Politecnico di Milano

School of Industrial and Information Engineering

Master Degree in Space Engineering

FLEXIBLE DEVICES FOR ACTIVE SPACE DEBRIS REMOVAL: THE NET SIMULATION TOOL

Supervisor: Prof. Michéle LAVAGNA
Co-Supervisor: Dr. Riccardo BENVENUTO

Master Thesis of:
Samuele SALVI 783543

Academic Year 2013 - 2014

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Space debris mitigation and remediation are urgent and growing issues to be faced for the future of space operations and exploitation durability. The Active Debris Removal topic focuses on trading-off, designing and building up mechanisms to be mounted on board of active chasers that can rendezvous and interact with uncooperative tumbling targets, either transferring them on a graveyard orbit or ensuring a safe disposal. In contrast to rigid capture mechanisms, tethered-net solutions are characterized by a safe capturing distance, a passive angular momentum damping effect and the highest flexibility to whichever unknown shape, material and attitude of the target to interface with.

This thesis work provides a complete physically based mathematical model (*NeST*), to simulate the entire multibody system dynamics during the capture and the disposal, in order to support the system design. A particular attention is put on the flexibility and interfaceability of the simulator to whichever user defined inputs and external third-party software. Noteworthy is the built-it collision detection and contact algorithms, optimized ad-hoc to increase the simulator performance. Further numerical and model optimizations are presented and validated, analyzing their impact on the precision and the benefits on the computational speed. The simulator is finally numerically validated and an intensive analysis campaign is done to characterize the tether-net system dynamics, with the ultimate aim of scaled tests in micro gravity environments. Results of the capture analyses in the real operative scenario let to appreciate the higher fidelity level to the physics of the problem, compared to the simplified simulators available. The energy dissipated at contact, able to passively reduce both the linear and the angular momentum content of the tumbling target, is a definitive example of the great role played by this numerical tool. These performances could not be otherwise predicted with such precision. This confirms the importance of *NeST* for this ADR concept, to pave the way in the international Clean Space initiative trade-off.

Keywords:
Active space Debris Removal, flexible structures, tether-net systems, collision detection and contact, multibody problem, numerical optimizations.

SOMMARIO

La mitigazione e la riduzione del numero di detriti spaziali rappresentano questioni urgenti da affrontare per poter garantire il futuro dello spazio e il suo utilizzo. La rimozione attiva dei detriti richiede fasi di studio, scelta della tecnologia e il suo sviluppo. Tali sistemi vengono progettati per essere montati su satelliti controllati, in grado di raggiungere oggetti non cooperanti e con assetto talvolta difficile da prevedere e infine garantire il loro trasferimento in un orbita cimitero o un completo rientro controllato in atmosfera. Differentemente dai sistemi rigidi, la soluzione cavo-rete garantisce un'interazione a distanza, uno smorzamento passivo dell'oscillazione, ma soprattutto la massima flessibilità di azione per qualsiasi forma, materiale e assetto del detrito.

Questo lavoro ha come obiettivo la costruzione di un modello matematico completo (*NeST*), in grado di ben rappresentare tutte le dinamiche in gioco nelle fasi di cattura e rientro in atmosfera, poi integrato in un simulatore per supportare le fasi di progetto del sistema. Un'attenzione particolare è data alla fase di collisione e contatto della rete, in questo lavoro vengono presentate diverse strategie numeriche progettate specificatamente per ottimizzare le prestazioni, in termini di precisione e velocità di integrazione. Viene data inoltre una struttura modulare al codice in modo da potersi interfacciare con il più generico ingresso deciso dall'utente ed eventuali librerie esterne in sostituzione di ogni modello. Viene fornito un piano di validazione numerica e sperimentale. Per supportare quest'ultima, è stata fatta un'approfondita analisi di caratterizzazione del sistema rete sia in ambiente operativo che in microgravità. Dai risultati delle analisi di cattura si può apprezzare la qualità della modellazione e la sua fedeltà alla fisica del problema, soprattutto se confrontati con simulatori attuali disponibili, ben più semplificati. L'energia dissipata dall'interazione di contatto e scivolamento della rete, in grado di smorzare notevolmente il moto non cooperante dell'oggetto, è un esempio del ruolo importante svolto da *NeST*. Questo lavoro apre così nuove vie per uno studio più approfondito del sistema nel contesto competitivo mondiale Clean Space.

Parole chiave:
rimozione attiva di detriti spaziali, strutture flessibili, sistemi di cattura a rete, algoritmi di collisione e contatto, simulatore multi-corpo, ottimizzazioni numeriche.

ACKNOWLEDGMENTS

First of all I would like to thank Prof. Michèle Lavagna for the great support and passion she puts in all she does. This experience turned out to be truly beyond expectations, starting from the friendship and relations that were born.
I particularly need to thank Riccardo for the great support in this work and experience. Then I cannot miss to thank all the office team, in particular Lorenzo, Daniele and Riccardo for the great friendship that has grown and the companionship in this adventure. I think I will hardly forget it, especially the great experience in Toronto together with Paolo and Alessandro. I am sure these moments could be enough to prove why all this work was worth it.

I thank Paul Scharff for the artistic and friendly interpretation in the back cover, of what cleverly represents the concept and all this thesis work.

Then of course, I am in debt with all my family and friends for their support and patience shown. This thesis work took me one year and lot of things happened. Great friends have passed, friends who were the first people I shared this passion with. But also great and unexpected facts occurred, from the top conclusion of my scouts experience to the discovery of the best life companion ever met.

Therefore, I cannot avoid to thank you Letizia for your true support and contribution in what proved to be a team success.

Ad Alberto e Marco,
insuperabili compagni.

“Tutta la differenza fra costruzione e
creazione è esattamente questa:
una cosa costruita si può solo amare;
ma una cosa creata si ama prima che esista.”

G. K. Chesterton

Chapter 1

INTRODUCTION

1.1 Background and Motivation

1.1.1 The Space Debris Issue

The term *Space Debris* characterizes the collection of objects in orbit around Earth with no useful purpose associated. These consist of everything from spent rockets stages and defunct satellites to small and micrometric fragments. As their orbit often overlap with active operative satellites, any debris represents a potential collision risk. In this sense, the solution adopted up to now implied the monitoring of their size and location in order to better plan Collision Avoidance Maneuvers (CAMs). More difficult is to directly obtain the attitude, which can be only estimated by propagated dynamics computations. Of course this approach does not solve the problem at all and it represents only an expensive countermeasure.

The space debris issue affects all the main operational orbits, such as LEO, MEO and GEO, but in a different way. Debris density decreases rapidly for outer orbits. MEO altitudes are characterized by few satellites and a limited number of orbits, used mainly for telecommunication. Although GEO is the most exploited operational orbit, it has the smaller space debris density with respect to LEO. This last, in fact, represents the topic orbit for the debris issue since it is the perigee of transfer orbits (such as GTO) and thus early rocket stages may be released there. Secondly, LEOs are characterized by a wide range of inclinations and RAAN, therefore at fixed altitude a lot of different objects coexist and have intersecting orbits. Most orbital debris are located within 2,000km altitudes and the highest concentration is at 800-900km.

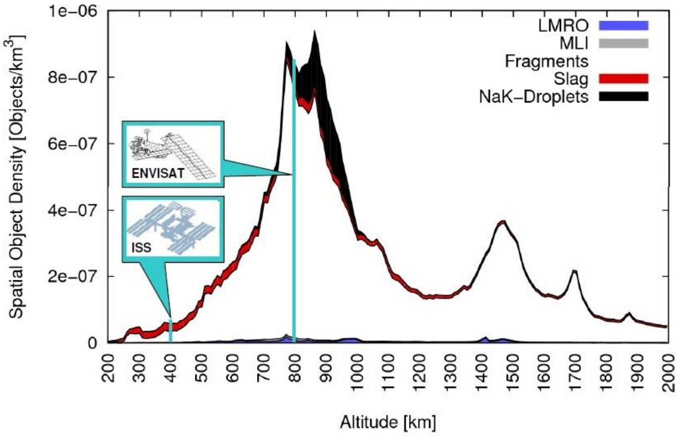


Figure 1.1: spatial debris density with altitude in 2009. Envisat and ISS highlighted [60].

In terms of hazards they represent, micro-debris are less dangerous since not always catastrophic. They can be counteracted with shields to absorb the impact energy. However, these shields cannot be mounted on solar panels, hence degrading their performance and reducing the operational life of the satellite.

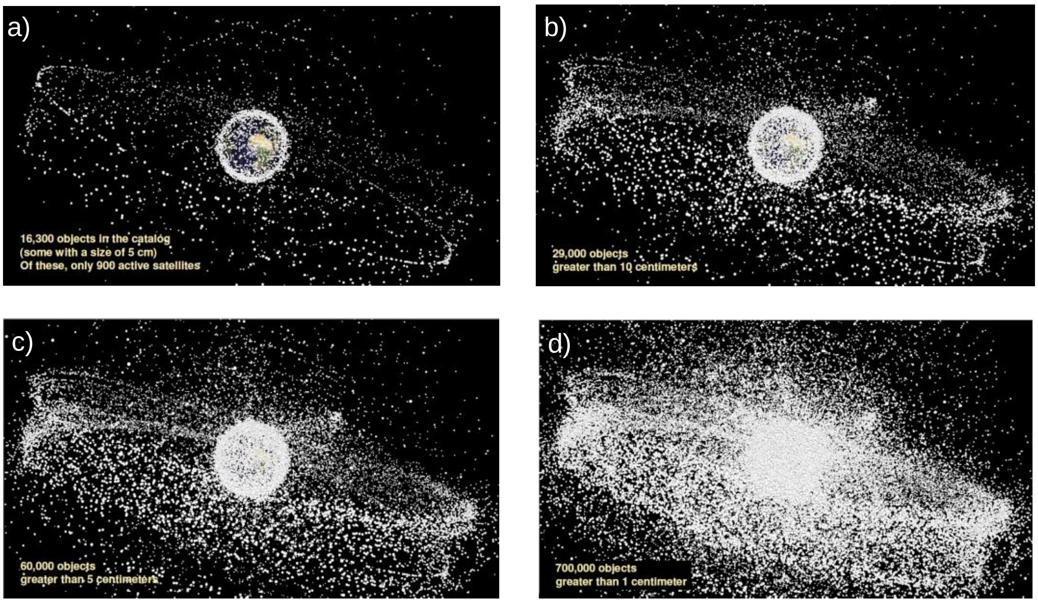

a)
16,300 objects in the catalog
(some with a size of 5 cm)
Of these, only 900 active satellites
b)
29,000 objects
greater than 10 centimeters
c)
60,000 objects
greater than 5 centimeters
d)
700,000 objects
greater than 1 centimeter

Figure 1.2: space debris spatial distribution according to their dimensions: a) 16,300 dead satellites, b) 29.000 objects greater than 10cm, c) 60.000 objects greater than 5cm and d) 700,000 objects greater than 1cm. [60]

Defunct satellites represent the worst case, due to the high dimensions and masses and their catastrophic effect at impact, capable of breaking up a spacecraft. Moreover, being uncontrolled, they cannot preform CAMs and every time they impact with other objects, a cloud of micro-debris is further generated and the more are the debris orbiting, the higher is the probability to generate new debris in a domino effect (*Kessler Syndrome*).

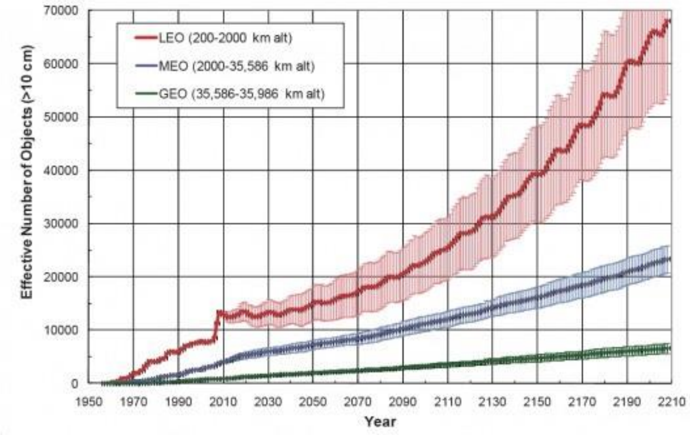


Figure 1.3: number of debris prediction with time.

Therefore, the hot topic of the debris issue concerns:

- LEO orbits
- Defunct satellites

In that sense, the most illustrative example taken as reference for this thesis work, is represented by *Envisat* (Fig. 1.4), which is the largest Earth-observing satellite ever put into space [26 x 10 x 5]m, weighting more than 8t at launch and located on a SSO at 790km altitude (Fig. 1.1). ESA lost contact on April 8th 2012. Despite the satellite is on a stable orbit, there is concern about its orbital evolution, collision with other satellites or, even more dangerous, an uncontrolled atmospheric re-entry.

1.1.2 Active Debris Removal Strategies

Besides a disposal policy for new space segments, Active Debris Removal (ADR) is a necessary step, in particular for large debris, in order to:

- prevent collisions with operative satellites;
- increase the operative life of the active satellites, reducing the number of CAMs;
- prevent the exponential generation of further debris.

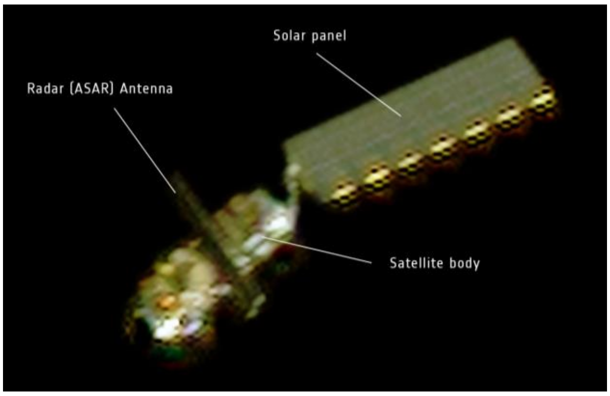


Figure 1.4: Envisat monitored from Earth.

Real missions concerning space debris have never been realized since they are costly and imply complex operations. Cost is related to the momentum required for the de-orbiting, complexity is due to the way to provide that momentum to the tumbling and non-cooperative object. The ADR topic focuses on trading-off, designing and building operational devices to be placed on board of an active chaser. This last has to rendezvous with and grapple the target object and ensure a controlled and safe disposal. Different techniques have been proposed for this challenging task (depicted below), such as:

- Robotic arm;
- Tether-net system;
- Tentacles and Harpoon;
- Drag sails;
- Electro-dynamic tether;
- Gas plume momentum thruster.

Among these, the last three involve a disposal time too long, the ionic thruster in particular, although being the most flexible to whichever target being the interaction contactless, is too costly and it is limited by the thruster’s lifecycle.

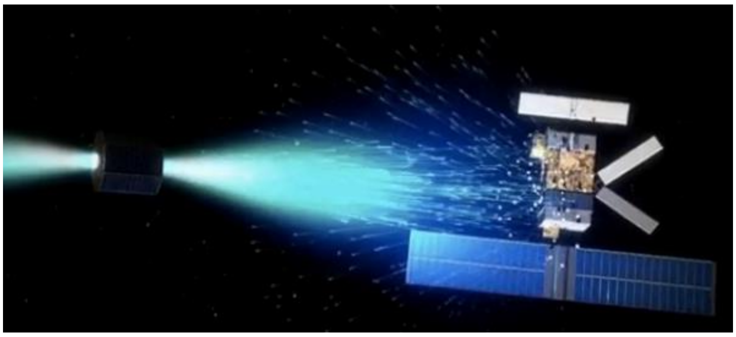


Figure 1.5: Ionic thruster concept (*credits ESA*)

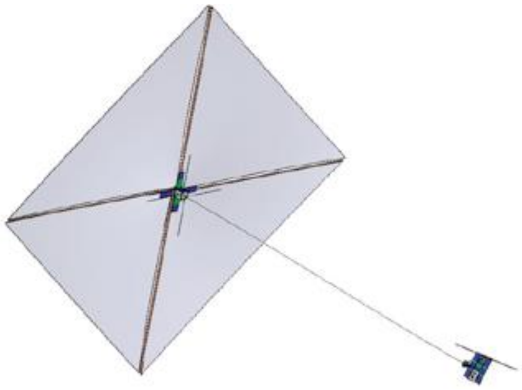


Figure 1.6: drag sail concept (*credits ESA*)

The harpoon is completely inappropriate due to the inevitable probability to generate new debris.

Therefore, the remaining alternatives can be organized in two categories:

- → Rigid devices
- → Flexible devices

Rigid devices, like the robotic arm, let to have a great control on the deorbiting, but they are too much sensitive from the tumbling state of the object, which represents a strong limiting requirement. They further require close proximity rendezvous (increasing the collision risk) and concentrated loads on the target (limiting the admissible interfaces).

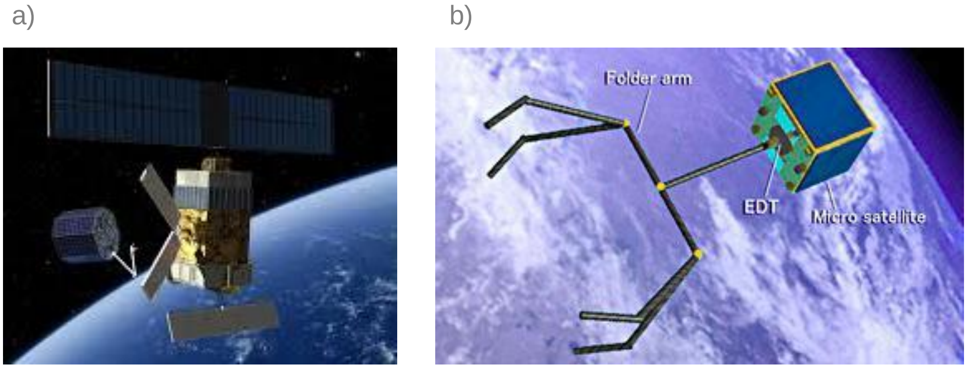


Figure 1.7: a) robotic arm and b) tentacles concepts (*credits ESA and JAXA*)

1.1.3 Flexible ADR Devices: the Tether-Net System

Flexible ADR solutions were studied in the past decade by Astrium from a systemic point of view, as a part of the ROGER study [24], and they have recently been the topic of an ESA CDF study [25] in the Clean Space program [61], looking for green technologies and eco-design.

Figure 1.8: Clean Space logo. (*Credis ESA*)

Among the flexible devices, the tether-net system is the most interesting and promising from many points of view, fulfilling the Clean Space drivers. It consists in capturing the non-cooperative free tumbling object throwing impulsively a net stored in a canister and connected to the chaser via a long tether. The net is deployed and driven by the pulling action of a set of lumped masses, named *bullets*, shot by a pneumatic system.

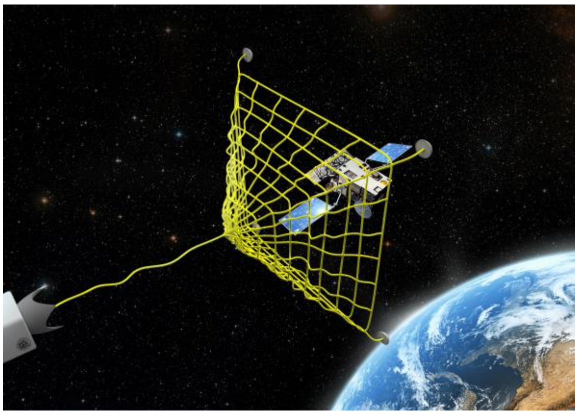
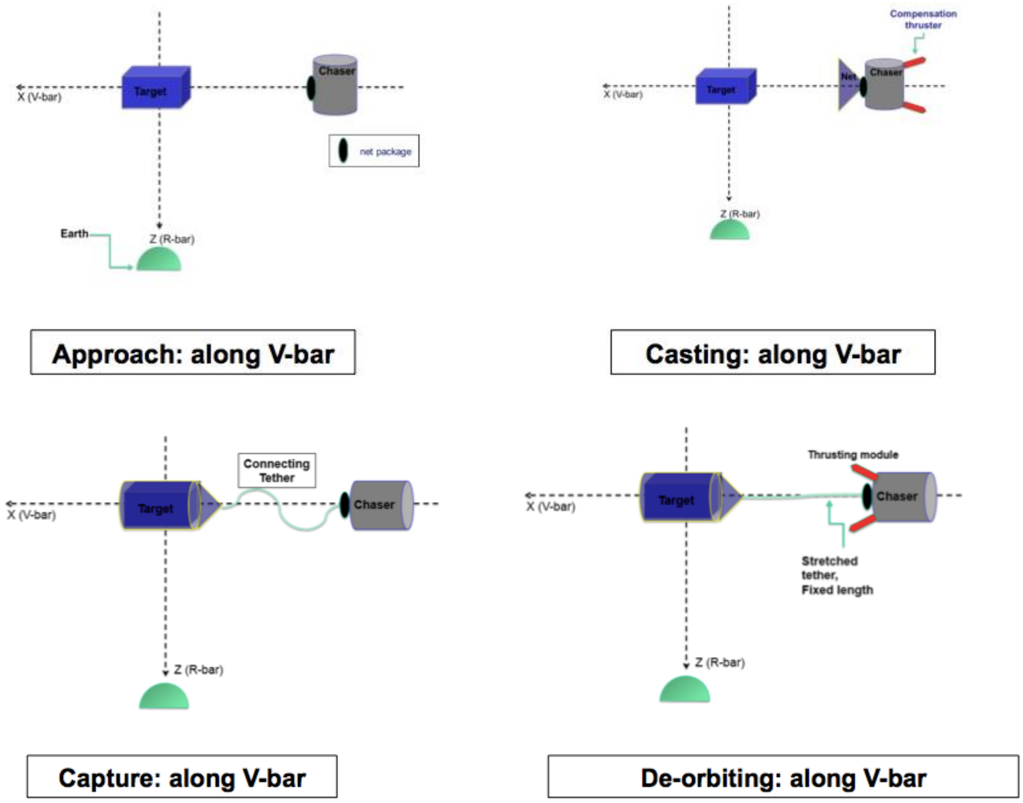


Figure 1.9: tether-net concept.

The capture is ensured by an event-driven *closing mechanism*, activated after the complete wrapping of the net over the target. Rendezvous and disposal pull are performed along the V-bar in order to avoid relative dynamics.

No particular constraints are imposed on the capture instant, but to ensure a relative distance at least equal to the tether length.
The disposal thrusting phase is engaged as soon as the net is closed and the tether is tensioned.



Approach: along V-bar

Casting: along V-bar

Capture: along V-bar

De-orbiting: along V-bar

Figure 1.10: representation of the capture and disposal phases.

In contrast with rigid capture mechanisms, tether-net solution therefore presents the following benefits:

- interfaceability with any possible target, whichever the dimensions, attitude, state and material;
- no synchronization required for the capture;
- isotropic loads applied, instead of concentrated ones;
- highest and safer capture distances, implying lowest collision risk;
- high tolerance on the pointing budget;
- passive damping of the tumbling state of the object;
- existing and off-the-shelf equipment;
- low mass budget and costs.

The only drawback, is related to the control, holding the flexible interface linking the target to the chaser and working only under traction. Therefore, in order better study and refine the control issue, the dynamics of this complex flexible multibody system needs to be modelled and accounted for the design.

1.2 Thesis Outline

To this end, the objective of this thesis work is to build up a complete numerical tool (*NeST*: Net Simulation Tool) able to simulate all the capture and disposal phases, precisely modelling all the physical dynamics occurring, with the ultimate aim of supporting the entire study and design of the controlled disposal.

The system will be analyzed in detail in each component, mathematically and numerically modelled step-by-step. A trade-off of the modelling of every physical dynamics will be discussed and validated both individually and overall at the end.

A particular attention will be given to the contact dynamics, since never treated before in such a detail for this system. The desired objective is to completely characterize the dynamics of such a complex multibody system and numerically quantify the expected benefits theorized up to now, proving the effective need of such a precise numerical tool.

The main drivers for the design are precision and efficiency, with a strong attention on the minimization of the computational workload. Therefore mathematical and numerical optimizations will be presented. For this reason an ad-hoc collision detection algorithm is designed and implemented in order to obtain maximum benefits in that sense.

Finally, the simulator will be physically validated and the tether-net concept characterized in details with a parametric sensitivity analysis. A final analysis in appendix supports the experimental validation of the ADR concept in a micro-gravity environment.

Chapter 2

MATHEMATICAL MODEL

2.1 Overview

We are going to present in this Chapter the overall Net-ADR system architecture and step-by-step the way each subsystem has been mathematically modelled. We will introduce the equations of motion, discuss and trade-off the constitutive laws, validate all the modelling approximations. It will be described first the different operative environment scenarios and the way they influence the dynamics.

2.2 Goals and Requirements

The main goal is to design a complete mathematical model of the dynamics, physically based and able to provide high fidelity results in a reasonable time lap. Further specific goals are:

- The possibility to be experimentally validated;
- Interfaceability with external libraries and third-party software;
- Simplified and user-friendly approach for tuning the numerical and model parameters.

This is translated in the following requirements for the design:

- Flexibility to three environment scenarios:
    - Ground test;
    - Parabolic test;
    - Simulation on orbit.
- Numerical robustness;
- Computational efficiency coming from optimization strategies;
- High precision of the dynamics integration;
- Physically-bases design of the subsystems models;

- Identification of few but strategic and essential characteristic parameters for each model.

2.3 Reference Frames and Equations of Motion

Before starting characterizing the dynamics equations, we need to define the reference frames that are at the bases of the model. Dealing with an orbiting system, we identify three reference frames:

- Earth Centered Inertial (**ECI**);
- Local Vertical Local Horizontal (**LVLH**);
- Principal Inertia (**PI**) or BODY reference frame.

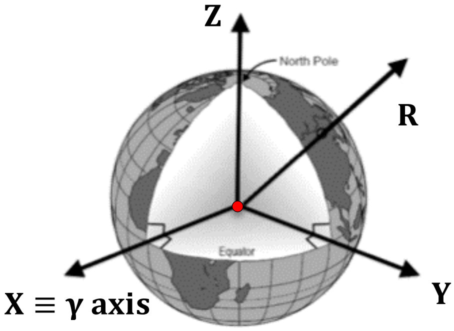


Figure 2.1: ECI frame representation.

ECI frame (Fig.2.1) is an Inertial reference frame, centered in the Earth and time independent by definition. It is exploited to define the orbital equations of motion to be integrated in the dynamics. It is represented in capital letters as well all the variables expressed in this reference frame (e.g. orbital position $\mathbf{R}$). It is so defined:

- **X** axis: permanently fixed towards the vernal equinox (γ-axis or Aries-axis);
- **Y** axis: lying in the equatorial plane, right-hand side orthogonal to **X** and **Z**;
- **Z** axis: parallel to the Earth rotation vector.

LVLH (Fig.2.2) frame is exploited to express typical user parameters describing the initial attitude of the Chaser and the Target. It is also used to express orbital and attitude perturbations. It is centered in the body CoM (Center of Mass) and it respects the following convention:

- **R-bar**: axis directed towards the Earth center;
- **H-bar**: axis opposite to the orbit angular momentum $\boldsymbol{h_{orb}}$;
- **V-bar**: axis right-hand side orthogonal to H-bar and R-bar.

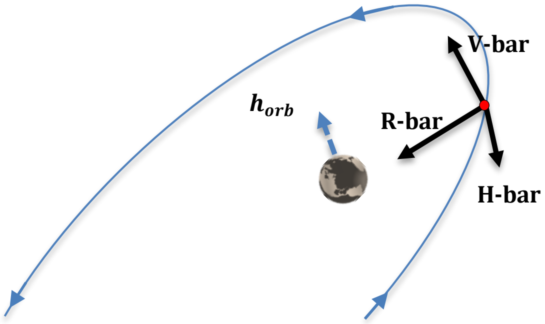


Figure 2.2: LVLH frame representation.

BODY frame (Fig.2.3) is characteristic of each rigid body in the simulation. It is defined by the Principal Inertia axes and centered in the CoM of the body. Quantities in this local reference frame are referred in lower case (e.g. $\boldsymbol{r}$).

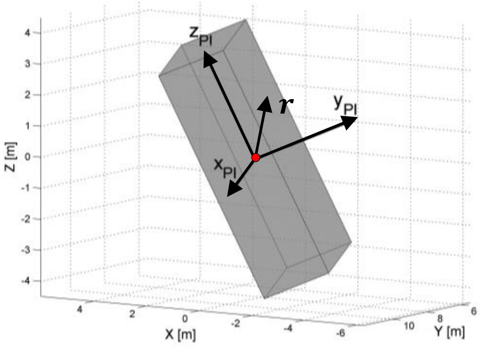


Figure 2.3: BODY frame representation in the INERTIAL one.

The linear dynamics is expressed in the ECI frame in order to avoid the modelling of fictitious forces. It is governed by the Newton's Second Law of motion and so defined for the mass *i*:

$$m_i\,\ddot{\boldsymbol{R}}_{\boldsymbol{i}} = \boldsymbol{F}_{\boldsymbol{i}} \tag{2.1}$$

Where $\vec{\mathbf{F}}_{\boldsymbol{i}}$ is the force acting on the mass *i*. It is the resulting action coming from both internal and external actions:

$$\boldsymbol{F}_{\boldsymbol{i}} = \sum_{\boldsymbol{j}} \boldsymbol{F}_{\boldsymbol{i,j}}^{\boldsymbol{ext}} + \sum_{\boldsymbol{j}} \boldsymbol{F}_{\boldsymbol{i,j}}^{\boldsymbol{int}} \tag{2.2}$$

The angular dynamics is expressed in the BODY frame and is governed by the Euler's equation:

$$\boldsymbol{I}_{\boldsymbol{i}}\,\dot{\boldsymbol{\omega}}_{\boldsymbol{i}} + \boldsymbol{\omega}_{\boldsymbol{i}} \times \boldsymbol{I}_{\boldsymbol{i}}\,\boldsymbol{\omega}_{\boldsymbol{i}} + \dot{\boldsymbol{I}}_{\boldsymbol{i}}\,\boldsymbol{\omega}_{\boldsymbol{i}} = \boldsymbol{T}_{\boldsymbol{i}} \tag{2.3}$$

Where $\boldsymbol{I_i}$ is the inertia matrix of the rigid body i, $\boldsymbol{\omega_i}$ is the angular velocity and $\boldsymbol{T_i}$ is the acting torque. This last is the resultant as well of external and internal actions:

$$\boldsymbol{T_i} = \sum\nolimits_{\boldsymbol{j}} \boldsymbol{T_{i,j}^{ext}} + \sum\nolimits_{\boldsymbol{j}} \boldsymbol{T_{i,j}^{int}} \tag{2.4}$$

The attitude is expressed through quaternions, because of their numerical robustness and computational efficiency. A quaternion is defined as:

$$\boldsymbol{q} = \begin{Bmatrix} s \\ v_1 \\ v_2 \\ v_3 \end{Bmatrix} = \begin{Bmatrix} \cos\frac{\theta}{2} \\ e_1 \sin\frac{\theta}{2} \\ e_2 \sin\frac{\theta}{2} \\ e_3 \sin\frac{\theta}{2} \end{Bmatrix} \tag{2.5}$$

Where the first element s is defined as the scalar part and the remaining three elements represent the vector part $\boldsymbol{v}$. It can be also expressed as function of the Euler axis components e_1, e_2 and e_3 and the Euler angle θ which is the angle required to rotate from an Inertial frame to the current one, around the Euler axis. Vector rotation from frame *A* to frame *B* is so performed:

$$\begin{bmatrix} 0 \\ \boldsymbol{v_B} \end{bmatrix} = \boldsymbol{q_{BA}} \begin{bmatrix} 0 \\ \boldsymbol{v_A} \end{bmatrix} \boldsymbol{q_{BA}^*} \tag{2.6}$$

Where $\mathbf{v}_{\boldsymbol{A}}$ and $\mathbf{v}_{\boldsymbol{B}}$ are vectors in the frame expressed in their index, while $\boldsymbol{q_{BA}}$ and $\mathbf{q_{BA}^*}$ are respectively the quaternion to rotate from frame *A* to *B* and its conjugate (*B* to *A*). The simulator automatically integrates the kinematics of quaternions and gives as output the attitude from ECI to BODY frame as $\boldsymbol{q}_{ECI\rightarrow BODY}$.

2.4 Environment Modelling

The modelling of the environment provides a full set of external forces and torques *j* acting on the body *i*, respectively $\boldsymbol{F_{i,j}^{ext}}$ and $\boldsymbol{T_{i,j}^{ext}}$**.** We identify three possible operational environments, according to the Simulator requirements:

- Simulation on ground;
- Simulation on a parabolic flight (*zero-g* tests);
- Simulation in a real operative orbital environment.

Therefore in the following paragraphs we define all the possible external actions characterizing the aforementioned scenarios.

2.4.1 Gravity Field

Gravity field modelling is responsible for three main external actions:

- Gravity force;
- J2 perturbation force;
- Gravity Gradient perturbation torque.

The gravity force is of course to be considered in the orbital motion since it defines the orbital dynamics itself and also ground test simulations (Eq. 2.7). It is directed towards the Earth center along **R-bar**.

$$\boldsymbol{F_{G,i}} = -\frac{\mu\, m_i}{|\boldsymbol{R_i}|^3}\boldsymbol{R_i} \tag{2.7}$$

Where

- μ is the Earth gravitational constant ($G \cdot M_{earth} = 3.986\text{e}14\ \text{m}^3/\text{s}^2$);
- m_i is the *i-th* element mass $[kg]$;
- $\boldsymbol{R_i}$ is the vector position of the *i-th* element $[m]$.

Like all the external forces, $\boldsymbol{F_{G,i}}$ is expressed in the **ECI** frame. For simulations on ground, the Newton's gravitational law automatically comes out from Eq. 2.7, summing the Earth radius (6378e3 m) to the initial ground-relative position.

Taking into account also the non-constant intensity of the gravitational field at different latitudes, an orbital perturbation arises due to Earth oblateness. It is the so-called J2-perturbation, and it is a long term orbital perturbing force acting on the body CoM. As a consequence, it does not affect the attitude and it is not considered in both ground and *zero-g* simulations. J2 is the first zonal harmonic coefficient that dominates over all the others describing the Earth mass distribution and so the gravity field. The arising perturbing force is modelled in the LVLH reference frame:

$$\boldsymbol{F_{J2,i}} = -3\, m_i \frac{J_2\, \mu\, R_{earth}^2}{|\boldsymbol{R_i}|^4} \left\{ \begin{array}{c} (\sin(i))^2 \cdot \sin(\omega+\theta) \cdot \cos(\omega+\theta) \\ -\sin(i) \cdot \cos(i) \cdot \sin(\omega+\theta) \\ 3/2\,(\sin(i))^2 \cdot (\sin(\omega+\theta))^2 - 1/2 \end{array} \right\} \tag{2.8}$$

Where:

- J_2=1.08271e-3;
- R_{earth}=6378e3 m;
- i= orbit inclination [rad];
- ω= orbit anomaly of pericenter [rad]
- θ= orbit true anomaly [rad]

The orbital parameters i, ω and θ are retrieved from the orbital position $\boldsymbol{R_i}$ and velocity $\boldsymbol{V_i}$. Rotation in the **ECI** frame is required in order to integrate the force in the dynamics. This is done defining the force components (**LVLH** frame) in the inertial frame:

$$\begin{cases} \boldsymbol{x}_{LVLH,i} = \boldsymbol{V\ bar} = \ \boldsymbol{y}_{LVLH,i} \times \boldsymbol{z}_{LVLH,i} \\ \boldsymbol{y}_{LVLH,i} = \boldsymbol{H\ bar} = -\dfrac{\boldsymbol{R_i} \times \boldsymbol{V_i}}{|\boldsymbol{R_i} \times \boldsymbol{V_i}|} \\ \boldsymbol{z}_{LVLH,i} = \boldsymbol{R\ bar} = -\dfrac{\boldsymbol{R_i}}{|\boldsymbol{R_i}|} \end{cases} \tag{2.9}$$

Finally, the Gravity Gradient. It affects only the body attitude since it is related to the influence of the gravity field on the particular geometry and mass distribution of the orbiting object. Therefore, it is active only for rigid bodies and in an orbital scenario. It is directly related to the inertia matrix $\boldsymbol{I_i}$ of the body and its attitude with respect to the **R-bar**. This last in particular is expressed through direction cosines $c_{k,i}$.

$$\boldsymbol{T_{GG,i}} = 3\frac{\mu}{|\boldsymbol{R_i}|^3}\begin{Bmatrix} (I_3 - I_2)c_2c_3 \\ (I_1 - I_3)c_1c_3 \\ (I_2 - I_1)c_1c_2 \end{Bmatrix}_{\boldsymbol{i}} \tag{2.10}$$

Where I_1, I_2 and I_3 are the principal inertia moments, that are the eigenvalues of a more generic form of the inertia matrix, therefore rewritten with respect to the principal inertia axes (**PI**) as:

$$\boldsymbol{I_i} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}_{\boldsymbol{i}} \tag{2.11}$$

The direction cosines are the ones obtained expressing **R-bar** in the **BODY-PI** frame directly through the quaternions $\boldsymbol{q}_{ECI\rightarrow PI}$ integrated in the dynamics**:**

$$\boldsymbol{c_i} = \left[-\frac{\boldsymbol{R_i}}{|\boldsymbol{R_i}|}\right]_{\boldsymbol{PI}} \tag{2.12}$$

Estimating the order of magnitude of this perturbation for a LEO orbit of 200km radius and an orbiting body whose maximum $\Delta I \approx 1e3$, we obtain $|\boldsymbol{T_{GG,i}}| = o(10^{-3})$ Nm.

2.4.2 Atmospheric Drag

Atmospheric perturbations imply drag (**ECI** frame), so defined:

$$\boldsymbol{F_{D,i}} = -\frac{1}{2}\sum_j A_{i,j}\, c_{D,j}\, \rho\, (\boldsymbol{N_{i,j}} \cdot \boldsymbol{V_{i,j}})\boldsymbol{V_{i,j}}\ ,\quad (\boldsymbol{N_{i,j}} \cdot \boldsymbol{V_{i,j}}) \geq 0 \tag{2.13}$$

Where:

- *i,j* index refers to body *i*, face *j*, that is why the resulting force comes out from the sum of all the forces acting on the surfaces;
- $A_{i,j}$ is the area of the surface *j* [m^2];
- $c_{D,j}$ is the drag coefficient characteristic of the surface *j*;
- ρ is the air density at altitude *h* [kg/m^3];
- $\boldsymbol{N_{i,j}}$ is the normal exiting unit vector defining the surface *j*.

$\boldsymbol{V_{i,j}}$ is the velocity in ECI frame of the body *i* at the surface *j*, computed as:

$$\boldsymbol{V_{i,j}} = \boldsymbol{V_{0,i}} + \left[\boldsymbol{\omega_i} \times \boldsymbol{r_j}\right]_{ECI} \tag{2.14}$$

Where:

- $\boldsymbol{V_{0,i}}$ is the velocity of the body CoM in ECI frame [m/s];
- $\boldsymbol{\omega_i}$ is the body angular velocity in BODY frame [rad/s];
- $\boldsymbol{r_j}$ is the relative position of the centroid of the surface *j* with respect to the CoM, expressed in BODY frame [m].

$\boldsymbol{F_{D,i}}$ acts on the centroid of the surface which does not always overlap with the body CoM, so it is responsible as well of the arising of a corresponding torque:

$$\boldsymbol{T_{D,i}} = \sum_j \boldsymbol{r_j} \times \left[\boldsymbol{F_{D,i,j}}\right]_{BODY} \tag{2.15}$$

Atmospheric drag concerns all the three scenarios as it depends on the air density and system velocity.

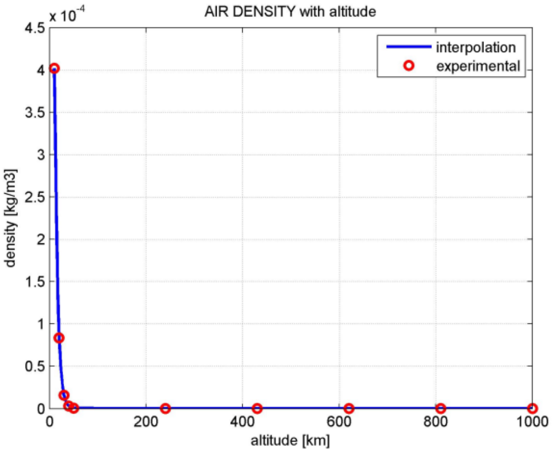


Figure 2.4: air density profile with altitude. Experimental data and interpolation exploited within the dynamics are shown.

In the orbital scenario we have very high velocities $(o(10^3)\, m/s)$ although relatively low density, while on ground velocities are much lower ($o(10^0)\, m/s$), but relatively high density. The atmospheric density can be exponentially modelled with altitude h according to the isothermal law of the International Standard Atmosphere (ISA):

$$\rho(h) = \rho(0)\, e^{-\frac{h}{h^*}}, \qquad \begin{cases} \rho(0) = 1.225\, kg/m^3 \\ h^* = 7{,}11\, km \end{cases} \tag{2.16}$$

However in order to achieve maximum physical fidelity, we prefer to use experimental density values as function of the altitude which are interpolated and then retrieved during the simulation (Fig. 2.4). For the very low values at high altitudes ($o(10^{-17})\, \frac{kg}{m^3}$), we decide to limit the integration of the drag below 1,000 km altitude.

2.4.3 Radiation Pressure

The influence of the radiation pressure exists only for orbiting objects and it is responsible for the arising of both perturbing forces and torques, so affecting both the orbital motion and the attitude. We identify the Earth and the Sun as the main sources and we express the overall perturbing force (in ECI frame) acting on the *i-th* body as follows:

$$\boldsymbol{F}_{R,i} = \sum_{i} \boldsymbol{F}_{i,j}^{SUN} + \sum_{i} \boldsymbol{F}_{i,j}^{EARTH} \tag{2.17}$$

Where the *i,j* indexing refers to the force acting on the face *j* of the body *i*.
The Earth is responsible for a direct radiation emission and the reflection of the solar radiation (albedo), both depending on the altitude and directed as the object position (opposite to **R-bar**). Empirical data of the specific power are retrieved and interpolated to get a continuous model in the simulator (Fig 2.5). The sum of the two contributions provides the overall specific power W_E coming from Earth. Radiation pressure is then retrieved from the specific power through the light speed:

$$P_E = \frac{W_E}{c} \tag{2.18}$$

Where c is the speed of light in vacuum (3e8 m/s).
Due to the low values at high altitudes, we choose to limit the integration of the power radiated and reflected by the Earth up to an altitude of 8e3 km. We make also the following reasonable approximation assumption:

$$\boldsymbol{R}_i + \left[\boldsymbol{r}_j\right]_{ECI} \cong \boldsymbol{R}_i \tag{2.19}$$

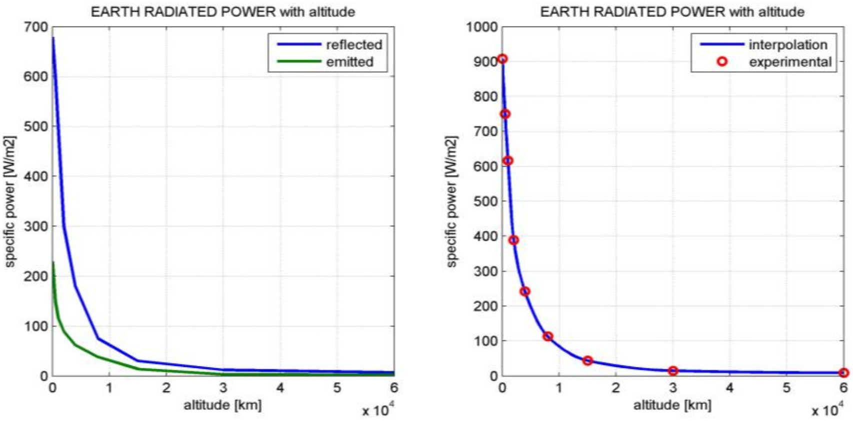


Figure 2.5: Earth radiated specific power with altitude.

Where:

- $\boldsymbol{R}_i$ is the position vector of the body orbiting around the Earth (**ECI** frame);
- $\boldsymbol{r}_j$ is the position of the of the surface centroid where the force is applied.

The expression of the perturbing force acting on a surface j of the body I, due to the Earth contribution is therefore derived:

$$\boldsymbol{F}_{i,j}^{EARTH} = \begin{cases} \boldsymbol{0} & , \quad (\boldsymbol{R}_i \cdot \boldsymbol{N}_{i,j}) > 0 \\ -P_{\boldsymbol{E}}\, A_{i,j}\, c_{R,j} \left(\dfrac{\boldsymbol{R}_i}{|\boldsymbol{R}_i|} \cdot \boldsymbol{N}_{i,j}\right) \dfrac{\boldsymbol{R}_i}{|\boldsymbol{R}_i|}, & (\boldsymbol{R}_i \cdot \boldsymbol{N}_{i,j}) \leq 0 \end{cases} \tag{2.20}$$

Where:

- $A_{i,j}$ is the surface area [m^2];
- $c_{R,j}$ is the reflection coefficient;
- $\boldsymbol{N}_{i,j}$ is the exiting normal unit vector defining the surface (**ECI** frame).

For what concerns the solar direct emitted radiation, we can assume it to be constant both in time and in direction. This can be done because of the inverse square relation holding between the radiation intensity and the distance from the source and the high relative distance from the Sun. Therefore, the following approximations can be done:

$$\begin{cases} W_S \cong const \\ \boldsymbol{R}_{SUN} + \boldsymbol{R}_i + [\boldsymbol{r}_j]_{ECI} \cong \boldsymbol{R}_{SUN} \end{cases} \tag{2.21}$$

Where:

- W_S is the solar direct radiation specific power (1358 W/m^2);
- $\boldsymbol{R}_{SUN}$ is the position of the Earth with respect to the sun in **ECI** frame;

The expression of the perturbing force acting on a surface *j* of the body *i*, due to the Sun contribution is therefore derived:

$$\boldsymbol{F}_{i,j}^{SUN} = \begin{cases} \boldsymbol{0} & , \quad \left(\boldsymbol{R}_{SUN} \cdot \boldsymbol{N}_{i,j}\right) > 0 \\ -P_{\boldsymbol{S}}\, A_{i,j}\, c_{R,j} \left(\frac{\boldsymbol{R}_{SUN}}{|\boldsymbol{R}_{SUN}|} \cdot \boldsymbol{N}_{i,j}\right) \frac{\boldsymbol{R}_{SUN}}{|\boldsymbol{R}_{SUN}|}, & \left(\boldsymbol{R}_{SUN} \cdot \boldsymbol{N}_{i,j}\right) \leq 0 \end{cases} \tag{2.22}$$

Where $P_{\boldsymbol{S}}$ is derived from W_S through Eq. 2.18.
Finally, in order to take into account the eclipse eventuality, two further conditions must be alternatively satisfied:

$$\begin{cases} (\boldsymbol{R}_i \cdot \boldsymbol{R}_{SUN}) < 0 \\ |\boldsymbol{R}_i \times \boldsymbol{R}_{SUN}| > R_{EARTH} \end{cases} \tag{2.23}$$

Where R_{EARTH} is the Earth radius, defining the eclipse shadow cylindrical dimensions. Concluding, we derive the overall resulting acting radiating torque for the body *i*, that is:

$$\boldsymbol{T}_{R,i} = \sum_j \boldsymbol{r}_j \times \left[\left(\boldsymbol{F}_{i,j}^{SUN} + \boldsymbol{F}_{i,j}^{EARTH}\right)\right]_{BODY} \tag{2.24}$$

Radiating torque is expressed in *BODY* frame, so it requires the radiating forces to be rotated accordingly.

2.5 Rigid Bodies

The Net-ADR system involves the modelling of the following elements:

• Chaser	Rigid model
• Target	
• Bullets	
• Net	Flexible model
• Tether	

The first three are modelled as rigid bodies, meaning that no internal flexibility is accounted, nor internal sloshing. Therefore it follows:

$$\dot{\boldsymbol{I}}_{\boldsymbol{i}}\, \boldsymbol{\omega}_{\boldsymbol{i}} = \boldsymbol{0} \tag{2.25}$$

$$\sum\nolimits_{\boldsymbol{j}} \boldsymbol{T}_{\boldsymbol{i,j}}^{\boldsymbol{int}} = \boldsymbol{0} \tag{2.26}$$

Eq. 2.3 and Eq. 2.4 simplify to:

$$\boldsymbol{I}_{\boldsymbol{i}}\,\dot{\boldsymbol{\omega}}_{\boldsymbol{i}} + \boldsymbol{\omega}_{\boldsymbol{i}} \times \boldsymbol{I}_{\boldsymbol{i}}\,\boldsymbol{\omega}_{\boldsymbol{i}} = \boldsymbol{T}_{\boldsymbol{i}} \tag{2.27}$$

$$\boldsymbol{T}_{\boldsymbol{i}} = \sum_{\boldsymbol{j}} \boldsymbol{T}_{\boldsymbol{i,j}}^{\boldsymbol{ext}} \tag{2.28}$$

2.5.1 Chaser

The Chaser model is reduced to a parallelepiped rigid body, so defined by a mass M_C and three dimensions [a_1 a_2 a_3]. Principal Inertia matrix $\boldsymbol{I_C}$ is automatically obtained:

$$\boldsymbol{I}_C = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} \tag{2.29}$$

Where the principal inertia moments are:

$$I_i = \frac{1}{12}\, M_c\left(a_j^2 + a_k^2\right) \tag{2.30}$$

Alternatively, a different inertia matrix can be provided to simulate a non-homogeneous mass distribution. To account for drag and solar pressure, the normal exiting unit vector for each surface is derived due to the simple geometry (green vectors in Fig. 2.6), while single drag and reflectivity coefficients (c_D and c_R) are associated to all of the six surfaces and defined by the user. In the same way the thrusting and attitude control laws can be provided, introducing a further term $\boldsymbol{F}_C^{thrust}$ in the external forces acting on the Chaser as well as a control torque $\boldsymbol{T}_C^{control}$.

