10 shows oscillations on the incidence caused by the 3rd stage separation and the engine ignite (1). Oscillation is dampened when the scramjet equivalence ratio progressively increases (2). The engine is then switched off and the unstable vehicle starts amplifying its oscillations (3).

This result shows a complete LEA flight with a classic trajectory simulation combined with a precise calculation of the separation phase.

Monte-Carlo simulations are then run to calculate risks of failure of the scenario.

4. Conclusion

SPLITS model has been implemented to model a complex separation. The main innovation lies in its high degree of modularity, which allows a quick adaptation to several mechanical configurations. A separation phase can now be treated in an exhaustive way with both aerodynamic and mechanical interactions. The tool can be used for two complementary purpose: trajectory establishment and joint efforts monitoring. SPLITS has been validated both numerically and physically through numerical simulation and ground tests. LEA demonstrator and its hypersonic separation has been our first SPLITS use-case.

References

1. Tartabini, P. V., Roithmayr, C. M., Toniolo, M. D., Karlgaard, C., and Pamadi, B. N., Modeling Multibody Stage Separation Dynamics Using Constraint Force Equation Methodology (2011)