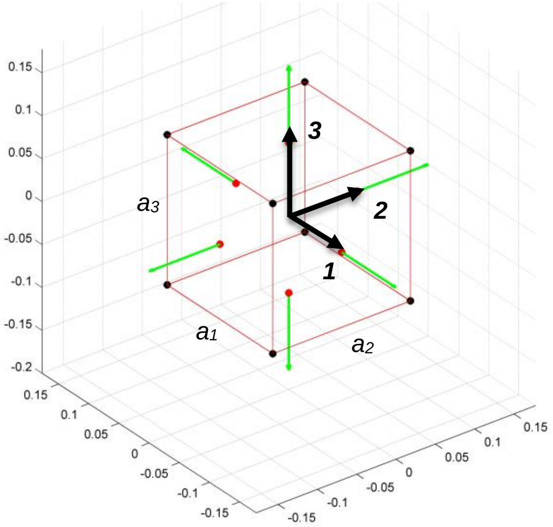


Figure 2.6: Chaser geometry definition in BODY frame **[1 2 3]**.

2.5.2 Target

Differently, the Target is a non-cooperative and uncontrolled body. Since it is the main topic of the simulation, after the Net, it is much deeply defined in all its aspects. First of all the shape. Mass M_T and inertia $\boldsymbol{I_T}$ need to be both directly provided by the user and in addition a full mesh characterization of the surfaces is to be defined in input. What is important to remark is that given few strategic geometrical parameters, it is possible to retrieve all the required information.
The Target is treated as a most generic 3D geometry composed by planar surfaces defining the mesh. The geometry is uniquely defined once provided:

- the position $\boldsymbol{r_j}$ of all the vertices *j* of the mesh elements in *BODY* frame with respect to the Target CoM;
- the mesh elements borders as a set of double indices referring to the bounding vertices.

The strategy consists in defining the borders in a counter-clockwise order as shown in Fig. 2.7.

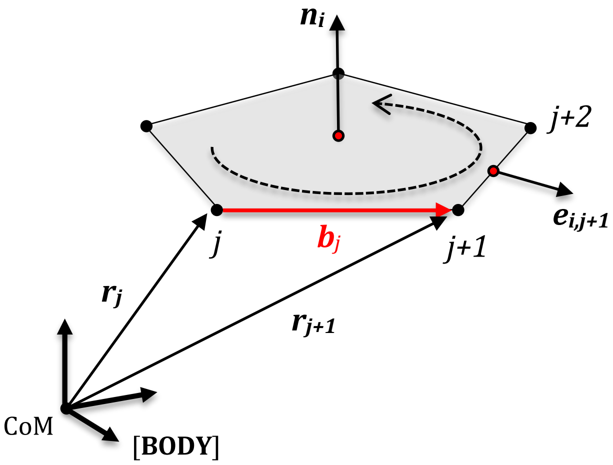


Figure 2.7: Target *i-th* mesh element definition.

The border $\boldsymbol{b}_j$ is defined by vertices *j* and *j+1*, such that:

$$\boldsymbol{b_j} = \boldsymbol{r}_{j+1} - \boldsymbol{r}_j \tag{2.31}$$

Therefore, normal exiting unit vector $\boldsymbol{n_i}$ characterizing the mesh element *i* can be derived:

$$\boldsymbol{n_i} = \frac{\boldsymbol{b}_j \times \boldsymbol{b}_{j+1}}{\left|\boldsymbol{b}_j \times \boldsymbol{b}_{j+1}\right|} \tag{2.32}$$

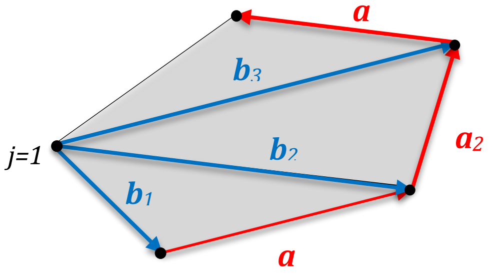


Figure 2.8: Target *i-th* mesh area computation.

And similarly the normal exiting unit vector $\boldsymbol{e}_{i,j}$ of the border $\boldsymbol{b}_j$ is retrieved:

$$\boldsymbol{e}_{\boldsymbol{i},\boldsymbol{j}} = \frac{\boldsymbol{b}_j \times \boldsymbol{n}_{\boldsymbol{i}}}{|\boldsymbol{b}_j \times \boldsymbol{n}_{\boldsymbol{i}}|} \tag{2.33}$$

Finally, what misses is the computation of the mesh element area. It is obtained from Eq. 2.34, supporting Fig. 2.8.

$$A_i = \sum_{j=2}^{N-1} \frac{1}{2} \left\| \boldsymbol{a}_{j-1} \times \boldsymbol{b}_{j-1} \right\| \tag{2.34}$$

Where, N is the number of borders and:

$$\begin{cases} \boldsymbol{a}_{j-1} = \boldsymbol{r}_{j+1} - \boldsymbol{r}_j \\ \boldsymbol{b}_{j-1} = \boldsymbol{r}_j - \boldsymbol{r}_{j=1} \end{cases} \tag{2.35}$$

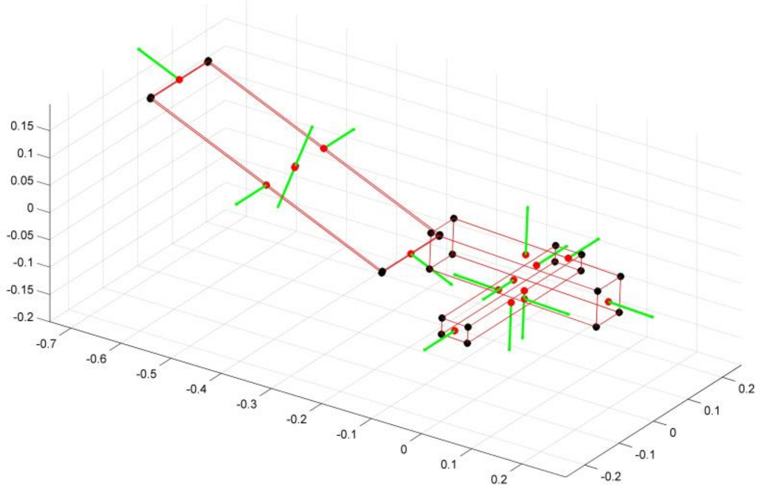


Figure 2.9: example of Target geometry definition.

Finally each surface mesh element i is associated to its characteristic physical parameters adopted in the models (from the environmental perturbations to the contact reactions). A representation example of the Target geometry characterization is given in Fig. 2.9, where black dots are the vertices, red segments the borders and green arrows the mesh elements exiting normal unit vectors $\boldsymbol{n}_i$.

2.5.3 Bullets

Finally, the Bullets. They deserve a different treatment: they are all equal, cylindrical shaped and so modelled once given the mass m_b and the dimensions (radius r_b and height h_b). The Inertia matrix $\boldsymbol{I}_b$ is obtained as same as in Eq. 2.29 and the principal inertia components are:

$$I_1 = I_2 = \frac{1}{12}\, m_b (3r_b^2 + h_b^2) \tag{2.36}$$

$$I_3 = \frac{1}{2}\, m_b\, r_b^2 \tag{2.37}$$

However, it is still possible to define directly their inertia matrix if they are non-homogeneous in mass distribution. Although, not an operative condition, it is also possible to define each Bullet differently from the others. This is useful for the test campaign, in order to simulate non-symmetrical behavior and perform a sensitivity analysis on the maximum admissible tolerance on the mass difference between the bullets. Bullet mass can be alternatively provided as a function of the Net mass, introducing the concept of Bullet Ratio (BR), useful for parametric analysis and scaled simulations

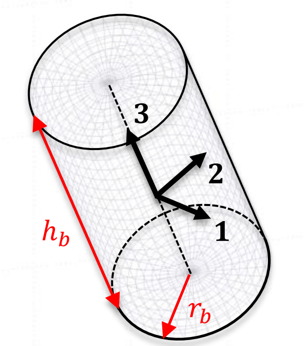


Figure 2.10: Bullet geometry definition. BODY frame displayed **[1 2 3]**.

$$BR = \frac{N_b\, m_b}{m_{tot}} \tag{2.38}$$

Where N_b is the number of Bullets and m_{tot} is the overall mass of the Net system. Each Bullet is linked to the Net through one or more connections named *bulletLinks* and to the *closingLinks*, as long as a closing mechanism is included and the winch is stored inside the same Bullet (as it will be describe in Section 2.7). All Bullets are shot by means of a pneumatic ejection system which acts axially along the third BODY axis, being responsible of the introduction of the external force $\boldsymbol{F}_b^{shot}$ (Section 2.6).

For what concerns the other external actions, we neglect the ones related to the gravitational field and radiation pressure, due to the very small dimensions and the small forces and torques arising as result. On the other hand, for the importance on the ground testing campaign, drag has been considered and the related quantities and directions are defined below as well. Focusing on the cylindrical shape, we define two components for the resulting drag force: $\boldsymbol{F}_D^{base}$ acting on the cylinder base and $\boldsymbol{F}_D^{lateral}$ acting on the lateral surface. Recalling the drag force expression from Eq. 2.13, in the case of the drag force acting on the front-base, $\boldsymbol{N}$ is coincident with the third BODY axis and the force is applied in the base center: $\boldsymbol{r}_{base} = \frac{h}{2} \cdot \widehat{\boldsymbol{3}}$. Velocity of that point is computed through Eq. 2.14, while the drag coefficient is retrieved from the interpolation of experimental data (Fig. 2.11) according to the Bullet dimensions and the nose shape (flat or curved).

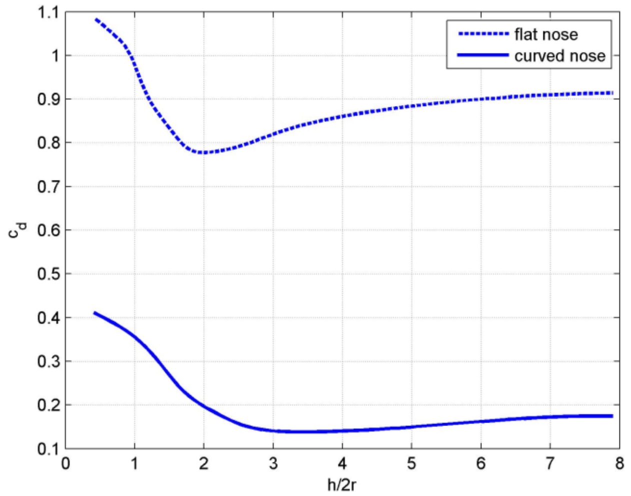


Figure 2.11: Bullet front section drag coefficient with the aspect ratio. Flat nose and curved nose compared.

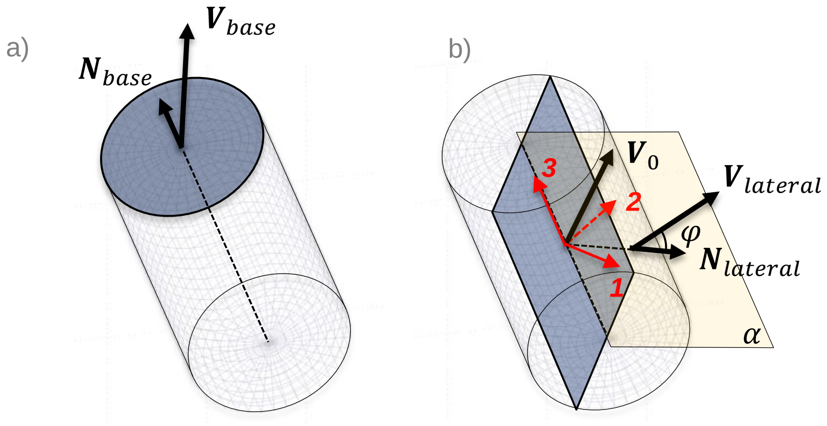


Figure 2.12: Bullet cross-section areas and velocity for the two drag force contributions: a) base, b) lateral.

Differently, for the lateral surface we compute the normal unit vector:

$$\boldsymbol{N}_{lateral} = \frac{[\mathbf{3} \times \boldsymbol{V_0} \times \mathbf{3}]_{ECI}}{|\mathbf{3} \times \boldsymbol{V_0} \times \mathbf{3}|} \tag{2.39}$$

$\boldsymbol{V_0}$, $\boldsymbol{V}_{lateral}$ and $\boldsymbol{N}_{lateral}$ lie on the same plane α, orthogonal to the interested area, which is the cross-section one displayed in Fig. 2.12b. Therefore:

$$\boldsymbol{r}_{lateral} = r_b\, \boldsymbol{N}_{lateral} \tag{2.40}$$

$\boldsymbol{V}_{lateral}$ is again retrieved at the first iteration through Eq. 2.14.

The lateral drag coefficient is instead differently defined and more rigorously derived, since it will be also used for the Net drag model. While for ground tests an empirical value can be easily provided, for orbital simulations the physics changes. In fact, for bodies at hypersonic velocities in the rarefied atmosphere $(Mach\ no. = o(10^1))$, good results are given exploiting a drag coefficient calculation method based on Newton's impact theory [56]. It is considered that the impact of particles of gas with the body is inelastic and there is a damping phenomenon to take into account. Aerodynamic drag coefficient does not depend on the size of the body and for cylindrical shapes it takes the following expression [57]:

$$c_D = \frac{2}{3}\bar{k} \tag{2.41}$$

The incidence angle φ with respect to $\boldsymbol{N}_{lateral}$ is computed as:

$$\cos(\varphi) = \boldsymbol{V}_{lateral} \cdot \boldsymbol{N}_{lateral} \tag{2.42}$$

While factor k depends on velocity (Mach no. M_∞) and air property (adiabatic index χ):

$$\bar{k} = \frac{2}{\chi\, M_\infty^2}\left[\left(\frac{\chi+1}{2}M_\infty^2\right)^{\frac{\chi}{\chi-1}}\left(\frac{\chi+1}{2\chi M_\infty^2-\chi+1}\right)^{\frac{1}{\chi-1}}-1\right] \quad (2.43)$$

Finally, the resulting drag force (**ECI** frame) and drag torque (**BODY** frame), both applied at the Bullet CoM, can be computed:

$$\boldsymbol{F_D} = \boldsymbol{F}_D^{base} + \boldsymbol{F}_D^{lateral} \quad (2.44)$$

$$\boldsymbol{T_D} = \boldsymbol{r}_{base} \times \left[\,\boldsymbol{F}_D^{base}\right]_{BODY} + \boldsymbol{r}_{lateral} \times \left[\boldsymbol{F}_D^{lateral}\right]_{BODY} \quad (2.45)$$

2.6 Flexible Bodies

The primary most challenging topic of the simulator is of course the modelling of the flexible systems represented by Net and Tether. A **Net** is defined as a system of knotted threads composing the meshes. We define a mesh as the minimum dimension closed patterns, which the Net is made of (Fig. 2.13), while we call *knot* the physical joint between two or more threads. There are many different possible geometries of the mesh and of the overall shape of the Net, such as 2D and 3D geometries. Two examples are here provided (Fig. 2.14) and further will be presented in Section 5.4.1.

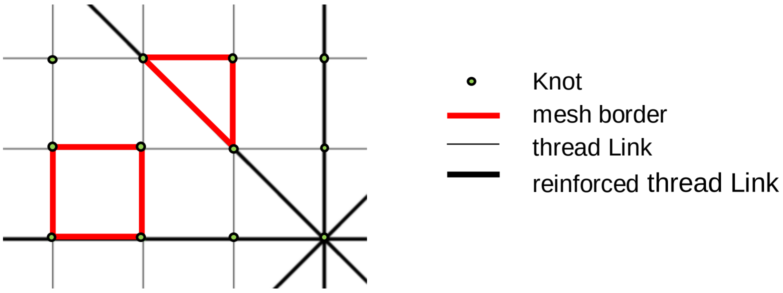


Figure 2.13: Two examples of Net mesh geometries: square and triangular. Knots highlighted.

The Net is deployed and dragged by an arbitrary number of lumped masses (Bullets), positioned at the inlet cross-section area and linked to the so-called *Net-mouth* in a direct way or through additional links named *bulletLinks*. The Bullets are shot through a pneumatic Net-gun device or Shooting System, whose design is reported in Section 2.6. Before being shot, the Net is folded and stored in a sealed *canister*, to which the Net and Bullets are constrained until it is opened.

In order to finalize the capture and safe disposal, the Net is provided with a *Closing Mechanism*, consisting of a set of links running along the *Net mouth* perimeter, passing through an arbitrary number of rings and winding on spinning winches at the Net inlet section (Fig 2.14). These threads can be seen as variable length links (differently from all the others), whose *length-law* is strictly related to the closing system design (Section 2.7).

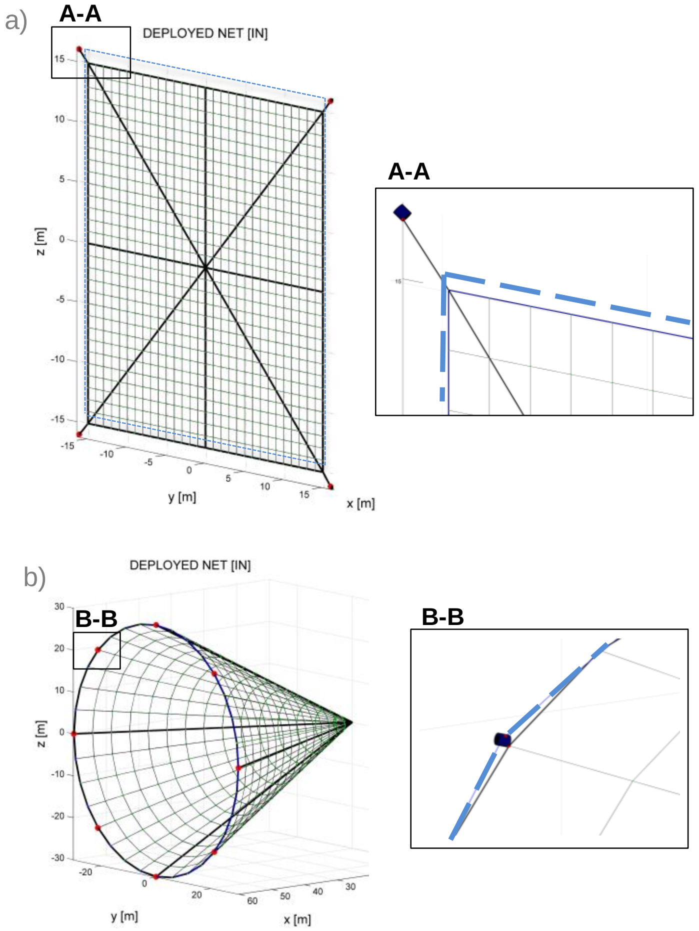


Figure 2.14: a) *Planar* Net, with *bulletLinks* and *closingLinks* directly connected to the Net corners. b) *3D conical shape* Net, with 8 bullets, no *bulletLinks* and with *closingLinks* passing through rings along the whole *Net-mouth* perimeter and connecting to the Bullets.

Finally, the Net is linked to the Chaser by means of a **Tether**, through a direct or multiple links. The Tether is in charge of connecting the Chaser to the Target once wrapped by the Net, in order to let it possible the disposal and ensure a safe control at a certain distance. The higher is the length, the safer is the capture, reducing the risk of collision between Chaser and Target. Moreover, the longer is the Tether, the lower is the resulting stiffness of the link joining the two-bodies deorbiting system (relying only on traction and null compression), which in some cases provides benefits to the disposal dynamics. Therefore, Tether length plays an important role in the control design, and this Simulator aims at providing the tools for this task, taking into consideration (most important) the complex interaction between Net and Target instead of having fixed connection points on the Target itself.

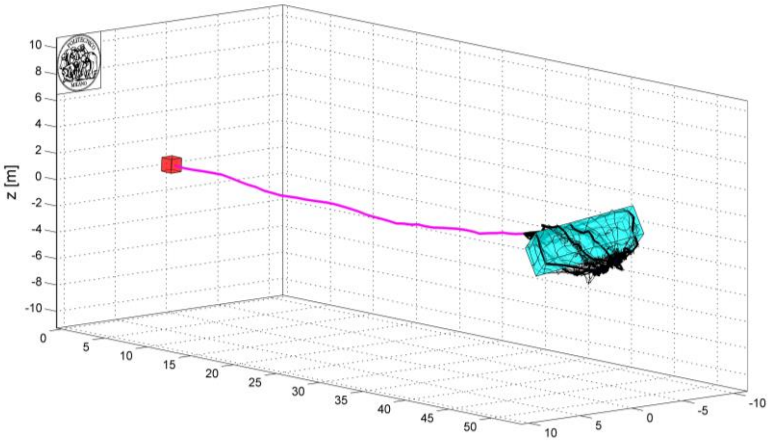


Figure 2.15: example of the primary role of the Tether in the disposal (single Tethered system).

2.6.1 Complete Dynamics Model

Modelling and simulation of structures with such a huge change in displacement configurations (Net folded, deployed and wrapped over the Target) and loads (shock loads from a zero-tension state), requires a particular attention and study. The peculiarities of this flexible system are:

- the high degree of flexibility;
- the big displacements involved.

If we add the further issues of the orbital dynamics propagation and contact dynamics, this becomes a very stiff problem that requires a much more deep study on both the mathematical model and the numerical one. Smart modelling, implementation and simulation strategic techniques are required. Therefore, what we want to do, is to build up a mathematical model for these flexible structures that, starting from the physical nature of the problem, is capable of providing high precision results in a reasonable time. Facing the flexible problem, we have to deal with Partial Differential Equations (PDE) depending on both time and space. To solve numerically the problem, we need to find a way to avoid the spatial dependency and convert it into an equivalent set of time-dependent Ordinary Differential Equations (ODE). The main SoA solutions are:

- Assumed Modes Method (AMM);
- Lumped Parameters Methods (LPMs).

The AMM [58] is very effective in describing both the transversal and the longitudinal vibrations relying on *shape functions* satisfying the boundary conditions of the problem. However, AMM turns out to be suitable for systems made of only few flexible elements, such as for tethered satellite systems. Therefore, with a Net made up of hundreds of flexible elements, AMM is reasonably discarded. Furthermore AMM has been demonstrated to be numerical unstable when describing the dynamics of a simple tether deployment [59]. Instability is related to the large configuration changes. This involves the necessity to redefine time-depending *shape functions* and so the redefinition of the problem from ODE system back to PDE system of equations, holding no more the space-time decoupling. Therefore, we discard the continuous approach to the problem for the following main reasons:

- Instabilities arising with large displacements;
- Complexity and inefficiency coming from the inclusion of the essential boundary conditions in the equations.

LPMs become the reasonable alternatives. Theses approaches are proven from literature to be reliable exactly for Net structures (fishing applications) [19] [20] [22]. There is not a unique LPM strategy. Generally speaking, Lumped Parameters Methods are displacements approaches that involve the approximation of a continuous system as a discrete system of lumped masses and viscoelastic properties at discrete geometrical points (nodes); elements are connected and loaded at these nodes. We start describing the model following the most general approach, that is the interpretation of the Nest as a discrete system of beams according to the Euler-Bernoulli beam model. According to this model, shear is put to zero to solve the problem and then it is generally recovered afterwards through the Timoshenko beam model. In our case, we deal with threads approximated by strings, which have no transversal stiffness by definition, so no shear forces need to be taken into account or recovered afterwards. As explained previously, this is a displacement approach, meaning that all the unknowns are expressed in terms of displacements. The constitutive law, arising from the equilibrium system of equations associated (Eq. 2.46), can be resumed with the stiffness matrix of Eq. 2.47 and with the conventions of Fig. 2.16.

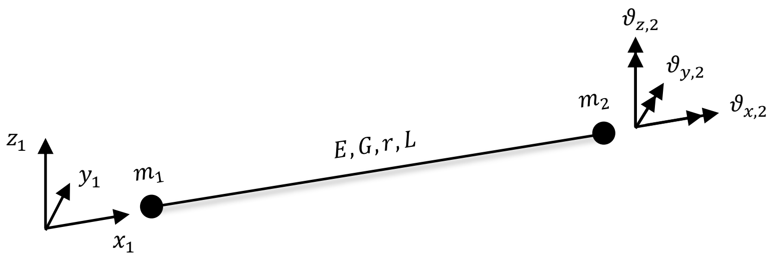


Figure 2.16: beam element DOF conventions at the two bounding nodes (1) and (2).

$$\boldsymbol{F} = \boldsymbol{K} \cdot \boldsymbol{u}, \qquad \boldsymbol{F} = \begin{Bmatrix} F_{x,1} \\ F_{y,1} \\ F_{z,1} \\ T_{x,1} \\ T_{y,1} \\ T_{z,1} \\ F_{x,2} \\ F_{y,2} \\ F_{z,2} \\ T_{x,2} \\ T_{y,2} \\ T_{z,2} \end{Bmatrix}, \qquad \boldsymbol{u} = \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \\ \theta_{x,1} \\ \theta_{y,1} \\ \theta_{z,1} \\ x_2 \\ y_2 \\ z_2 \\ \theta_{x,2} \\ \theta_{y,2} \\ \theta_{z,2} \end{Bmatrix} \tag{2.46}$$

Assuming the threads as slender cylinders of radius r and length L, and whose Young's and shear moduli are E and G respectively, the components of the stiffness matrix $\boldsymbol{K}$ are reported.

$$\boldsymbol{K} = \left[\begin{array}{cccccc|cccccc} X & 0 & 0 & 0 & 0 & 0 & -X & 0 & 0 & 0 & 0 & 0 \\ & Y_1 & 0 & 0 & 0 & Y_2 & 0 & -Y_1 & 0 & 0 & 0 & Y_2 \\ & & Z_1 & 0 & -Z_2 & 0 & 0 & 0 & -Z_1 & 0 & -Z_2 & 0 \\ & & & S & 0 & 0 & 0 & 0 & 0 & -S & 0 & 0 \\ & & & & Z_3 & 0 & 0 & 0 & Z_2 & 0 & Z_4 & 0 \\ & & & & & Y_3 & 0 & -Y_2 & 0 & 0 & 0 & Y_4 \\ \hline & & & & & & X & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & Y_1 & 0 & 0 & 0 & -Y_2 \\ & & & & & & & & Z_1 & 0 & Z_2 & 0 \\ & & SYM & & & & & & & S & 0 & 0 \\ & & & & & & & & & & Z_3 & 0 \\ & & & & & & & & & & & Y_3 \end{array} \right] \tag{2.47}$$

X is the axial stiffness per unit length:

$$X = \frac{EA}{L}\ [N/m] \tag{2.48}$$

S is the torsional stiffness per unit length:

$$S = \frac{GJ_p}{L}\ [Nm/rad] \tag{2.49}$$

The bending stiffness components are the following:

$$\begin{gathered} Y_1 = Z_1 = 12\frac{EJ}{L^3} \quad ; \quad Y_2 = Z_2 = 6\frac{EJ}{L^2} \\ Y_3 = Z_3 = 4\frac{EJ}{L} \quad ; \quad Y_4 = Z_4 = 2\frac{EJ}{L} \end{gathered} \tag{2.50}$$

The equivalency $Y_i = Z_i$ is possible under the aforementioned assumptions of circular shape of the cross section. If further assumptions of isotropic material were done, it would be possible to directly compute the axial, bending and torsional stiffness from the Young's and shear moduli, explicitly defining the area, the specific moment of inertia and the specific polar moment of inertia, that in the case of circular sections are in the order:

$$A = \pi\, r^2 \tag{2.51}$$

$$J = \frac{\pi}{4}\, r^4 \tag{2.52}$$

$$J_p = 2J = \frac{\pi}{2}\, r^4 \tag{2.53}$$

Shear stiffness, neglected up to now, may arise from pre-stress. We talk about *Geometric Shear Stiffness*, which is exactly equal to the pre-stress axial force T_0, and we get:

$$T_y = \frac{T_0}{L}\, \Delta v \tag{2.54}$$

$$T_z = \frac{T_0}{L}\, \Delta w \tag{2.55}$$

The important assumption at the base of the modelling is the coincidence of the mass axis with the elastic axis. This lets to decouple the Axial, Bending and Torsional dynamics, as can be clearly seen in Eq. 2.47.

At this step, we would have to build up the model of the Net system and then include all the essential boundary conditions (defining the Net mesh). This wold involve the definition of the stiffness matrix (Eq. 2.47) for each element in its local reference frame, then its rotation in the ECI frame at each time instant and finally the composition of a single huge stiffness matrix of the whole system [6N x 6N], where N is the number of all the system discretizing nodes:

$$\boldsymbol{K}_{SYSTEM} = \sum_{i=1}^{N} \boldsymbol{\Omega}_i^T \cdot \boldsymbol{K}_i \cdot \boldsymbol{\Omega}_i \tag{2.56}$$

The conclusion is that the computational work load would be really huge as well as the dimensions of the variables built, so we better find a solution to extrapolate the essential boundary conditions from the model and impose them externally *a-posteriori*. This is possible redefining the model and the primary goal: convert the single body problem with the definition of its constitutive law, into a multi-body problem made of single elements constitutive laws (Fig. 2.17). In this way we define independently each thread element as a free rigid body (Eq. 2.1 and Eq. 2.3) with its mass m_i and inertia $\boldsymbol{I}_i$ (still in the element-local frame) and impose the essential boundary conditions as penalty reactions, defining the geometric pattern of the Net beside the model. Axial, Bending and Torsional stiffness are decoupled as explained before and, thanks to this, they are concentrated as lumped axial, flexural and torsional springs at the joints (Fig. 2.17a). Penalty forces to be applied at the connections are computed taking into account both stiffness and energy dissipation. Therefore, we model viscoelastic joints as spring-damper systems. Axial, flexural and torsional stiffness are retrieved from the material constitutive law, specifically from Eq. 2.48, Eq. 2.49 and Eq. 2.50, respectively EA, EJ and GJ_p. Energy dissipation coefficient needs to be retrieved from material tests as well (Section 2.6.2).

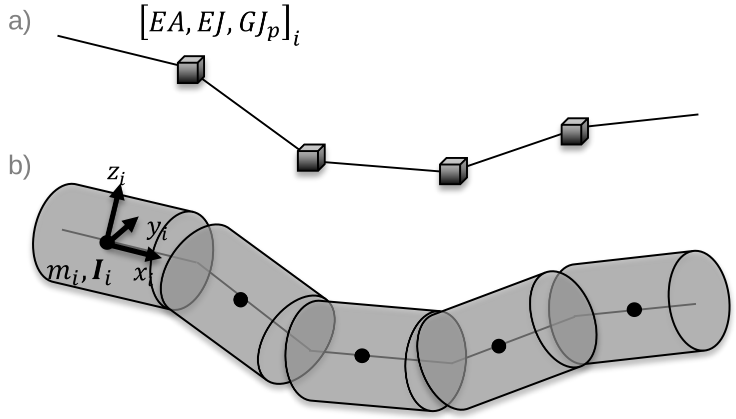


Figure 2.17: LPM applied to a thread discretized with multiple rigid-body nodes (b) and connected through viscoelastic joints (a).

Different solutions in modelling the joints are possible, such as Maxwell, Zener, Kelvin-Voigt and Hunt-Crossley. The first two are differential laws, the latter are algebraic. Difference between Maxwell and Kelvin-Voigt models is that they are equivalent to spring-damper systems respectively in series and in parallel (Fig. 2.18). This representation has to be intended also for bending and torsional dynamics.

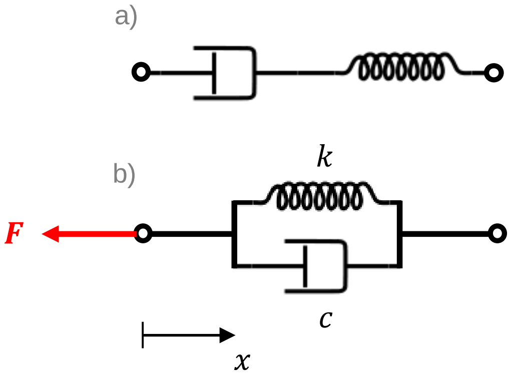


Figure 2.18: a) Maxwell, b) Kelvin-Voigt viscoelastic models comparison.

We choose algebraic laws, and in particular the Kelvin-Voigt one (Eq. 2.57), in line with the drivers of the Simulator design, that is simplicity in construction, easiness of handling the parameters and minimization of the computational work-load. Kelvin-Voigt model is expressed as follows for the axial action:

$$F = -k\,x - c\,\dot{x} \qquad (2.57)$$

Meaning that at the connection joints, it acts an axial force opposite and proportional to the deviation from the nominal distance x and relative velocity $\dot{x}$ projected on the axial direction. The proportional coefficients are the axial stiffness k corresponding to Eq. 2.48 and axial damping coefficient c, both to be found through the material characterization.

The same viscoelastic modelling is assumed for bending and torsional joints, where x stands for the corresponding relative bending and torsional angle. Corresponding stiffness values could be retrieved from Eq. 2.49 and Eq. 2.50, but they are better directly found from material testing as follows.

2.6.2 Net Threads Material Characterization

In order to finalize the model of the Net and the Tether, we need to find the aforementioned Axial, Flexural and Torsional stiffness as well as the corresponding damping factors. This is done in order to have directly real validated numbers and avoid geometrical and homogeneity assumptions done before. Tests are required and essentials also because of the difference between the declared properties of the fibers and the braid.

The goal is to validate the model proposed in the previous paragraph and possibly improve it acting on focused simplifications (according to the characterization of the specific material). The Net fiber material characterization campaign can be summed up with Tab. 2.1 and Tab. 2.2.

Table 2.1: Technora braid estimated values of axial, torsional and bending stiffness.

Quantity	Symbol	Estimated Value	Units
Axial stiffness per unit length	$\boldsymbol{EA}$	$9.84 \cdot 10^{3}$	N
Torsional stiffness per unit length	$\boldsymbol{GJ_p}$	$2.94 \cdot 10^{-6}$	$\frac{Nm}{rad}m$
Bending stiffness per unit length	$\boldsymbol{EJ}$	$1.34 \cdot 10^{-6}$	Nm^2

Table 2.2: Technora braid estimated values of axial, torsional and bending damping ratio.

Quantity	Symbol	Estimated Value
Axial damping ratio	$\boldsymbol{\zeta_{ax}}$	0.106
Torsional damping ratio	$\boldsymbol{\zeta_t}$	0.079
Bending ratio	$\boldsymbol{\zeta_b}$	0.014

What can be clearly noted are the almost 10 orders of magnitude of difference between the axial stiffness and the bending and torsional ones. According to these results, simplifications may be done neglecting the corresponding dynamics and drastically reducing the complexity of the elastic model, the whole flexible structure and the entire Simulator. It is necessary to remark that this assumption holds as long as dealing with Net threads materials with such low bending and torsional stiffness.

This is verified for most of synthetic fibers with high bending flexibility and small cross-section, which are exactly the ones desired from our mission requirements (it does not hold for metallic wires, but they are not even considered for these missions).

2.6.3 Reduced Dynamics Model

Besides the considerations up to now explained on the bending and torsional dynamics, a further implication of the low value of the bending stiffness is the compression instability, which implies a reconsideration of the rigid bodies LPM designed so far. This issue is related to the so-called *Euler Buckling load*, so defined:

$$N_0^* = -\frac{EJ\pi}{L^2} \tag{2.58}$$

It is the minimum compression axial load responsible for the structural instability of the thread of length L and bending stiffness EJ. Having a very low value of such stiffness and very high loads in all directions for the threads to withstand (think of the huge compression loads in order to store the Net inside the Canister), this instability would occur and could not be handled by the model, which cannot avoid compression to be taken into account.

Therefore a reduced model is not only advised, but necessary to handle structural axial intrinsic instability. It is advised because in this way we reduce by a factor of two the overall system of dynamics equations, neglecting the attitude of nodes and their dynamics, and so the computational work load.

The model is revised neglecting the inertia of the nodes and their attitude. Nodes are then converted into lumped masses with only 3 DOF instead of the 6 previous. These masses m_i are located now at the joints, and equivalent to half of the mass of the connecting threads j:

$$m_i = \sum\nolimits_j \rho_{l,j} \frac{L_j}{2} \tag{2.59}$$

Where $\rho_{l,j}$ is the linear density (kg/m) of the thread j, while L_j its length.

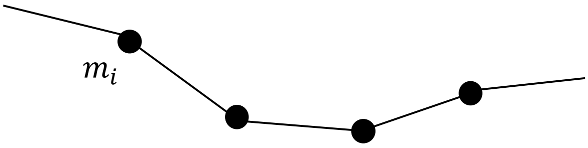


Figure 2.19: concentrated masses Lumped Parameters Model of a thread.

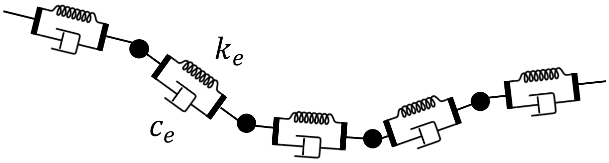


Figure 2.20: axial viscoelastic constitutive law reduction.

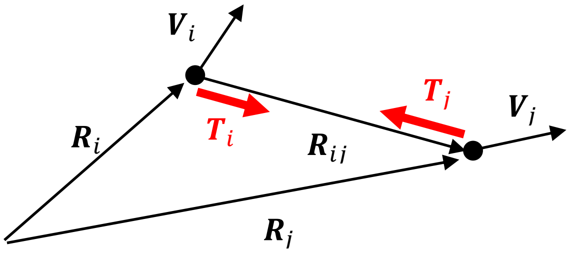


Figure 2.21: axial traction computation, supporting.

The viscoelastic law is reduced to a simple axial penalty traction force, so avoiding compression and axial instability. Therefore, for the thread element *e*, the tension force acting on the bounding node *i* is derived from Eq. 2.57:

$$\boldsymbol{T}_i = \begin{cases} [-k_e(|\boldsymbol{R}_{ij}| - L_e) - c_e(\boldsymbol{V_{ij}} \cdot \widehat{\boldsymbol{R}}_{ij})]\widehat{\boldsymbol{R}}_{ij} & if|\boldsymbol{R}_{ij}| > L_e \\ \boldsymbol{0} & if|\boldsymbol{R}_{ij}| \leq L_e \end{cases} \qquad (2.60)$$

Where:

- $\boldsymbol{R}_{ij}$and $\boldsymbol{V_{ij}}$ are the relative distance and velocity vectors of nodes *i* and j;
- L_e is the nominal thread element length;
- k_e is the axial stiffness (Tab. 2.1);
- c_e is the damping coefficient derived from the damping factor ζ_{ax} (Tab. 2.2):

$$c_e = 2\zeta_{ax}\sqrt{k_e m_e} \qquad (2.61)$$

Where m_e is the thread element mass, computed from the thread linear density ρ_e:

$$m_e = \rho_e L_e \qquad (2.62)$$

The same opposite force is finally applied to the other node j:

$$\boldsymbol{T}_j = -\boldsymbol{T}_i \qquad (2.63)$$

Concluding, the great advantage of dealing with high flexibility order structures is that they can be treated with the most simple lumped masses multi-body approach, simplifying not only the viscoelastic constitutive law, but the whole Simulator computing work load and memory. This is thanks to the drastic reduction of the DOF (where 50% of hundreds of thousands of DOF is a great improvement) and the reduction of the dynamics equations associated (Euler's equations and relative attitude computation).

Finally, but most important, a LPM-based model so defined allows to obtain an explicit system of ODEs, integrating accelerations at each time step.

What we have done at the end, is a re-definition of such a single complex flexible body that is the Net, as a multi-body system made of a discrete number of flexible elementary bodies, extrapolating the essential boundary conditions defining the Net mesh structure. Viscoelastic forces are treated a-posteriori with a penalty method and applied directly at nodes. This simplify also the introduction of the external forces as explained below.

2.6.4 External Forces

Thanks to the representation of the Net as a flexible multi-body system discretized with thousands of nodes, it is simply required to apply the **gravity** force to each node to have as a further consequence a gravity gradient equivalent effect on the flexible system. This is an example of the effect of the discrete representation of a generic body, which does not require the modelling of a complex time varying equivalent inertia matrix, since this inertia is considered (by definition) as the way mass is distributed, which is indirectly carried in a discrete way by the nodes position and their mass.

Neglecting solar radiation, **atmospheric drag** is implemented, because of its importance in the case of simulations on ground or *zero-g* flights. To do so, we recall the same considerations made for the Bullets drag modelling. As Bullets, Net threads are cylindrically shaped, but differently from the former, these are not rigid bodies, so the drag force is retrieved assuming an ideal direct link between two discretizing thread nodes (Fig. 2.22). Drag force is made of the only lateral component, differently from the Bullets. Therefore, the drag coefficient value is computed as before from Eq. 2.43.

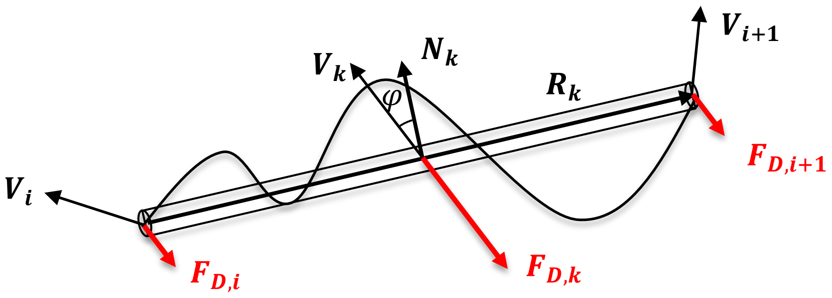


Figure 2.22: drag force geometrical approximation of the thread link connecting nodes *i* and *i+1*.

The velocity of the thread is obtained as the mean velocity of the two bounding nodes:

$$\boldsymbol{V}_k = \frac{\boldsymbol{V}_i + \boldsymbol{V}_{i+1}}{2} \tag{2.64}$$

So, for the *k* thread element, the drag force expression of Eq. 2.13 simplifies to:

$$\boldsymbol{F_{D,k}} = -\frac{1}{2}\,\rho\,\boldsymbol{V_k}\,d_k\,c_D\,|\boldsymbol{R}_k \times \boldsymbol{V_k}| \tag{2.65}$$

Where d_k is the thread diameter and $\boldsymbol{R}_k$ the relative distance of node i*+1* from node *i* (**ECI** frame):

$$\boldsymbol{R}_k = \boldsymbol{R}_{i+1} - \boldsymbol{R}_i \tag{2.66}$$

Finally, the resulting drag force acting on the *i-th* thread element is split and applied to the bounding nodes:

$$\boldsymbol{F}_{D,i} = \sum\nolimits_k \frac{1}{2}\boldsymbol{F_{D,k}} \tag{2.67}$$

Torque-effects arise indirectly from the unbalanced forces acting on the nodes, due to the several different adjacent contributions from other threads.

2.7 Shooting System

The Net-shooting task is accomplished by a pneumatic system, which interfaces with the Bullets. This system, in the specific, is based on compressed gas ejection in the case of ground and parabolic tests, while in the real operative case it relies on Cold Gas Generators (CGG).

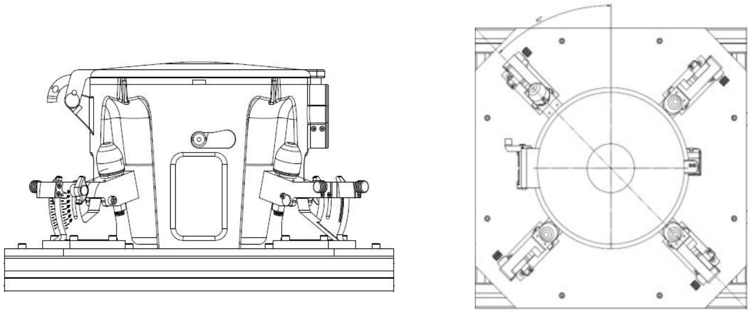


Figure 2.23: side and top view of the experimental Canister and the Shooting System. Four Bullets case. *(courtesy of PRODINTEC).*

Its main task is to provide the exact amount of pressure for a specific time impulse in order to obtain as output the desired ejection velocity of the Bullets. It is required to:

- Minimize the Δv error between the Bullets;
- Minimize the time delay between the shots, aiming at a precise synchronization.

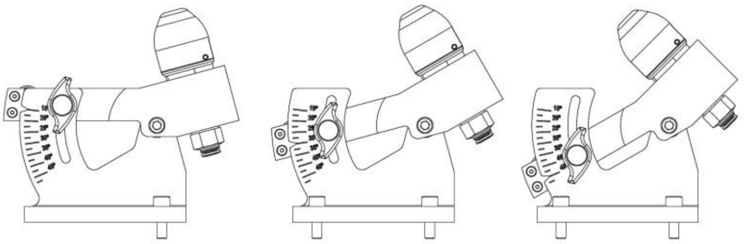


Figure 2.24: side view of the Bullets experimental titling platform. Set for three difference divergence angle *(courtesy of PRODINTEC).*

Bullets are placed around the Canister (Fig. 2.23) on a tilting platform (Fig. 2.24) that can be regulated before the shot, tuning the divergence angle α_{div}. A second shooting parameter is the ejection velocity v_b of the Bullets, coming from the whole capture dynamics design and it is accomplished through the regulation of the system pressure and duration of the impulse.

The pressurized gas system is much more flexible and easier to handle. It provides a tunable constant pressure for all the impulse duration. Being controlled by electric valves, it also ensures a perfect synchronization between the Bullets shots. So, generally speaking, all the three tasks are perfectly respected. The drawback, that limit its usage in the operative mission, is the weight of the pressurized tank, feeding line, valves and electrical system, resulting a more complex system.

Cold Gas Generators become the operative solution in the orbital context, since they are standalone lightweight systems, cutting down the mass budget of 94% with respect to the previous choice. The main drawback is that they are difficult to synchronize (if operated separately). So, a solution could be their reduction to a single gas generator feeding all the Bullets (as the pressurized gas feeding line).

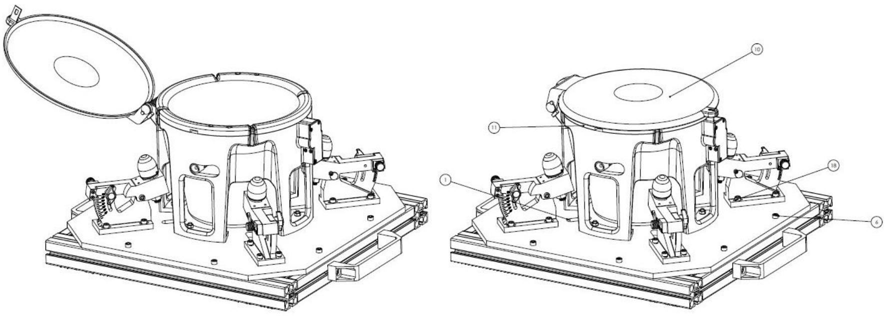


Figure 2.25: experimental Canister open and closed configuration *(courtesy of PRODINTEC).*

In both cases, the Shooting System and Canister are modelled as a system that constraints the whole Net and all the Bullets, until it is give the electrical signal to open the Canister (Fig. 2.25) and shoot the Bullets. At that time, the Net

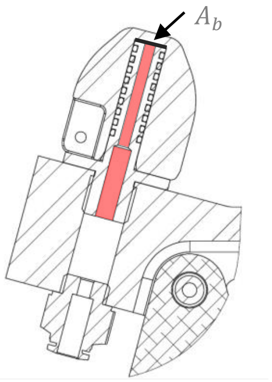


Figure 2.26: experimental Bullet section. Highlighted the pressure chamber *(courtesy of PRODINTEC).*

dynamics is left unconstrained and the Bullets are given a specific axial force F_{shot} for a time impulse $\Delta t_{impulse}$. In the case of pressurized Shooting System, the pressure is assumed constant ($\bar{P}$) for all the time impulse and given the desired Bullets ejection velocity v_b, we easily retrieve the axial force and the associated shooting time duration:

$$F_{shot} = \bar{P} A_b \tag{2.68}$$

$$\Delta t_{impulse} = \frac{m_b\, v_b}{F_{shot}} \tag{2.69}$$

Where m_b and A_b are respectively the Bullet mass and the cross-section area of the Bullet's pressure chamber (where the pressure acts). In the case of Gas Generators, a more detailed model is designed, taking into account the whole gas expansion process. Assuming an isentropic expansion:

$$P\, v^{\gamma} = const \tag{2.70}$$

The related work is:

$$W = m_{GG}\, P_1\, v_1^{\gamma} \left(\frac{v_2^{1-\gamma} - v_1^{1-\gamma}}{1-\gamma} \right) \tag{2.71}$$

Where:

- γ is the specific heat ratio of the generated gas;
- m_{GG} is the mass of gas generated;
- P_1, v_1 and P_2, v_2 are the pressure and specific volume at the beginning and at the end of the expansion.

The initial conditions are retrieved from the nominal declared pressure $\bar{P}$, volume $\bar{V}$ and generated gas mass m_{GG} (defining the gas generator) through Eq. 2.70:

$$P_1 = \bar{P} \left(\frac{V_1}{\bar{V}} \right)^{\gamma}, \qquad v_1 = \frac{V_1}{m_{GG}} \tag{2.72}$$

Where V_1 is the pressure chamber initial volume (Fig. 2.26). Eq. 2.71 is solved for the final specific volume v_2, once imposed the work that the gas has to do with its expansion, which is equal to the one done by the Bullet to achieve the desired velocity v_b:

$$W = \frac{1}{2}\, m_b\, v_b^2 \tag{2.73}$$

Then, P_2 can be derived from Eq. 2.70 as before and the initial and final forces which the Bullet is subjected to, are so computed:

$$F_i = P_i A_b \tag{2.74}$$

Concluding, imposing the desired shooting velocity, we are able to derive the axial force to impose and the duration of application in the numerical model:

$$F_{shot} = \frac{F_1 + F_2}{2} \tag{2.75}$$

$$\Delta t_{impulse} = \frac{v_b\, m_b}{F_{shot}} \tag{2.76}$$

2.8 Closing Mechanism

The closing mechanism task is to finalize the complete capture of the Target, sealing the Net around, trying to satisfy the requirements of maximum reliability and safety for the thrusting phase and the whole disposal. Its further functionalities are:

- Support the Target wrapping in presence of a significant spinning attitude;
- Speed up the wrapping phase after the Net impact over the Target;
- Increase the clamping robustness avoiding slippages of the Net on the Target, which lets to have a more direct control on the Target attitude.

It is advised to be:

- Passive
- Event driven
- Technology ready.

The solution is found exploiting winches to wind the perimeter edges of the Net and unlocked at a tagged time-span after the first contact with the Target. These further links may pass through perimetric rings in correspondence of all or some of the Net knots or even connect directly the winches (Fig. 2.14). Winches are located at Net inlet section corners and are activated by sensors measuring the strain of strategic threads (strain gages). As baseline they are integrated inside the Bullets, but in the Simulator they can be placed separately for analyses.

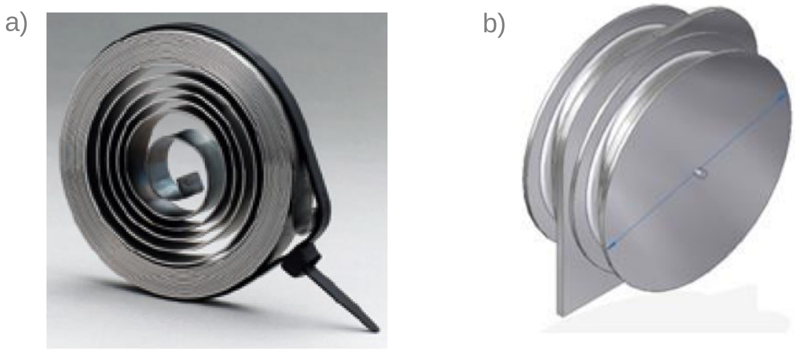


Figure 2.27: a) example of spiral spring, b) representation of the counter rotating system.

The winding mechanism, it is passive (as required), based on two counter-rotating spiral springs mounted on the same shaft (Fig. 2.27), each of them winding half of the perimeter edges of the Net. The advantages of this configuration are:

- No tumbling or gyroscopic effects arising, due to the nullifying effect of the counter-rotating system;
- Half the winding speed is necessary, comparing the solution that collects the entire Net edge in a single winch;
- Failure proven, due to the double redundancy over the same thread: if a winch encounters problems in activation or winding, there still the other supplying the winding (Fig. 2.28). Of course, closing times are doubled in this particular case, but the closure is ensured at the end.

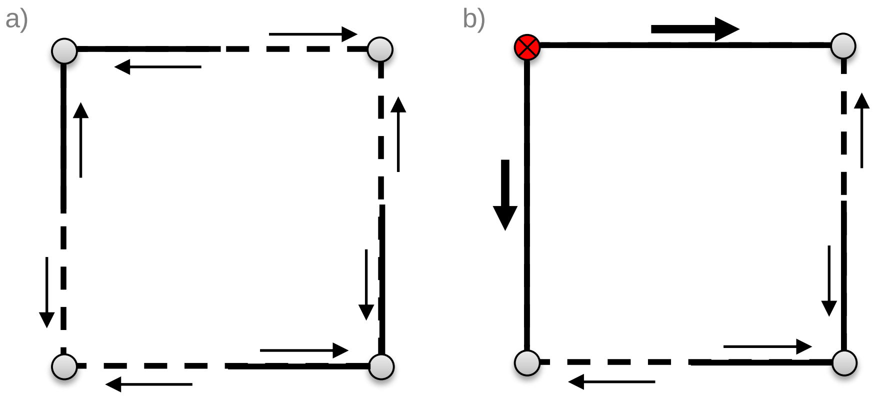


Figure 2.28: a) baseline winding scheme with 4 winches at the corners, b) single winch failure dynamics, adjacent winches support the winding at the same velocity, doubling the overall time.

So, being the winding speed the main driver of the spring dimensioning, the system required performance is reduced, as well as the complexity and the mass budget. Finally, for the closure activation, here we propose two strategies:

- Passive system relying on the pulling effect of the *bulletLinks* on a winch placed inside the Bullet;
- Electric impulse from strain gages positioned on strategic threads.

The *bulletLinks* have the highest sensitivity for the impact, so they will certainly be interested by those stress peaks. This is due to the fact that they are the only threads with the highest mass difference between the bounding nodes (Bullets). The first solution is passive, as advised from the requirements, but poorly feasible for three reasons:

- Need to avoid initial puling effect caused by the Bullet shot;
- Need to activate simultaneously all the Bullets at closure;
- Difficult to tune the closure synchronization.

Therefore, it would necessarily require an integration of electronics, with the exception of special simple dynamics, as long as validated through simulations. Hence, the second solution seems more reasonable. Analyzing the problem, we conclude that the strategic threads, to be monitored for the activation of the mechanism, cannot be defined always a-priori since they depend on the Net shape (2D or 3D), mesh geometry and the interaction with the Target. They need to be found through dynamics simulations according to the synchronization desired. So, a first degree of freedom of the synchronization is the identification of the threads to monitor, while a second parameter of tuning is represented by the activation time delay (once the triggering strain level has been exceeded). This is done through the activation of auto-resettable pin puller via an electrical capacitive circuit, able to delay the activation of the winch. The final issue to solve, is to ensure the simultaneous activation of all the winches. The input should be the same for all the actuators, implying them to have access to all the strain gages. What we want to avoid, in fact, is the eventuality that some winches may not be activated due to the not reached strain level measured by the others. This proves the importance of preliminary dynamics simulations to identify the trigger level and the threads where to put the strain gages. A redundant strategy consists in placing extensometers in correspondence of the *bulletLinks*, because of their reliability. It is considered a back-up solution since, despite the possibility of not having the desired synchronization, it provides a secure closure at the end (decreasing the risk of capturing failure). The closing mechanism is therefore modelled providing:

- The closing time range $\Delta t_{closing}$
- The time delay Δt_{delay} of the closure, starting from the first impact
- The percentage $c_{\%}$ of *Net-mouth* area reduction after the closure

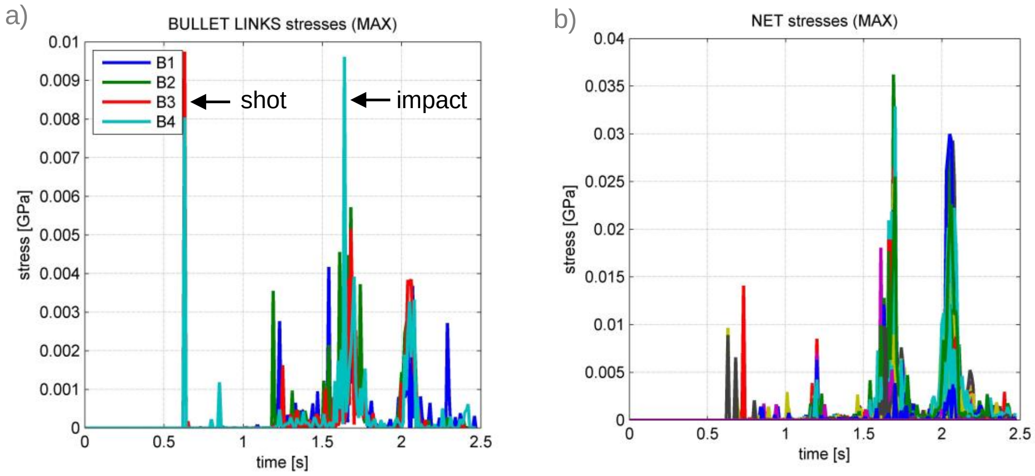


Figure 2.29: a) *bulletLinks* maximum stresses with time, b) Net threads maximum stresses with time for comparison. Shot and impact are easily identified monitoring the *bulletLinks* (a).

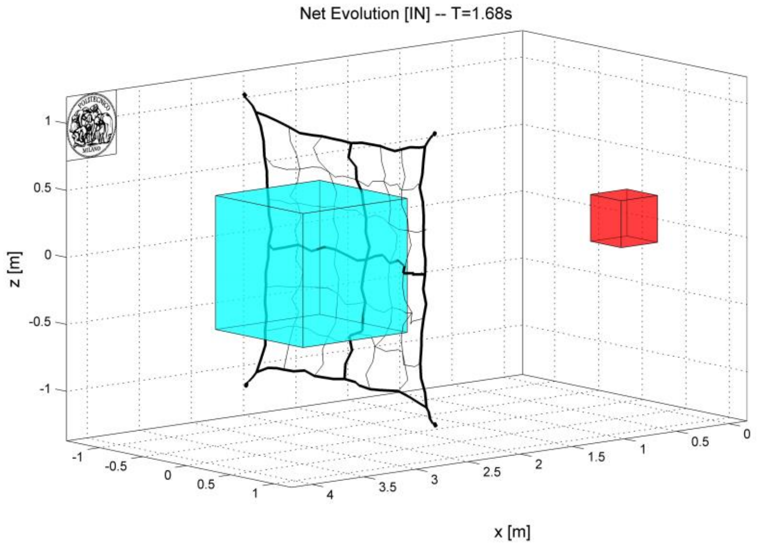


Figure 2.30: impact time instant representation, supporting the analysis on the threads stresses with time.

The impact time instant (t_{impact}) is effectively the one directly coming as output of the collision detection S*imulation Block*. Time synchronization with the threads stresses is post-processed after the simulation. In this way we do not monitor at each integration time step all the stresses, since this define an algebraic loop that cannot be integrated in *Simulink*. The winding effect is modelled defining a length law reduction (linear) for the closing links, which is computed in the pre-processing from the data provided:

$$L(t) = \begin{cases} L_0 & , t < t_{impact} \\ L_0 - v_w \left(t - t_{impact}\right), & t_{impact} \leq t < (t_{impact} + \Delta t_{closing}) \\ L_f & , t \geq (t_{impact} + \Delta t_{closing}) \end{cases} \tag{2.77}$$

Where v_w is the winding velocity:

$$v_w = \frac{L_0 - L_f}{\Delta t_{closing}} \tag{2.78}$$

L_0 and L_f are respectively the initial and final length of the thread to wind up. The final length is derived from the *Net-mouth* area (A_0) reduction after the closure (A_f) so defined:

$$c_{\%} = \frac{A_0 - A_f}{A_0} \tag{2.79}$$

So, the final length is obtained from the initial one and the area reduction:

$$L_f = L_0 \sqrt{1 - c_{\%}} \tag{2.80}$$

As long as the length is reduced, the corresponding mass of the winded portion of the links is added to the related winch:

$$\Delta m(t) = \rho_L \big(L_0 - L(t)\big) \tag{2.81}$$

Being ρ_L the linear density of the thread (kg/m).
Winches are treated as lumped masses as long as they are alone; if they are integrated inside the Bullets, then also the inertia of this last will be dynamically updated during the iteration:

$$\boldsymbol{I}(t) = \boldsymbol{I}_0 \left(1 + \frac{\Delta m(t)}{m_0}\right) \tag{2.82}$$

Being m_0 and $\boldsymbol{I}_0$ the initial Bullet mass and inertia matrix.

Chapter 3

COLLISION DETECTION

3.1 Overview

Collision detection is one of the most appealing and open issue in multi-body simulations, that is why there are a lot of approaches, each optimized case by case and it does not exist a unique way to solve it.

Most notable papers dealing with the subject are Von Herzen, Barr and Zatz [1] and Moore and Wilhelms [2]; in robotics, a number of papers of interest are Canny [3], Gilbert and Hong [4], Meyer [5] and Cundall [6]. Preparata and Shamos [7] describes many approaches in computational geometry to the problem. All involves the concept of hierarchical structures (or trees) of bounding volumes [28], that are volumes surrounding specific subsets of nodes. The main differences among these approaches consist in the way these volumes are defined. The objective of such technique is to smartly select which nodes to analyze for collision detection as the dynamics of the nodes evolves, having as final objective the maximization of detection precision and minimization of the computational cost.

Moreover for the recently increasing interest in the problem, a plenty of open-source algorithms have been developed and for this reason, we have analyzed their theoretical approach to the subject. They are most diffused in the 3D animation and in the virtual reality scenarios such as gaming. Most important libraries in this sense are *Chrono parallel multi-physics library* [35] and the ones developed by the American research group *GAMMA* [15]. Among these codes, they have been studied in particular the ones concerning the collision detection and contact between clothes and rigid bodies (as most resembling the physics of the Net wrapping on the Target), such as *iCollide* [13] [34] and *SWIFT* [12]. Further important ones, although less *physical*, are *Bullet* [14] (mainly used in 3D animation) and *SOFA* [11]. In the next sections, we will describe how these different State of the Art strategies has been analyzed to build up an *ad-hoc*

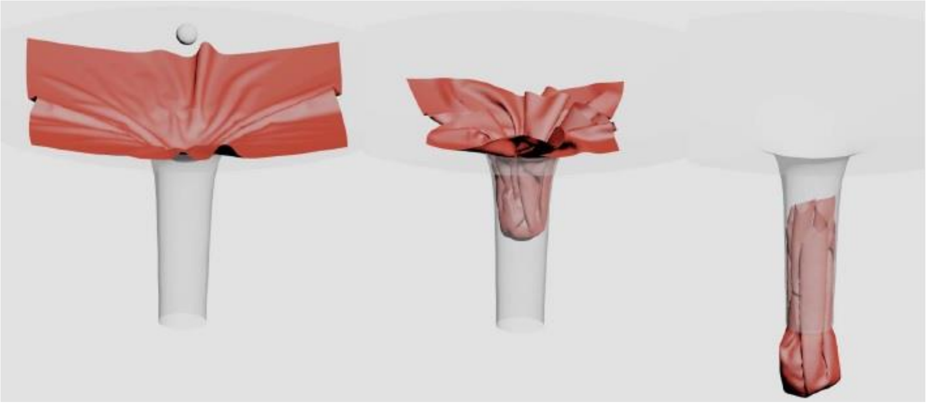


Figure 3.1: *iCollide* benchmark test of a cloth passing into a funnel under the pressure of a ball [13].

collision detection algorithm for the specific flexible multi-body problem, capable of satisfying the requirements of efficiency and precision imposed.

3.2 Strategies Survey

At the highest level we can distinguish two types of collision detection algorithms:

- *Static*
- *Dynamic*

The first one works with a fixed subset of nodes belonging to each of the two (or more) colliding bodies, and it performs a point-by-point cross-check [27]. It is the simplest one, but requires a huge computational cost when dealing with lots of degrees of freedom (e.g. higher than 10^3). Dealing with *n* nodes, the code would require $o(n^2)$ checks.

The second approach works with dynamic subsets of nodes, at different levels, starting with rough precision, refining with levels. This is known as bounding box hierarchy theory [28]. It is more complex, but computationally much more efficient. A *bounding box* or *control volume* is defined as a geometric volume, representative of a specific number of nodes of a colliding body. It is cleverly exploited in order to drastically reduce the number of collision queries to be done at each integration time step, so instead of the *n-points* checks, a subset *m-points* checks is to be performed, where m<n and increasing dynamically with the collision refinement (up to *n*). If two bounding boxes are found not to overlap (so incoherent), no further comparisons involving the contents of the boxes are needed. A naive pairwise comparison requires $o(m^2)$ work and it is too inefficient, unless the number of bodies is small. Computational geometry algorithms exist

that can solve this problem in time *o(m·log(m)+k)* where *k* is the number of pairwise overlaps. Using coherence, we can achieve substantially better performance, therefore it is further analyzed and studied in the following sections.

3.2.1 Bounding Volumes Theory

We are able to identify two classes of control volumes, which are *geometric* and *graphic-based*. The first one opens to different geometries, ordered in complexity:

- Spherical Bounding Volumes (SBV);
- Axis Aligned Bounding Boxes (AABB);
- Oriented Bounding Boxes (OBB);
- Oriented Convex Polyhedra (OCP).

The main advantages in using this class of volumes, are the easiness of construction and the low coherence-check computational cost.

Spherical Bounding Volumes are correctly defined implying the assumption of the minimum spherical volume enclosing a subset of nodes [26] [29].

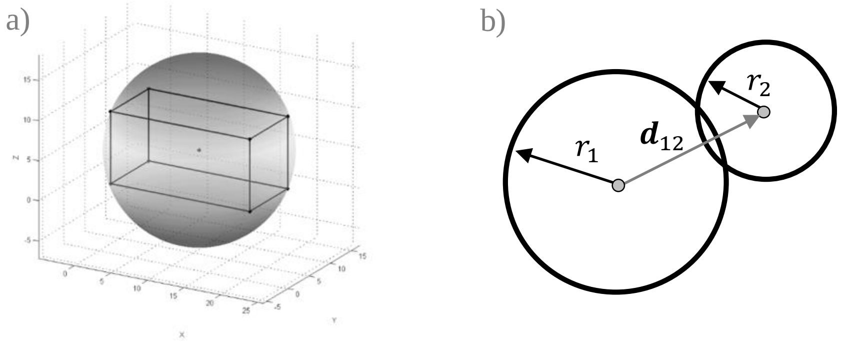


Figure 3.2: a) Spherical Bounding Volume enclosing a specific volume geometry, b) coherence condition.

The coherence between two spheres is identified when the distance between their centers is lower than the sum of the radii: $|\boldsymbol{d}_{12}| < (r_1 + r_2)$.

Axes Aligned and *Oriented Bounding Boxes* are defined, in the same way, as the minimum cuboid volumes enclosing a specific subset of points [17] [18], and they differ according to the reference frame axes with respect to which their edges are oriented:

- INERTIAL frame (for AABB), which is fixed in time
- BODY frame (for OBB), which is time variant

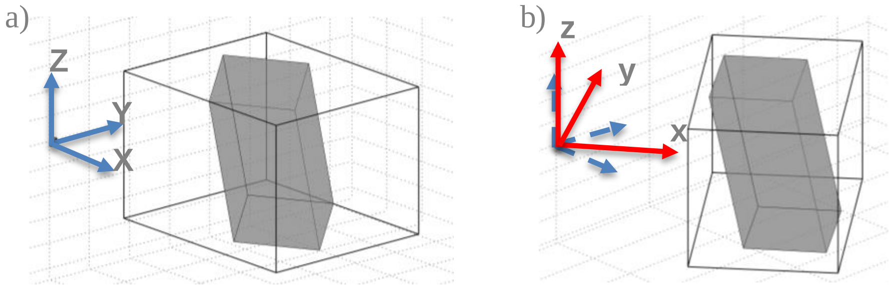


Figure 3.3: comparison between: a) AABB aligned with the ECI frame and b) OBB oriented with the BODY frame.

The second strategy, despite requiring the further orientation computational work, let to better follow the dynamics and surely optimize the coherence checks. This last is proven simply comparing the three spatial coordinates of the minimum and maximum vertices of the Boxes.

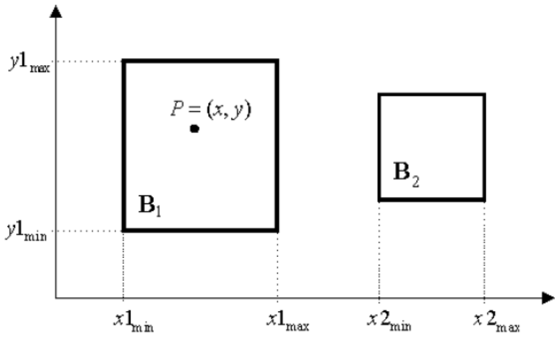


Figure 3.4: bi-dimensional AABB and OBB coherence cross-checks.

The coherence in 3-dimensions space is validated as long as all the three following relations are simultaneously satisfied:

$$\begin{cases} x1_{min} < x2_{max} \ \ and \ \ x1_{max} > x2_{min} \\ y1_{min} < y2_{max} \ \ and \ \ y1_{max} > y2_{min} \\ z1_{min} < z2_{max} \ and \ \ z1_{max} > z2_{min} \end{cases} \tag{3.1}$$

These bounding boxes are computationally the simplest of all linear bounding containers, and the one most frequently used in many applications. At runtime, the inequalities do not involve any arithmetic, and only compare raw coordinates with the pre-computed *min* and *max* constants.

Oriented Convex Polyhedra are more complex volumes, that aim at being more close to the geometry of the body to enclose, minimizing the volume enclosing the subset of nodes [32] [33]. Also known as *Convex Hull*s, they are built at each time step in a recursive way, in order to be the smallest convex 3D volumes enclosing the assigned set of points. Two polyhedra do not inter-penetrate if and only if a separating plane between them exists. Using this type of volumes involves necessarily the introduction of some arithmetic, so non-trivial expressions and a higher computational cost in the algorithm at each time step.

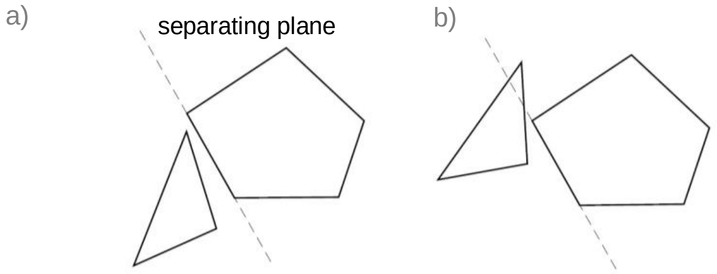


Figure 3.5: bi-dimensional OCP coherence cross-check via the separating plane concept. Not crossing in case a), overlapping in case b).

Alternatives in between the Boxes and the Polyhedra, are *bounding diamonds*, *bounding octahedron* and *cube-octahedron*. They simply consist of OCP with a lower order of complexity; so they are treated the same, but with a lower computational effort with respect to the ideal *Convex Hulls*.

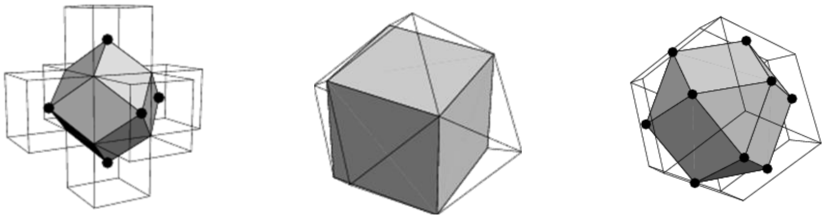


Figure 3.6: lower order *Convex Hulls*.

Finally, dealing with graphic-based volumes, we have: *Voxels*, *Octrees* and *Binary Space Partitioning Trees* (BSP). *Voxel* are simply 3D pixels, which are exploited to represent whichever geometry in space. The higher the refinement the higher the precision, but the longer the computational time.

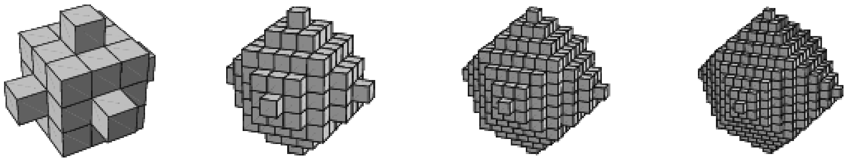


Figure 3.7: *Voxel* representation of a dodecahedron , with increasing refinement to the left.

On the other hand *Octrees* and *BSP* are directly hierarchical tree data structures in which each volumetric node subdivide the 3D space in further nodes. The difference is that each *Octree* is further subdivided in eight octants, while each *BSP* each node has two children. They are strategically used in virtual reality and gaming because are able to ensure the real time computation velocity [16]. This speed is possible only because the precision is rough, if treated as they are conceived for. In other words, they ensure a high speed over precision ratio. If a higher precision in collision detection is desired, their efficiency inevitably decreases.

3.2.2 Hierarchies Definition and Build-up Approach

The data structures associated with hierarchical collision detection is *bounding volume trees*, which can be seen as hierarchies of successive refinement, or as levels-of-detail (LODs). The method they are conceived for is called *Sweep and Prune*, also known as *Divide & Conquer* approach. Whenever a collision is detected between two control volumes, sub-volumes are invoked for both and the collision algorithm is run to find the next collision pairs. This procedure is repeated until the resolution of the contact, imposed a priori, is reached or no intersections occur. The hierarchical collision detection algorithm can be only as good as its associated algorithm for constructing the hierarchies, so this is a crucial point. The goal is to discard quickly as most pairs of BVs not intersecting, for this reason BVs should enclose their associated set of children as tight as possible. Reminding that there is no absolute definition of *optimal* BV hierarchy, we can identify however a set of guiding criteria that should be followed in order to refine and optimize the bounding boxes hierarchy:

N_b

- **Minimize** the total volume of all BVs;
- **Minimize** their overlap with their neighbors ;

T_b

- **Maximize** the simplicity of building of the BVs;
- **Maximize** the simplicity of cross-checking the BVs coherence.

The first two contribute to decrease the number of collision checks (N_b), the last two let to reduce the time required by each check (T_b). Hence, we can define a cost function to minimize, that is the computational time:

$$T = N_b\, T_b \tag{3.2}$$

Therefore *tightness* and *simplicity* can be assumed as the main guideline.

- *Tightness* definition is still an open issue; one criterion could be the ratio Volume/Area, however for cube and spheres it is not such a suitable criterion: for all of them it is =1/6·Diameter.
- Spherical BVs should be preferred since they minimize the total volume of the BV hierarchy.

Finally, in order to build a BV tree we have identified three methods:

- INSERTION method: start with a trivial BV tree (1 volume), then add the others;
- **BOTTOM-UP** method: start defining the final mesh, then group them defining the hierarchies and reaching the root volume;
- TOP-DOWN method: set the root and then subdivide into groups defining the children.

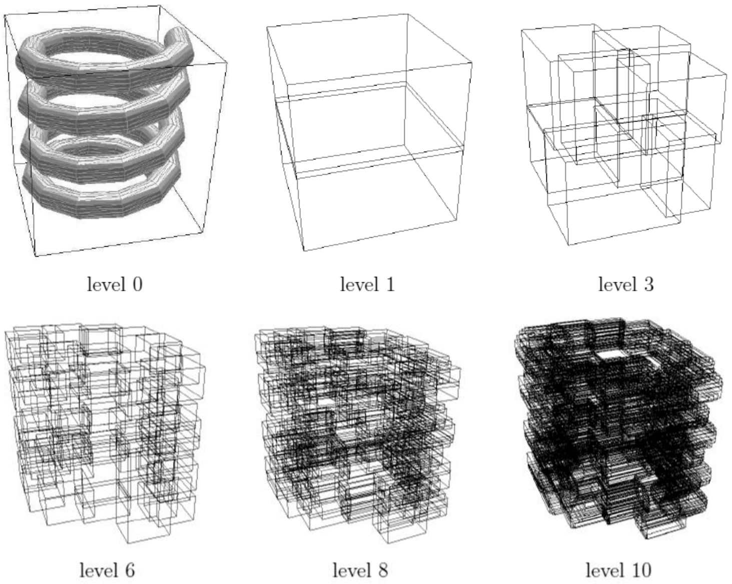


Figure 3.8: example of different BV levels creation in a hierarchical structure [15].

3.2.3 Collision Detection Interface with Contact

We have different ways to determine the collision reactions, all related to strategy chosen to compute the impact time instant:

- use the softness of the material to calculate a force, which will resolve the collision in the following time steps like it is in reality;
- estimate the time of collision by linear interpolation, roll back the simulation, and calculate the collision by the more abstract methods of conservation laws;
- iterate the linear interpolation (Newton's method) to calculate the time of collision with a much higher precision than the rest of the simulation.

In other words, we have to choose one of the two ways: detect the collision *a-posteriori* (after the collision occurs) or *a-priori* (before the collision occurs). In the *a posteriori* case, the physical simulation is advanced by a small time step, then it is checked if any objects are intersecting, or are somehow so close to each other that we deem them to be intersecting. This method is *a-posteriori* because we typically miss the actual instant of collision, and only catch the collision after it has actually happened. In the *a-priori* methods, we would need to write a collision detection algorithm able to predict very precisely the trajectories of the physical bodies, so that the physical bodies never actually interpenetrate. For each method there are advantages as well as drawbacks, and they are discussed in more details in the following section.

3.3 Trade-off Analysis

Recalling the goals of the whole simulator:

- o high precision;
- o high computational speed;
- o maximum flexibility to whichever dynamics of the Net-system scenario;
- o maximum interfaceability with external codes and libraries.

The requirements for the Collision Detection Algorithm become:

- high efficiency of the Collision Detection strategy;
- high efficiency of Collision Detection code;
- possibility to parallelize the code to run faster;
- independency of the Collision Detection algorithm from the Contact one, so that they are autonomous blocks in the code, that can be switched with external libraries (interfaceability requirement);

The architecture chosen is, as already explained, the dynamic one, involving a hierarchical collision detection strategy. Synergy with the characteristic dynamics of the problem, is not only advisable, but necessary in order to be able to build up an ad-hoc optimized algorithm (still flexible for whichever correlated scenario). *Graphic-based* control volumes have been discarded for the low precision and the excessive adjustment difficulties that would be involved. Therefore we analyzed the ***geometric-based*** control volumes. The higher is the complexity of the Bounding Boxes geometry, the higher is the computational cost in the collision detection pre-processing at each time instant, but the lower could be the number of coherence checks (if they were strategically defined and optimized).

Tab. 3.1 reports goals and requirements for the trade-off process on the choice of the BV geometry. In general, dealing with huge number of independent rigid bodies, it is preferred to exploit *Oriented Convex Polyhedra*, also for robustness

Table 3.1: Bounding Volumes geometries trade-off.

GOALS	REQUIREMENTS
It is easy to test that: 1. a point is outside the BV 2. 2 BVs are disjoint.	Have a small number of easy-to-compute inequalities to test inclusion of a point in the BV (*simplicity* condition). Good candidates are **SBV** and **OBB**.
It is efficient to compute and store the BV.	Minimize the information and time required to build the BV (*simplicity* condition). Good candidates are **SBV** and **OBB**.
The geometric object is closely approximated by the BV.	Minimize the volume of the single BV (*tightness* condition) and the overlap with its neighbors. Good candidates are **OCP**.
The free dynamics of the Net is maximized in time, before engaging the Contact checks (time expensive).	Optimize the Collision Detection BV geometry for the deployment of the whole Net. Good candidates are **OCP**.

to whichever dynamics of the involved colliding bodies. However, we can take advantage of the characteristic dynamic evolution of the system:

- The bodies involved in the impact are only two (Net and Target);
- The dynamics has a preferential direction (Net-shooting direction).

Therefore, satisfying the same BV tightness and keeping the architecture robustness, OCP can be strategically simplified with ***Net-Oriented Bounding Boxes*** drastically reducing the aforementioned cost function (Eq. 3.2), maximizing the simulation speed in the first phase.

In a second phase, when hierarchies are invoked, ***Spherical Minimum Bounding Volumes*** should be preferred for the easiness of construction, coherence test and the minimum storage requirement. All this is translated in a computing speed-up. The hierarchical structure is built up with a ***bottom-up strategy***. The main reason is the simplicity of the architecture, since the body mesh provided by the external user becomes directly the finest hierarchy Level, then lower Levels are automatically created simply grouping subsets of higher Levels. Therefore a multi-geometry hierarchy architecture is chosen, where both Net-OBB and SMBV are involved and a bottom-up building strategy advised.

Table 3.2: *a-posteriori* vs. *a-posteriori* strategy comparison.

METHOD	PROS	CONS
a-priori	1. Increased fidelity and stability	1. Difficult to separate the physical simulation from the collision detection algorithm. 2. A numerical root finder is required to estimate the impact time.
a-posteriori	1. a simple list of physical bodies is fed to the algorithm, and the program returns a list of intersecting bodies. 2. it is completely independent from the Contact algorithm. 3. It is one dimension simpler than the a priori algorithm: the time variable is absent.	1. Problems in the *fixing* time step, where intersections need to be corrected. If the discrete step is not related to object's relative speed, the collision could go undetected, resulting in an object which passes through another (if fast enough).

For the requirements imposed and the pros/cons balance (Tab. 3.2), it is straightforward the choice of the ***a-priori* strategy**, with the only attention to the time step regulation at impact, in order to prevent unwanted interpenetration of the colliding bodies.

3.4 Collision Detection Algorithm Build-up

The Collision Detection strategy implemented is strictly related to the Net evolution dynamics, and therefore optimized in that way.

The highlighted phases are:

- *Free-dynamics* phase
- *Impact and wrapping* phase

3.4.1 Free-Dynamics Phase

This phase is the one that characterizes the full deployment of the Net, between the Bullets shot and the impact with the Target (Fig. 3.9).

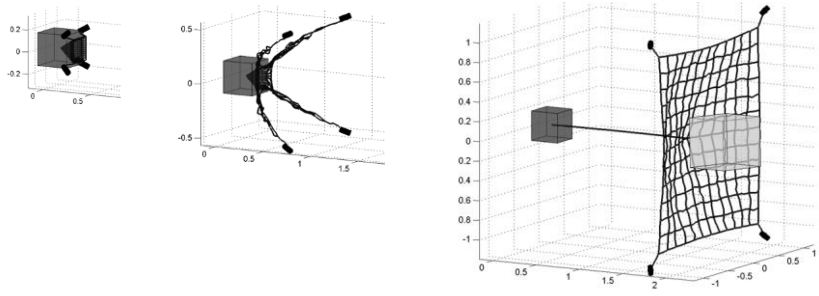


Figure 3.9: Net *free-dynamics* phase.

In order to obtain the maximum delay before engaging contact checks (computational bottle-neck), the Collision Detection has to be simplified as much as possible, simplifying the bounding volumes number and geometries (Eq. 3.2). Therefore, the winning strategy consists of only two Net-Oriented Bounding Boxes (Fig. 3.10), one enclosing the whole Net, and the other the entire Target. *Net-Oriented* refers to the boxes alignment to the Net shooting direction, corresponding to the x-axis of the Net reference frame.

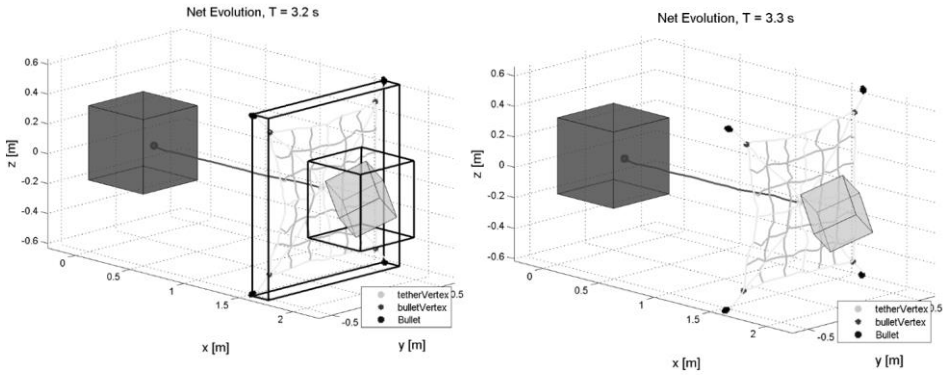


Figure 3.10: time instant when Hierarchies are actually engaged, exploiting Net-OBB in the *free-dynamics* phase.

Therefore, we call *LEVEL#0* the lowest hierarchy level for the Net and the Target, so defined:

- Net: it includes all its knots, nodes and bullets. Tether excluded;
- Target: it includes all its mesh nodes.

Being directly related to the Chaser attitude, Net-OBB is also robust to whichever initial conditions and scenario.

3.4.2 Net Bounding Volumes and Hierarchies

The second phase is characterized by the Net impacting and wrapping over the Target (Fig. 3.11).

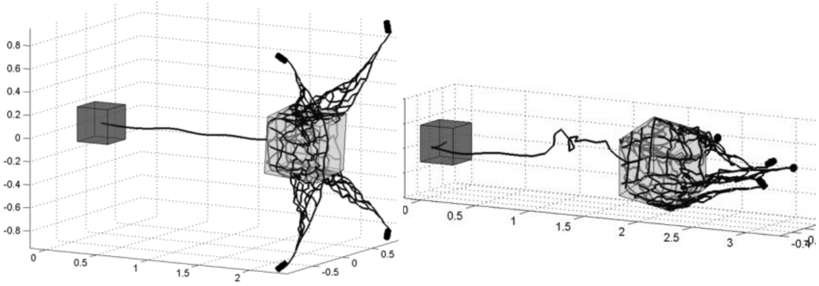


Figure 3.11: Net *impact and wrapping* phase.

For the proximity to it, a higher precision in collision detection is required: that is when hierarchies are engaged. For the Net we define the finest level as the single discretization node, so grouping up we are able to build a parenthood. For the Target the nodes considered are the ones defining the mesh. Spherical Minimum Bounding Volumes (SMBV) are exploited for the hierarchical architecture. An example of the Net hierarchical structure is presented in Fig. 3.12, where a Planar A geometry is discretized with four LEVELS. Tether is treated similarly, as a single thread.

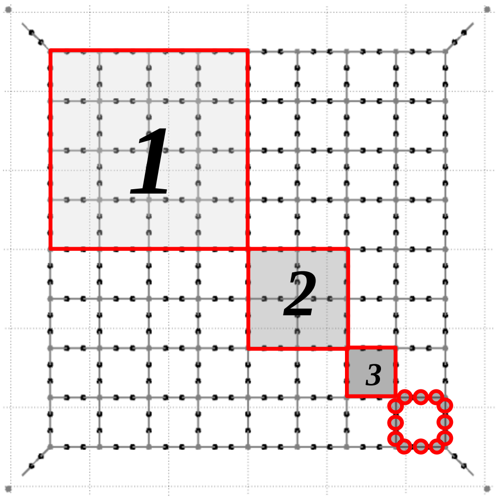


Figure 3.12: Net hierarchies. Four LEVELS in a Planar A geometry Net.

3.4.3 Net Threads Interpenetration-Avoidance

Interpenetration characterizes the portion of Net threads between two discretizing nodes, when crossing sharp edges (Fig. 3.13). Therefore, to face this problem we have to make these thread elements *physical*.

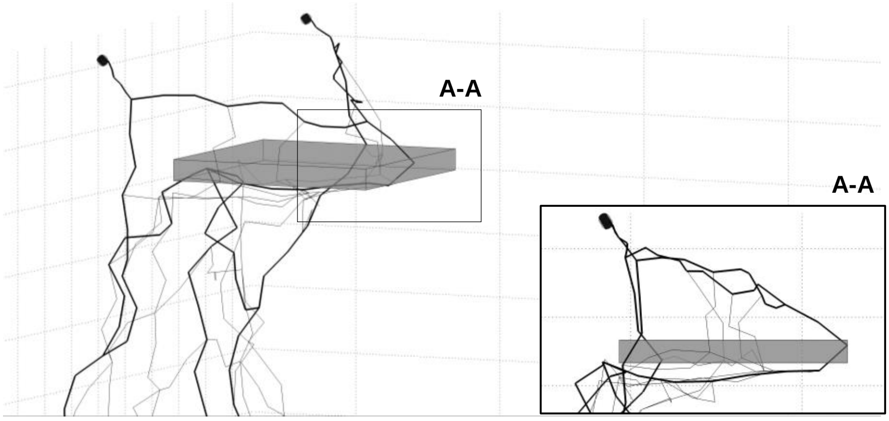


Figure 3.13: Interpenetration problem of a thread element with a sharp edge.

We have three alternatives to achieve this goal:

- Increase the number of discretizing nodes;
- Discretize each element with *dummy nodes*, used only for collision and not as DOF;
- Define a Representing Spherical Volume (RSV) for the thread element.

Representing Spherical Volumes totally differ from Spherical Bounding Volumes in their scope. While SBV are exploited to identify the specific subset of nodes to further consider in the Collision Detection, RSV physically represent these nodes. So, RSV give a 3D dimension to the otherwise dimensionless nodes. Trade-off is schematically outlined in Tab. 3.4, where a score 1 to 3 (*low* to *high*) is assigned to each solution.

Table 3.3: interpenetration avoidance solutions trade-off.

	Precision	**Efficiency**	**Robustness**
Increase DOF	3	1	1
Introduce *dummy nodes*	3	2	1
Dummy nodes with RSV	1	2	3
Define real nodes RSV	**1**	**3**	**3**

The first choice is the simplest, since it does not need any further algorithms, but it is completely inefficient for the computational speed. Moreover it is still not robust for sharp edges, which ideally involve a continuous model of the thread (implying an infinite number of discretizing nodes). The second choice implies the introduction of *dummy nodes* used just for the collision detection of the thread element, so no further DOFs are added. The drawback lies in the necessity to build further hierarchies, otherwise efficiency decreases again. The third alternative becomes the best, since:

- It introduces no further real nodes, nor *dummy* ones;
- No additional hierarchies are required;
- Representing Spherical Volumes let to keep the highest robustness.

The first two points are translated into a computational sped-up, while the last involves necessarily a precision loss. This is the price to pay for efficiency. The exact error to take into account, is a collision detection offset equal to the RSV radius; the higher the discretizing nodes, the lower the radius and the higher the precision. The radius is computed at each time instant as the minimum half-length among all the adjacent elements (Fig. 3.14). This solution is much better in terms of precision, rather than fixing the radius at the beginning as the maximum half-length. Moreover in this way we are able to ensure that no overlap between adjacent volumes will ever occur. If this was not ensured, collision would be detected and unwanted compression reactions would rise.

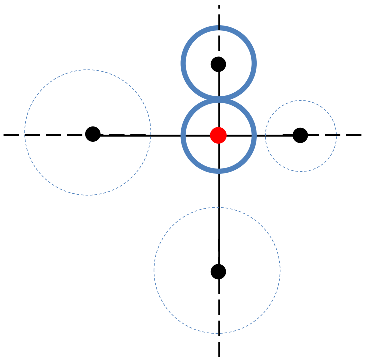


Figure 3.14: focus on a Net mesh. RSV defined by the minimum half-length of the adjacent element threads (bold). Emphasized case.

Integrated with the Collision Detection algorithm, we report the effectiveness of the Interpenetration Avoidance strategy adopted through a benchmark test. No interpenetration occurs at all, with the only price to pay equalto to a contact offset.

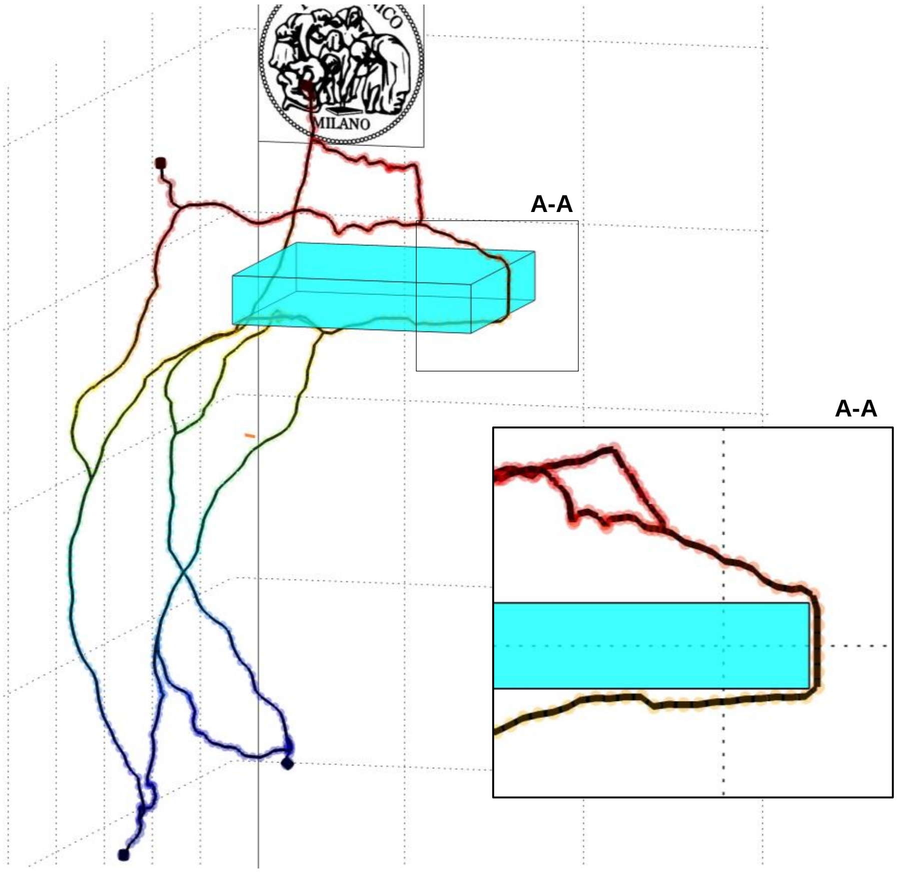


Figure 3.15: Benchmark test for Net threads Interpenetration Avoidance algorithm. Impact of a single thread on a sharp edge.

3.4.4 Bullets Bounding Volumes and Hierarchies

For what concerns the Bullets, they need to be treated as rigid bodies also in the Collision Detection and Contact algorithms and not only as their CoM, otherwise we would miss the whole impact dynamics. Specifically we would have:

- Bullets interpenetration with the Target (unreal);
- Absence of contact torques at the impact (unphysical).

Bullets have been therefore treated separately from the rest of the Net nodes and knots. From the specifics of the project, they are designed cylindrical shaped.
The strategy we adopt consists in the discretization and approximation of those bodies with compositions of Representing Spherical Volumes (RSV), as in the case of Net threads.

The criteria we follow, is related to the optimum shape ratio definition, involving the minimization of the ratio between the representing spherical volume and the enclosed solid geometry. This involve the definition of a cost function to minimize, or alternatively the corresponding error. This volumetric error is translated to the maximum linear difference between the radius of the representing volume and the solid surface. The goal in the Bullets hierarchy build-up is to find the optimum number of discretizing spheres representing the cylindrical body.

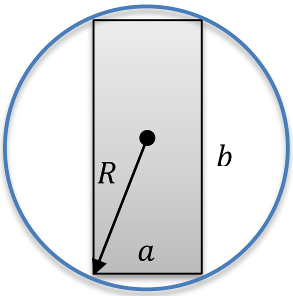


Figure 3.16: 2D view of a Bullet enclosed in a spherical volume.

Calling a and b the cylinder dimensions (holding the condition: $b > a$), we define:

$$R = \sqrt{{}^{a^2}/_4 + {}^{b^2}/_4} \tag{3.3}$$

$$\varepsilon = R - {}^{a}/_2 \tag{3.4}$$

Where R is the radius of the representing sphere and ε is the maximum error. Defining the bullet shape ratio α and its integer approximation *f*:

$$\alpha \triangleq {}^{b}/_a \tag{3.5}$$

$$f \triangleq floor(\alpha) \tag{3.6}$$

We can reformulate the error as a function of only one Bullet dimension:

$$\varepsilon = \sqrt{{}^{a^2}/_4 + {}^{(\alpha\cdot a)^2}/_4} - {}^{a}/_2 = {}^{a}/_2\left(\sqrt{1+\alpha^2} - 1\right) \tag{3.7}$$

So, we obtain the minimum error due to this sphere-representation, that is for unitary α (sphere enclosing a cube):

$$\varepsilon_{\%} = {}^{\varepsilon}/_a \cdot 100 = \left[{}^{1}/_2\left(\sqrt{1+\alpha^2} - 1\right)\cdot 100\right]_{\propto=1} = 20\% \tag{3.8}$$

We can now find the critical shape ratio $\bar{\alpha}$ for the discretizing volumes:

$$\bar{\alpha} = \alpha_i :/ \{\forall\, \alpha_i > \bar{\alpha} \;\rightarrow\; \varepsilon > \varepsilon_{min}\} \tag{3.9}$$

Therefore it follows that if $\alpha_i > \bar{\alpha}$, we need to increment the number of discretizing spheres. Another way to see $\bar{\alpha}$, is:

$$\bar{\alpha} = \alpha_i :/ \{\varepsilon(f) = \varepsilon(f+1)\} \tag{3.10}$$

Where:

$$\varepsilon(f) = \sqrt{\left(\frac{a}{2}\right)^2 + {\alpha_i}^2\left(\frac{a}{2f}\right)^2} - {}^{a}/_{2} \tag{3.11}$$

The solution is found solving numerically the following non-linear equation coming from Eq. 3.10:

$$\sqrt{\frac{1}{4} + \left(\frac{\bar{\alpha}}{f}\right)^2} - {}^{1}/_{2} = \sqrt{\frac{1}{4} + \left(\frac{\bar{\alpha}}{2(f+1)}\right)^2} - \frac{\bar{\alpha}}{2(f+1)} \tag{3.12}$$

The solution varies with the overall Bullet shape ratio, but keeps around the value of $\bar{\alpha} - f \approx 0.2$. As can be seen in Fig. 3.17 in the case of a Bullet shape ratio between 1 and 2, the dotted red line stresses the convenience of using one single element discretization up to a shape ratio of 1.2 and a two element discretization after.

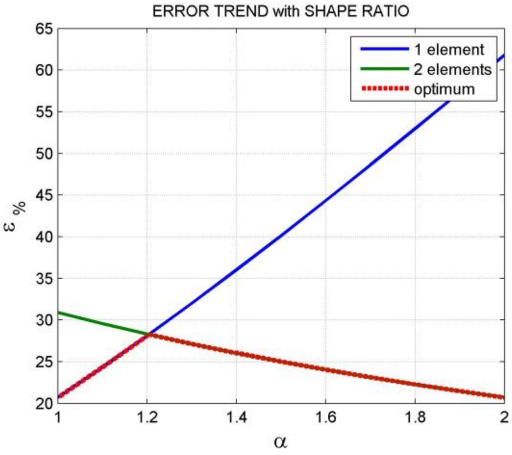


Figure 3.17: analysis of the error trend with the Bullet shape ratio in the case of $1 < \alpha < 2$.

Therefore, once fixed the optimum f according to the Bullet shape ratio given by the external user, the algorithm automatically discretizes the geometry according to the aforementioned criteria (Fig. 3.18).

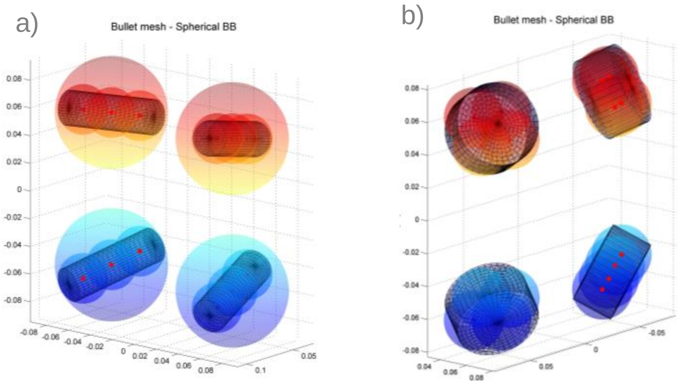


Figure 3.18: example of flexibility of the approach for two different Bullet shapes. Hierarchies plotted in a) while in b) the only representing spheres.

Here we report the benchmark tests for the algorithm adopted in the Bullet Collision Detection. The first one compares the precision of the strategy with respect to the simple reduction of the Bullets to their CoM. As can be seen from Fig. 3.19b, no interpenetration occurs and the Contact is able to give both forces and reacting torques.

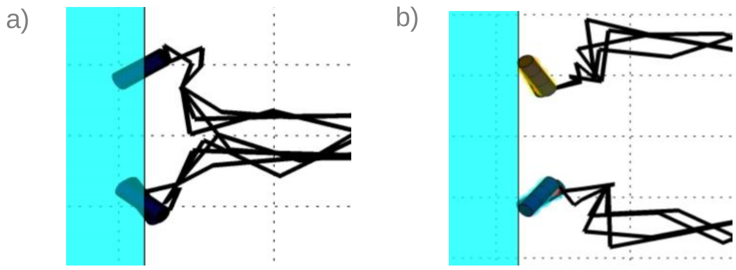


Figure 3.19: Long cylindrical Bullets. Comparison between a) the simple Bullet CoM Collision Detection and b) the engagement of the RSVs.

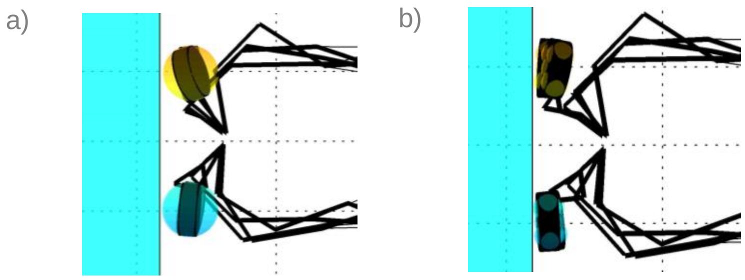


Figure 3.20: Flat cylinders. Comparison between a) one single sphere approximation for each Bullet and b) intelligent RSV discretization.

Fig. 3.20 shows the advantages obtained through the discrete spheres Bullet representation, instead of a single sphere. The precision in this case consists in minimizing the offset in the Collision Detection.

3.4.5 Target Characterization and Impact Detection

For the required consistency with the Net hierarchical Bounding Volumes strategy adopted, the Target is processed in the same way of the Net, that is exploiting hierarchical SMBV. A first analysis deals with discretizing the mesh of the Target faces and then grouping them in order to grow hierarchies up (Fig. 3.21). The main advantage of this choice is that the more are the BV representing the Target faces, the tighter they are on the Target, so minimizing the total enclosing volume. So, the advantage of this strategy is to delay the Contact algorithm engagement, refining at most the impact area.

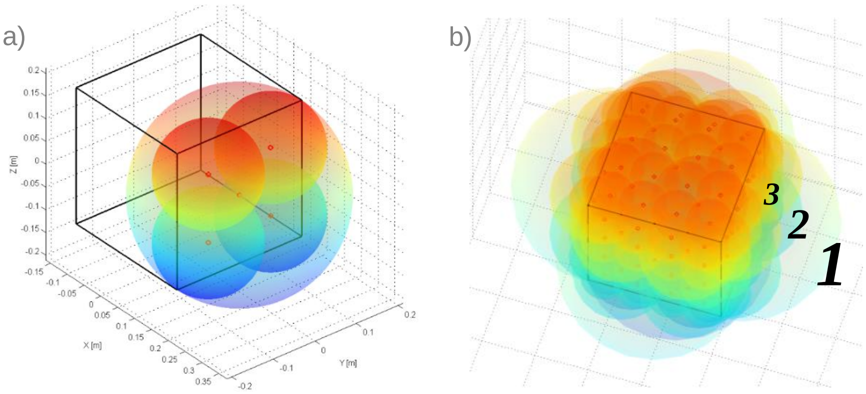


Figure 3.21: a) Target face meshes discretization and BV build-up. b) Target hierarchies displayed.

However, from simulations this strategy turns out to be very time expensive, because the higher is the number of discretizing BV, the more are the coherence checks that need to be done. So, instead of doing a single check between a Net node and a face of the aforementioned plate, hundreds of checks are performed, that recall all the BV of the plate face. The computational work becomes really important in this case, and it has been empirically proven.

Therefore an improved strategy is designed, that involves grouping all the near meshes with the same exiting unit vector (orientation). This is done automatically pre-processing whichever Target mesh provided by the external user. This strategy proves to be much more computationally efficient and time saving. Each face is characterized by its own SMBV, to which it is associated an exiting normal unit vector (red arrows), the SMBV centers (red points) and edge normal unit vectors (green arrows).

→ The face normal unit vectors ($\boldsymbol{n}$) let to interpret the spheres as effectively hemispheres (as represented), in order to avoid the impact detection occurs from the wrong side of the face.

→ The face edge normal unit vectors ($\boldsymbol{e}$) let instead to properly identify the impact to occur exactly on the face, reducing to *zero* the error on the impact point.

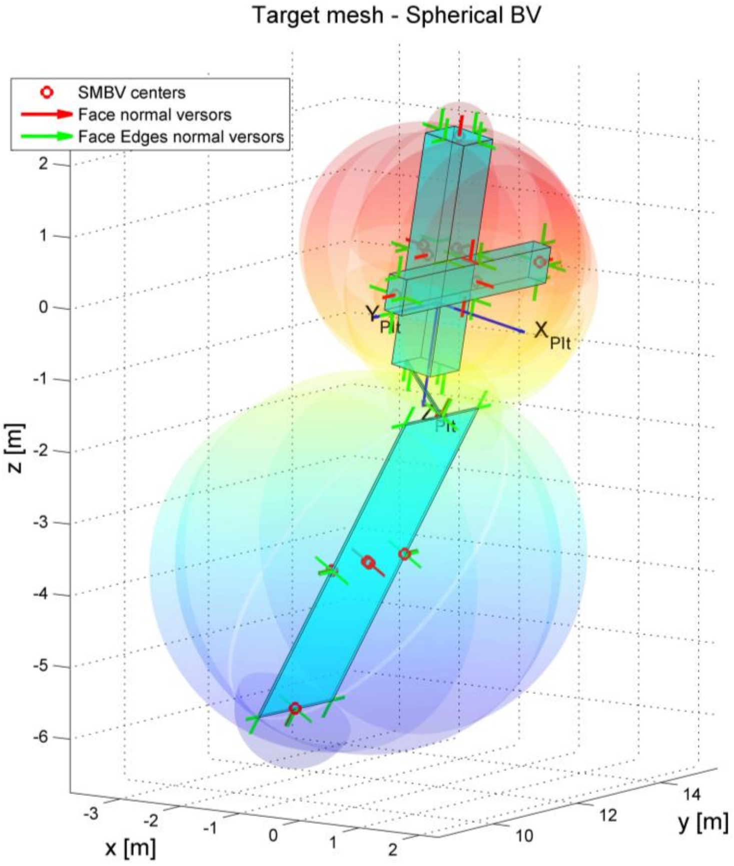


Figure 3.22: Target pre-processed for the Collision Detection.

Finally the Collision Detection algorithm, once reached the highest hierarchy level, performs the following structured checks to prove the impact, given a generic mesh shape (Fig. 3.23). The impact of a point P with a surface can be detected only comparing two different temporal instants, involving the validation of both *collision eligibility* and *penetration effectiveness*.

Collision eligibility is proven at time t if and only if the point P is inside the hemisphere:

$$\begin{cases} |\boldsymbol{d}| < R \\ (\boldsymbol{d} \cdot \boldsymbol{n}\,) \geq 0 \end{cases} \tag{3.13}$$

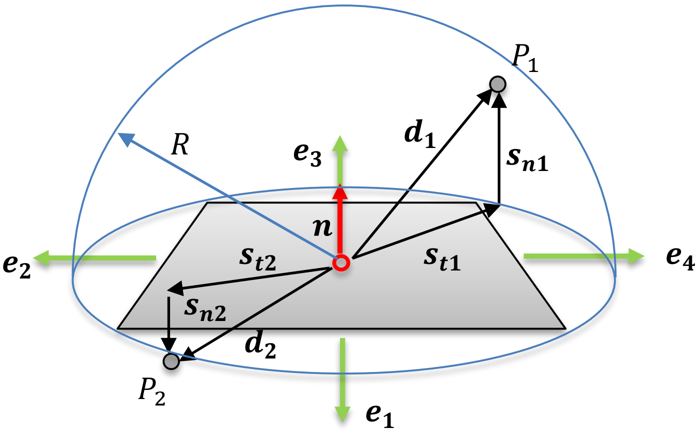


Figure 3.23: Impact Detection conditions. 1) registered, but not colliding, 2) registered and impacted.

Penetration effectiveness is then proven at time *t+1* if and only if the point is *eligible* from the previous time step and:

$$\begin{cases} \boldsymbol{d} \cdot \boldsymbol{n} < 0 \\ \boldsymbol{s_t} \cdot \boldsymbol{e_i} < 0, \quad (i = 1{:}n_{edges}) \end{cases} \tag{3.14}$$

3.4.6 Auto-Collision Problem

Generally speaking, the collision scenarios involve:

- Net vs. Target (main collision problem);
- Net vs. Net (auto-collision problem).

Auto-Collision could be neglected to speed-up the simulation in the case rough guess results are wanted, since not occurring in the case of a good deployment and affecting only the wrapping phase. It needs to be included if maximum precision is desired in the deployment and mostly in the wrapping phase.
The Auto-Collision issue only affects the Net system, so specifically threads and bullets. We have already defined all the required tools, since:

- → for Net and Tether we exploit the Representing Spherical Volumes introduced before to avoid threads interpenetration (Subsection 3.4.3);
- → for the Bullets, the same RSV discussed before to deal their Special Collision Detection approach (Subsection 3.4.4).

So, we keep the same Bounding Volumes and hierarchies as before, the only thing required is to extend the coherence checks for each node to all the nodes, including the Net, Tether and Bullets. Hierarchies are engaged in the same way, as well as coherence checks. In Fig 3.24 we report a benchmark test comparison between including or not the Auto-Collision checks between four Bullets colliding. We also report how the hierarchies are dynamically engaged and higher levels are considered in the coherence checks for proximity detection (first and second frames). All the reacting forces and torques arising, are directly applied to the discretizing nodes, centers of their RSV. These nodes are already DOF, so no further forces and torques equivalencies need to be done to derive the resulting action at the eventual bounding nodes (as it would be needed with *dummy* nodes).

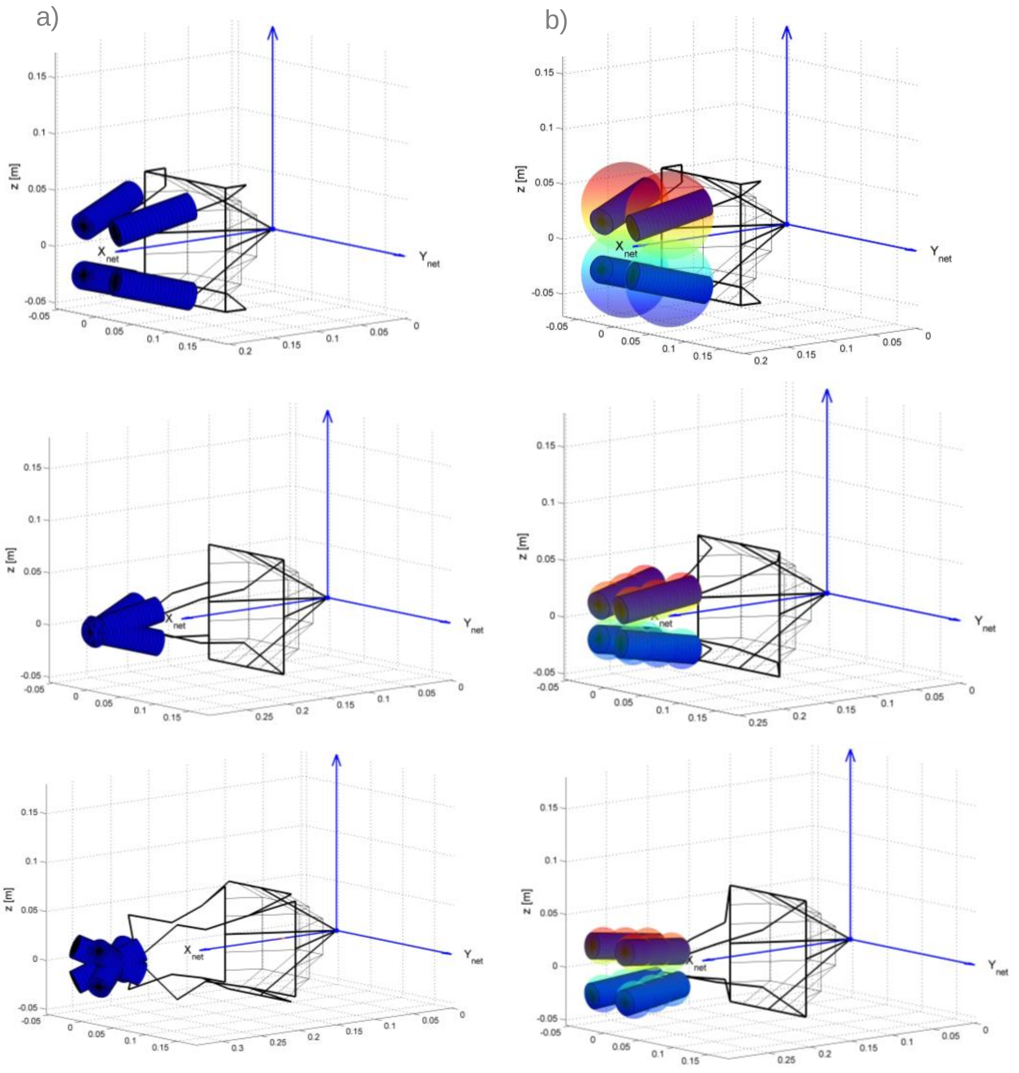


Figure 3.24: benchmark test for Auto-Collision problem. a) Bullets simplified to their CoM, b) Bullets processed with hierarchies and RSV.

3.5 Optimizations and Computational Speed-up

Bottlenecks haven identified with all that processes involving any check with all the system DOFs. So, one is the **Viscoelastic-Block**, as it processes all the relative distances between the Net nodes, the other is the **Collision-Detection Block** for its necessity to check their impact. In the second case, the computational work is further more for two reasons:

- To the DOF nodes, they are added the surface meshes nodes of all the rigid bodies;
- Auto-Collision is responsible for the same workload of the direct Collision, so if n is the total number of nodes, n^2 becomes the total number of checks including Auto-Collision, so times are square compared to the VEF-Block.

This last issue has already been faced with the introduction of hierarchies, but hierarchies are determinant to speed-up only the approaching phase of an object towards another, because they act as active filters for nodes selection in the proximity checks. Therefore, as soon as these objects are totally close-fitting (like the Net wrapped on the Target), a point-to-point cross-check is unavoidable since all the highest levels are already engaged, so the speed-up requirement comes out again. We have two ways to increase the computational speed of the code:

1. **Optimize** the numerical model;
2. **Parallelize** the computing process.

Parallelization by definition involves a *for-loop* task to be performed in parallel by a set of workers or clusters. This implies the following two important limitations:

- The code is not sequentially executed, so no information can pass between two different iterations;
- The parallelization is applicable only to the outer *for-loop*, so nested cycles are computed by the same cluster of the dominant iteration.

These two aspects strongly limit the optimization design, so much of the effort has been done looking for parallelizable solutions.

First simplifications is in the *free-dynamics* phase, the aim is to reduce to the minimum the number of nodes to consider in the Collision Detection. So, for the sequentiality of the Net deployment and impact, Tether is excluded from the pot of nodes considered in the Net-OBB build-up, and this holds whichever the deployment. Robustness is so preserved.

Secondly, a further improvement is in the choice of the strategic nodes to exploit for the Net-OBB build-up, both for the Net and the Target.

As can be seen from the free dynamics simulations, the strategic points include necessarily the bullets and the tether vertex. However, if considering only these, it could happen that in the case the Net collides with the target after the predicted

time (maximum deployment), a particular evolution of its shape would let some central nodes to anticipate both the bullets and the vertex (Fig. 3.25).
This would lead to a delayed activation of the next level of the collision detection, with computational bugs due to an excessive interpenetration of the nodes and causing unrealistic high contact reactions.

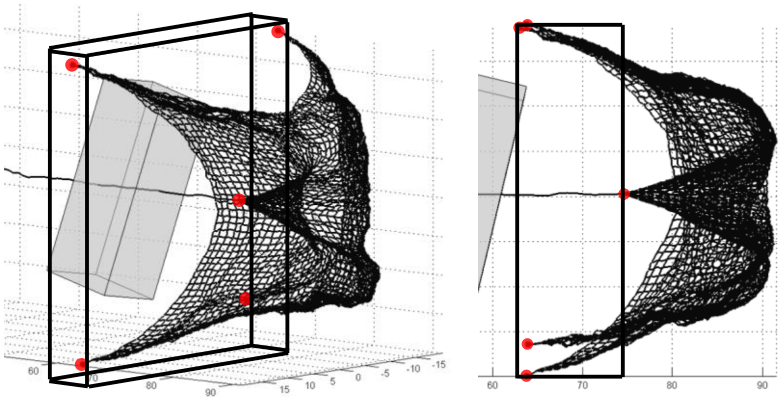


Figure 3.25: critical Net deployment and the associated WRONG Net-OBB that would be created in the case only Bullets and Tether vertex were considered (red dots).

Therefore the optimal choice is to include a *cross-set* of knots (Fig. 3.26) that, for the symmetrical behavior of the net deployment, represents best the Net shape evolution in time.

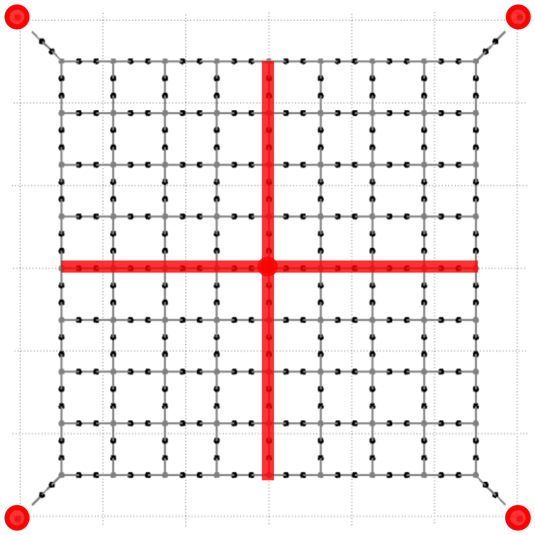


Figure 3.26: strategic set of knots finally considered for the Net-OBB build-up (highlighted in red).

The speed-up gained is of one order of magnitude (**10x**) in the case of thousands of DOFs simulations.

Concerning **Bullets**, they should be considered with all their RSV and not only as their CoM, otherwise in the eventuality one Bullet impacts on the Target before the Net nodes, the unwanted interpenetration bug would occur (Fig. 3.27).

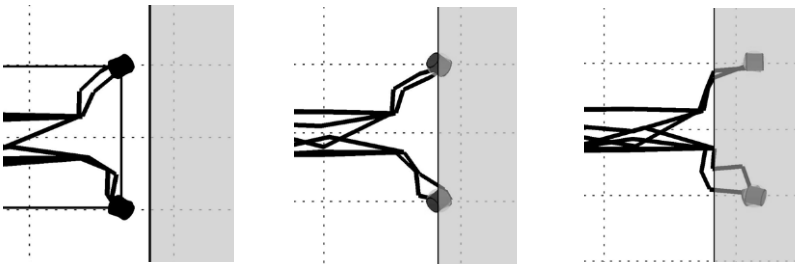


Figure 3.27: Bullets interpenetration bug due to the missing inclusion of the Bullets RSV in the computation of the Net-OBB.

Since not only the impact, but also any coherence check is detected *a*-posteriori, there is still the possibility to have excessive interpenetrations according to the integration step size and the characteristic velocity of the dynamics. Therefore, to prevent the delayed engagement of the hierarchies, a margin is applied on the *x-dimension* of the Net-OBB:

$$ds = v_{CoMx} \cdot dt \tag{3.15}$$

Where:

- v_{CoMx} = predicted average x-component of the Net CoM velocity.
- dt = maximum integration time step.

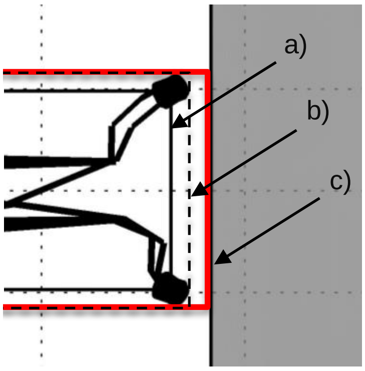


Figure 3.28: Comparison of the three Net-OBB configurations: a) considering the Bullets CoM, b) including the Bullets RSV and c) setting a safety margin in the capture direction.

For the **Target**, the strategy consists in defining the minimum and maximum coordinates in the Target body frame (respectively *P_1* and *P_2* in Fig. 3.29). These two points will be the only ones considered in the rotation in Net frame during the simulation. The advantage consist in reducing whichever number of nodes of the mesh to just two points, so nullifying the sensitivity of the process to the Target geometry complexity and mesh discretization.

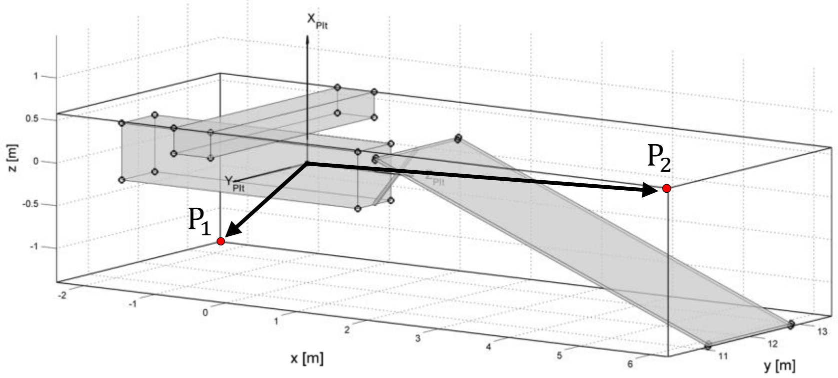


Figure 3.29: Target simplification through minimum (P_1) and maximum (P_2) coordinates in Target Body frame (blue frame centered in Target CoM).

The last parallelizable improvement is for the construction of the Spherical Volumes adopted in the hierarchical SMBV build-up. This is crucial for the computing time when dealing with huge numbers of DOF. This task cannot be solved in a closed way, so there is not an analytical solution.

A ***deterministic incremental algorithm*** [21][31] is chosen (instead of the most diffuse recursive procedures) since it costs computationally less and grants a precision with an error around 5% from the exact minimum bounding sphere. The procedure is called *Fast Approximate Bounding Ball*, and consists in starting from two initial points and include randomly all the other nodes. If at step *k* the node *P_k* is already inside the sphere (Fig 3.30a), the code moves on to step *k+1*. If the point *P_{k+1}* is outside the sphere defined at step *k* (Fig 3.30b), a new Ball *B_{k+1}* and center *C_{k+1}* are defined in order to include it.

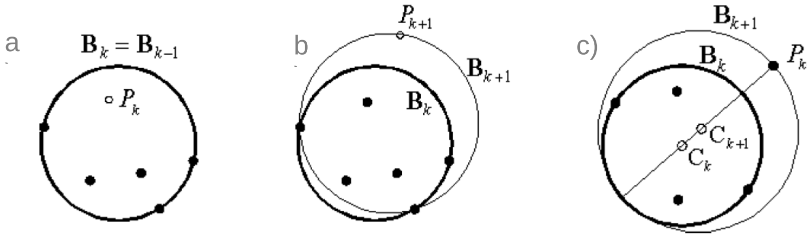


Figure 3.30: Fast Approximate Bounding Ball procedure.

The characteristic of B_{k+1} is to be tangent to B_k, passing through P_{k+1} (Fig 3.30c). Most of the efficiency depends on the choice of the two starting points, and we have two ways:

a) Random starting points;
b) *Min* and *MAX* of the set of points.

Choice (b) turns out to be more efficient, although the extra loop at the beginning to find the *min* and *MAX* of the set of points. The reason is that it grants to define a lower volume with respect to choice (a): the radius has been demonstrated to be 7% lower on average. The importance of the minimum volume definition lies in the further computational time gain due to delay in hierarchies engagement and useless collision checks.

Further optimizations are applied together with all the aforementioned ones, but differently from them, these cannot be exploited in a *parallel-computing* architecture. Therefore, they need to be considered as an alternative optimization strategy to parallelization for the computational speed-up.
First is the exploitation of *time-coherence*, that is the assumption that once a node has been detected to be in a certain position at time *t*, it will be reasonably near in space at time *t+1*. For this reason it is introduced a matrix called *Collision Matrix* (***CM***), necessary to record both:

- The *collision eligibility* of each node with respect to a surface;
- The identification number of the surface on which the node has impacted or it is meant to impact.

Therefore ***CM*** is a matrix with dimensions$[(n+b)\times(n+b+N)]$, where n is the overall number of Net nodes, b is the number of Bullets RSV centers and N are the Target faces. It is a matrix of *ones* and *zeros* to respectively record the collision eligibility or not. Each cell *(i,j)* corresponds to node *i* impacting on node or face *j*, and since a node cannot impact on itself, the diagonal is *zero*.
Due to the low density of ***CM*** (very big matrix of most zeros), it is better organized in three matrices:

- ***CM1***= $[(n+b)\times N]$ for the main collision problem (Net and Bullets nodes versus Target faces);
- ***CM2***= $[n\times n]$ for the Net Auto-Collision problem;
- ***CM3***= $[b\times b]$ for the Bullets Auto-Collision problem.

This subdivision also gives the possibility to engage each of the three functions (main Collision, Net Auto-Collision, Bullets Auto-Collision) according to the precision of the outputs and computational speed desired. The efficiency improvement due to the ***CM*** introduction is that once a node has been detected to impact over a specific surface at time *t*, at the next time step *t+1* it is no more verified its coherence with all the surfaces, but with the only one recorded at time *t*. If it is satisfied, then hierarchical and contact checks are pursued, otherwise it is verified the coherence with the adjacent faces (Fig. 3.31). This is a further improvement in the speed-up, since it is much more efficient than simply resetting the memory of the ***CM*** relative to the node under study.

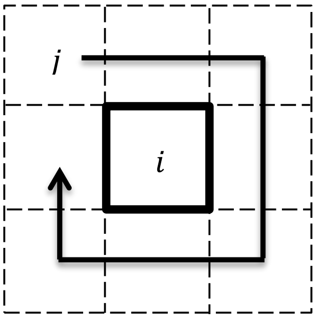


Figure 3.31: in the case of missing impact of the node *k* on face *i*, they are checked all its adjacent faces *j* before resetting ***CM***(*k*,*i*).

Therefore, in order to improve the algorithm efficiency during the integration, it has been transferred more workload to the pre-processing, since it needs to be run only once at the beginning of the simulation. Moreover, differently from the Auto-Collision problem where a *double for-loop* is required over all the nodes, in the case of Target *composed faces*, we can do a further efficiency improvement.

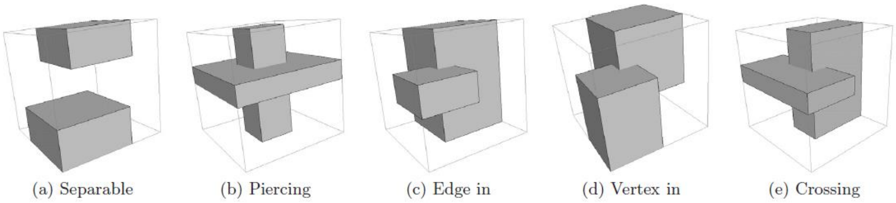


Figure 3.32: example of the so-called *composed faces*.

We can automatically identify for each face also the number c of convex or superimposed faces that need to be controlled together during the Collision Detection, so further less checks are performed ($c \ll N$). Composed faces are automatically identified during the Target pre-processing, if the both the checks in Eq. 3.16 are satisfied:

$$\begin{cases} (\boldsymbol{d_{ij}} \cdot \boldsymbol{n_i}) > 0 \\ |\boldsymbol{d_{ij}}| < (r_i + r_j) \end{cases} \tag{3.16}$$

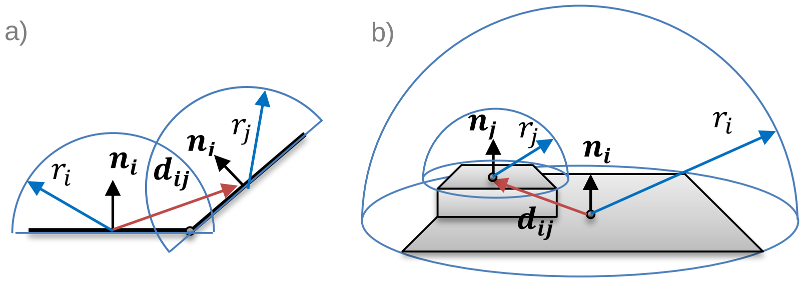


Figure 3.33: examples of: a) concave surfaces, b) composed surfaces.

Where:

- $\boldsymbol{d}_{ij}$= distance between the centers of SMBV of faces *i* and *j*
- $\boldsymbol{n}_i$= unit vector normal and exiting from face *i*
- r_i, r_j= radii of SMBV of faces *i* and *j*

For what concerns Bullets Auto-Collision, as previously mentioned, a ***CM3*** matrix is built-up according to strategic assumptions.

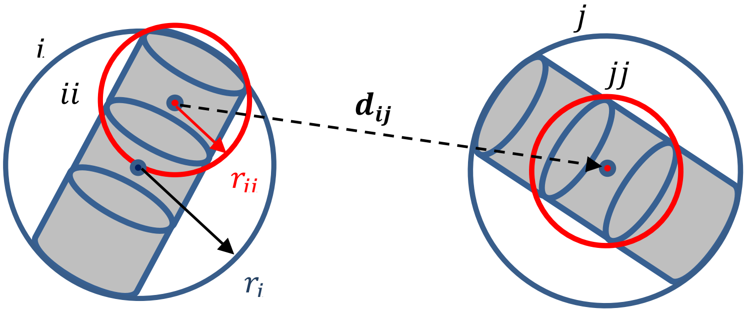
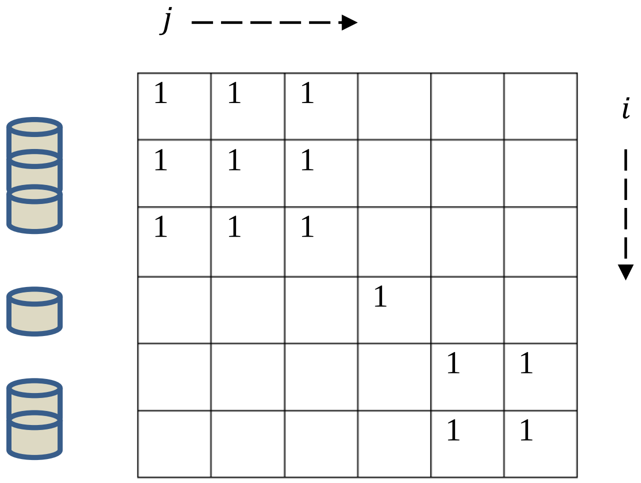


Figure 3.34: Bullets Auto-Collision Detection approach.

Calling *i* and *j* the loops on the *LEVEL*#1 of the Bullets, *ii* and *jj* those on *LEVEL*#2, we state that:

1. It is reasonable that each single RSV *ii* can impact at least with only one RSV *jj* of the Bullet *j* (so, multiple impacts with other Bullets are still admitted). Therefore, as soon as an impact is detected, the loop is interrupted.
2. For the Newton’s third law of motion, once a collision is detected and the contact force is computed between two RSVs *ii* and *jj*, the same force is applied as reaction on the second Bullet, avoiding the same pair check *ii*-*jj* in future:

$$\boldsymbol{F}_{jj} = -\boldsymbol{F}_{ii} \tag{3.17}$$



1	1	1			
1	1	1			
1	1	1			
			1		
				1	1
				1	1

Figure 3.35: architecture of the ***CM3*** matrix. *i* and *j* are the indexes of the RSV of the Bullets. Blank cells are the *zeros*.

The ***CM3*** is therefore defined as a *zeros* square matrix [$b \times b$] (where b is the overall number of RSVs of *LEVEL#2*). The impact detection is recorded as a *state-change* from *0* to *1*. Therefore, in order to avoid the Auto-Collision checks between the RSVs of the same Bullet, these are directly set to *1* at the beginning (Fig. 3.35).

Chapter 4

CONTACT

4.1 Contact Dynamics Modelling

The physical phenomena occurring during the impact of two or more bodies are topics of continuous research.

The objective of our study is to develop a contact dynamics modelling integrated in the simulator to be validated by experimental tests. The goal is to design a general-purpose contact model to be used instead of the non-elastic contact models implemented in previous versions of the simulator, ensuring the highest precision and fidelity to the physical phenomena, identifying a minimum number of tunable parameters.

4.1.1 Contact Model Requirements

The aforementioned goals are translated in the following requirements for the mathematical and numerical models to design:

- **Realistic** force: the model should be able to reproduce physically observed behaviors. The fidelity level should allow accurate simulation of the dynamics of the Net impacting and wrapping over the Target.
- **Easily identifiable model parameters**: related to physical properties and that can be easily obtained through experimental tests.
- **Compatibility** with explicit fixed-step and variable-step ODE solvers.
- **Capability** to handle whichever objects with complex shapes.
- **Interfaceability** with possible third-party collision detection algorithms.
- **Robust** implementation: the user is not required to hand-tune the parameters related to the implementation.
- **Flexible** implementation framework: allowing different levels of complexity in the contact model to be easily switched.

4.1.2 Contact Literature Survey and Trade-off

At the very low level, we can distinguish between:

- *non-elastic* contact;
- *elastic* contact.

The first one assumes a non-elastic impact occurring between two colliding bodies. Non-elastic means that all the impact energy is absorbed in the impact and so dissipated. The effect is that the colliding objects get stuck after the impact. It is hardly exploited, except for very rough simulations. Elastic contact is of course the most reasonable to be used in contact modeling.

We can classify two types of elastic contact models:

- *Rigid-body* contact models;
- *Regularized* contact models.

The *rigid-body* approach, also known as perfect-elastic contact model, has the strong assumptions of infinite contact stiffness, translating into an impulsive contact dynamics. It is only concerned with the description of the global impact response, i.e. the motion after the impact depending on the motion at the beginning. Impact hypotheses are applied such as Newton's kinematic [36] or Poisson's impulse [37] or Stronge's energy based impact hypothesis [38] to derive a complete set of equations. Hereby, these are obtained balancing the system's momenta before and after the impact, so without explicitly considering contact compliances. This approach assumes the collision to be an instantaneous and discontinuous phenomenon, that means a discrete event. Friction is accounted for through the introduction of a friction impulse. For lasting contact (below a threshold velocity), the method leads to an explicit closed-form solution for the normal contact force. So, the *rigid-body* approach is better than the *non-elastic* model, but still does not represent entirely the contact physics, neglecting the small displacements during collision and the finite impact time. Consequently, rigid body models, assuming the stiffness to be infinite and the impact duration to be instantaneous, provide an inadequate reference for validating the performance of the contact.

The second class of models are the *regularized* contact models. The regularization consists in the reformulation of the problem, changing the nature of the impact from a discontinuous process into a continuous one. The contact forces are described as a function of the contact deformation by smothering the discontinuity of the impact and friction forces in the constraints [39]. This approach is also referred as *penalty method*, since the model returns a measure of the constraint violation, the larger the violation, the higher the penalty. In contrast to the contact models based on the rigid body assumption, compliant models describe the rate-dependent normal and tangential compliance relations over time. These models can be easily integrated within the existing simulation based on ODE solvers. It goes without saying that this model intrinsically includes all the aforementioned contact models; the higher the stiffness-to-damping ratio, the lower the contact time. Setting a *zero-damping* penalty we approximate to the rigid body contact model (*impulsive* contact), while setting *zero-stiffness* we have only energy absorption and the approximation to the non-elastic contact model (*sticky* contact). Therefore, this formulation provides the required degree of freedom necessary to regulate and adjust the parameters according to the experimental results. The compliance properties are directly related to the colliding object geometry and the material Young's modulus, satisfying the requirements. Therefore, we can now look in more details at the literature regarding the implementation of compliant contact models in multi-body simulations. These models are essentially linear or non-linear spring models used to predict the force acting to separate the colliding bodies. This force is identified as *normal contact force*, to distinguish from the frictional forces acting in the tangential direction to the contact surface.

We identify two main approaches:

- *volumetric* contact model;
- *point* contact model.

The *volumetric* contact model is based on a compliant approach, using the information related to the volume of inter-penetration of the colliding bodies. This volume is defined as the one obtained from the intersection of the two bodies (*volume of interference*). The contact force is proportional to the volume of interference and it is applied at its centroid. The advised contact model is the *Winkler Elastic Foundation Model* [41]. This model substitutes the lumped contact force with a simple approximation of the contact pressure distribution. To find the pressure at any point for the given geometric contact profile, the solution of an integral equation of pressure would be requested. Fig. 4.1 shows the foundation model, where k_f is the elastic modulus of the foundation and h_f is the arbitrarily chosen depth of the foundation mattress.

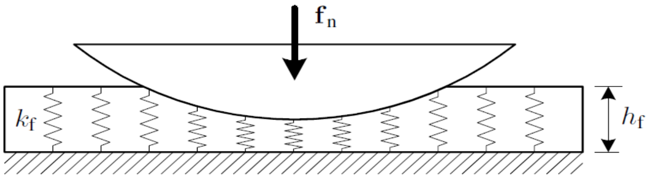


Figure 4.1: Winkler Elastic Foundation model [41].

Instead of exploiting the costly FEA (Finite Element Analysis), a good solution is the one proposed by Hippmann [43]. He uses this approach to deal with complex geometries assuming them as composed of polygons (PCM, Polygonal Contact Model) and suggests methods of how to obtain the foundation depth and

stiffness. The point-to-point contact force is obtained by numerically summing up the force contribution from each contacting polygon (Fig. 4.2). The local polygon normal magnitude is found as a function of the polygon area and the interpenetration depth, while its direction is determined arbitrarily using some heuristic rules. The model also includes a simple tangential friction model.
Hasegawa and Sato [44] use a similar approach, defining a more sophisticated friction model with state transitions between static and dynamic friction forces.

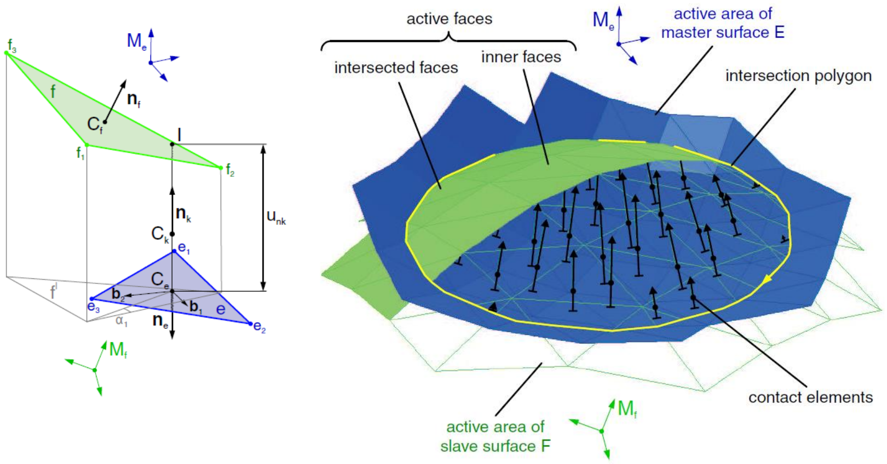


Figure 4.2: Polygonal Contact Model approach theorized by Hippmann [43]. a) Polygons interpenetration, C_k is the centroid, b) Winkler Elastic Foundation model applied to the PCM, pressure distribution displayed.

However, analyzing our problem, we can observe that with strategic approximations we can simplify the contact model from a volumetric to a point level. This can be done thanks to the way we chose to model the main impacting body. The Net in fact is not treated as a single body, but as a flexible multi-body, so intrinsically discretized with lumped masses linked through the action of viscoelastic forces. Therefore the impact can be seen occurring at the point level instead of between the whole area of the Net and the Target. If we add the further assumption of non-deformable Target, we can reasonably discard the costly and complex volumetric contact in favor of the point contact model. The *point* contact model theory is valid as long as the contact region is small compared to the dimensions of the colliding bodies, and this holds for the aforementioned Net modelling. The contact model takes the form of a lumped-parameter spring. The local deformation can be parameterized as a function of the penetration depth between the non-deformed bodies. This produces a single algebraic expression relating the inter-penetration to the normal contact force (spring model). The explaining theory is the classic *Hertz theory* [42], which is explained in the following subsection. For direct central frictionless impacts**,** *Hunt and Crossley*

[47] and *Lankarani and Nikravesh* [49] are valid contact models, which integrate the Hertz theory (spring model) with a damper (spring-damper model in order to take into account the energy dissipation in the normal direction of the impact.

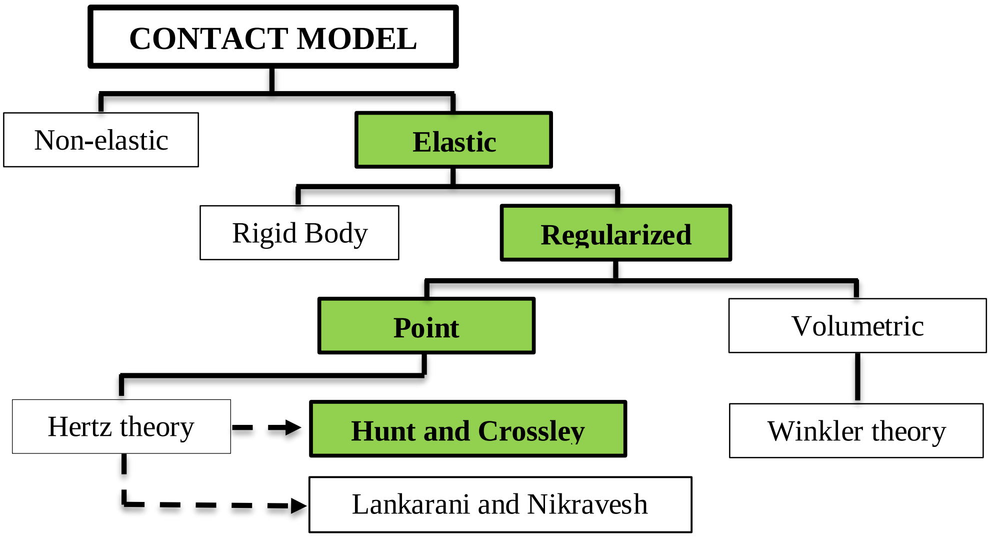


Figure 4.3: Contact problem trade-off scheme.

To implement a general point contact model, tangential forces must also be modelled, such that oblique frictional impacts can be treated as well. There are many friction models in literature based on the so-called *Bristle Theory* [30], approaching the problem at the microscopic level invoking precisely the modelling of the surfaces with elastic bristles. However, bristle friction models are valid in a context where the normal load acting between the two bodies is constant. Therefore for modelling impacting bodies, this simplification is not appropriate. We choose instead a more empirical model, which is a regularized version of the *Coulomb's law* of dry friction. The model proposed is based on the so-called *Dwell Time Dependency theory* of friction, which theorizes a time depending behavior of the stiction forces below a velocity threshold.

4.1.3 Hertz Theory

Hertz theory is based on some assumptions on the colliding bodies:

- Homogeneous;
- Isotropic;
- Linearly elastic.

Contact models derived from Hertz theory are restricted to contact between bodies with non-conformal geometries, i.e. their shapes have sufficiently dissimilar profiles, such that the contact regions are small in comparison to the size of either bodies.

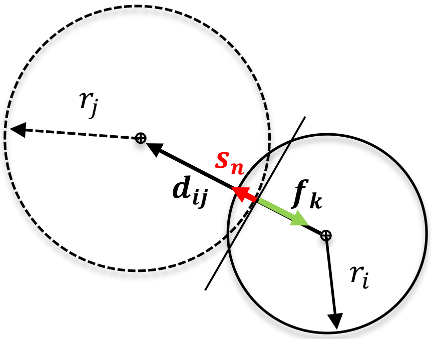


Figure 4.4: colliding spheres conventions.

The Hertz contact force law [45][46] states that for two spheres i and j in direct-central impact (Fig. 4.4), the normal contact force f_k is a function of the relative local interpenetration $\boldsymbol{s}_n$.

$$f_k = k \cdot |\boldsymbol{s}_n|^p \tag{4.1}$$

The normal interpenetration modulus and unit vector are computed as:

$$|\boldsymbol{s}_n| = |\boldsymbol{d}_{ij}| - (r_i + r_j) \tag{4.2}$$

Where:

- $\boldsymbol{d}_{ij}$ = distance between the two colliding spheres centers.
- r_i , r_j = radii of the two spheres.
- p = 3/2 in the Hertz theory, it varies for other contact models (Fig. 4.5)
- k = is the equivalent stiffness of the two impacting spheres and depends on their material stiffness and spherical radius:

$$k = \frac{4}{3\pi(h_{\mathrm{m},i} + h_{\mathrm{m},j})} \left(\frac{r_i \cdot r_j}{r_i + r_j}\right)^{0.5} \tag{4.3}$$

Where the material properties $h_{\mathrm{m},i}$ and $h_{\mathrm{m},j}$ for the spheres i and j are:

$$h_{\mathrm{m},i} = \frac{1 - \nu_i^2}{\pi E_i} \tag{4.4}$$

The variables ν_i and E_i are respectively the Poisson's ratio and the Young's modulus of the sphere i.

To model the contact force occurring between a sphere and a plane, one of the radii is set to infinity and the Eq. 4.3 simplifies to:

$$k = \frac{4}{3\pi(h_{\mathrm{m},i} + h_{\mathrm{m},j})}(r_i)^{0.5} \tag{4.5}$$

Where *i* refers to the properties of the sphere and *j* to the plane as defined above. Compliant contact models based on Hertz's Theory are typically expressed in the form of an equation relating the depth of inter-penetration of two bodies to the contact force (Fig. 4.5).

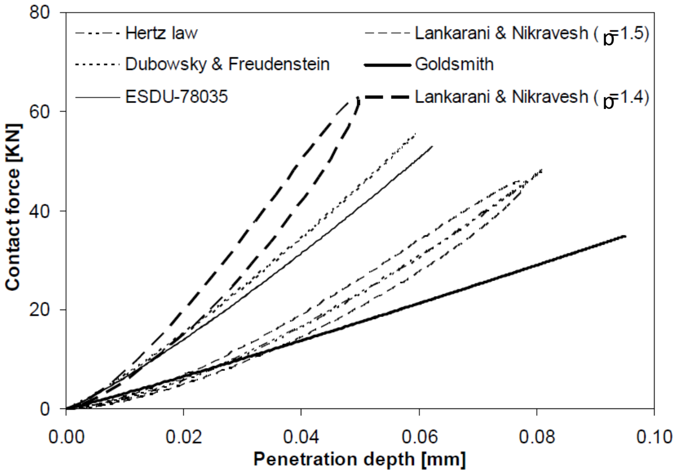


Figure 4.5: Normal Contact Force laws derived from the Hertz theory.

4.1.4 Normal Contact Force Model

The normal contact force model derived in this section belongs to a special class of non-linear spring models. Given that the Hertz non-linear spring behavior is a well understood phenomenon, the analysis presented here will concentrate on the mechanism through which the energy is dissipated. The model proposed are the ones derived by Hunt and Crossley [47] and Lankarani and Nikravesh [49]. The compliant normal-force expression, proposed by Hunt and Crossley for direct central and frictionless impact, is a non-linear spring-damper model.

$$f_n = -\lambda|x|^p\dot{x} + k|x|^p \tag{4.6}$$

Where:

- k = equivalent stiffness (Eq. 4.3).
- p = empirically coefficient related to the impacting geometries [50];
- $x, \dot{x}$ = respectively the penetration depth and velocity;
- λ = hysteresis damping factor.

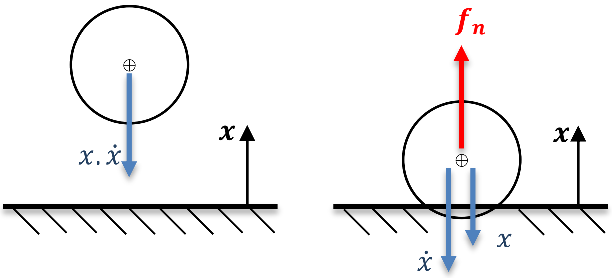


Figure 4.6: impact sequence: interpenetration depth and velocity sign convention in local reference frame (normal to the impacting surface).

This damped model is consistent with the expectation that the total contact force should vanish when the penetration depth goes to zero. This means that no impulsive behavior of the contact force dynamics appears at impact.

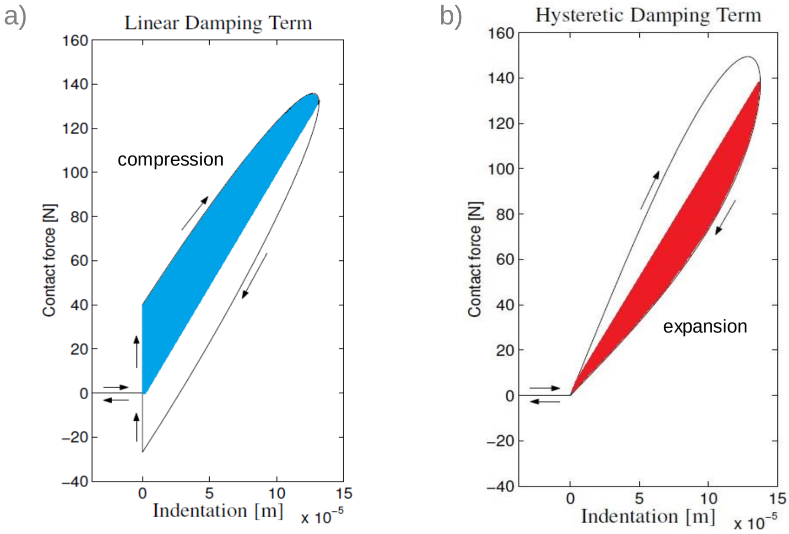


Figure 4.7: linear vs. hysteretic damping in normal contact force model [50].

In the case of the spring with the linear damping term, the contact force can be seen to be positive and non-zero at the start, negative at the end of the impact phase (Fig. 4.7a). This behavior contradicts two characteristics that we would expect from a consistent model:

- Contact force equal to zero at zero-penetration;
- Contact force always positive, to avoid *sticking* effect.

Assuming that the energy dissipated (area inside the loop) during the compressive phase (blue) and the one dissipated during the expansion phase (red) are equal, Hunt and Crossley approximate the hysteresis damping factor as:

$$\lambda_{HC} = \frac{3}{2}\left(1 - \mathrm{e}_{eff}\right)\frac{k}{v_n^i} \tag{4.7}$$

Where e_{eff} is the effective restitution coefficient, defined as:

$$\mathrm{e}_{eff} \triangleq -\frac{v_n^{end}}{v_n^i} \tag{4.8}$$

v_n^i and v_n^{end} are respectively the impacting normal velocities at the beginning and at the end of the contact. It has been shown that at low impact velocities and for most materials, the restitution coefficient can be approximated with an empirical formulation [51]:

$$\mathrm{e}_{emp} = 1 - \alpha\, v_n^i \tag{4.9}$$

Where α is an experimental parameter that usually varies in between the range [0.08-0.32] s/m. It is one of the degrees of freedom of the model, and it needs to be tuned wit experimental tests. Therefore if we approximate $\mathrm{e}_{eff} = \mathrm{e}_{emp}$, we easily get rid of v_n^{end}, so we can define all the contact parameters at the beginning of the impact. The important consequence is that now Eq. 4.7 becomes:

$$\lambda_{HC} = \frac{3}{2}\alpha\, k \tag{4.10}$$

And so Eq. 4.6:

$$f_n = \left(1 - \tfrac{3}{2}\alpha\, \dot{x}\right) k|x|^p \tag{4.11}$$

Lankarani and Nikravesh performed a similar analysis, assuming that the amount of energy dissipated is much less then the one stored in compression. This assumption allows to evaluate the integral of the energy dissipated during the impact phase, and assuming as before a symmetric distribution of the energy dissipation around the hysteresis loop, they obtained their expression for the hysteresis damping factor:

$$\lambda_{LN} = \frac{3}{4}\left(1 - e_{eff}^2\right)\frac{k}{v_n^i} \tag{4.12}$$

Or again assuming $\mathrm{e}_{eff} = \mathrm{e}_{emp}$ we obtain:

$$\lambda_{LN} = \frac{3}{2}\alpha\, k\left(1 - \frac{\alpha\, v_n^i}{2}\right) \tag{4.13}$$

Among these two models, we choose the first one (Hunt and Crossley), because of the independency of the formulation from the velocity at the impact instant.

A further reason for this choice can be found with the analysis of the approximating error associated to these two models. To do so, we define a damping factor a as the ratio between the the hystsresis damping factor and the equivalent stiffness:

$$a = \frac{\lambda}{k} \tag{4.14}$$

The contact normal force (Eq. 4.6) becomes:

$$f_n = (1 - a\,\dot{x})k|x|^p \tag{4.15}$$

The two colliding bodies are assumed to impact only when $x < 0$, therefore in order to prevent from negative forces, we must impose:

$$1 - a\,\dot{x} \geq 0 \tag{4.16}$$

Deriving the equation of relative motion governing the impact, we write:

$$m\,\ddot{x} = -f_n \tag{4.17}$$

So the equation of motion can be re-written including Eq. 4.15:

$$m\,\ddot{x} + (1 - a\,\dot{x})k|x|^p = 0 \tag{4.18}$$

Where m is the *reduce mass* of the two impacting bodies i and j, so defined:

$$m = \frac{m_i\,m_j}{m_i + m_j} \tag{4.19}$$

This equation of relative motion has been integrated by Stronge [38] leading to the following result:

$$\frac{1 + d/e_{eff}}{1 - d} = e^{d(1+1/e_{eff})} \tag{4.20}$$

Where d is a non-dimensional damping factor defined as:

$$d = e_{eff}\,a\,v_n^i \tag{4.21}$$

Eq. 4.20 represents the closed-form non-linear solution to the contact problem, derived by Stronge. Among all the possible values of d, a valid solution is the one that satisfies the condition expressed by Eq. 4.16. So, this implies $d < 1$ for any e_{eff}. At the same time the damping factor cannot be negative, otherwise this would cause energy creation during the impact. Therefore the valid solution must be for $0 < d < 1$. We derive the solution for this range of d, solving Eq. 4.20 using a numerical bisection algorithm with an initial guess $d_{guess} = 1 - e_{eff}^2$.

The solution is presented and compared with both Hunt and Crossley and Lankarani and Nikravesh models in Fig. 4.8.

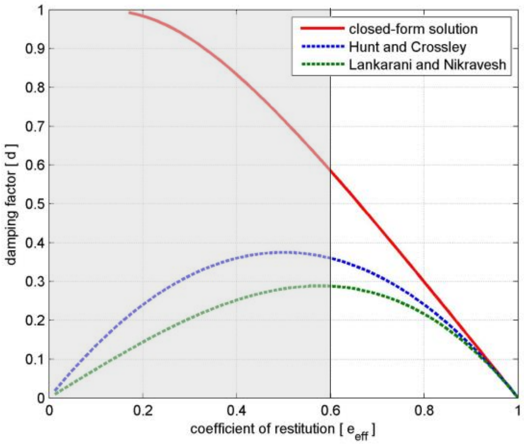


Figure 4.8: dimensionless damping factor d as a function of e_{eff}.

The expression of the damping factor d in the two models has been derived once rewritten the hysteresis damping factor:

$$\lambda = \frac{k\, d}{e_{eff}\ v_n^i} \tag{4.22}$$

Using Eq. 4.7 and Eq. 4.13 together with Eq. 4.22, we obtain the relations between the damping coefficient and the restitution coefficient for the two models:

$$d_{HC} = \frac{3}{2} \mathrm{e}_{eff}\left(1 - \mathrm{e}_{eff}\right) \tag{4.23}$$

$$d_{LN} = \frac{3}{4} \mathrm{e}_{eff}\left(1 - e_{eff}^2\right) \tag{4.24}$$

The damping factor d goes to zero at $e_{eff} = 1$ and this is the region where it is better approximated by the two analyzed models. In particular, we can see that Hunt and Crossley model better approximate the closed-form solution near the unity.

4.1.5 Contact Model Validation

As we want to validate the Hunt and Crossley normal-force contact model, we set up a simple 2D dynamic model of a bouncing ball falling and impacting on a plate. We consider a spherical ball of mass 0.1 kg falling from a height of 0.2 m and perform a parametric analysis on the contact model tuning coefficients, such as k and α, respectively for the equivalent stiffness and damping.

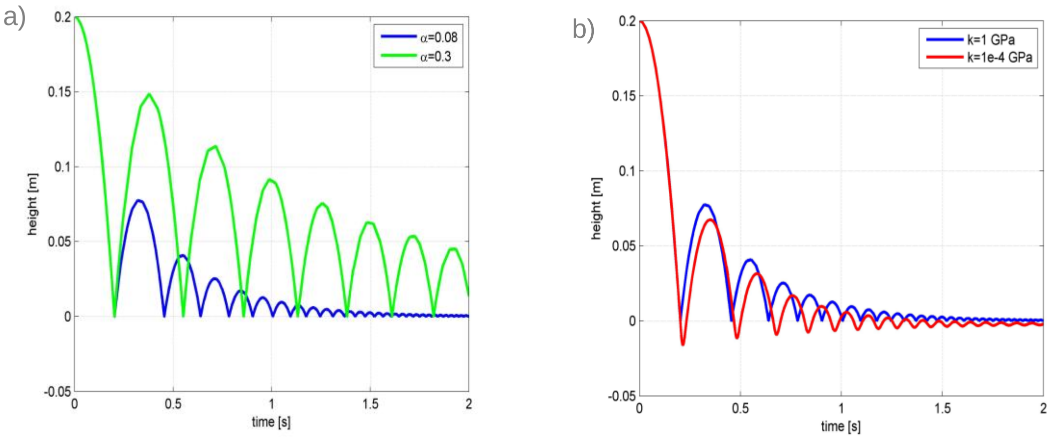


Figure 4.9: bouncing ball dynamics varying: a) the damping experimental parameter α, b) the stiffness *k*.

We clearly see from Fig. 4.9a that for an equivalent stiffness of 1GPa, the higher is *α*, the stronger is the damping effect, resulting into a more *sticky* surface. The further implication is the higher energy dissipation rate. In Fig. 4.9b, we note how the penetration depth becomes important due to the lower equivalent stiffness. The energy dissipation rate keeps almost the same due to the unchanged value of *α*. Analyzing better the first case (k=1GPa and α=0.08), we report the contact forces with time and a focus of the single impulse.

Zooming the evolution of the resulting contact reaction with time, we note the hysteresis, evident from the non-symmetric shape (Fig. 4.10b).

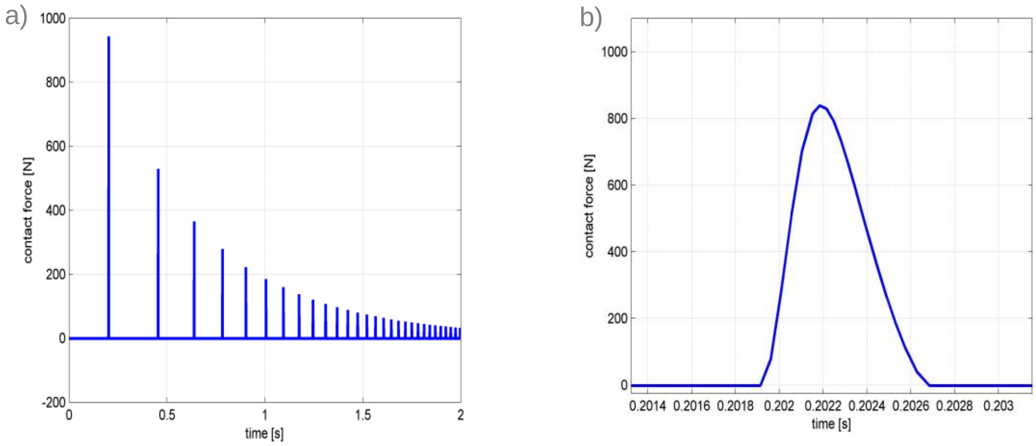


Figure 4.10: contact forces with time and focus of one impulse shape.

Therefore, we definitively verify through this last representation that:

- Contact force is always positive;
- Non-impulsive behavior of the contact force at the beginning and at the end of the impact;
- Existence of a hysteresis cycle.

Finally we compute the energies: kinetic E_k, potential U, elastic E_s and dissipated D ones:

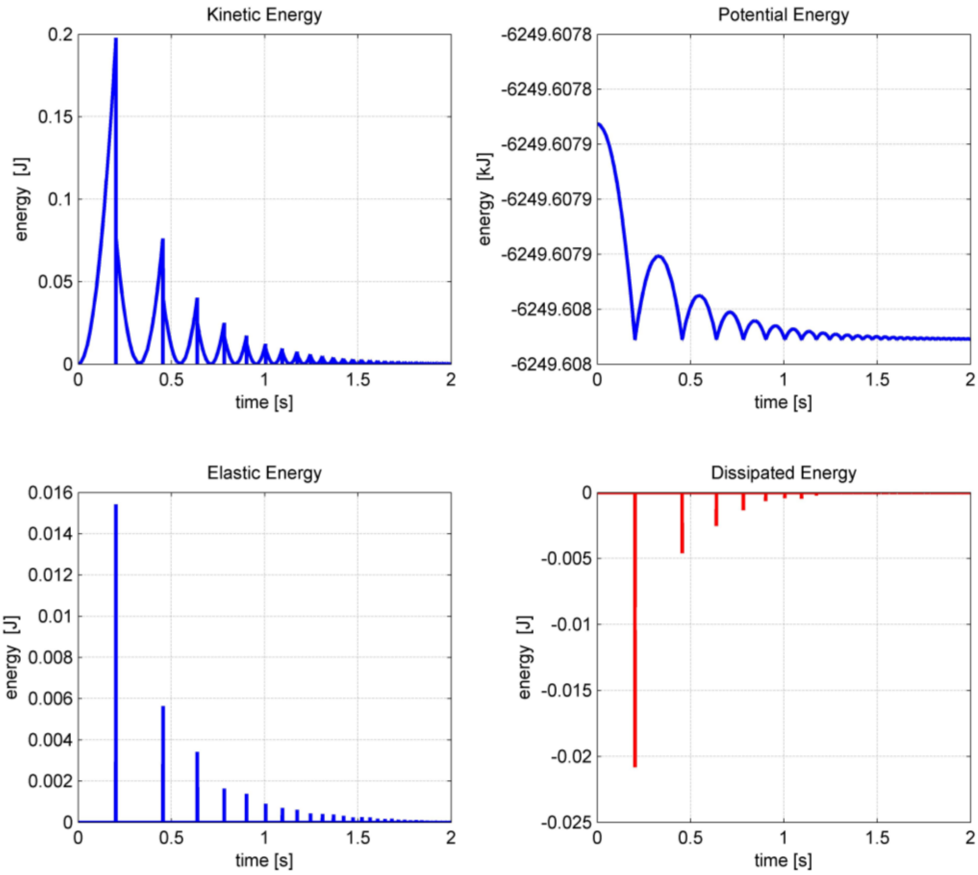


Figure 4.11: bouncing ball test: kinetic, potential, elastic and dissipated energies.

The model is finally validated by the total energy conservation reported graphically in Fig. 4.12 and resulting from the overall energy balance:

$$E_{tot} = E_k + U + E_s - D \tag{4.25}$$

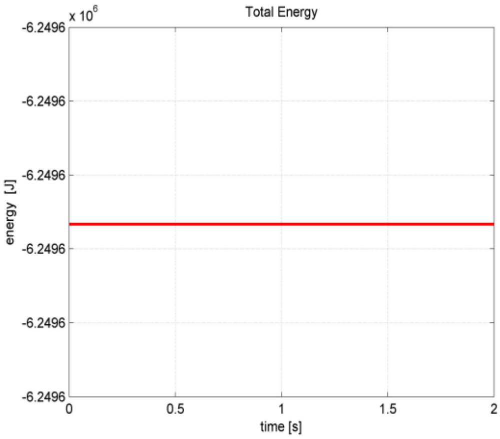


Figure 4.12: bouncing ball test: total energy conservation.

4.1.6 Friction Model

Differently from the normal contact force, previously obtained using scalar equations, the surface friction force is a vectorial quantity by nature.
The acting direction of the friction force is always tangent to the surface and opposite to the velocity by definition.

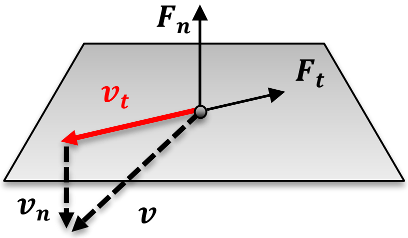


Figure 4.13: tangential contact force direction convention.

There are two levels of acting: static and dynamic, whose transition is not continuous. It is called static friction, the one occurring below a force threshold proportional to the normal contact acting force F_n, according to the Coulomb's static friction coefficient μ_s:

$$f_{static} \leq \mu_s\ |\boldsymbol{F_n}| \tag{4.26}$$

Experimental observations [53] have shown that the full magnitude of the stiction force does not come into effect as soon as the relative velocity become zero. Instead, the maximum static friction force gradually increases over time and eventually reaches the upper limit $f_{static} = \mu_s\ |\boldsymbol{F_n}|$. This is the so-called *dwell-time dependency* mentioned before.

The important advancement in this theory is the conversion of the force-based transition from static to dynamic, into a velocity threshold definition [54] (algebraic loop otherwise). Calling v_t the tangential velocity modulus and v_ε the velocity threshold, the friction force modulus F_t is so defined:

$$f_t = \begin{cases} \mu_d \, |\boldsymbol{F_n}| & : v_t \geq v_\varepsilon \\ \mu_d \, |\boldsymbol{F_n}| \, \dfrac{v_t}{v_\varepsilon}\left(2 - \dfrac{v_t}{v_\varepsilon}\right) & : v_t < v_\varepsilon \end{cases} \tag{4.27}$$

Here, μ_d is the Coulomb's dynamic friction coefficient. It is a regularized version of the Coulomb's law of dry friction (Fig. 4.14). If the slip velocity falls below the threshold v_ε, the friction force is faded out quadratically.

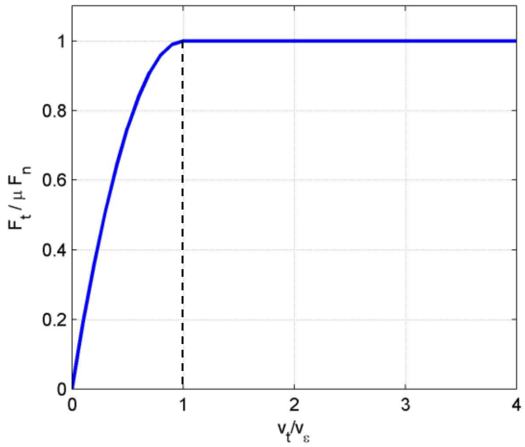


Figure 4.14: Regularized version of Coulomb's friction law

Transition velocity v_ε provides a further tuning parameter for the experimental validation. As a first guess it should be set equal to the Stribeck velocity v_S, that is the velocity at which the stick-slip effect occurs (in the Stribeck curve). It has been proven experimentally [55] that a velocity threshold in the range between 10^{-4} and 10^{-6} m/s is a good compromise between the accuracy and computational robustness and effectiveness.

In conclusion, the introduction of the tangential friction model completes the definition of the resulting contact force:

$$\boldsymbol{F_{contact}} = f_n \, \boldsymbol{n} + f_t \, \boldsymbol{t} \tag{4.28}$$

Acting normal direction $\boldsymbol{n}$ and tangential direction $\boldsymbol{t}$ are defined in more details in the next section.

4.2 Integration of the Contact Model in the Simulator

Contact models derived from Hertz theory have been analyzed to be used for modelling the contact forces in the multi-body dynamics simulation. Their key feature is that they can be expressed directly in terms of geometrically identifiable quantities, such as penetration depth and velocity. As a result, the implementation of a Hertz theory-based contact model into a multi-body dynamics is fairly straightforward. Same considerations hold for the dwell-time dependency friction model.

4.2.1 Acting Directions of the Contact Forces

To include the contact model into a multi-body dynamics simulation system, it is necessary first to specify the point where the contact force will be applied. Since the contact region is assumed to be very small, the torques generated by the local deformations in the contact region are neglected (coherently with the assumptions of the *point* contact model previously explained). Second, the contact force should act in the direction that minimizes the constraint violation and, hence, the normal direction should be determined as well as the tangential one. Therefore, to finalize the implementation of the model into the algorithm we need to retrieve:

- The contact point;
- The interpenetration depth and velocity;
- The direction of the contact forces.

The contact point is exactly identified by the collision detection algorithm (Chapter 5). The determination of the contact normal, direction along which the resultant force preventing penetration must act, is a parallel problem to the determination of the penetration depth and velocity. Our problem has to deal with contact between simple geometries, which the whole system has been simplified with. These geometries are spheres and planes and the possible combinations occurring are:

- Sphere on Sphere;
- Sphere on Planar Surface.

We treat as spheres:

- Net nodes and knots for the main contact problem. Ideally they are non-dimensional points in the dynamics integration, but in the contact they are given a specific dimension, that is a spherical one with the diameter of the associated Net thread;
- RSV of the Net nodes and knots for the auto-collision problem and threads interpenetration avoidance;
- RSV of the Bullets for the main collision and auto-collision problems.

Planar surfaces are the ones which the Target is discretized in, and they are directly associated with their exiting normal unit vector $\boldsymbol{n}_i$ (Section 2.5.2).
Analyzing the aforementioned two possible interactions, we derive the parameters required to define the normal and tangential contact forces.

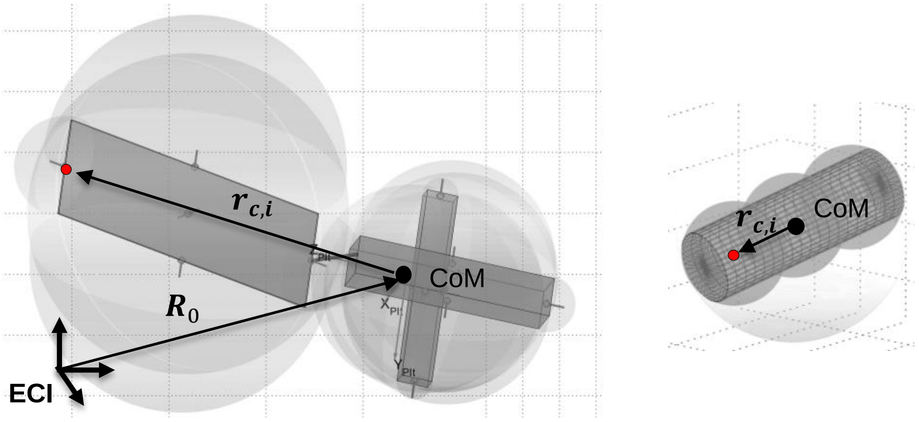


Figure 4.15: Target and Bullet contact surface centers position.

Calling $\boldsymbol{R}_i$, $\boldsymbol{R}_j$ and $\boldsymbol{V}_i$, $\boldsymbol{V}_j$ respectively the position and the velocity of the two impacting geometries centers *i* and *j* in the inertial frame, we define the relative distance and velocity of *i* with respect to *j*:

$$\boldsymbol{d} = \boldsymbol{R}_j - \boldsymbol{R}_i \tag{4.29}$$

$$\boldsymbol{v} = \boldsymbol{V}_j - \boldsymbol{V}_i \tag{4.30}$$

If these centers are already DOF of the system (Net nodes and knots), positions and velocities are directly retrieved from the state vector, otherwise (for Target mesh faces and Bullets RSV in Fig. 4.15) they are derived as follows:

$$\boldsymbol{R}_i = \boldsymbol{R}_0 + \left[\boldsymbol{r}_{c,i}\right]_{\boldsymbol{ECI}} \tag{4.31}$$

$$\boldsymbol{V}_i = \boldsymbol{V}_0 + \left[\boldsymbol{\omega_0} \times \boldsymbol{r}_{c,i}\right]_{\boldsymbol{ECI}} \tag{4.32}$$

Where $\boldsymbol{R}_0$, $\boldsymbol{V}_0$ are respectively the position and velocity (**ECI** frame) of the CoM of the rigid body, while $\boldsymbol{\omega_0}$ is the angular velocity of the body in local frame. The relative position of the *i-th* geometry center is $\boldsymbol{r}_{c,i}$ and it is defined in the pre-processing in local **BODY** frame. Quantities in local frame are then rotated in the inertial frame according to the actual attitude at time *t* retrieved from the state vector.

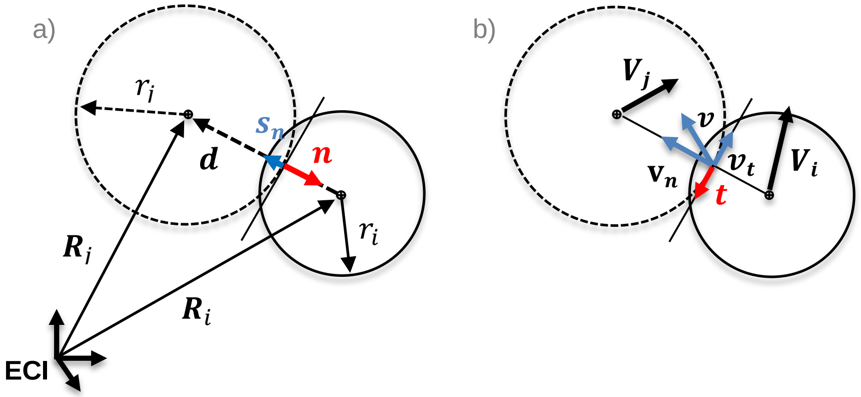


Figure 4.16: colliding spherical geometries quantities.

In the case of two impacting spheres (Fig. 4.16), the normal unit vector characterizing the normal contact force acting on *i*, is so retrieved:

$$\boldsymbol{n} = -\frac{\boldsymbol{d}}{|\boldsymbol{d}|} \tag{4.33}$$

Called r_i and r_j the radii of the spheres, the normal penetration depth s_n is:

$$s_n = |\boldsymbol{d}| - (r_i + r_j) \tag{4.34}$$

The relative velocity is:

$$\boldsymbol{v} = \boldsymbol{V}_i - \boldsymbol{V}_j \tag{4.35}$$

From which we derive the normal and tangential components:

$$\boldsymbol{v_n} = (\boldsymbol{v} \cdot \boldsymbol{n})\, \boldsymbol{n} \tag{1.36}$$

$$\boldsymbol{v_t} = \boldsymbol{v} - \boldsymbol{v_n} \tag{1.37}$$

The tangential unit vector defining the friction acting direction is therefore obtained:

$$\boldsymbol{t} = -\frac{\boldsymbol{v_t}}{|\boldsymbol{v_t}|} \tag{1.38}$$

To cope with numerical problems arising at small relative velocities, a sign relaxation convention is exploited.

$$\boldsymbol{t} = \begin{cases} -\dfrac{\boldsymbol{v_t}}{|\boldsymbol{v_t}|} & , \quad |\boldsymbol{v_t}| \geq v_{th} \\ -\dfrac{\boldsymbol{v_t}}{v_{th}}\left(\dfrac{3}{2}\dfrac{|\boldsymbol{v_t}|}{v_{th}} - \dfrac{1}{2}\left(\dfrac{|\boldsymbol{v_t}|}{v_{th}}\right)^3\right), & |\boldsymbol{v_t}| < v_{th} \end{cases} \tag{1.39}$$

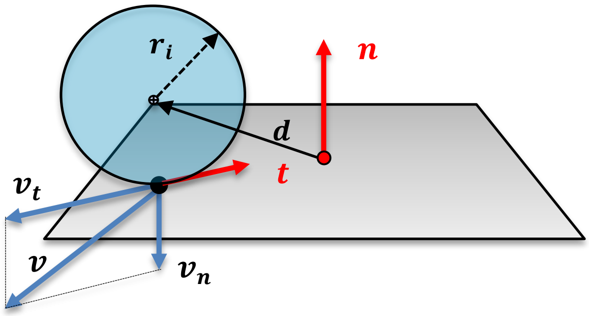


Figure 4.17: contact components in planar collision.

The reference threshold velocity v_{th} considered is a fraction of the Stribeck velocity v_s. As a guideline, the v_{th} should be at least one tenth of v_s. This minimizes the impact of the sign relaxation on the system dynamics when the relative velocity is below the threshold v_ε (Eq. 4.27), since the friction model will be already switched into the *stick* mode in this velocity range.

In the second impacting type, as before mentioned, the normal unit vector is directly available from the impacting planar surface of the Target. All the formulae just derived are still valid, except for the computation of the penetration depth:

$$s_n = \boldsymbol{d} \cdot \boldsymbol{n} - r_i \tag{4.40}$$

Once computed the overall resulting contact force acting on body *i* (Eq. 4.28), it is applied as same with opposite direction on the body *j* (Newton’s third law):

$$\boldsymbol{F}_j = -\mathbf{F}_i \tag{4.41}$$

Finally, although the contact models implemented do not take into account local torques, this does not prevent them to arise on rigid bodies *k* due to 3D geometry. In particular, the overall contact reacting force $\boldsymbol{F}_k$ and torque $\boldsymbol{T}_k$ are computed as the resultant of the *N* impacts of the nodes *i*:

$$\boldsymbol{F}_k = \sum_{i=1}^{N} [\boldsymbol{F}_i]_{ECI} \tag{4.42}$$

$$\boldsymbol{T}_k = \sum_{i=1}^{N} [\boldsymbol{r_i} \times \boldsymbol{F}_i]_{ECI} \tag{4.43}$$

Where $\boldsymbol{r_i}$ is the impact point of the node *i*. Force $\boldsymbol{F}_k$ is propagated in **ECI** frame, while torque $\boldsymbol{T}_k$ is directly computed and recorded in in local **BODY** frame.

4.2.2 Contact Model Parameters

The advantage of the contact models chosen, is that we are now able to tune the contact forces through few essential physical coefficients. This satisfies the required flexibility for the contact model and the physical meaning too. These are at the end:

- the equivalent stiffness k (Eq. 4.3), which is automatically computed in the pre-processing (with the exception of the radii), exploiting Eq. 4.4 if this is not specified;
- the hysteresis damping factor λ, automatically computed from Eq. 4.10 if not directly given.

As said, these can be directly provided and need to be tuned from the experimental validation campaign. For what concerns the tangential friction force, it is required to define the dynamic Coulomb's friction coefficient μ_d and the threshold velocity v_ε (Eq. 4.27). The crucial consideration before implementation, is how to manage the interaction between different bodies, since all these parameters do not characterize the single surface in an absolute way, but its interaction and relation with the others. We have two strategies to face the problem. The one more canonical involves the definition *a-priori* of all the possible interactions (Fig. 4.18a). In the scheme we have represented all these combinations, distributing to Target, Net and Bullets the characteristic parameters of the interactions, avoiding redundancies to optimize the memory (*A* stands for the *Auto-Collision* case). The Target can be characterize in each of its surface, with a different set of contact parameters. The same characterization potentiality is left to each Net thread and Bullet. This lets to have a very high level of detail in the contact characterization, that may not be required. If so, a second approach is proposed (Fig. 4.18b). It consists in a simplification of the model, providing a single set of contact parameters for each surface. Then, as soon as an interaction occurs between two bodies, the equivalent resulting coefficients are derived. This is done summing in series the stiffness and the damping coefficients according to an equivalent first order model:

$$\mathrm{k}_{eq} = \frac{\mathrm{k}_i\,\mathrm{k}_j}{\mathrm{k}_i + \mathrm{k}_j} \tag{4.44}$$

$$\mathrm{c}_{eq} = \frac{\mathrm{c}_i\,\mathrm{c}_j}{\mathrm{c}_i + \mathrm{c}_j} \tag{4.45}$$

While for the friction coefficient and threshold velocity, an average value between the colliding surfaces ones is set. This second solution is a rough simplification, useful to be exploited when less attention to the contact is required. This does not imply a total loss of the physical meaning of the phenomenon, but only a simplification to a first order model.

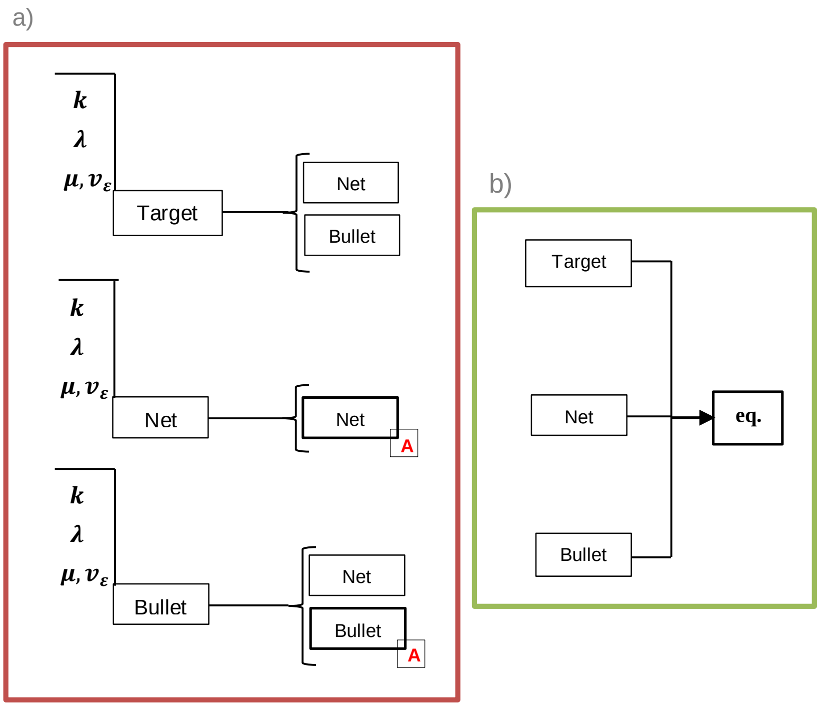


Figure 4.18: strategies to manage the interactions between different materials: a) all possible combinations accounted, b) simplified equivalent model.

4.2.3 Simulation Results and Models Comparison

Before implementing the elastic model so far theorized, a more simple non-elastic one was numerically built up and simulated. The resulting impact dynamics is reported in Fig. 4.19 in order to be critically compared with the more precise and realistic elastic model finally implemented. It is very interesting to note the *sticky* effect of this model, which is emphasized simulating a spinning Target. The Net keeps stuck to the first impact relative position on the Target. This is obviously unacceptable for the simulation requirements, being totally unrealistic. This difference can be also qualitatively appreciated simulating the dynamics of a conical Net impacting and wrapping over a tumbling Target (Fig. 4.20). The sticky effect fades away, as we can see a Net blowing return after the closure, showing the effectiveness of the Hunt and Crossley normal force modelling and dwell-time Coulomb’s dry friction. Certainly the correctness of the model cannot be only justified through a *physical-alike* sensation of the impact dynamics. Simulator requirements enforce a complete physical modelling of the problem. This implies necessarily an experimental validation, possible through a 3D reconstruction of the dynamics.

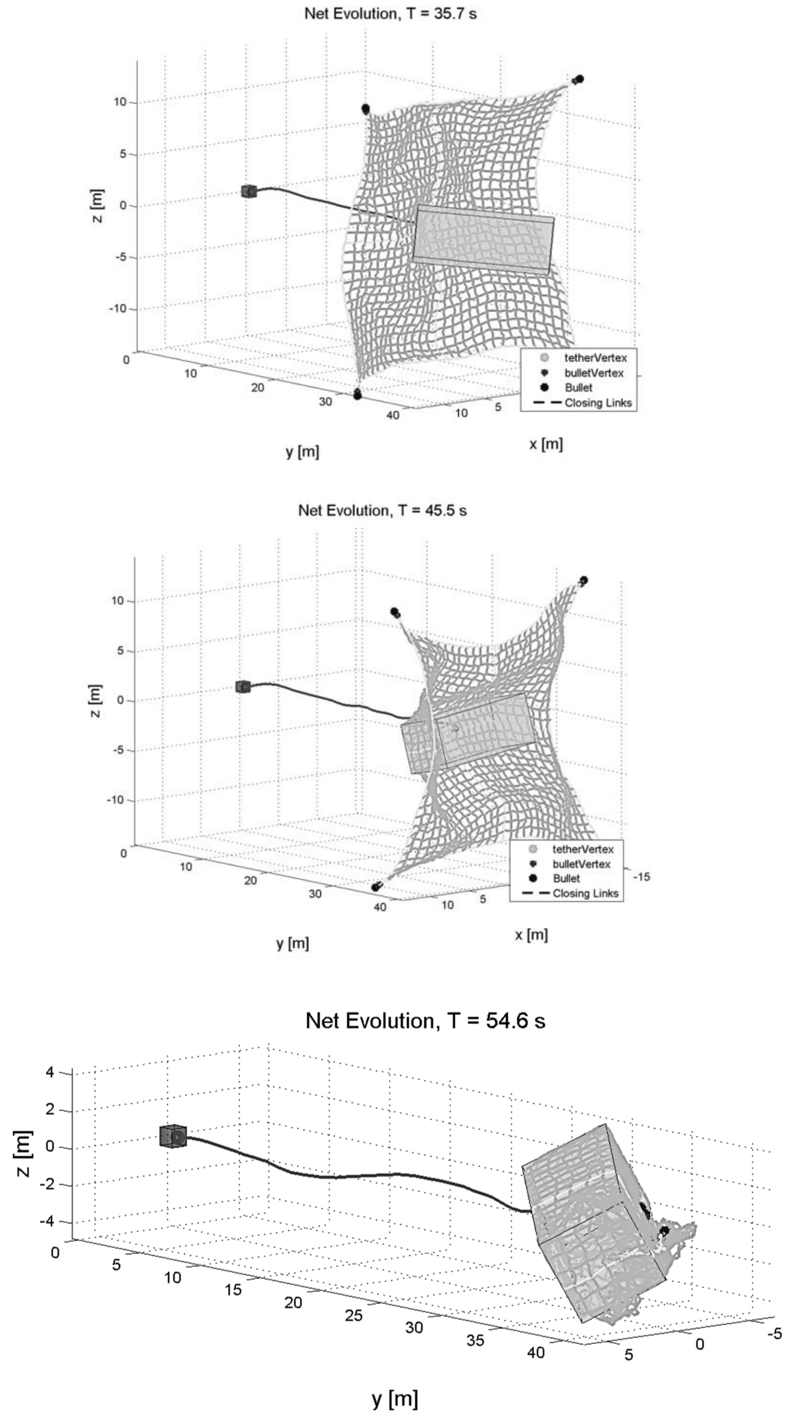


Figure 4.19: simulated dynamics of a Planar Net impacting and wrapping over a spinning Target according to a non-elastic contact model.

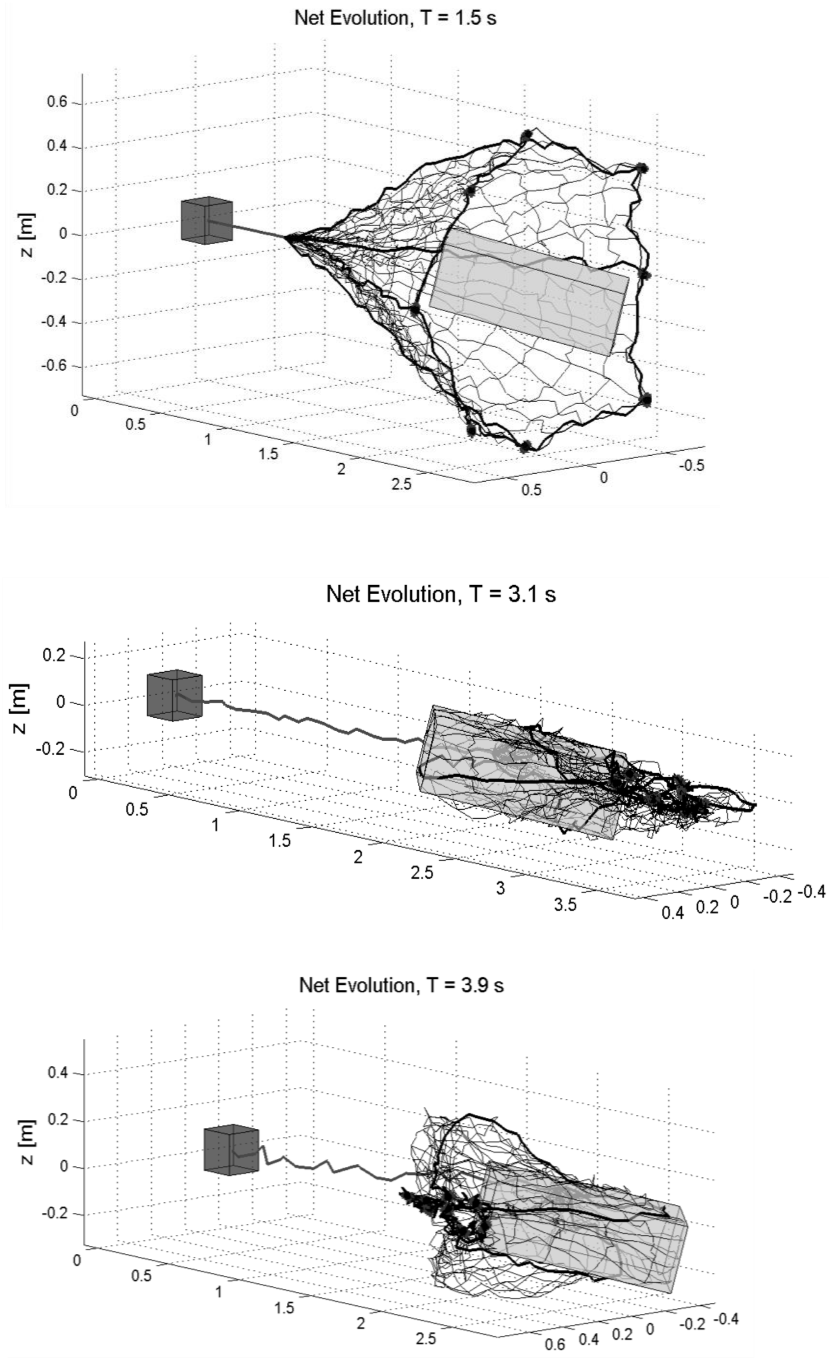


Figure 4.20: simulated dynamics of a Conical Net impacting and wrapping over a spinning Target according to the finally implemented elastic contact model.

In order to provide a complete set of tools for the contact model parameters recovery, contact forces and torques are post-processed and graphically displayed (Fig. 4.21).

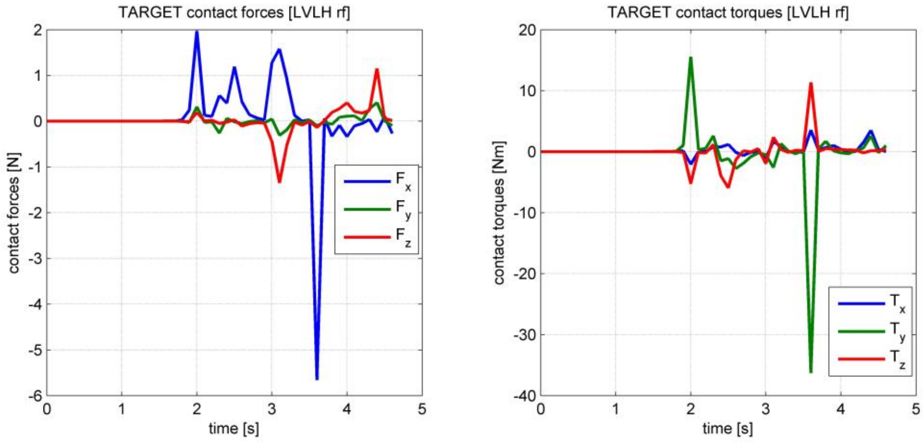


Figure 4.21: forces and torque acting on the Target due to the Net impact and wrapping. They are expressed in LVLH reference frame.

A final important topic is also the possibility to study the risk associated to the impact of the Bullets. For this reason, Bullets' contact forces over the Target are post-processed as well (Fig. 4.22).

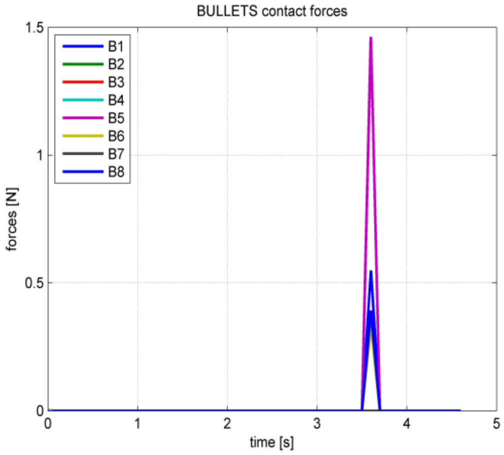


Figure 4.22: modulus of the impact forces of the Bullets hitting the Target.

Physical validations are provided in Section 6.5, simulating the full capture in all its phases and checking the effective energy dissipation modelled.

Chapter 5

SOFTWARE ARCHITECTURE

5.1 Overview

In the following Chapter it is presented the architecture of the Simulator modelled so far. User and third-party software interfaces are provided as well as the predefined models that can be easily accessed. Trade-offs for the solver have been performed and optimizations discussed. Finally, identified the bottlenecks of the dynamics, we propose an optimization path, promising a great jump in performances, since analyzed and tested on small scale.

5.2 Goals and Requirements

The Simulator objectives can be summed up with high performances (meaning precise results in reasonable time), flexibility, robustness and interfaceability.
Therefore, the requirements become:

- A simple and easy to handle user-interface;
- The most general definition of the simulation parameters, so that they can be alternatively provided directly by the user;
- A modular structure, so that functions and models can be easily identified and possibly substituted by third-party software or user-defined models;
- Optimization strategies for data allocation;
- Fast, accurate and robust integration solver;
- Code architecture that let possible to be coded in other simulation environments and eventually interface with graphics third-party software;
- The possibility to support analyses for experimental tests on ground and in a *zero-g* environment.

5.3 Software Architecture

The Simulator is completely designed in Matlab Simulink and it is composed of two main subsystems:

- *Pre-processing* ;
- *Dynamics Propagator*.

The *Pre-Processing* is required to set and prepare all the simulation parameters. It is characterized by three classes of parameters:

- *INPUT* parameters;
- *USER* parameters;
- *PROPAGATION* parameters.

They correspond to three different levels of elaboration of the simulation data, as shown below. For all the *INPUT* and *USER* parameters, a check is performed to preserve consistency of the data and avoid singularities.

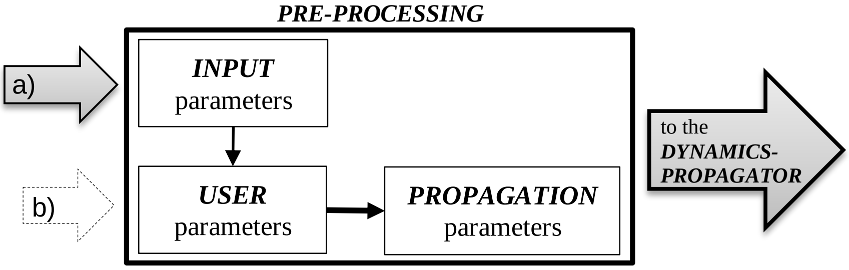


Figure 5.1: pre-processing structure and alternative ways to define the *PROPAGATION* parameters: a) provide simple *INPUT* parameters, b) directly provide *USER* parameters.

PROPAGATION parameters are the output of the last elaboration level of the *pre-processing* and are the ones used inside the Dynamics Propagator. They are optimized in order both to carry the minimum necessary information and to speed up the simulation. This is possible according to the reduction of the allocated memory and the introduction of strategic structures to exploit in the most efficient way the key data. *USER* parameters can be given directly from the user or automatically retrieved from *INPUT* parameters. These last are easy to handle *tuning* parameters of pre-defined Net geometries, Tether, Chaser and Target.
The *Dynamics Propagator* is the core of the software. It is made of three subsystems (Fig. 5.2):

- System Physical Model (SPM);
- Dynamics Integration Solver (IS);
- Collision Detection (CD).

The SPM includes the equations of the dynamics and all the models explained up to now:

- Equations of linear and angular dynamics;
- Environment model;
- Viscoelastic constitutive laws for flexible elements;
- Contact model.

It works with PROPAGATION parameters and allows directly to input external forces and torques profiles from third-party software or customizable *Simulink blocks*. All the dynamics is propagated through the state vector $\boldsymbol{X}$ and its derivatives $\dot{\boldsymbol{X}}$ and $\ddot{\boldsymbol{X}}$. The state vector contains the position of all the masses, the angular velocity and the attitude for the rigid bodies. Whichever parameter of the dynamics can be set as output to be saved in the *workspace* and then post-processed at the end of the simulation.

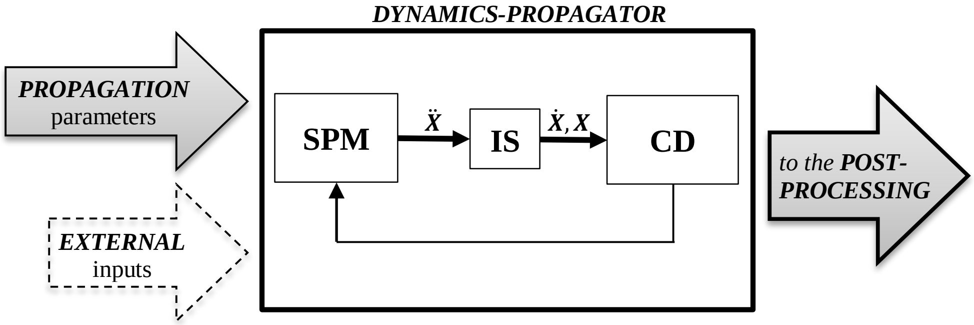


Figure 5.2: Dynamics-Propagator architecture.

5.4 User Interface and Pre-Processing

There are two ways to set up the Simulation:

➔ Exploit pre-defined configurations (*INPUT* parameters);
➔ Provide directly pre-processed information (*USER* parameters).

5.4.1 INPUT Parameters

There is the possibility to characterize 5 possible pre-defined **Net** shapes:

- *Planar-A*: planar net with orthogonal threads
- *Planar-B*: planar net with radial threads and orthogonal rings
- *Planar-C*: planar net with orthogonal threads, diagonal extra links
- *Pyramidal*: 3D shape pyramidal net with radial strings and transversal rings
- *Conical*: 3D shape conical net with radial strings and transversal rings

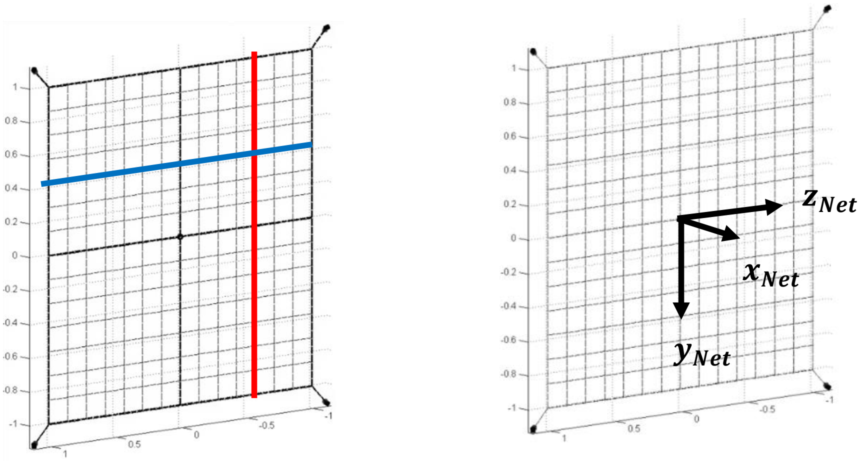


Figure 5.3.a: **Planar-A** shape, with and without reinforced

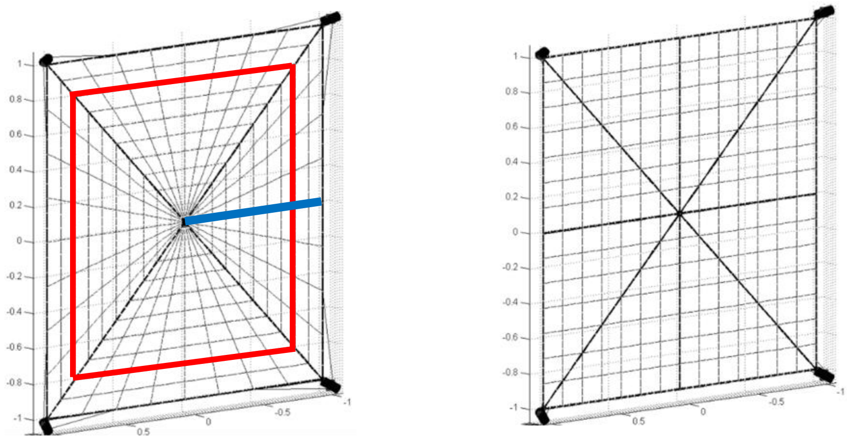


Figure 5.3.b: **Planar-B** and **Planar-C** shapes, with reinforced threads.

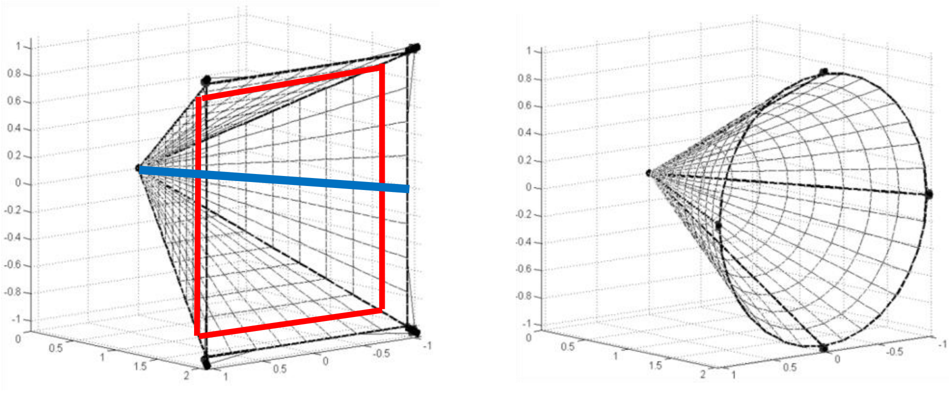


Figure 5.3.c: 3D shapes: **pyramidal** and **conical** with reinforced threads.

A Net is a mesh of knots modelled with the following elements:

- *Knots*: physical interlacement between threads of the mesh (depending on the manufacturing technology: weaved or knotted)
- *Links*: any thread between two knots. A link is subdivided into nodes (lumped masses discretizing the thread mass) and thread elements
- *Closing Links*: variable length links belonging to the Closing Mechanism
- *Bullet Links*: links connecting the Bullets to the Net
- *Bullets*: massive elements in charge of dragging the net to deploy
- *Tether*(s): link(s) defining the connection between Net and Chaser

For each Net type, it is necessary to define:

→ The dimensions [x y z] of the deployed configuration, in the Net reference frame. First dimension is to give for 3D shape geometries.

→ The mesh geometry:

  - The number of *strings* (longitudinal threads). For Planar-B, Pyramidal and Conical geometries, it needs to be a power of 2, for the others it must be odd.
  - The number of *rings* (transversal threads), it must be odd for Planar A and C.

→ The threads properties:

  - Diameter of threads and reinforced threads [m]
  - Linear density [kg/m]
  - Young's modulus [Pa]
  - Breaking stress [Pa]
  - Damping factor

  Reinforced threads are located at the net outer perimeter (*Net-mouth*) and at mid-links for all the geometries. In the case of Planar-B, Pyramidal and Conical shapes, the reinforced links are the diagonals instead of the mid-links, while for Planar-C both the diagonals and the mid links are reinforced.

→ The discretization:

  - The number of discretizing nodes for each link;
  - The knot ratio, as the mass ratio between the real mass (due to the manufacture process) and the one obtained from the discretization (Eq. 2.59), for the Net knots, the Tether connection and the Bullets connections.

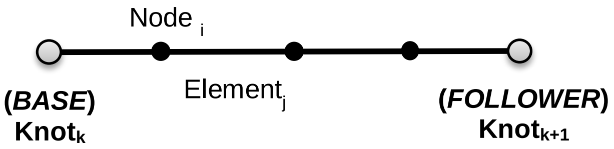


Figure 5.4: thread Link discretization with nodes and thread elements.

Bullets are cylindrical shape designed, so radius and height are to be provided. The Bullets mass can be directly defined or alternatively providing the Bullet Ratio (Eq. 2.38). A default number of 4 bullets is considered for all the Net shapes, but for the Conical one, which can have up to 8 Bullets. Moreover, it is required to set the connection relative position (Bullet **BODY** frame) of:

- Net to the Bullet;
- Closing Links (if any) to the winch inside the Bullet.

There is the possibility to include the bullet Links or attach the Bullets directly to the Net, as well as including or not the Closing System.

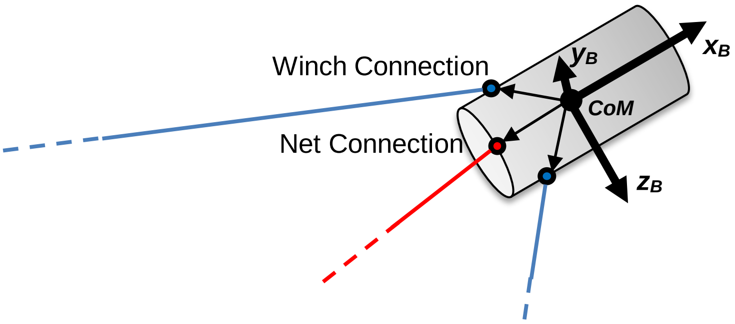


Figure 5.5: definition of the Bullet's connections.

The **Closing Mechanism** has been modelled with two possible configurations:

- Interlaced;
- Free.

The first one passes through all the knots of the existing outer perimeter or ring. The second one directly connects the Bullets each other (Fig. 5.6). In the case there are no Bullet Links, the winches are automatically placed inside the Bullets, otherwise they stands at the Net corners alone. The closing law, consisting of a time law for each closing link length, is retrieved from the user provided parameters defined in Section 2.8.

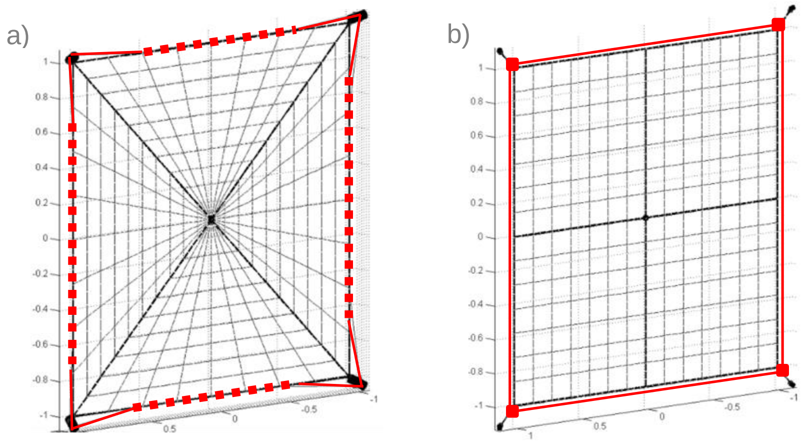


Figure 5.6: two alternative Closing Links integrations: a) interlaced, b) free.

The **Shooting System** parameters include:

- Divergence angle of each Bullet from the shooting direction (Net x-axis);
- Initial velocity modulus of each Bullets and relative shot delay;
- Dimensions of the Canister and initial position of the Bullets;
- Net folding pattern: pyramidal or random;

A single Tethered system is accounted. Tether can be included or not, according to the Orbit simulation and ground or parabolic tests.

Tether is defined in the same way of the Net threads, once specified the deployed length. Initially it is stored fully winded inside the Chaser. By default Tether connection with Chaser is fixed, so no winch and length control law are accounted.

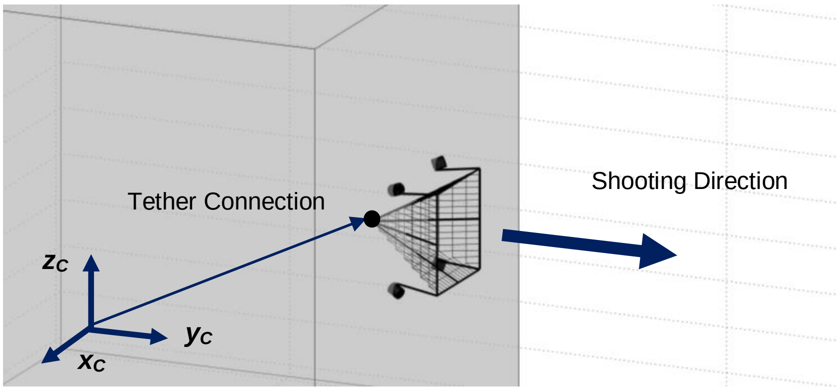


Figure 5.7: Net Canister positioning on the Chaser and shooting direction.

As explained in Chapter 2, the default **Chaser** configuration is a homogeneous cuboid rigid body:

- It is defined by mass and three dimensions, inertia is automatically computed;
- Initial position and velocity can be directly given or derived from Keplerian Parameters for orbital environment. Initial attitude and angular velocity are to be defined as well;
- It is further required to provide the Tether-connection position (in local **BODY** frame) if Tether present, as well as the shooting direction;
- To account for ground and parabolic flight tests, it is possible to introduce rotational and in translational constraints;
- For the simulation of the disposal, it is possible to define also the thrusting law, in terms of constant thrust modulus (that will be applied opposite to **V-bar**) and thrusting times (pre-tension and disposal burn).

The **Target** is defined exactly as the Chaser, except for the possibility to build up a much more complex body as a composition of simple cuboids, giving their dimensions, mass, CoM position and relative attitude.

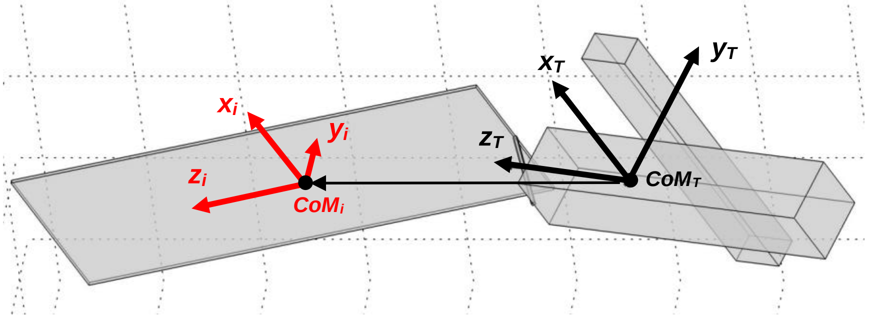


Figure 5.8: Target shape composition.

The operational **environment** is set among three pre-defined options:

- Orbit;
- Ground testing;
- Parabolic flight testing.

Testing scenarios are automatically recognized fixing the rotational and translational DOF of the Chaser. Initial positions are then interpreted as relative from the Earth surface, so that the gravity constant comes out automatically from the gravitational law modelled. Parabolic flight tests, in particular, are identified if further the gravitational constant is put to *zero* and it is defined the aircraft pitch rate.

5.4.2 USER Parameters

The *USER* parameters consist of 3 structures: Net, Chaser and Target. These structures are the elaboration of the *INPUT* ones in a most general formulation, so that they can be provided directly from an external user. Linear positions and velocities are expressed in **ECI** frame, while the angular velocities are in local BODY frames. The attitude of the rigid bodies is defined as **ECI-to-BODY** rotation quaternions.

The *Net* structure contains:

- The initial position, velocity and mass of all the knots and nodes.
- The characterization of the Links and Closing Links, with the definition of all the properties of the thread and their relation with respect to the knots.
- The Bullets initial conditions, masses, inertiae and surfaces characterization. Within this substructure it can be defined whichever other rigid body, besides the proper Bullets.
- The Tethers parameters, including the possibility to have them with a variable length, stating the knot where a winch is connected (e.g. winch mounted on the Chaser).

The *Chaser* structure contains its:

- Mass and Inertia;
- Initial conditions (linear and angular velocities, position and attitude);
- Linear and angular constraints;
- The position of the Net Canister and the relative orientation with respect to the Chaser body frame;
- The Tethers connection points;
- The surfaces mesh and material properties (contact parameters and environment interaction properties).

The meshing approach consists in discretizing the overall 3D geometry of the body through elementary planar elements. These are defined by nodes (vertices) and borders (Section 2.5.2).

The *Target* structure includes the same substructures of the Chaser ones.

5.4.3 PROPAGATION Parameters

USER parameters are then checked in definition and consistency before further elaborations. This is possible thanks to strategic redundancies in the parameters provided by the user. Then, *PROPAGATION* parameters can be created. These variables will be given to the Simulink model and propagated through the dynamics in time. *PROPAGATION* parameters can be defined as the minimum amount of information necessary to propagate the dynamics and structured in such a way that the simulation speed benefits. Since one of the bottlenecks of the dynamics propagation is the integration of the accelerations and the velocities, a vectorization of the full state is the solution. Therefore, we define a unique state vector $\boldsymbol{X}$ containing the linear and the angular state of all the bodies:

$$\boldsymbol{X_i} = \left\{ \begin{matrix} \begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix}_{ECI} \\ \begin{Bmatrix} \int \omega_x \\ \int \omega_y \\ \int \omega_z \end{Bmatrix}_{BODY} \\ \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix}_{ECI \to BODY} \end{matrix} \right\} \tag{5.1}$$

It has dimensions [DOFx1], where:

$$DOF = 10\, nRB + 3\, nK \tag{5.2}$$

- nRB = number of rigid bodies;
- nK = number of nodes and knots.

Initial conditions $\boldsymbol{X_0}$ and $\dot{\boldsymbol{X}}_{\boldsymbol{0}}$ are computed from *USER* parameters and provided to the numerical integrators.

$$\boldsymbol{M_i} = \left\{ \begin{matrix} \begin{Bmatrix} m_i \\ m_i \\ m_i \end{Bmatrix} \\ \begin{Bmatrix} I_x \\ I_y \\ I_z \end{Bmatrix} \\ \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix} \end{matrix} \right\} \tag{5.3}$$

Consequently, a related *mass matrix* should be associated, but since it would be diagonal (thanks to the **PI-BODY** frame adopted), it is vectorised to optimize memory. Division then is performed element-wise in the simulation to retrieve accelerations from forces and equivalent torques (coming from Euler's equations) to be then integrated. Quaternions second derivatives are not obtained from that division, they come out from the angular accelerations. The *mass vector* is hence made of the masses of the nodes and the Principal Inertia moments of the rigid bodies. It has dimensions [DOFx1]. Initial mass $\boldsymbol{M_0}$ is computed and will be propagated in time taking into account winches and reels actions, responsible for mass re-distribution.

The whole Net definition is translated in a single matrix called *Topology matrix*, which characterizes every single thread Link with all the parameters necessary to compute the viscoelastic forces. It is a [11 x *nLinks*] matrix (where *nLinks* is the overall number of Links) defining:

- The base knot index pointer in the state vector;
- The follower knot index pointer;
- The first node index pointer;
- The winch knot index pointer (if any, otherwise is [0]) characterizing all the variable length Links;
- The index pointer of the connection point on a rigid body (for torques computing);
- The axial stiffness *EA* (Eq. 2.48);
- The damping coefficient *c* (Eq. 2.61);
- The nominal element length L_0;
- The number of discretizing nodes;
- The breaking strain $\mathcal{E}_{max}$;
- The drag coefficient c_D.

Collision detection geometrical parameters (identified in Section 3.4.5) as well as the contact properties (defined in Section *4.2.2*) are stored in a single structure each for Net, Bullets, Chaser and Target.

5.5 Dynamics Propagator

After the *pre-processing* has been launched once, the initial state is then propagated through the dynamics for a specific time range, set *a-priori* by the user. The *Simulink* model is organized in functional blocks as shown in Fig. 1.2, receiving the *PROPAGATION* parameters at the beginning, interfacing with any possible external inputs and providing the final state as output, as well as any variable set by the user to be saved in the *Matlab workspace* (e.g. forces and internal stresses).

5.5.1 Interface with External Inputs

To better understand the way the Dynamics Propagator interfaces with external inputs, we explode the SPM block (Fig. 5.9). It is composed of four distinct subsystems providing the forces and torques acting on the system:

- internal viscoelastic forces for the Net threads;
- external forces and torques coming from the environment modelling and the contact interaction between the bodies.

The fourth block (Parabolic Flight) models the apparent relative accelerations arising in an ideal *zero-g* parabolic flight (Appendix A).
The forces and torques computed for each body are then propagated inside a single vector [DOF x 1].

$$\boldsymbol{F_i} = \left\{ \begin{matrix} \begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix}_{ECI} \\ \begin{Bmatrix} T_x \\ T_y \\ T_z \end{Bmatrix}_{BODY} \\ \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix} \end{matrix} \right\} \tag{5.4}$$

The dynamics block contains the linear and angular dynamics equations of motion. It receives the $\boldsymbol{F}$ vector and derives the accelerations $\ddot{\boldsymbol{X}}$ as output.

The System Physical Model block interfaces with the external inputs through:

→ *Essential Boundary Conditions* (EBC), in terms of nominal lengths $L_{0,i}(t)$ of the threads i. This happens for variable length Links, which have a time-varying length law externally defined. The size of the $L_{0,i}(t)$ parameter is [*nLinks* x 1], that is defining all Links. Application examples are Closing Links and Tether controlled in length (after the capture, to control the oscillations during the disposal).

→ *Natural Boundary Conditions* (NBC), in terms of profiles of forces and torques directly given to the dynamics block within $\boldsymbol{F_i}(t)$ ([DOF x 1] vector). Examples of NBCs are provided by the Shooting System and Chaser Thrusting: the first provides the ejection forces applied to the Bullets at the shot instant, the second the thrusting law for the controlled disposal. They can be easily substituted by whichever user-defined functional blocks in Simulink or coming as external input source. An example, is the integration of an Attitude Control Subsystem (ACS), working in feed-back with the state and providing control torques as input.

→ Finally, *linear* and *angular accelerations* can be directly provided. This becomes useful when dealing with parabolic flight tests, in order to account for perturbing accelerations and apparent accelerations (arising in an assumed *zero-g* environment). Perturbing accelerations are directly added to the $\ddot{\boldsymbol{X}}$ vector, while apparent accelerations are computed once provided in input to the Parabolic Flight block the pitch angle profile $\alpha_{pitch}(t)$ of the aircraft.

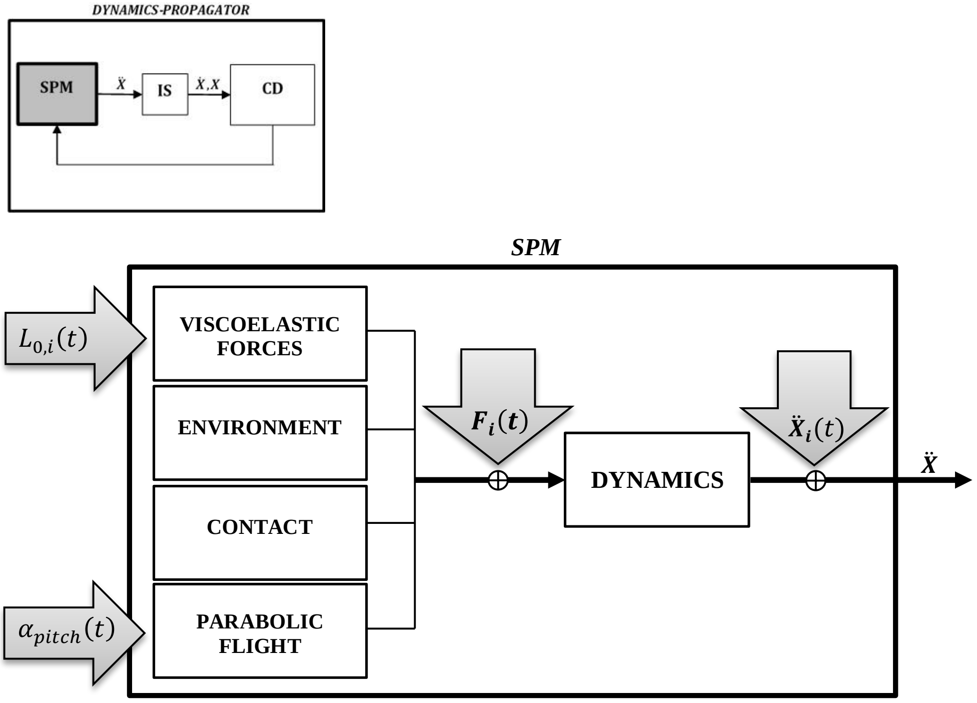


Figure 5.9: System Physical Model explosion in the constitutive subsystems. Interface with the external inputs.

5.5.2 Integration Solver Trade-off

The Integration block is made of the integrating solver $\left(\frac{1}{s}\right)$ supported by two subsystems processing the quaternions. Quaternions derivative $\dot{\boldsymbol{q}}$ is retrieved from the angular velocity, whose components define the skew-symmetric $\boldsymbol{\Omega}$ matrix:

$$\dot{\boldsymbol{q}} = \frac{1}{2}\,\boldsymbol{\Omega} \cdot \boldsymbol{q} \tag{5.5}$$

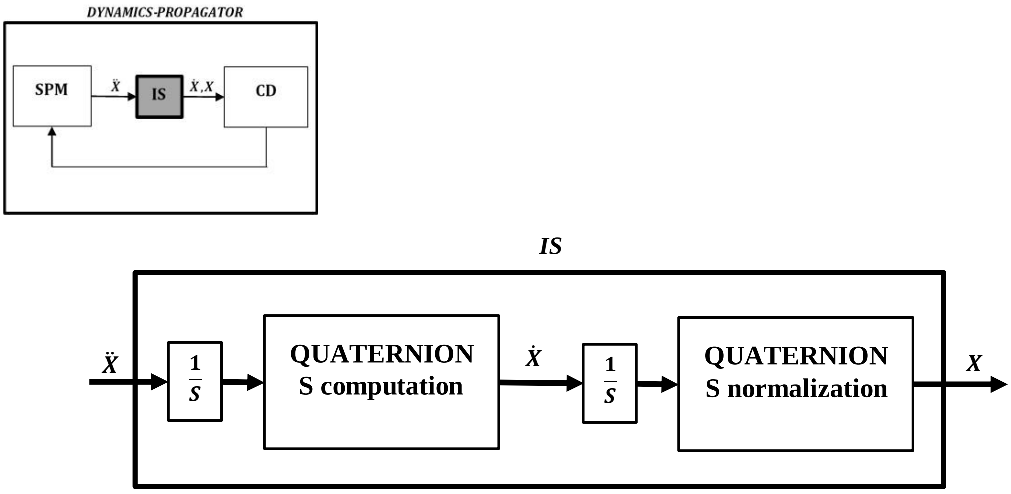


Figure 5.10: Integration Solver structure.

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ & 0 & \omega_x & \omega_y \\ & skew & 0 & \omega_z \\ & & & 0 \end{bmatrix} \tag{5.6}$$

Quaternions are then integrated inside the whole state vector $\boldsymbol{x}$ an d normalized:

$$\boldsymbol{q} = \frac{\boldsymbol{q}'}{|\boldsymbol{q}|} \tag{5.7}$$

The integrator has been chosen after a trade-off among a complete list of *Simulink* built-in solvers, comparing the timing profiles and the accuracy of the results. The optimization of these parameters, in fact, is the goal of the whole project and the class of the integration solver exploited is crucial for the achievement of that goal. The dynamics is really easy to be solved as long as it is unconstrained, involving the integration of the Newton's second law and the Euler's equations. The time step can be as large as the user requires. However, as soon as threads elongations or contacts occur, the dynamics become more stiff to be solved and the integration time step is required to be much smaller. For this reason discrete solvers with fixed time step are avoided and built-in ODE solvers are analyzed (Tab. 5.1).
ODE23 and **ODE23t** have been discarded because of:

- timing inefficiency for the required accuracies;
- absence of a numerical damping.

They are optimized for mild stiffness dynamics. The same conclusions are drawn for **ODE113**, suitable mainly for particularly stringent error tolerance problems, involving relatively high computational times.

Table 5.1: List of Integration Solvers and associated integration methods analyzed for the trade-off.

SOLVER	***INTEGRATION METHOD***
ODE23	Bogacki-Shampine
ODE113	Addams-Moulton
ODE45	Dormand Prince
ODE15s	Numerical Differentiation Formulas (NDFs)
ODE23s	Modified Rosenbrock
ODE23t	Trapezoidal
ODE23tb	Backward Differentiation Formulas (BDFs)

In the eventuality the problem becomes too stiff, implicit methods could be exploited instead of the explicit ones. This is the case of **ODE15s**, **ODE23s** and **ODE23tb**. However, for our specific case, they turn out to be efficient only with rough tolerances, timing becomes unacceptable.

Table 5.2: timing comparison between the integration solvers investigated.

SOLVER	***COMPUTING TIME***
ODE23	30.2 s
ODE113	27.7 s
ODE45	**25.2 s**
ODE15s	>60 s
ODE23s	>60 s
ODE23tb	>60 s

Times shown in Tab. 5.2 refer to a benchmark simulation dealing with the deployment and impact of a 758 DOF planar Net. The simulation has been run in *Normal Mode* on a 64bit machine with an i7 processor and a CPU speed of 3.2 GHz.

Lack of accuracy in the integration has been visually identified, noting unnatural deployment dynamics, disappearing once the relative tolerance between the integration steps is decreased. This happened with **ODE23** and **ODE113** solvers, which can solve the dynamics faster than **ODE45**, but lacking a lot in accuracy, so increasing the tolerance, the timing grows as well.
Therefore, the winning solver for computational efficiency, keeping the accuracy of the results, has turned out to be **ODE45** with the Dormand-Prince method. For completeness, the **ODE45** is an explicit Runge-Kutta (4,5) solver, a fifth-order

method performing a fourth-order estimation of the error. Simulations results using this solver are accurate and coherent. This integrator class has proven to be more accurate and faster than all the others for our specific dynamics.

5.5.3 Speed-up Optimizations

Besides the integration solver, here we report all the further strategies that have been identified and adopted in order to optimize the Simulator computational efficiency:

- → **vectorization** of all the signals (in the *Dynamics Propagator*);
- → **pre-allocation** of the parameters (in the *pre-processing*);
- → ***frame-based* data**, fixing a time step for the storage of the integration outputs instead of recording all the integration time steps. More so since we use a variable time step solver, which reduces the time step when the dynamics becomes stiff. This speeds up the code exponentially when dealing with huge amount of data, such as when with >50k DOF (real simulations with huge Net dimensions). This is reasonable since saving data in memory requires a non-null finite time. Sampling data in batches revealed to speed up the simulation ***2*x**.
- → Minimize the number of Simulink blocks that communicate and grouping them into bigger ones, treated as ***atomic* unit**;
- → Minimize the ***interpretation*** that Matlab has to do with operations between vectors and matrices, explicating the indexes.

Strong and very effective improvements have been identified translating and running the code in a **C++ environment**, exploiting the auto-coding. Performances are proved to increase up to ***10*x**, which is a great improvement. The ODE45 solver is auto-coded in C++ and due to its structure we can further speed up the propagation of the dynamics setting the relative and absolute tolerances to be automatically and dynamically set (impossible otherwise with the other solvers).

5.6 Future Work

Tests on this Simulator revealed a significantly improved performance when the C++ code is executed outside Matlab. On the other hand, the drawbacks of this choice are that:

- Operations are not interpreted, so they need to be fully explicated;
- The memory allocation is *static* and no more *dynamic* as in Matlab;
- It is necessary to design an ad-hoc variable time step solver.

The first two issues are solved exploiting *Matlab-Coder*, while the last one needs further work. The knowledge achieved of the Net dynamics equations allows us to include relevant intelligence in the solver. These different dynamics are here listed:

- Free motion = linear equations;
- Perturbations = slow dynamics (small forces applied for a long time);
- Viscoelastic actions, such as threads elongations and contact = very fast dynamics (large forces applied for a short time).

As before, slow dynamics will be solved faster with a higher time step, while fast dynamics will reduce the integration time step. This procedure will be supported by the computation of the relative error. As long as this error exceeds the tolerance fixed it encounters a failure and the time step is reduced. But, knowing the dynamics we will be able to predict the time step, drastically reducing the failures.

Finally the strongest improvement is the **parallelization** of the code. It is not possible to do it in Simulink directly. Two alternatives are shown:

- Rewriting the whole Simulator as a Matlab script;
- Codify the Simulator in C++ and then parallelize the code outside Matlab.

Second choice is advised. However parallelization does not provide any benefit if applied without due care to the code structure. The bottlenecks of the simulator have been identified in the Net viscoelastic forces computations (elongation and contact) and Collision Detection. Parallelization requirements has been shown in Section 3.4.7 and solutions were discussed specifically for Collision Detection algorithm (contact forces depend as well). Net viscoelastic forces computation would be the only issue to benefit since it does not include any nested for-loops, as it proceeds *point-by-point*. Therefore in this case the parallelization would imply the subdivision of the *N* threads to compute the elongation of, within *P* groups that will be treated separately from the equivalent number of processors or better clusters.

Chapter 6

VALIDATION AND ANALYSIS

6.1 Software Validation Plan

In order to prove the reliability of the Simulator, a validation plan has been set, consisting of:

- A *physical validation* of all the mathematical models implemented, through a post-processing analysis of the numerical results, proving their consistency and physical coherence;
- An *experimental validation* of the model and the physical parameters.

This last involves a comparison between the predicted numerical results of the dynamics and the effective experimental ones retrieved through a 3D reconstruction of the dynamics. The Software can be considered reliable and hence validated if it proves the ability of faithfully representing the dynamics (fixed an error tolerance), tuning the available physical parameters accessible by the user. A test campaign reproducing the Net dynamics in a relative *zero-g* environment has been planned and the simulator set-up support is discussed in the Appendix A.

6.2 Physical Validation

A first validation of each model consists in comparing the simulation results with the ones analytically computed and checking the consistency with energy and momentum conservation principles.

6.2.1 Dynamics

We consider as testing scenario a free tumbling spacecraft on a 500 km altitude LEO. The shape is a cuboid with dimensions [10 6 5] m and mass 2500 kg. In order to validate specifically the dynamics equations, all the perturbations are switched-off but gravity force.

First we validate the orbital motion, through the position [**ECI** frame] and velocity (**LVLH** frame) components in Fig. 6.1. We easily note a match of the computed orbital period and the resulting periodicity of the components. Altitude keeps constant and the only not-null velocity component is the one aligned with V-Bar, coherently with the null eccentricity of the orbit.

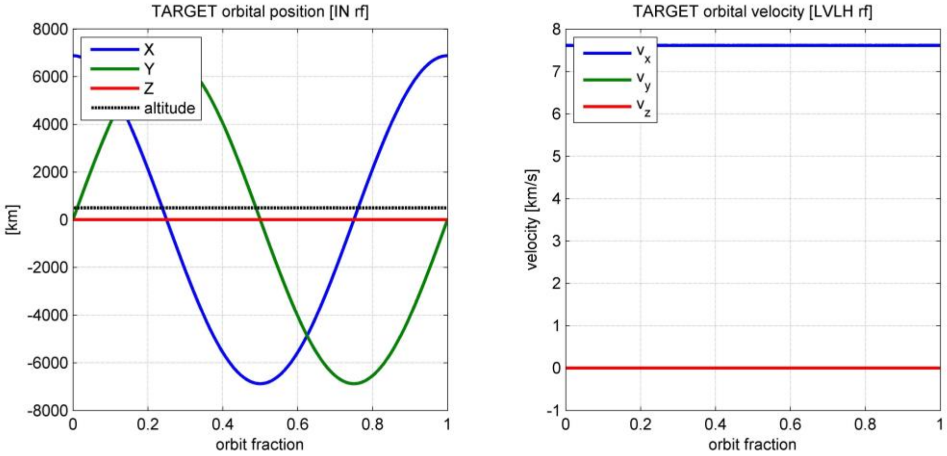


Figure 6.1: orbital position (ECI frame) and velocity (LVLH frame) components, in one orbital period.

Then we compute the attitude angular momentum $\boldsymbol{\Gamma}_{ATT}$ and the orbital specific angular momentum $\boldsymbol{h}_{ORB}$ so defined:

$$\boldsymbol{\Gamma}_{ATT} = \boldsymbol{I} \cdot \boldsymbol{\omega} \tag{6.1}$$

$$\boldsymbol{h}_{ORB} = \boldsymbol{R} \times \boldsymbol{V} \tag{6.2}$$

Where:

- $\boldsymbol{I}$ and $\boldsymbol{\omega}$ respectively the inertia matrix and the angular velocity, expressed in BODY frame;
- $\boldsymbol{R}$ and $\boldsymbol{V}$ are the orbital position and velocity vectors in ECI frame.

Since no external perturbations $\boldsymbol{T}_{EXT}$ have been included, we are able to validate the orbital and attitude dynamics as long as these quantities keep constant, according to the momentum conservation principle (Eq. 6.3).

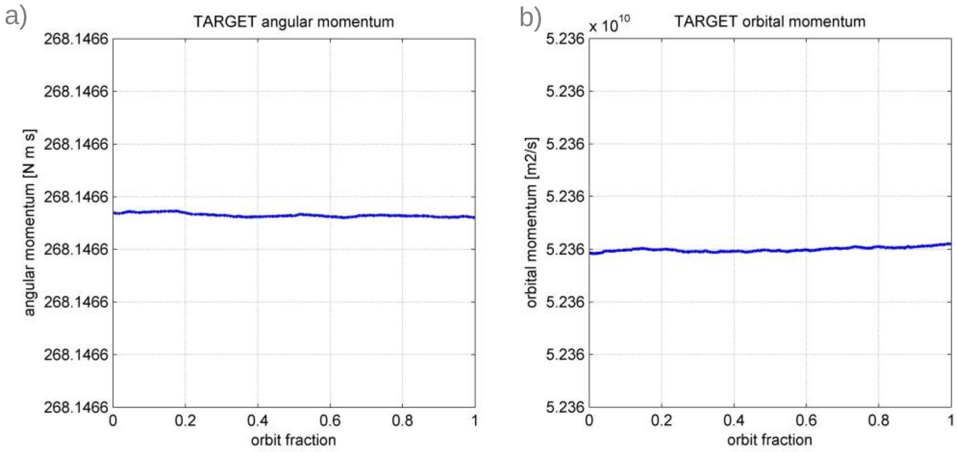


Figure 6.2: a) angular and b) orbital momentum conservation for orbital and attitude dynamics validation.

$$\frac{d\boldsymbol{\Gamma}}{dt} = \boldsymbol{T}_{EXT} = \boldsymbol{0} \tag{6.3}$$

Angular and orbital momentum conservation is proven and shown in Fig. 6.2, validating both the attitude and the orbital dynamics. No energy losses are modelled for the orbital dynamics nor for the attitude, so to prove the absence of any integration error that may induce fictitious losses, we check the energy conservation:

$$\frac{dE_{TOT}}{dt} = \dot{W}_{LOSS} = 0 \tag{6.4}$$

Where:

$$E_{TOT} = E_k + U \tag{6.5}$$

The kinetic and potential energies related to the orbital motion are respectively defined:

$$E_K = \frac{1}{2}\, m\, \boldsymbol{V} \cdot \boldsymbol{V} \tag{6.6}$$

$$U = -\mu \frac{m}{|\boldsymbol{R}|} \tag{6.7}$$

To validate the attitude dynamics propagation, the rotational kinetic energy is evaluated separately:

$$E_{K-ROT} = \frac{1}{2}\boldsymbol{\omega} \cdot \boldsymbol{I} \cdot \boldsymbol{\omega} \tag{6.8}$$

Both conservation principles are satisfied as shown in Fig. 6.3, validating once again both the dynamics.

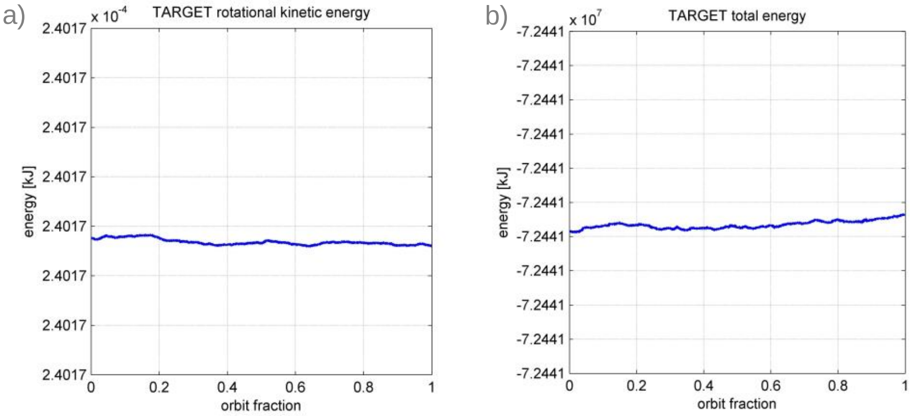


Figure 6.3: a) rotational kinetic and b) orbital energy conservation for attitude and orbital dynamics validation.

6.2.2 Environment

Considering the same orbiting body of the previous paragraph, we now set an inertial pointing attitude to investigate the perturbations effects. All the values are confirmed by the models and estimations already done in Section 2.4. What we want to focus on and analyze, is the coherence of the dynamics of these perturbations with the corresponding attitude and orbital position.

We appreciate that the *J2* components have frequencies multiple of the orbital one, in particular the perturbations along V-bar and R-bar have it double than the one along H-bar. This is explained from Eq. 2.8 noting that these components are expressed as function of the true anomaly ϑ (in our example the pericenter anomaly ω is null) and their frequency is appreciated as soon as they are rewritten as follows:

$$\begin{cases} a_{R-bar} = f(\sin^2(\vartheta)) = f(\cos(2\vartheta)) \\ a_{V-bar} = f(\sin(\vartheta)\cos(\vartheta)) = f(\sin(2\vartheta)) \\ a_{H-bar} = f(\sin(\vartheta)) \end{cases} \tag{6.9}$$

The *atmospheric drag* is coherently acting opposite the orbital velocity: the only non-null component is along V-bar (negative sign). Looking at the *radiation pressure*, we have the confirmation that the body experiences an eclipse for the 30% of the period, as expected from analyses (equatorial LEO at 500 km altitude). The validation holds as we notice that the only non-null component is the one opposite to R-bar, corresponding to the radiation emitted by the Earth.

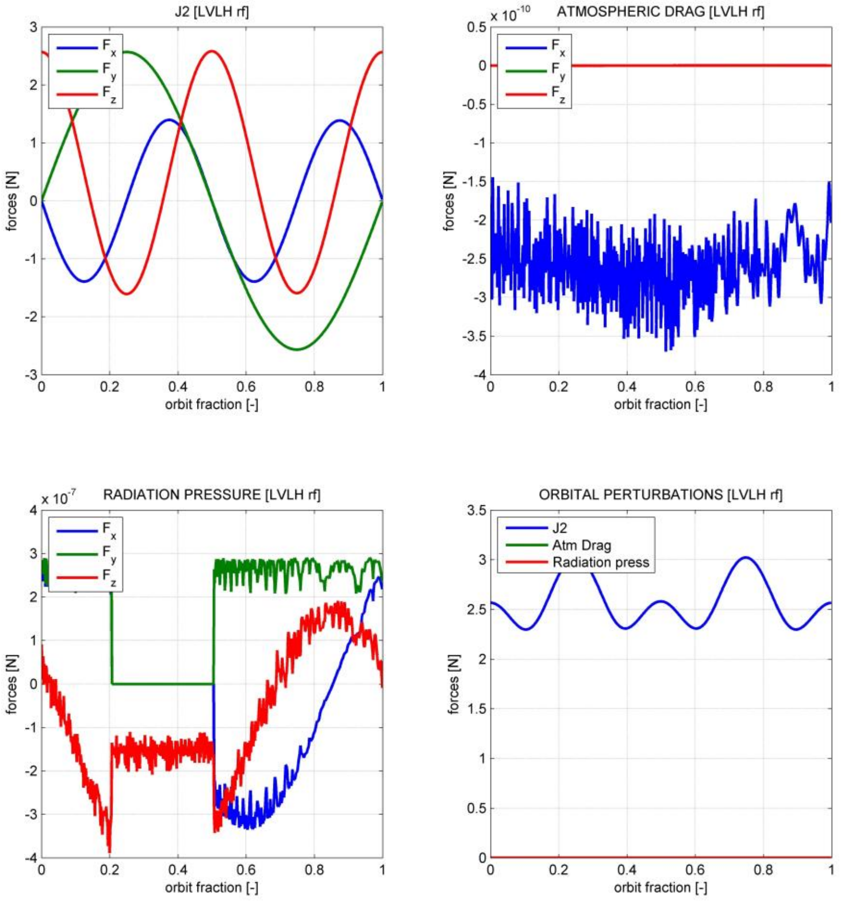


Figure 6.4: analysis of the orbital perturbation forces affecting an inertial pointing body, orbiting on an equatorial LEO at 500 km altitude.

At the effects, we note the predominant perturbation is the *J2*, being 7 to 10 orders of magnitude higher than all the others in the orbital scenario. Atmospheric drag and radiation pressure are computed as the resultant of the forces acting on each surface, therefore they contribute to the attitude perturbation too, generating the perturbation torques. The last remaining perturbation affecting the attitude is the *gravity gradient* torque, as it is directly retrieved from the inertia moments. Checks can be performed analyzing the configuration for which the gravity gradient stabilizes the system. We assume:

- Earth pointing attitude;
- Maximum inertia along the R-bar $(I_z > I_y > I_x)$.

From Fig. 6.5a we verify the stable oscillation of the R-bar in **BODY-PI** frame. Finally we can further analyze the limit case for which $I_x \approx 0$ and $I_y = I_z$.

Analytically solved, the perturbation analysis gives the following results on the direction cosines:

$$\boldsymbol{c} = \begin{cases} 1 \\ -\alpha_z \\ \alpha_y \end{cases} \tag{6.10}$$

Where α_y and α_z have a sinusoidal behavior with $f \cong 1/2T_o$ (Fig. 6.5b).

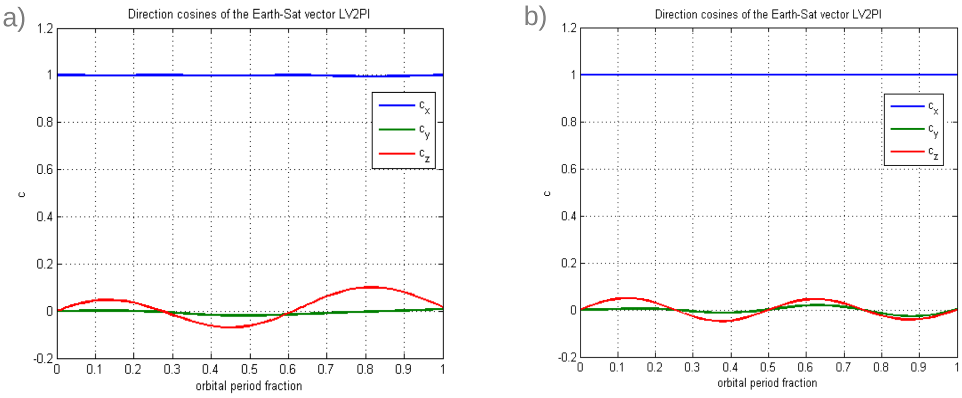


Figure 6.5: validation of the gravity gradient perturbation: Earth-pointing configuration.

Finally, we have that for ground tests the only perturbation worth considering is the atmospheric drag, affecting Net and Bullets. The proof of its importance on ground, rather than in orbit, is clearly shown in Fig. 6.6. The example reported refers to a [3x3] m Planar Net, shot at 3m/s in the two different environments.

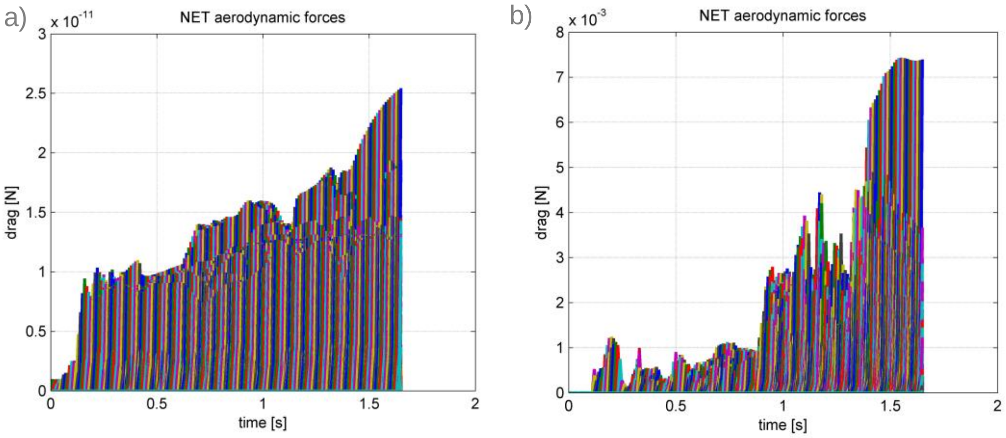


Figure 6.6: aerodynamic drag force magnitude, acting on the Net threads in the two scenarios: a) in orbit and b) on ground.

We appreciate a difference of 8 orders of magnitude. However, although each single thread is still affected by really low values *o(10^{-3})* N, this perturbation cannot be neglected. The reason is that these low values refer to a single thread and a real Net is made of thousands of links, so at the effect the overall disturbing action of the atmospheric drag reaches magnitudes of *o(10^{0})* N.
The same consideration between the two environmental scenarios can be done for the Bullets. In particular, differently from the Net threads, each Bullet is subjected to 3 magnitude orders higher forces (Fig. 6.7).

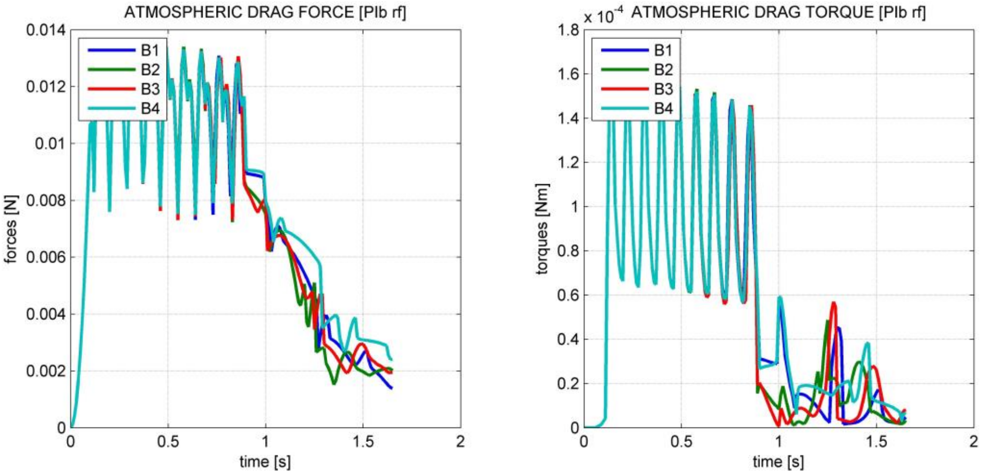


Figure 6.7: estimated aerodynamic forces and torques acting on four cylindrical Bullets shot on ground test.

6.2.3 Net Threads

A particular innovative feature of the Net threads modelling, is the possibility to have failures, meaning they break once reached their ultimate stress, without stopping the simulation and being entirely consistent with the viscoelastic constitutive law. Setting a breaking stress of 1.6 GPa, a benchmark simulation is run to test the effectivity of this further dynamics feature. A 80° divergence shot at 10m/s is simulated on a [2x2] m planar Net, with 20kg Bullets.

The arising stresses are both dynamically shown on the threads during the simulation (Fig. 6.8) and reported at the end (Fig. 6.9). From Fig. 6.8 we verify that the dynamics correctly propagates after the failure of most threads and all the bullet Links, which lets us to simulate not only the failures, but also their sensitivity on the whole capture dynamics. They surely would not occur during the deployment, but may happen after the wrapping of the Net on the Target, whenever sharp edges are present.

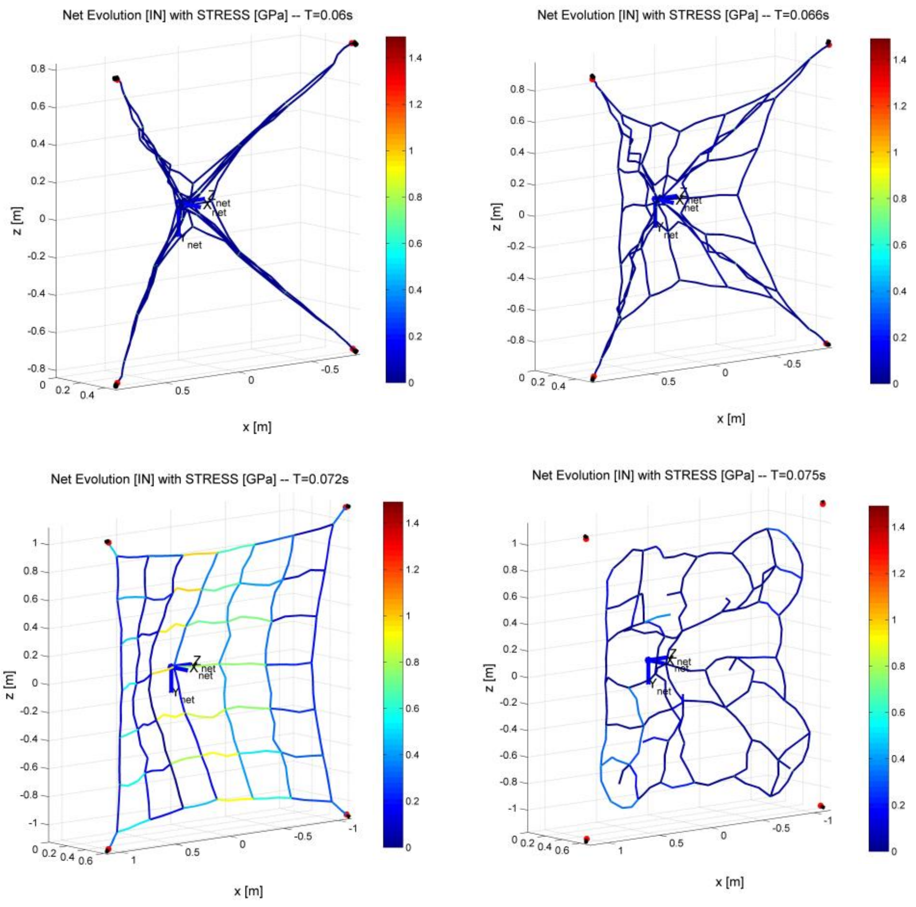


Figure 6.8: threads failure, benchmark test.

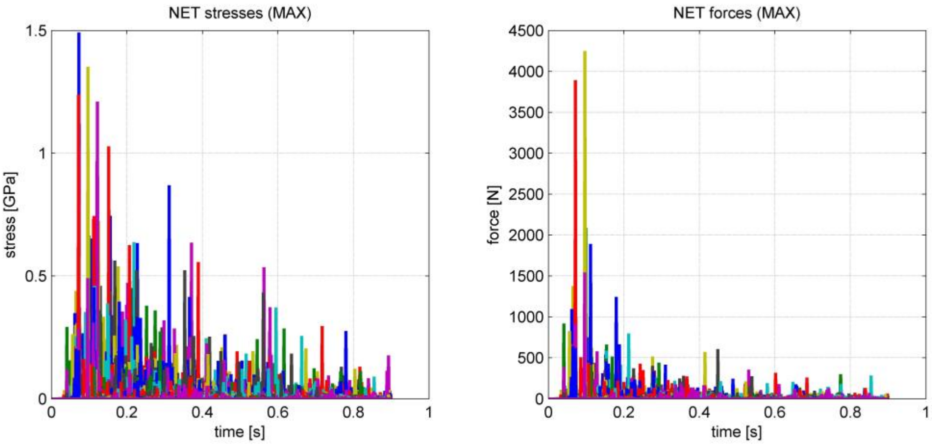


Figure 6.9: maximum traction stresses and forces acting on the Net threads during the benchmark test.

From Fig. 6.9 we verify that all the maximum stresses keep below the breaking value. The number of failures with time is post-processed and it is further possible to identify the exact ID of the broken threads (Fig. 6.10).

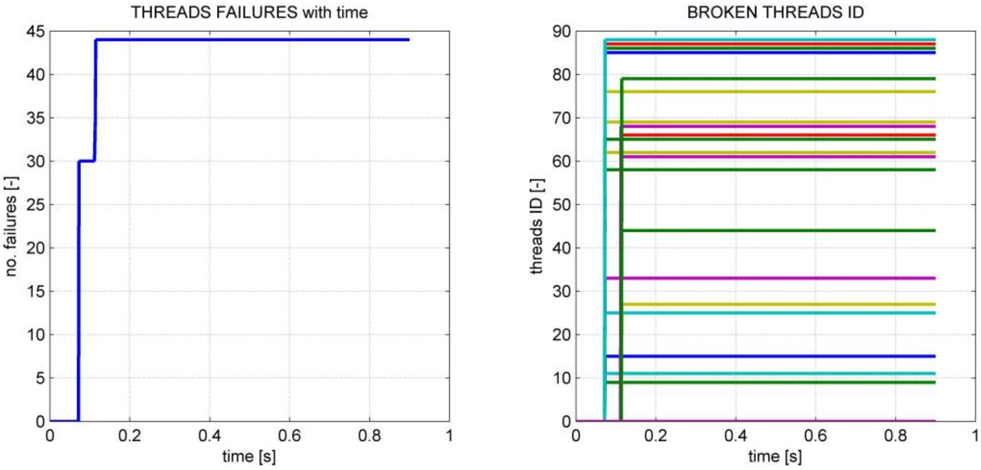


Figure 6.10: threads failures analysis: number of broken threads with time and ID of the corresponding broken threads.

6.2.4 Shooting System

An asynchronous shooting is simulated in orbit for the numerical validation. Four Bullets of mass $m_{bullet} = 0.3\ kg$ are shot with a delay of [0.5 0.5 0.7 0.9] s, imposing a shooting velocity $v_{shot} = 1m/s$ and a divergence angle $\alpha = 30°$.

We want to verify that the effective shooting velocity is coincident with the one set in input, since this last is not imposed directly as the initial condition, but physically induced by the corresponding force, considering the gas expansion of the CGG system (Section 2.7). This check is reported in Fig. 6.11 and confirms the aforementioned desired behavior. Moreover, besides the delay between the shoots, we can note that the final velocities of the Bullets are not impulsive but linear, thanks to the choice of dealing the shots with forces instead of velocities.

We check then the effects of the Bullets shots on the Chaser. We do not only expect to have, as reaction, the sum of the shooting forces acting on the Bullets, but also a tumbling perturbation due to the asynchronous shots.
The effects are the decrease of the linear orbital momentum of the Chaser coherently with the shots, and the arising of a non-null angular momentum in the attitude dynamics (Fig. 6.12).

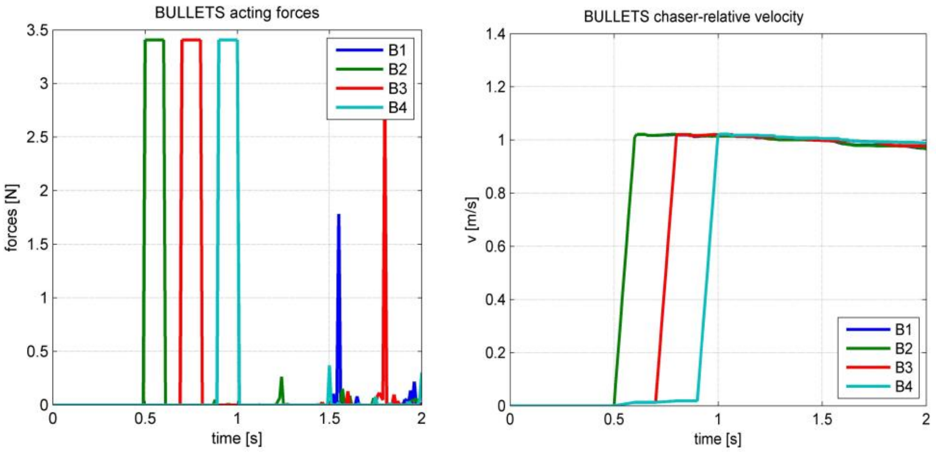


Figure 6.11: asynchronous shot of four Bullets: shooting forces and resultant axial velocity of the Bullets.

We definitely validate the Shooting System, proving that the overall impulse given to the Bullets equals the Chaser loss and we further obtain the Δv related:

$$\Delta v_{CHASER} = \frac{4\, m_{bullet}\, v_{shot} \cos(\alpha)}{M_{CHASER}} \tag{6.11}$$

Having $M_{CHASER} = 50kg$ in our shooting test, we obtain $\Delta v_{CHASER} = 0.023m/s$.

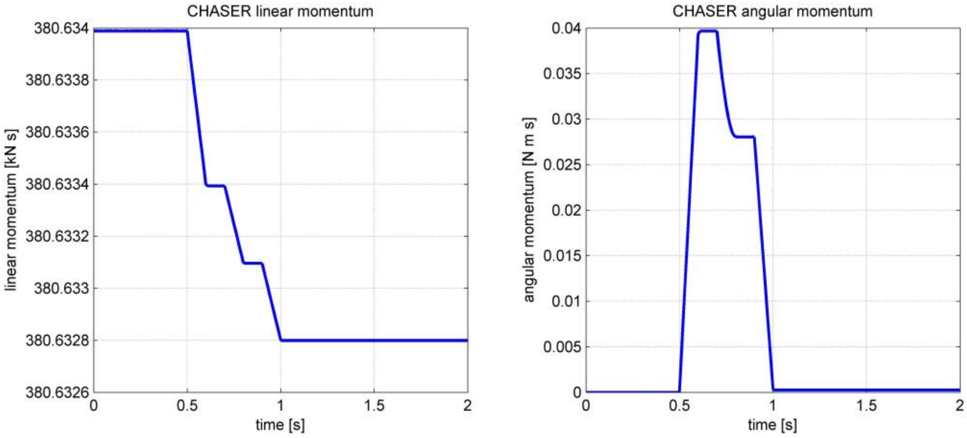


Figure 6.12: effects on the Chaser of an asynchronous shot of four Bullets: linear and angular momentum profiles.

6.2.5 Closing Mechanism

We recall that the closing law is applied to the nominal length of the Closing Links, being linear with time according to the spring winch model. Testing a [2x2]m planar Net and imposing a 99% closure of the Net-mouth in 1s, we verify the modelled closing dynamics.

We check that the final lengths are coherent with the Net-mouth reduction imposed (according to Eq. 2.80) noting that the Closing Links are correctly reduced by a factor of 10. At the same time, we verify that, as the winding process is engaged and the yarn is being collected on the reel mechanism, the mass of the winch effectively increases by the exact amount of the winded thread (90% of the 2m Link mass).

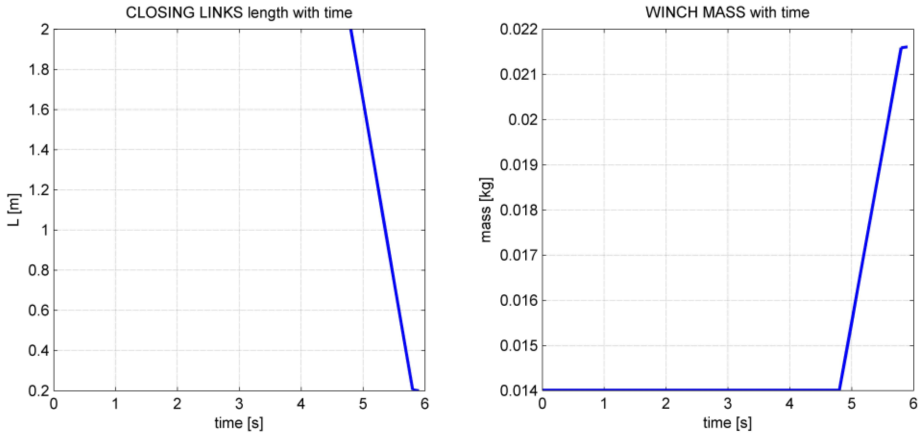


Figure 6.13: closure system validation: length profile of Closing Links and mass profile of the Winches (assumed massless at the beginning, the initial mass is the knot one).

We evaluate as well the stresses and the forces acting on each thread element of the Closing Links, in order to retrieve useful information for the winch sizing (Fig. 6.14). We observe that the Closing Links, as they run free adjacent to the outer Net-mouth perimeter, are not stressed and do not actively participate in the Net deployment until the closure is activated.

In order to provide an estimation of the winding force required by the winch, we post-process the stresses of all the Closing Links and identify at each time instant the equivalent axial force acting. In this way, we are able to define not only the force magnitude for the winch sizing process, but also the time profile required, providing important requirements also for the type of the spring to choose.

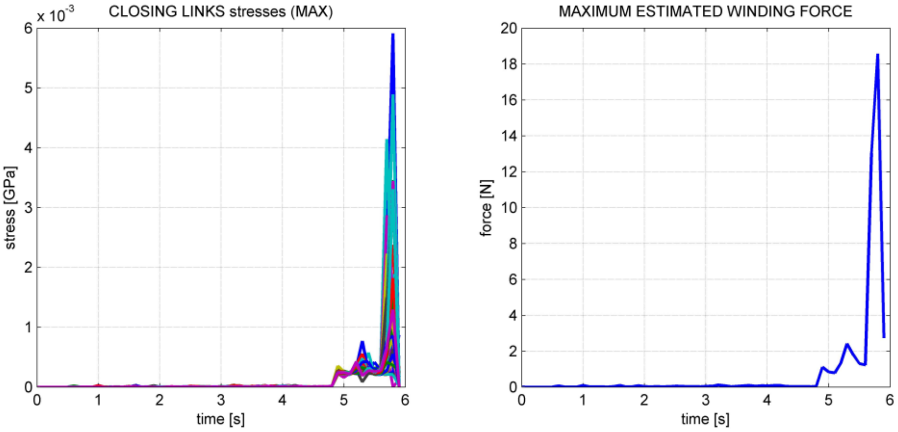


Figure 6.14: stress profile of all the Closing Links thread elements and the estimated winding force required.

6.2.6 Overall Dynamics Validation

Having the contact model been already validated aside in Chapter 4, we check the overall system dynamics, taking into account the interaction between all the dynamics described, modelled and individually validated.

To do so, we exploit the energy and momentum conservation principles, applied to a closed system. We recall the momentum conservation equation introduced before in Eq. 6.3, computing the overall momentum considering both the attitude and the orbital motion:

$$\boldsymbol{\Gamma_{TOT}} = \sum_{\boldsymbol{i}} \left[\boldsymbol{\Gamma}_{ATT,i}\right]_{ECI} + m_i\, \boldsymbol{h}_{ORB,i} \tag{6.12}$$

And we reformulate the total energy expression introduced in Eq. 6.5, to take into account all the contributions:

$$E_{TOT} = \sum_i E_{K,i} + E_{K-ROT,i} + U_i + E_{S,i} - L_{NC} + L_{EXT} \tag{6.13}$$

Where we further introduce:

- The elastic energy of the Net and Tether threads, so defined:

$$E_S = \frac{1}{2}\frac{EA}{L_0}(L - L_0) \tag{6.14}$$

- The contribution of all the non conservative perturbations L_{NC}, including the dissipation energy due to the damping and friction models, both in the threads constitutive law and in the contact.

- The contribution L_{EXT} due to external actions, such as the Net shot and closure and the disposal thrusting.

In order to validate the overall dynamics, through the energy continuity equation (Eq. 6.4) and the momentum conservation principle (Eq. 6.3), we set to *zero* all the dissipation sources and the environment perturbations. We simulate the entire capture dynamics, from the Net shot to the Chaser thrusting (Fig. 6.15). The thrusting is imposed to be continuous and lasting 5s soon after the Net closure.

We note a bouncing behaviour of the Chaser towards the Target during the thrusting phase and this is mostly evident since we have avoided all the dissipations, turning the Tether perfectly elastic. The elastic energy profile is reported in Fig. 6.16a, where we clearly identify the Chaser effective traction from the peak in the Tether elongation. In Fig. 6.16b we report the Chaser total energy, where we can appreciate the coherent constant trend between the shot and the thrusting phase (highlighted).

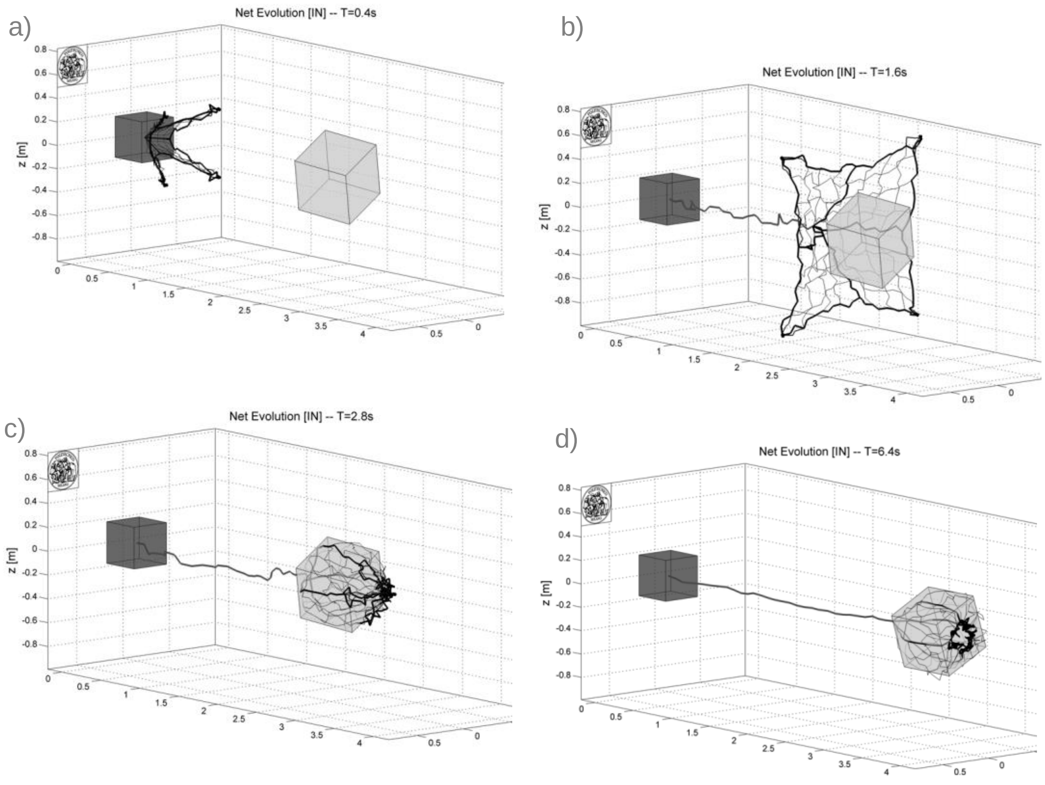


Figure 6.15: overall capture dynamics for the validation test: a) shot, b) impact, c) closure, d) thrusting.

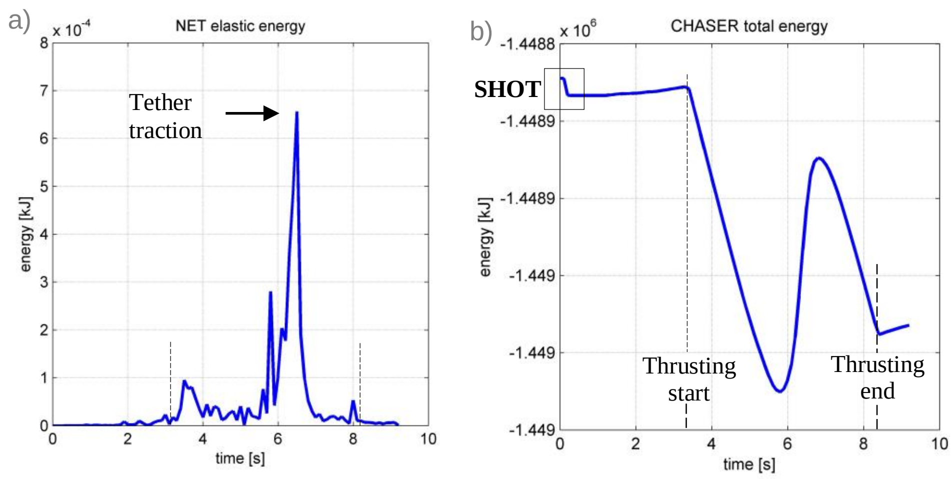


Figure 6.16: Net elastic energy and Chaser total energy profiles.

The overall dynamics is finally physically validated through Fig. 6.17, where we prove the overall momentum and energy conservation before and after the Chaser thrusting phase.

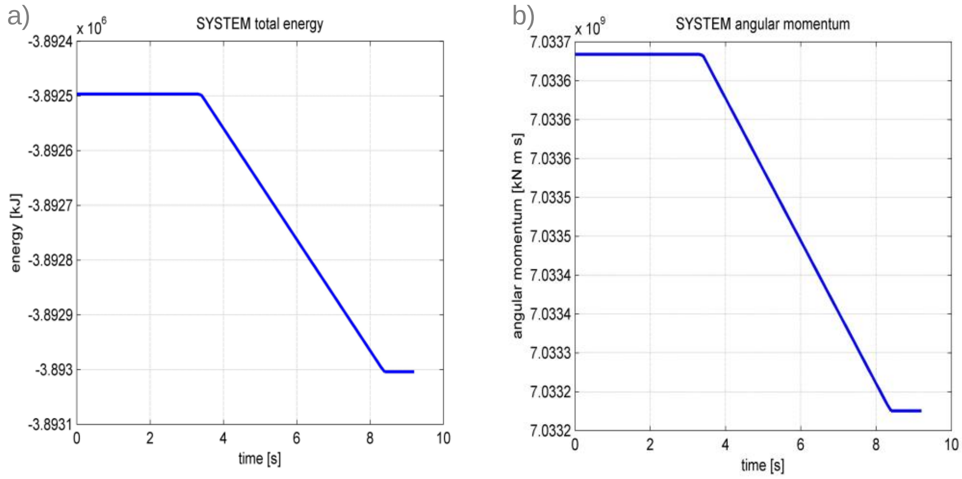


Figure 6.17: validation of the overall system dynamics through the energy (b) and momentum (a) conservation principles.

6.3 Parametric Sensitivity Analysis

We proceed with the analysis of the dynamics of the Net-based ADR system, focusing on two levels:

- Identify the *numerical parameters*, responsible for the quality and precision of the simulation results and their sensitivity on the dynamics;
- Characterize the *physical parameters*, proper of this system and analyze their sensitivity on the success of the capture.

We introduce the concept of *quality index* as the principal reference index to quantify the quality of the Net deployment. We define it assuming as reference scenario a perfectly deployed configuration, and computing the overall displacement error at each integration time step:

$$Q(t) \triangleq \left(1 - \frac{e(t)}{e_{ref}}\right) \cdot 100 \tag{6.15}$$

Where:

$$e(t) = \sum\nolimits_{i=1}^{N} \left\| \boldsymbol{x}_i^{ref} - \boldsymbol{x}_i(t) \right\| \tag{6.16}$$

$$e_{ref} = \sum\nolimits_{i=1}^{N} \left\| \boldsymbol{x}_i^{ref} \right\| \tag{6.17}$$

In order not to affect implicitly the quality index with the discretization level, the nodes considered are the only physical knot (whose number does not vary whichever the discretization). The position $\boldsymbol{x}_i(t)$ of the knots are computed in the same Inertial frame and relative to the time depending position of the *CoM* of the Net, so derived:

$$\boldsymbol{CoM}(t) = \frac{1}{m_{tot}} \sum\nolimits_{i=1}^{N} \boldsymbol{X}_i(t)\, m_i(t) \tag{6.18}$$

6.3.1 Parametric Characterization of the Net-System

It is necessary to remark that the success of a capture is directly related to the quality of the deployment at the impact. Important aspects that will be analyzed are the capture distance tolerance and the offsets, coming from the unbalances due to unpredicted failures. First, we identify the shooting parameters affecting directly the Net deployment and the capture quality. They all concern the Bullets and the Shooting System, in terms of:

- shooting velocity (v_b)
- divergence angle (α)
- number of Bullets and Bullet Ratio (Eq. 2.38)

Further sensitive parameters related to the Net are:

- shape (2D-3D)
- mesh size

The study is carried out varying each parameter once at a time, starting from the reference configurations in Tab. 6.1.

Table 6.1: reference Net configurations used for the analysis.

Type	Dimensions	Mesh	#Bullets	BR	Divergence	velocity
Planar4	[2 x 2] m	0.2 m	4	0.5	30 deg	1 m/s
Planar8	[2 x 2] m	0.2 m	8	0.5	30 deg	1 m/s
Conical8	[2 x 2 x 2] m	0.2 m	8	0.5	30 deg	1 m/s

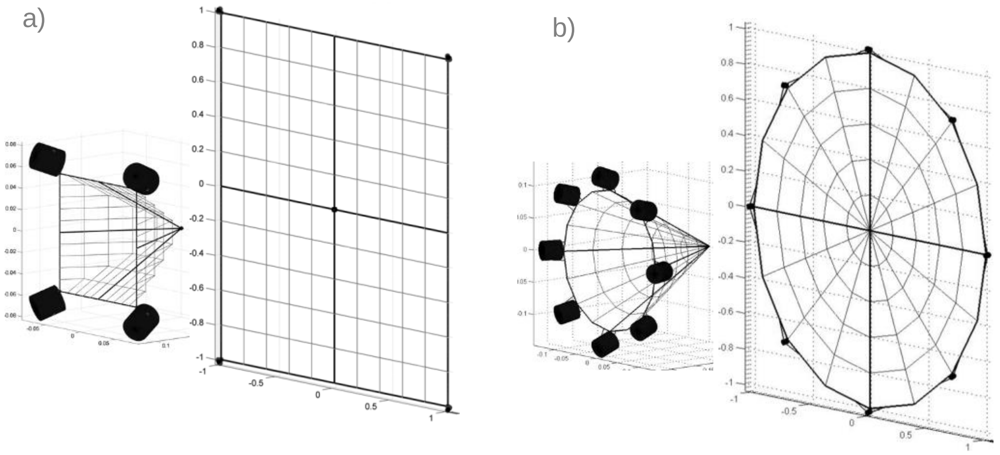


Figure 6.18: folded and deployed configurations of the reference Planar Nets, with a) 4 Bullets and b) 8 Bullets.

The first notable benefit coming from having more than 4 Bullets, is easily related to the geometrical aperture of the inlet cross-section area of the Net. This in particular has a strong influence on the 3D geometries as can be seen in Fig. 6.19. The importance of this feature is related to the enhancement of the capture success, decreasing the risk of missing capture: the higher the impact area, the lower is the sensitivity to unpredicted shooting failures or external perturbations.
The main advantage coming from the **3D-Net shapes**, is that they do not require any interaction with the Target in order to wrap and complete the closure.

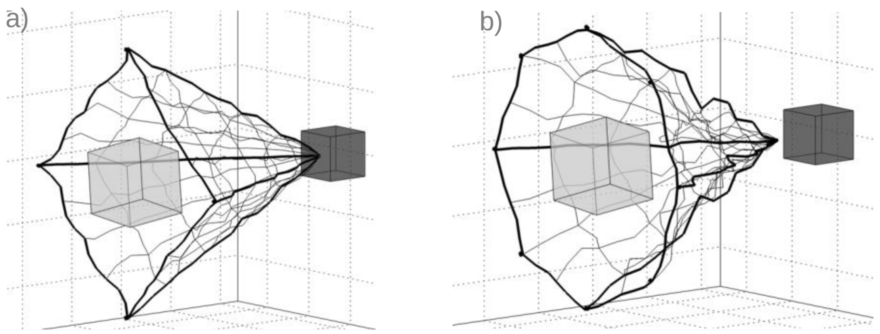


Figure 6.19: 3D conical Net geometry comparison: a) 4 Bullets, b) 8 Bullets.

The capture is ensured even at very low relative speeds and no calibrations are required in the case of unknown tumbling state of the Target. A further benefit is that these configurations let to minimize the capturing surface, saving mass. The only drawback lies in the manufacturing issue. Differently, the **planar geometries** are much easier to build, but require a direct interaction with the Target in order to wrap and close and are more sensitive to the aforementioned deployment parameters. For this reason they are here on characterized and compared, as they represent the most interesting configuration.

Generally, the capture distance could be estimated a-priori from geometrical properties, considering the divergence angle and the Net dimensions (Fig. 6.20).

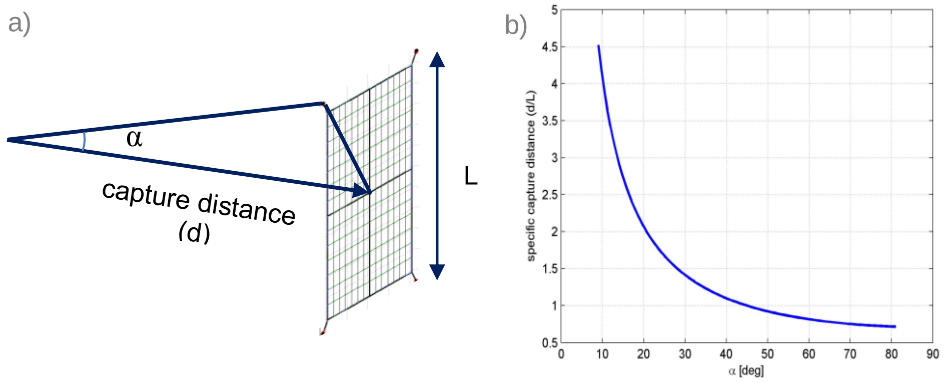


Figure 6.20: a) geometrical prediction of the capture distance. b) Specific capture distance (d/L) vs. divergence angle α.

The capture time is then trivially derived once estimated the velocity of the Net CoM, obtained from the momentum balance:

$$v_{CoM,x} = BR \cdot v_b \cdot \cos\alpha \qquad (6.19)$$

These simplified models are good enough to approximate the optimum capture instant. However, a better and more rigorous analysis can be done and it is discussed now.

Reporting in Fig. 6.21 the Q index trend with time (for the reference planar Net configuration in Table 6.1), we identify the exact capture distance as the one corresponding to the best deployment. Fixing a **70%** threshold on the deployment quality, we can define a tolerance on that capture distance, ensuring a quality at impact greater than the minimum limit imposed.

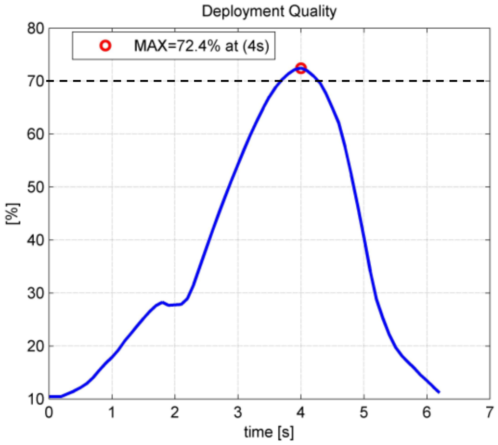


Figure 6.21: quality index trend with time, relative to the deployment of the Planar4 Net. Real capture distance and tolerance are retrieved from it.

The sensitivity analyses performed, take also into account inter-dependencies varying the shooting parameters for different meshes. The higher the mesh number means the higher number of strings and rings, so the lower are the lengths of the threads of the mesh. We see that the higher is the mesh, the better is the deployment in all the possible combinations and scenarios. Different behaviours characterize instead the dependency from the Bullet Ratio, the divergence angle and the shooting velocity.

For what concerns the Bullets ejection velocity, we do not record any influence on the deployment quality (Fig.6.22a). So we can cleverly exploit this parameter to tune the time scale of impact and wrapping. As a check we report in Fig.6.22b a prediction model on the wrapping time, compared with the simulation results:

$$\Delta t_{wrapping} = \frac{\pi \cdot R}{2 \cdot v_{CoM,x}} \tag{6.20}$$

Being R the half difference between the Net size length and the Target impact face size.

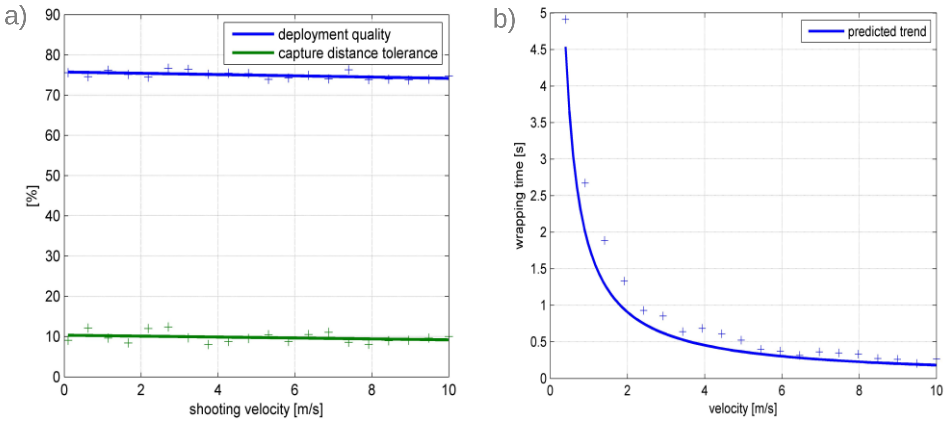


Figure 6.22: a) Bullet ejection velocity sensitivity analysis on the deployment quality and b) wrapping time estimation and real times.

Dealing with the *BR* sensitivity, we can see from Fig. 6.23a that the quality trend follows the Bullet Ratio with no upper-bound limits. This parameter is particularly important to provide a higher deployment robustness to low velocities, meaning that the lower the BR, the higher is the sensitivity of the quality to the low velocities. Although to the monotonic behaviour, an advisable value is to be found looking at the tolerance on the capture distance in Fig. 6.23b, choosing the one that ensures the maximum.

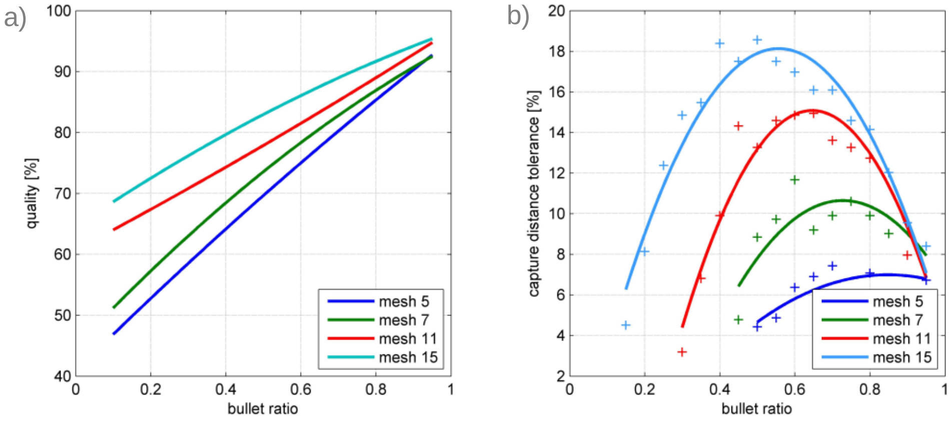


Figure 6.23: a) Bullet Ration sensitivity analysis on the deployment quality and b) capture distance tolerance (70% threshold).

Differently, the divergence angle has both an optimum value for the quality and the distance tolerance. From Fig. 6.24 we see that they are almost the same, being around 40 deg for all the mesh types. Boundaries on the divergence angles translates into limitations on the possible capture distances (Fig. 6.20b).

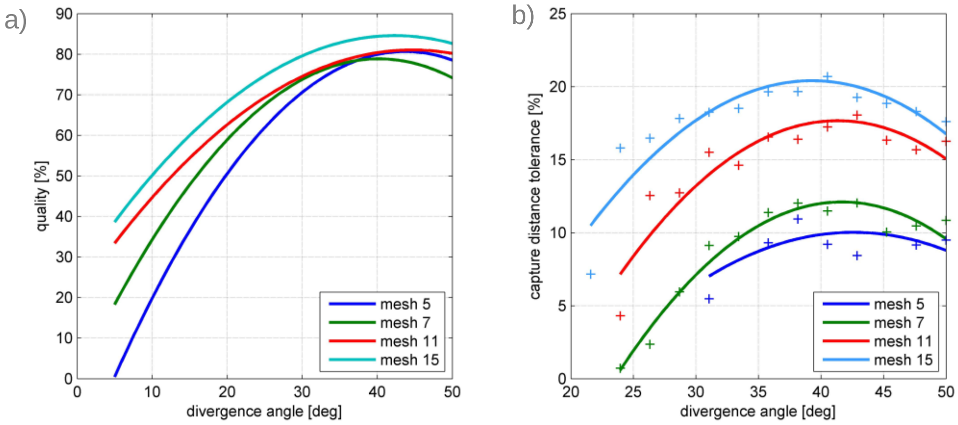


Figure 6.24: a) divergence angle sensitivity analysis on the deployment quality and b) capture distance tolerance (70% threshold).

Finally, we better analyse why high mesh levels let to have a better deployment. We recall that high mesh levels imply that, keeping the number of discretizing nodes per Link, the threads are much finer discretised. Therefore, focusing on the discretization level, we study its influence on the quality index for each mesh (Fig. 6.25), in terms of the percentage portion $l_{e,\%}$ of the element threads with respect to each Link length. The implication is:

➔ the higher is the mesh and the number of discretization nodes, the nearer we are to the exact continuous model.

So, we can see this quality gap (mesh fixed) as an index of the error of the discrete multi-body modelling.

Merging the discretization influence (mathematical model) with the mesh one (geometry), we define as sensitivity index, the ratio between the discretization length l_e and the overall Net size L:

$$L_{e,\%} = \frac{l_e}{L} \tag{6.21}$$

We conclude that the mesh influences the quality in two ways:

- from the mathematical one (discretization level)
- from the geometrical one (Link length ratio with to the Net size).

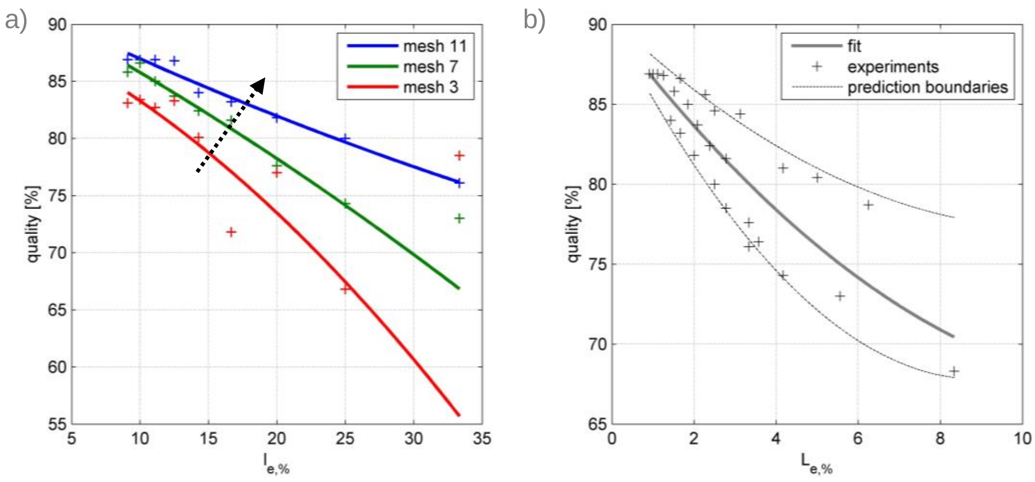


Figure 6.25: a) Link discretization and b) $L_{e,\%}$ sensitivity analysis on the deployment quality.

Merging these two effects, we have the trend in Fig. 6.25b, noting in particular the convergence of the data dispersion towards the fit curve at low $L_{e,\%}$ values, due to discretization error reduction.

6.3.2 Uncertainties Sensitivity

The Net deployment relies on the symmetry of the shooting, in terms of equal:

- Bullets mass $\bar{m}$;
- Divergence angle $\bar{\alpha}$;
- Shooting velocity $\bar{v}_b$;
- Shooting instant.

Therefore, an interesting and useful analysis consists in characterizing the system in terms of its robustness to unbalances and uncertainties of these parameters, introducing a stochastic relative error on each (10% to 90%). The shooting delay ratio considers, as reference time, the one required by a single Bullet to reach the Net length. Higher delays define a failure and are treated in the next section.

In Fig. 6.26 and Fig. 6.27 we report the analysis performed on different Net geometries (2D-3D) and Bullet numbers. The highest robustness belongs, in the order, to Concal8 and Planar8, keeping high quality values and small sensitivity to uncertainties. In particular from the divergence angle error and shooting delay. A higher influence have the Bullet mass and velocity differences.

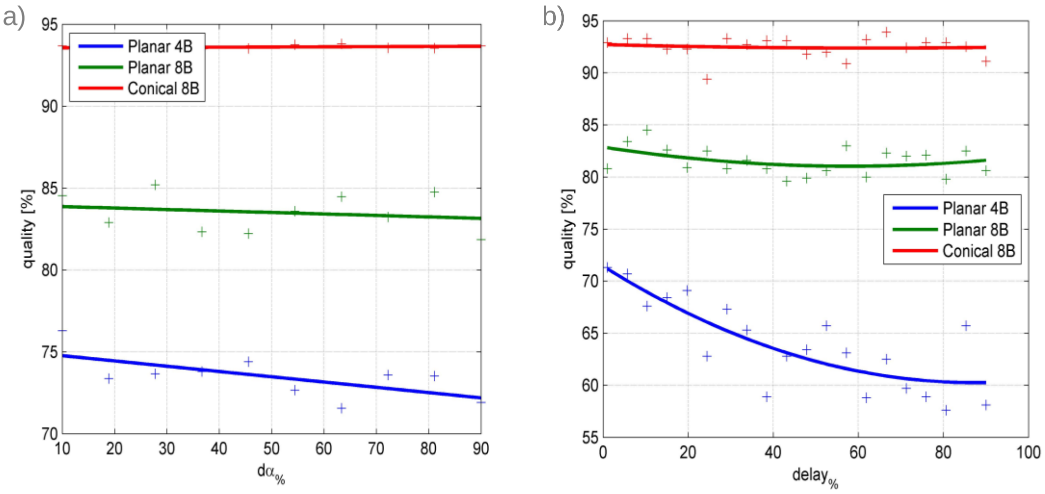


Figure 6.26: quality sensitivity to uncertainties on: a) divergence angle, b) shooting synchronization.

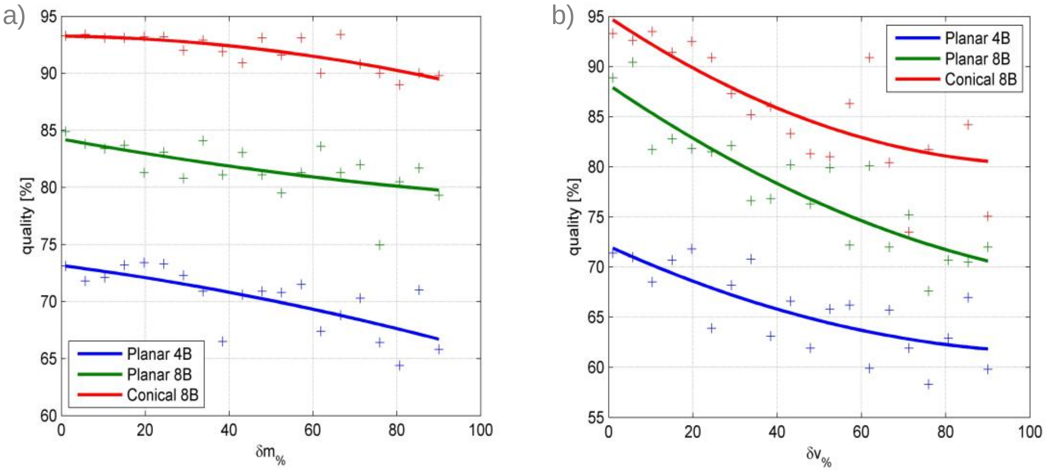


Figure 6.27: quality sensitivity to uncertainties on: a) Bullet mass, b) shooting velocity.

Being the Planar4 the most sensitive, a more detailed analysis is conducted on the longitudinal and transversal deviations. From Fig. 6.28a, we note that the reduction of the capture distance is governed by the mass deltas, while in Fig. 6.28b we see that the higher offset sensitivity is due to the shooting velocity unbalance. For this reason we appreciate shooting configurations that ensure highest capture distance tolerance, because in this way we are able to guarantee higher robustness of the system with respect to uncertainties.

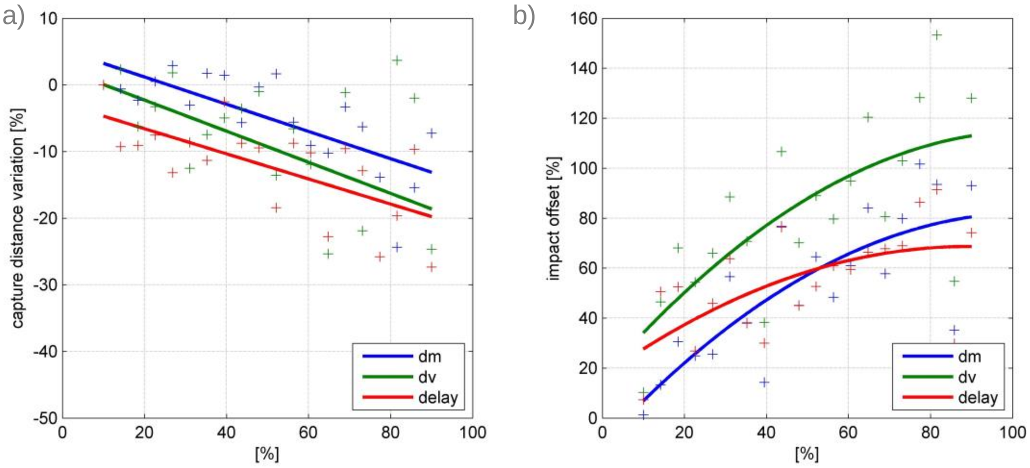


Figure 6.28: sensitivity of uncertainties on Bullets mass, velocity and synchronization, reflecting on: a) capture distance and b) lateral deviation at impact. Planar4 configuration.

A final sensitivity characterization is the one related to the folding pattern. Performing multiple simulations with Planar4 Net (most sensitive) on three different patterns, we do not find any substantial influence on the deployment quality, generalizing this robustness to all the configurations. The folding pattern analysed are shown in Fig. 6.29, starting from the most regular (pyramidal folding inside the canister) to the completely random folding. Quality differences registered do not exceed 5%.

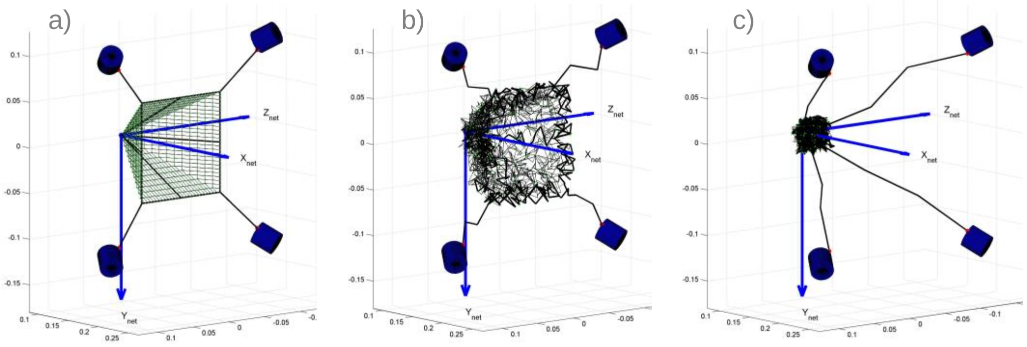


Figure 6.29: folding patterns: a) regular pyramidal, b) 50% random, (c) 100% random.

6.3.3 Failures Sensitivity

Two types of failures, corresponding to the 100% error respectively on the *velocity* and *mass*, define the extreme particular cases of the most critical unbalances previous analysed.

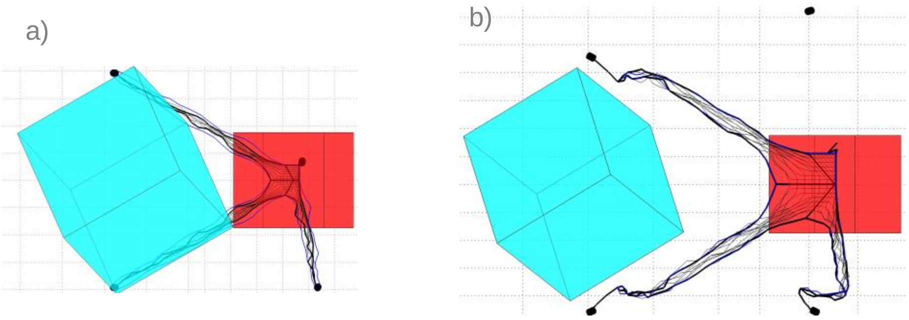


Figure 6.30: Bullet failures: a) shot miss and b) Link breakage.

These translate into:

- Bullet shooting failure;
- Bullet loss.

The single Bullet shooting failure involves not only an unbalanced shot, but also an inert mass to be dragged by the rest of the Net. This issue has the positive effect to still deploy the Net within a 20% margin on the capture distance (Fig. 1.30b), but the lateral deviation is so high that compromises the entire capture. The second type of failure is still worse (Fig. 1.31). Although it lets to have a lower lateral offset at impact (still unacceptable), the capture distance is drastically reduced (150%) in a way that cannot be covered by the available tolerances.

These issues are counteracted introducing more Bullets, hence reducing the single failure impact on the overall system. Analyses show that in order to ensure the capture, the shooting parameters should be set in order to grant a capture distance 40% higher and the Net dimensions should be 40% greater to face the lateral deviations. But according to Fig. 1.20, this last requirement involves itself an increase of the capture distance of 20%, therefore, the margin to apply to the capture distance reduces to 20%.

It is further possible to cleverly select shooting parameters in order to have a capture distance tolerance higher than ±20%. In this way, if a Bullet failure happens, the quality at the impact is the equivalent of having the target anticipated by 20% on the distance, so still in the quality tolerance.

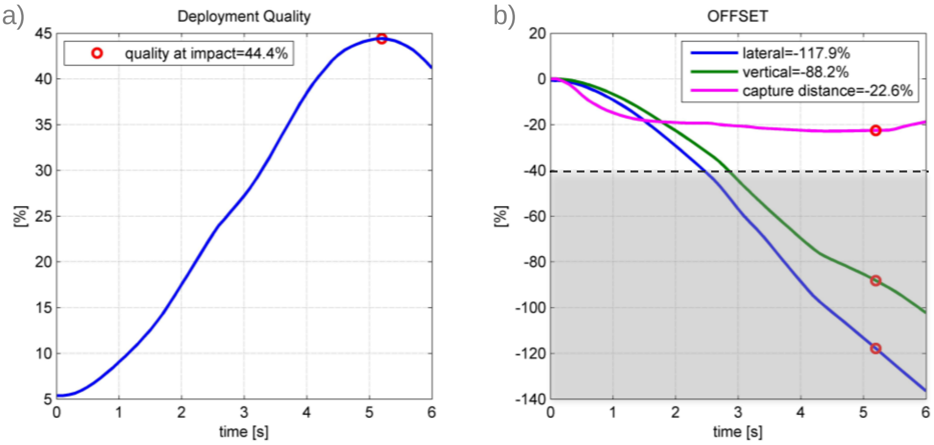


Figure 6.31: Planar4, single Bullet shot failure: 1a) deployment quality and b) deviations from the nominal shot without failures.

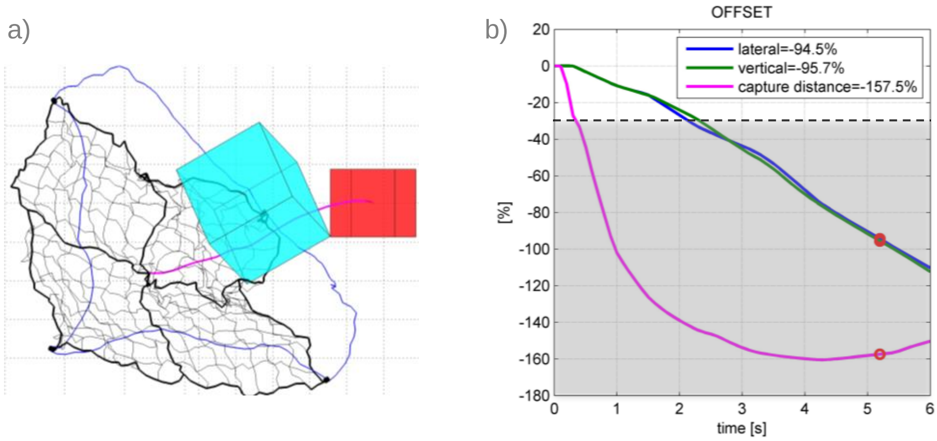


Figure 6.32: Planar4, single Bullet loss: a) impact instant and b) deviations from the nominal shot without failures.

It is important to remark that a complete failure involves not only a deviated trajectory, but a complete capture miss. For this reason, all the aforementioned analyses take into account also an active closure mechanism. We note that at the effects the closure is not enough to prevent the capture missing the first two cases (4 Bullets), whichever the closing type (both *free* and *interlaced* were simulated). Differently happens in the case of 8 Bullets, we can see from Fig. 1.32 that the closure enhances the wrapping although the Net deployment is not symmetric due to the Bullet failure.

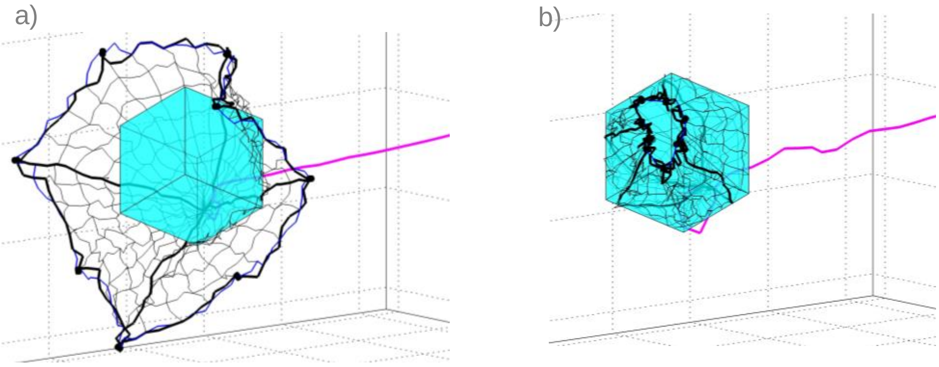


Figure 6.33: Planar8, single Bullet shot failure: a) impact instant, b) capture end.

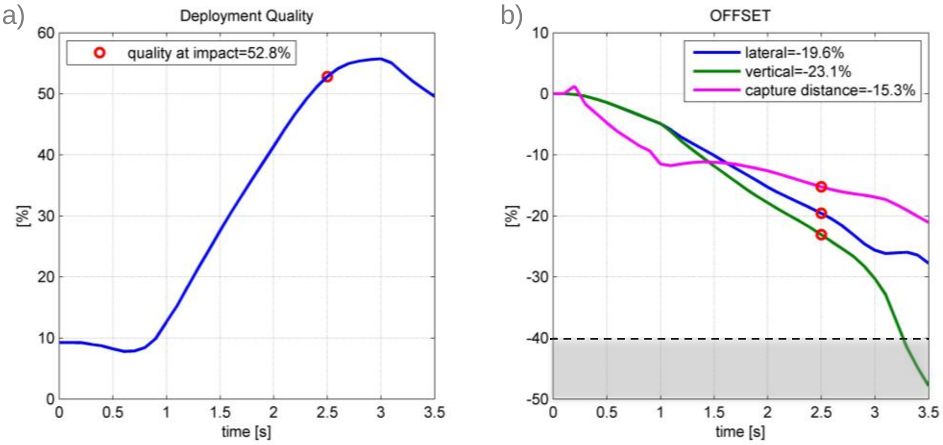


Figure 6.34: Planar8, single Bullet shot failure: a) deployment quality and b) deviations from the nominal shot without failures.

6.4 Ultimate Simulator Validation Test

Finally, we propose an ultimate validation test to appreciate the effectiveness and the benefits coming from the numerical tool developed so far (*NeST*).

The reference scenario considered, is the SoA simulator (*MUST*) that has been used up to now to simulate the tether-net system dynamics and control. It is a reduced model that avoids the complexity of the contact dynamics, fixing directly a discrete number of contact points on the Target. The Net is reduced to a number of Links equal to the number of the connection points, then linked to a single tether. To better simulate the traction dynamics, these points are placed rear the CoM of the object.

This simplified numerical tool has been built up specifically to design the disposal control law, since running much faster. The entire capture dynamics cannot be simulated and the simulation starts directly with the disposal traction (tether already tensioned).
Perturbations are avoided in order to focalize the attention totally on the flexible model and the contact interface. The Net exploited is a 2m *PlanarA*. The Target considered is a 05m dimension cuboid of 83kg, tumbling at [1 -5 1]deg/s. The thrusting law considered is a single continuous burn of 15N for 300s.

Here we report the *MUST* configuration and the corresponding simulation results:

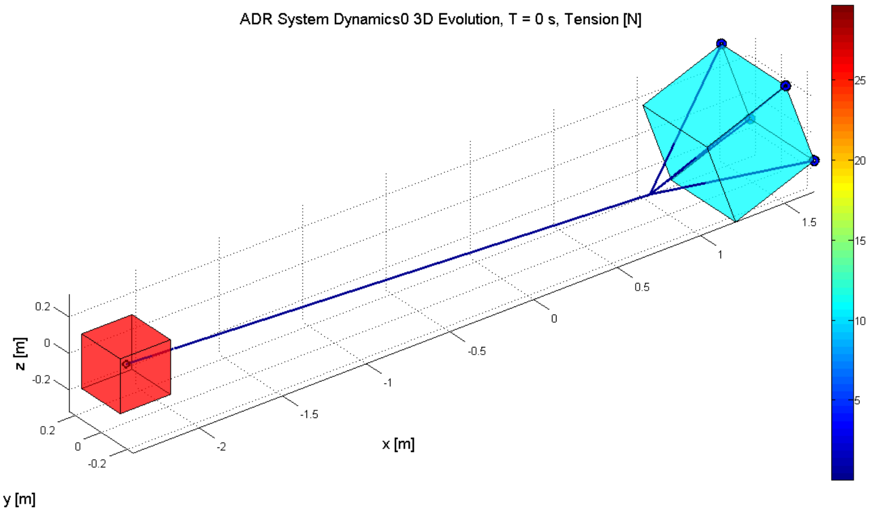


Figure 6.35: initial configuration of the reduced model (*MUST*)

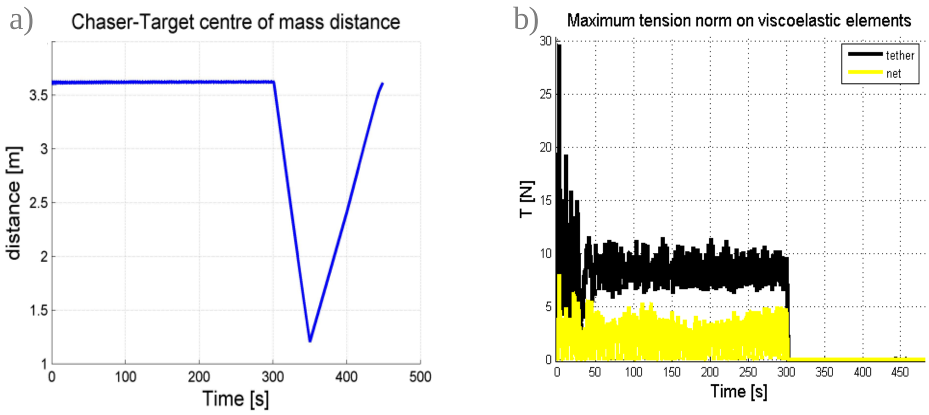


Figure 6.36: *MUST* simulation results: a) Target relative distance, b) maximum net and tether tensions.

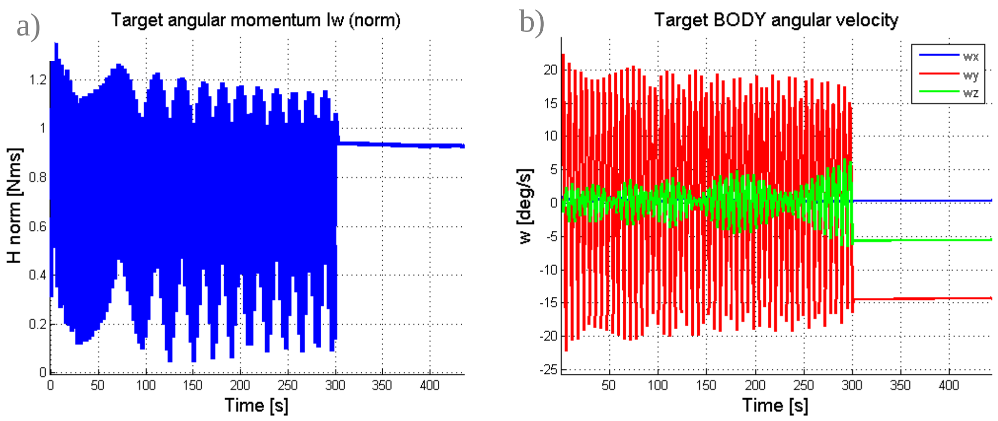


Figure 6.37: *MUST* simulation results: a) Target angular momentum modulus, b) Target angular velocity.

As we can see:

- We have a really modest initial bouncing dynamics, due to the simplified initial conditions and missing capture phase.
- Capture distance keeps constant during all the thrusting phase.
- The angular momentum starts oscillating around a stable configuration, without sensing any important damping effect.

More accurate and reliable dynamics results are obtained exploiting the complete numerical tool (*NeST*) developed so far in this thesis work. The thrusting phase is extended to 400s to appreciate the dynamics evolution.

Besides the initial bouncing of the Target towards the Chaser, involving stresses overshooting and oscillations, the final forces stabilize and the exact thrusting action can be clearly visible in the Tether transmitted force:

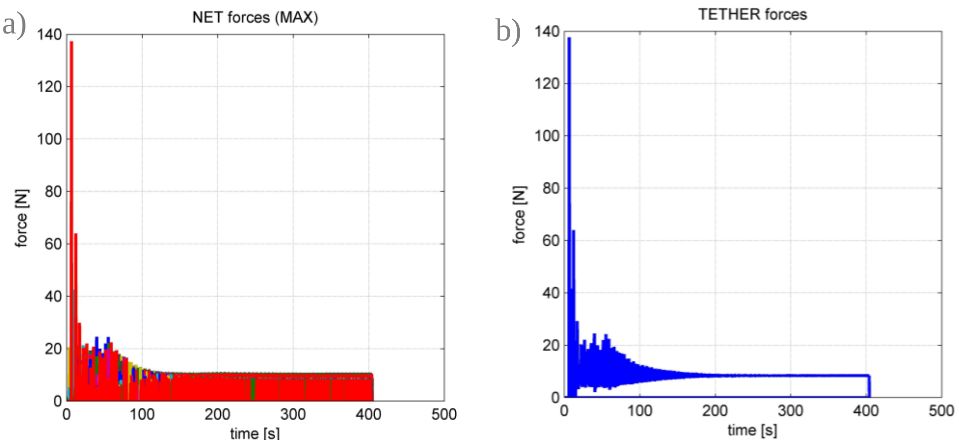


Figure 6.38: *NeST* simulation results: forces acting on a) Net and b) Tether during the capture and disposal phases.

The initial overshoot is due to the tether tensioning after the capture, which is a dynamics absent in *MUST*. However, we see that the initial bouncing, due to the initial shock, is then completely damped after 100s. This is notable in the relative distance and velocity components representation:

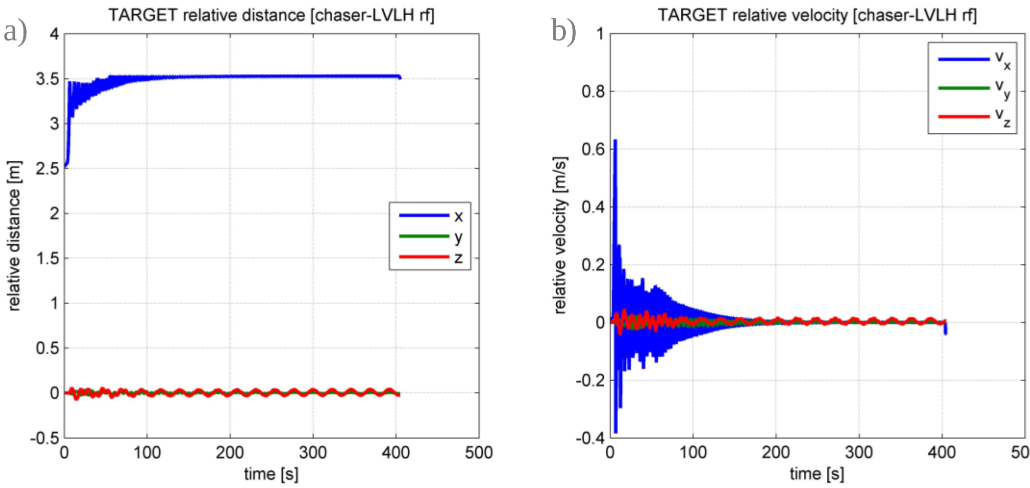


Figure 6.39: *NeST* simulation results: Target relative a) position and b) velocity.

We analyze the direct effect of the contact action on the Target in terms of forces and torques. We note coherently the exact transmitted traction along **V-bar** (F_x) and the reducing reaction with time due to the damped motion:

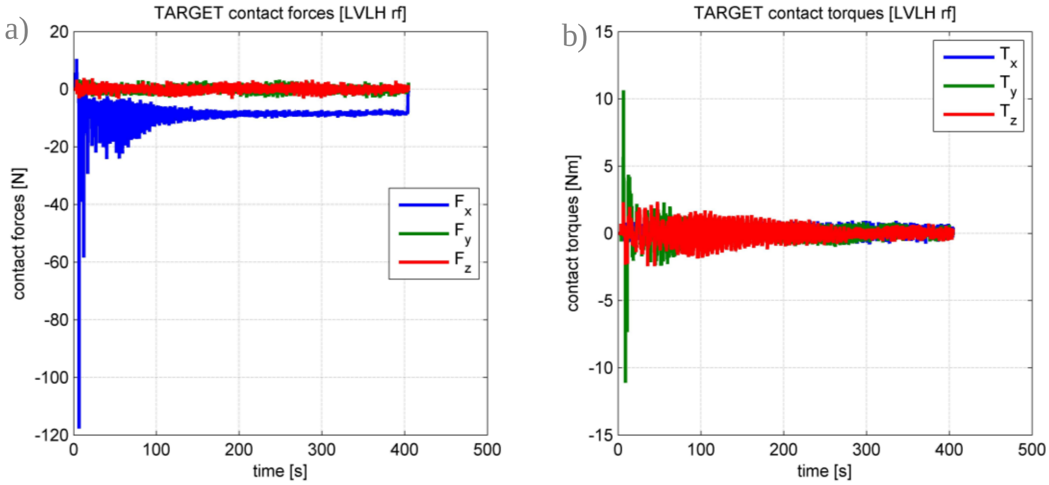


Figure 6.40: *NeST* simulation results: Target acting contact a) forces and b) torques.

Therefore we finally appreciate the contact model indirect effects, involving energy dissipation through slippage and friction. The initial angular momentum content and the further contribution due to the initial shock torques are clearly dissipated, proving the theorized effectiveness of the passive damping effect of the tether-Net concept. This could not be proved and precisely quantified up to now, demonstrating the importance of the implementation of the interfacing contact model for the evolution of the overall dynamics.

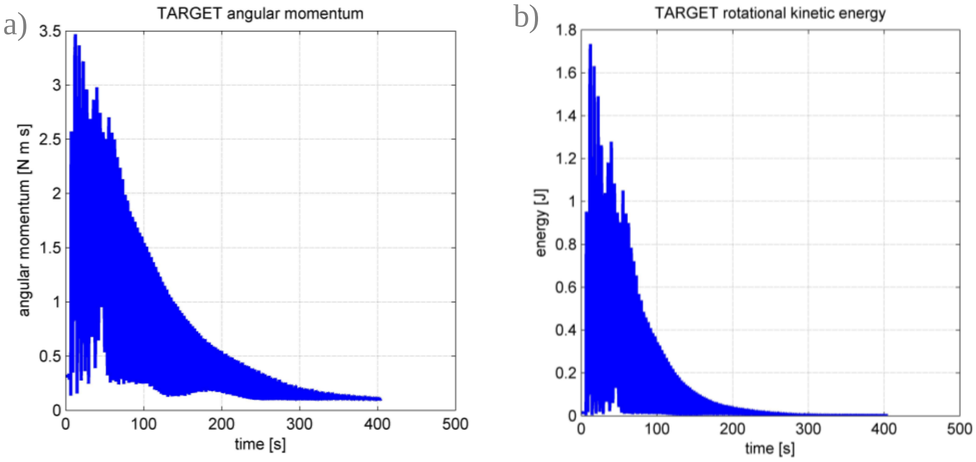


Figure 6.41: *NeST* simulation results: Target a) angular momentum and b) rotational kinetic energy.

Looking at the angular velocity of the Target we can clearly see the dynamic response of a damped system to a time step input, totally different from the previously predicted simulated dynamics.

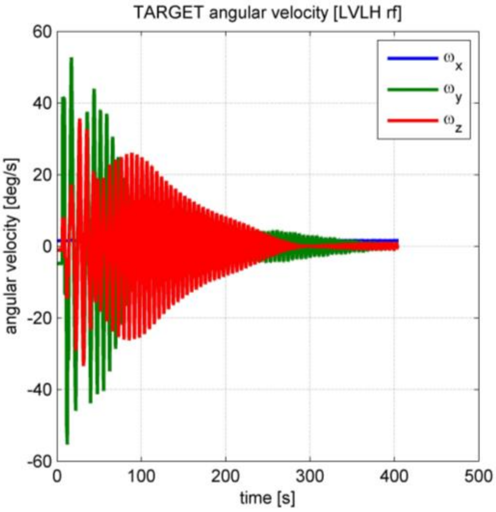


Figure 6.42: *NeST* simulation results: Target angular velocity trend with time. Equivalency with time step input dynamic response.

Here we report the last frame of the 400s of trusting simulated, showing a safe alignment of the Target, once damped all the longitudinal oscillations and dissipated the tumbling behavior.

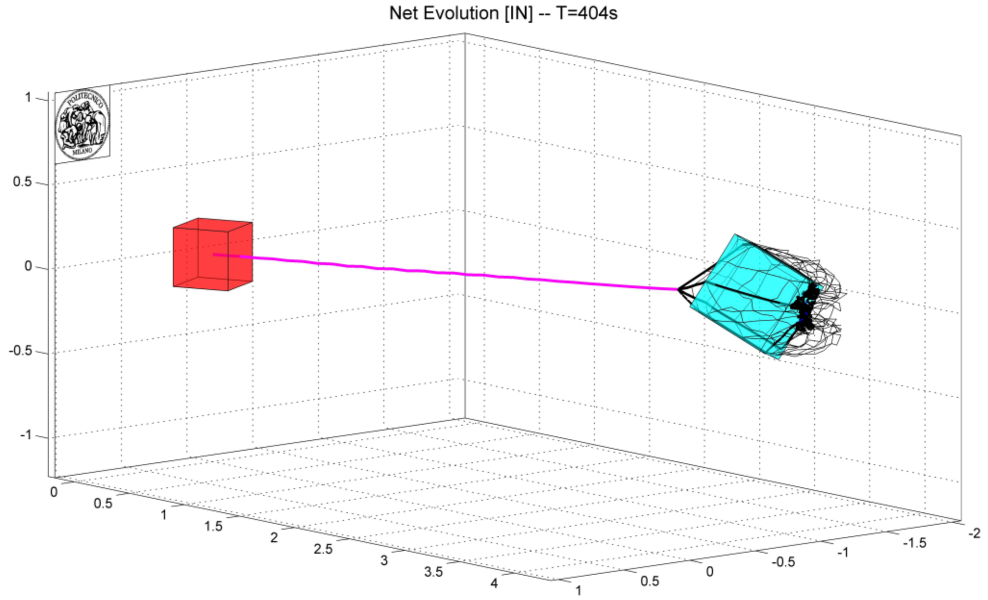


Figure 6.43: *NeST* simulation results: final time frame of the simulated dynamics, after the 400s of thrusting.

Chapter 7

CONCLUSIONS

7.1 Conclusions and Results

In this work, we designed a complete, physically based, mathematical model of a tether-Net Active space Debris Removal device. The main goal was to support the study of this flexible ADR concept in the international trade-off contest. It revealed to be a great tool to better learn, see and understand the overall behavior and the unforeseen dynamics arising from the interaction between two isolated bodies, becoming a single multibody system to deorbit. In particular, most of the attention was put on the interface between the Net and the free-tumbling object during the disposal pull, free to slip and not constrained a-priori. This could be studied only after modelling the *contact* and simulating the entire dynamics.

The Simulator design can be summed up in three phases:

- Modelling trade-off
- Mathematical and numerical optimization
- Validation and analysis

Most of the efforts was made in the identification of the bottlenecks of the model, in order to gain the best performances in terms of highest precision and quality of the results, minimizing the workload. From the numerical point of view, we focused on the *integration solver* and the *collision detection algorithm*. This last in particular was completely designed ad-hoc to satisfy the maximum performance and numerical robustness required. From the mathematical point of view, strong attention was put on the *viscoelastic models* of the flexible elements and the *contact* issues. The test campaign made for the characterization of the Net threads material, revealed to be decisive for the optimization of the model. This last was reformulated as a reduced parameters model, without loosing physical reliability and gaining decisively better computational performances.

A particular attention was also paid to minimize the set of parameters able to represent the physics of every phenomenon involved, among which the *contact* was the main topic. In this sense, we took into account as drivers for the design:

- The interfaceability of the numerical tool with external libraries and third-party software;
- A simplified and user-friendly approach for tuning the numerical and mathematical model parameters.

Besides the physical validation of the model, an experimental testing campaign in micro-gravity is planned. To this end, we designed that particular environment and we validated a reduced model for the experiment mock up. Simulations are advised before the tests for a rapid estimation of the Target positioning and orientation and after the 3D dynamics reconstruction, to retrieve and adjust all the parameters.

Important results were obtained thanks to this numerical tool, enabling a decisive quality and precision leap with respect to the reduced model tools exploited so far in the study of this particular and complex system. The fidelity to the physics of all the phenomena involved, paid back on the overall behavior at the effects.

Results shown, differently from the analyses carried out previously, the great role played by the contact modelling, proving energy dissipation reducing both:

- the axial bouncing dynamics of the Target during the pull;
- the angular momentum content of the Target.

These translate into benefits for the Net-device concept, since providing a passive damping aid. We have proved and we believe that the tether-Net ADR device has many promising chances to be finally exploited in real operative scenarios. This numerical tool provided not only some analyses supporting this thesis, but it plays a strategic role for the future analyses to come.

Thanks to this thesis work, two papers have been derived:

- R. Benvenuto, S. Salvi, M. Lavagna, *DYNAMICS ANALYSIS AND GNC DESIGN OF FLEXIBLE SYSTEMS FOR SPACE DEBRIS ACTIVE REMOVAL*, presented at IAA - DYCOSS (Rome, 2014) and published in ACTA Astronautica.

- M. Lavagna, R. Benvenuto, S. Salvi, *NET-BASED PAYLOAD ON BOARD AVUM ENHANCED PLATFORM TO EFFICIENTLY REMOVE LARGE DEBRIS FROM LOW EARTH ORBITS*, presented at IAC (Toronto, 2014).

7.2 Future Work

Future work involves:

- The translation of the Simulator into a C++ environment and the parallelization of the process, as designed and prepared for.
- An experimental campaign both to validate the Simulator and retrieve and tune the physical parameters of the models. To this end, a full 3D reconstruction support is required.

Appendix A

PARABOLIC FLIGHT ANALISYS

A.0 Goals and Requirements

The Simulator is planned to be experimentally validated, testing the Net deployment and impact in a micro-gravity environment. The goal is to prove the consistency of mathematical models exploited in the Simulator to simulate the dynamics, viscoelastic forces and contact reactions, with the physical results. To do so, 3D dynamics recovery is required in order to compare the simulation results and define the optimum mathematical parameters that minimize the error. The experiment is carried out exploiting parabolic flight tests, which let to recreate a relative near *zero-g* environment in a limited time window of almost 20s. Therefore, the mock up requires a trade-off of the Net characteristic and shooting parameters in order to satisfy constraints on both time and space scales.

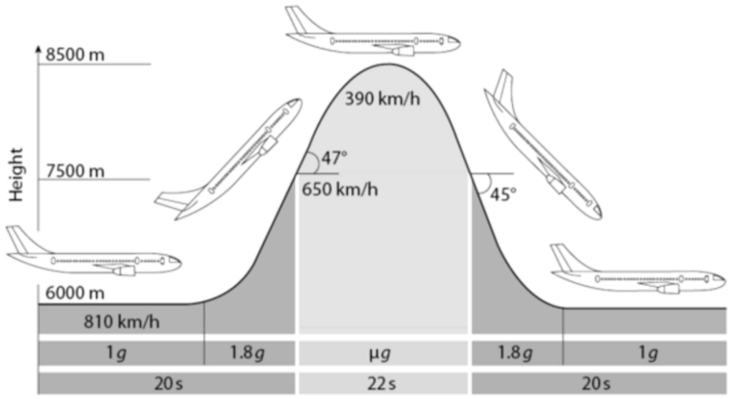


Figure A.1 typical parabolic flight profile *(courtesy of SpaceLand Omega SDP Consortium).*

A.1 Apparent Accelerations

The *microgravity* environment is simulated setting to *zero* the external gravitational field. However, this is a relative acceleration field. Therefore, in order to simulate the real *parabolic flight* environment, relative apparent accelerations must be modelled.

The predictable apparent accelerations arising, are related to the rotation of the aircraft around its CIR, within the parabolic trajectory. Parabola is so defined in order to get a near-zero relative acceleration inside the aircraft (*micro-gravity*). Considering an inertial reference frame **[X Y Z]***ECI* and a local one **[x y z]***PLANE* as shown in picture, the absolute acceleration of a free-moving body can be derived. The local frame is fixed to the aircraft and aligned with the **[roll-pitch-yaw]** frame.

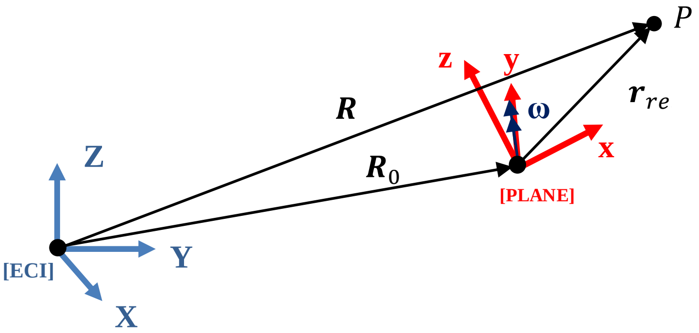


Figure A.2 reference frames convention and absolute position definition for a free-floating body.

Absolute kinematic quantities are here derived:

$$\boldsymbol{R} = \boldsymbol{R}_0 + \boldsymbol{r}_{rel} \tag{A.1}$$

$$\boldsymbol{V} = \boldsymbol{V}_0 + \boldsymbol{v}_{rel} + \boldsymbol{\omega} \times \boldsymbol{r}_{rel} \tag{A.2}$$

$$\boldsymbol{A} = \boldsymbol{A}_0 + \boldsymbol{a}_{rel} + 2\boldsymbol{\omega} \times \boldsymbol{v}_{rel} + \dot{\boldsymbol{\omega}} \times \boldsymbol{r}_{rel} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_{rel}) \tag{A.3}$$

Assuming a constant *pitch-rate* $\overline{\boldsymbol{\omega}}$, the absolute acceleration becomes:

$$\boldsymbol{A} = \boldsymbol{A}_0 + \boldsymbol{a}_{rel} + 2\boldsymbol{\omega} \times \boldsymbol{v}_{rel} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_{rel}) \tag{A.4}$$

The relative acceleration is so defined:

$$\boldsymbol{a}_{rel} = \boldsymbol{A} - \boldsymbol{A}_0 - 2\boldsymbol{\omega} \times \boldsymbol{v}_{rel} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_{rel}) \tag{A.5}$$

- $(\boldsymbol{A} - \boldsymbol{A}_0) = \boldsymbol{a}_{disturbing}$ is the unpredictable disturbing relative acceleration measured by the on-board accelerometer $\boldsymbol{a}_{IMU}$, to be cleared from the centrifugal component (once known the position where it is installed), in order to compute the *real* disturbing acceleration characterizing the parabolic flight:

$$\boldsymbol{a}_{disturbing} = \boldsymbol{a}_{IMU} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_{IMU}) \qquad \text{(A.6)}$$

- $\boldsymbol{r}_{IMU}$ is the position of the accelerometer on the aircraft wrt the CIR.
- $(2\boldsymbol{\omega} \times \boldsymbol{v}_{rel})$ is the *Coriolis* acceleration.
- $\left(\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_{rel})\right)$ is the *Centrifugal* acceleration.

Therefore, the *apparent* acceleration to be considered for a *zero-gravity* simulation is so reported:

$$\boldsymbol{a}_{apparent} = \boldsymbol{a}_{Coriolis} + \boldsymbol{a}_{centrifugal} + \boldsymbol{a}_{disturbing} \qquad \text{(A.7)}$$

Finally, ignoring the CIR position, the relative acceleration can be obtained coupling Eq. A.5 and Eq. A.6:

$$\boldsymbol{a}_{rel} = \boldsymbol{a}_{IMU} - 2\boldsymbol{\omega} \times \boldsymbol{v}_{rel} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \Delta\boldsymbol{r}_{rel}) \qquad \text{(A.8)}$$

Where:

$$\Delta\boldsymbol{r}_{rel} = \boldsymbol{r}_{rel} - \boldsymbol{r}_{IMU} \qquad \text{(A.9)}$$

A.2 Dynamics Constraints

The capture distance is limited by the dimensions of the experimental box:

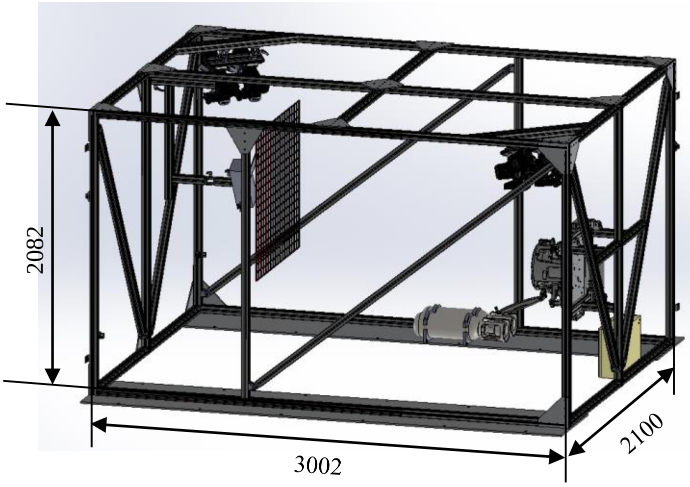


Figure A.2: experimental box dimensions *(courtesy of PRODINTEC)*.

While the time scale limits are dictated by the minimum time required by the 3D dynamics reconstruction and the margined duration of each single parabola:

Table A.1: Time and spatial constraints of the dynamics.

	Min	**Max**
Dynamics time [s]	3	18
Capture distance [m]	1	2

A.3 Preliminary Analysis

The configurations to test, are 0.6m and 0.9m Planar Nets with 2.5cm mesh. Preliminary constraints on the shooting parameters can be found.

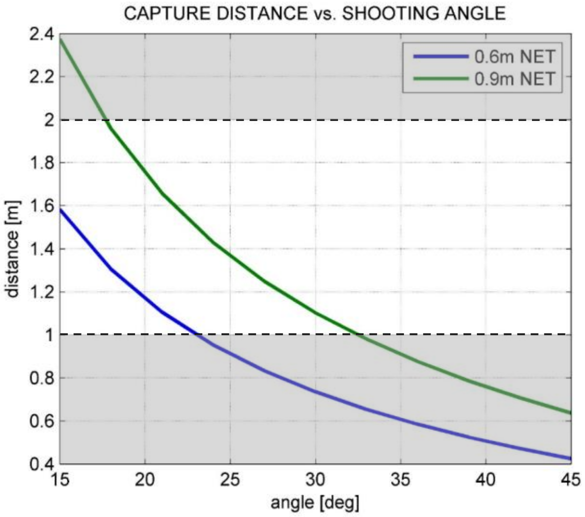


Figure A.3: constraints on the divergence angle arising from the capture distance limits.

The admissible divergence range for the two planar net geometries is:

- 0.9m NET: **[18°-33°]**
- 0.6m NET: **[15°-24°]**

According to the estimated capture distance (Fig. A.3), it is possible to predict the impact time (dotted line) and the whole capture duration including the wrapping time (Eq. 6.20). Imposing the time constraints, we further refine the choice of the admissible divergence angles and shooting velocities.

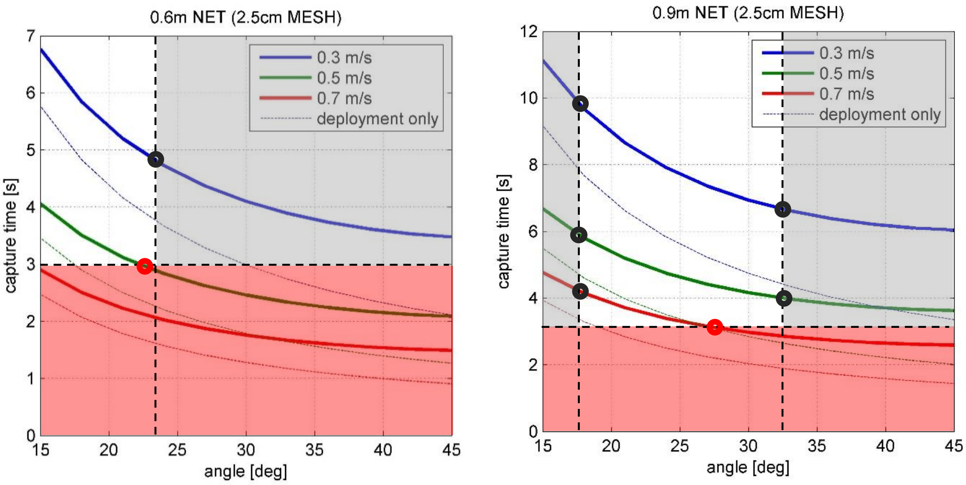


Figure A.4: constraints on the divergence angle arising from the capture distance limits combined with the temporal constraints.

Table A.2: final admissible range for shooting velocity and divergence angle, coming from spatial and temporal constraints.

	0.6m		**0.9m**	
Shooting velocity	*min*	*MAX*	*min*	*MAX*
0.3 m/s	15°	24°	18°	33°
0.5 m/s	15°	21°	18°	33°
0.7 m/s	-	-	18°	27°

A.4 Parametric Sensitivity Analysis

To simplify the problem in order to perform a sensitivity analysis, the following assumptions are done:

- Net dynamics reduced to its CoM;
- Net velocity is fixed to the predicted CoM velocity of the real dynamics.

Written in another way, the relative dynamics Eq. A.5 becomes:

$$\ddot{\boldsymbol{r}} = -2\boldsymbol{\omega} \times \dot{\boldsymbol{r}} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}) + \boldsymbol{a}_{disturbing} \tag{A.10}$$

The initial conditions associated are:

$$\boldsymbol{r}(t=0) = \boldsymbol{r}_{BOX} + \boldsymbol{r}_0, \qquad \dot{\boldsymbol{r}}(t=0) = \boldsymbol{v}_{CoM} \tag{A.11}$$

Where $\boldsymbol{r}_{BOX}$ is the position of the Box launching system on the aircraft and $\boldsymbol{r}_0$ is the CoM initial position with respect to this last.

Solved, the second order linear differential equation is converted into a system of two first order ones. The analysis is conducted on the velocity of the Net CoM, so taking into account the combination of all the shooting parameters (Eq. 6.19).

The *pitch* angular velocity is assumed constant (from the flight datasheet) and equal to 4 deg/s. This is computed knowing the total angular envelope from -45° to +45° in a temporal arch of 22s (Fig. A.1).

The shooting velocity modulus range considered is:

$$v_b = [\mathbf{0.3 - 0.7}]\ m/s \tag{A.12}$$

The lower value is limited by the shooting device technology.

The shooting divergence angle range is the one identified in Section A.2:

$$\alpha = [\mathbf{18 - 24}]\ deg \tag{A.13}$$

In this way the minimum and maximum values of Net CM velocity can be identified:

$$v_{CoM\_min} = 0.25\ m/s$$

$$v_{CoM\_max} = 0.64\ m/s$$

Therefore the analysis is conducted for the following range:

$$v_{CoM} = [\mathbf{0.2 - 0.7}]m/s \tag{A.14}$$

Here it is presented the sensitivity of the relative deviation of the dynamics, according to the spatial offset from the aircraft CoM in each of the three directions:

- *Longitudinal*, roll axis (x)
- *Transversal*, pitch axis (y)
- *Vertical*, yaw axis (z).

The simulation time is fixed to the one required by the unperturbed net to reach the end of the Box (2m), according to the Net CoM velocity. The results are presented as the *maximum relative deviation* in the three directions with respect to the ideal uniform linear dynamics.

In the first case, fixing the transversal and vertical offset to [0]m and varying the *longitudinal* one, we see that sensitivity is not symmetric. Since the dynamics is propagated along the roll axis in direction and versus, it is notable a much higher sensitivity to the velocity at lower speeds with negative offsets. At low shooting velocities it is strictly required to be within +5m distance from the aircraft CIR,

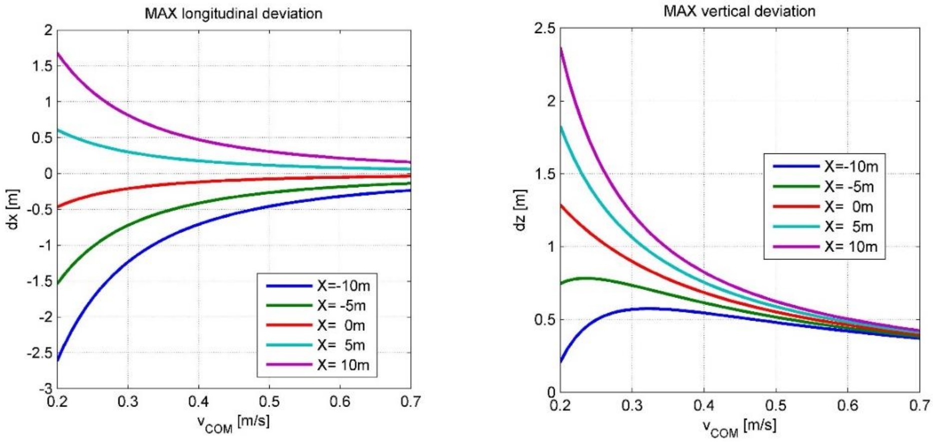


Figure A.5: Longitudinal and Vertical deviation sensitivity to the longitudinal OFFSET from the aircraft CoM.

withstanding a maximum longitudinal deviation of ±0.5m and a vertical one of +1.7m. The vertical deviation is always positive as long as the propagation of the dynamics is positively along the roll axis. In the other case, the behavior is opposite, but keeping the velocity sensitivity. No lateral deviation is recorded.

Transversal offset has non-influence on the dynamics, as long as shooting parallel to the pitch axis. Therefore the only transversal constraint comes from the possible not synchronous shot of the bullets or mounting error of the shooting mechanism, leading to a not symmetric deployment and a transversal deviation already analyzed in Section 6.3.2. Applied to the experimental most limiting case (0.9m Net), it is requested to ensure a maximum transversal deviation of 0.55m.

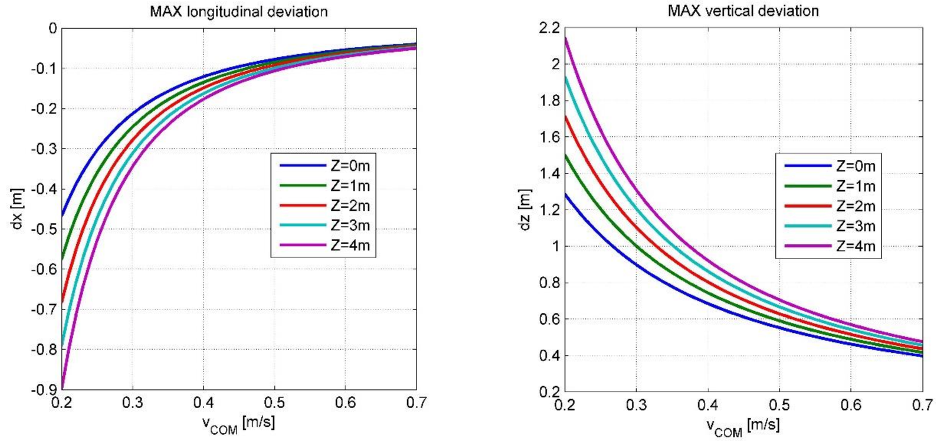


Figure A.6: Longitudinal and Vertical deviation sensitivity to the vertical OFFSET from the aircraft CoM.

Concluding, as long as the shooting direction is coincident with the roll axis:

- the vertical deviation is always positive;
- the longitudinal deviation keeps positive as long as the shooting point is at a positive distance from the CoM at least equal to the Box length. The deviation is negative in the other case;
- there is no lateral deviation, whichever the shooting velocity and direction orientation.
- Finally, but most important, the higher is the CoM velocity, the lower is the sensitivity to any offset and the lower is also the deviation too. So, need to do a trade off with the desired times of the dynamics and the deviations to face, as reported below.

A.5 Real Scenario Application Example

Avoiding, as proved, the transversal offset and assuming that the aircraft experimental ground is at a height of 2m from its CIR, it is significant to analyze the deviation sensitivity to the only longitudinal offset (assigned slot in the cabin). The reasonable offset range considered is ±10m.

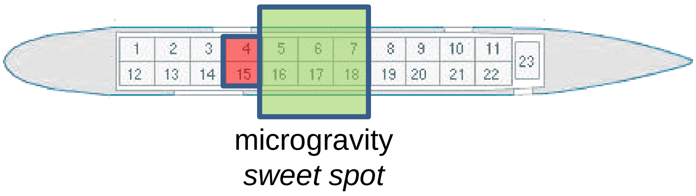


Figure A.7: Box position considered on the aircraft *(courtesy of SpaceLand Omega SDP Consortium).*

Imposing a maximum longitudinal deviation of **±1m** (50% of the dynamics available length) and **1m** as maximum vertical deviation (1.4m minus 0.4m that is half the maximum Net dimension), we find from Fig. A.5 that in order to face whichever longitudinal offset, the CoM velocity needs to be higher than **0.4m/s.**

Therefore, considering the shooting device at a distance **[5 2 2]**m from the aircraft CIR (red box in Fig. A.7), the trajectory and velocity profiles are derived (Fig. A.8) and we obtain the following estimations:

- Vertical deviation: **+0.73 m** (<1m).
- Lateral deviation due to the positioning/shooting error: **-0.03m**.
- Assuming that the Net deploys orthogonally to the trajectory, from the derivative of the vertical deviation we can also estimate the Net plane inclination, useful to orient the Target before the shot: **16deg**.
- Impact velocity: **0.5 m/s**.
- Time to impact: **4.5 s** (>3s).

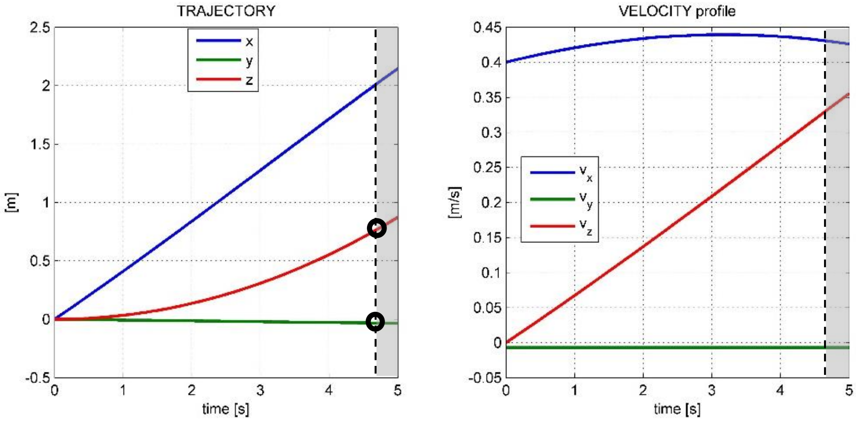


Figure A.8: trajectory and velocity profiles with time.
[5 2 2]m OFFSET and 0.4m/s CoM velocity.

Analyzing the acceleration profile (Fig. A.9), it is possible to separately study the two contributions of the apparent acceleration. This is a further confirmation of the always positive contribution on the vertical deviation, due to the positive contribution of both the apparent accelerations.

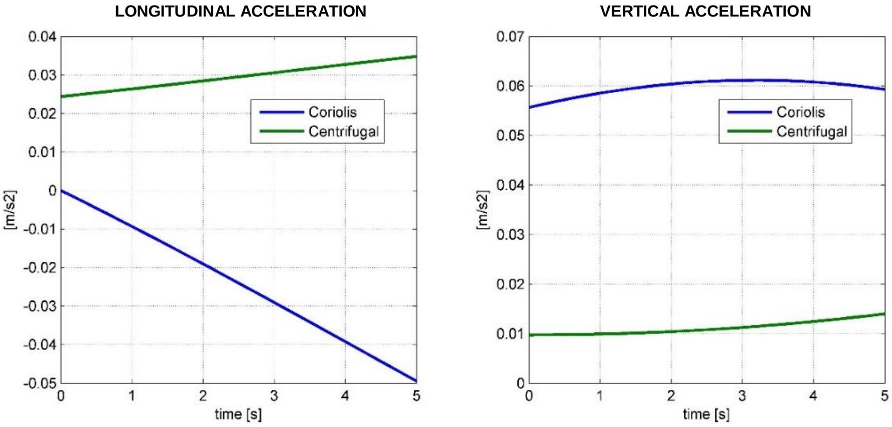


Figure A.9: (a) longitudinal and (b) vertical apparent accelerations profiles with time. Coriolis and centrifugal contributions.

A.6 Simplified Model Validation

Considering the previous parameters, we simulate the whole Net deployment and check the consistency of the mathematical reduced model with the complete one computing the error.

The Target is therefore positioned according to the previously predicted results (Fig. A.10).

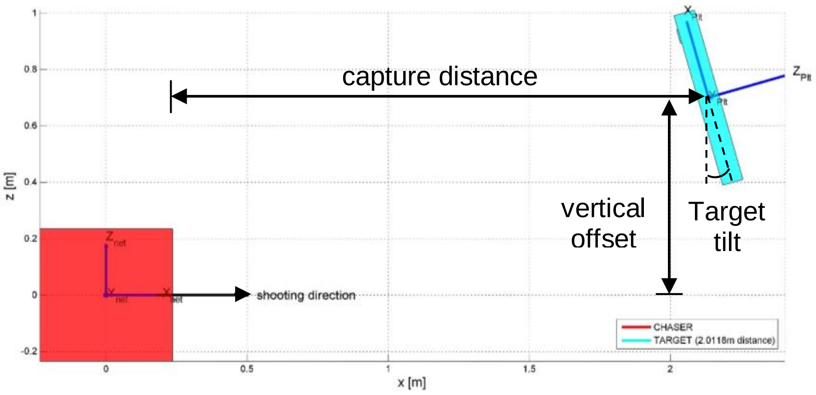


Figure A.10: Experimental set-up, Target position and attitude.

We report in **Fig. A.11** the comparison between the CoM trajectory predicted with the reduced model and the real one simulated considering the whole Net dynamics.

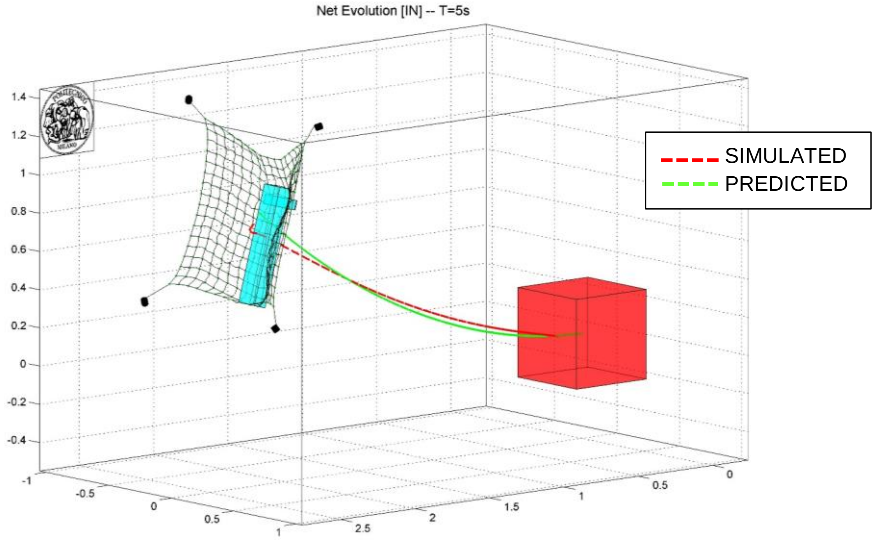


Figure A.11: Comparison between the simulated trajectory of the Net CoM (red) and the predicted one (green).

The real CoM trajectory almost perfectly matches the predicted one.

Target positioning can be defined successful, including the inclination of the Net at the impact instant with the target.

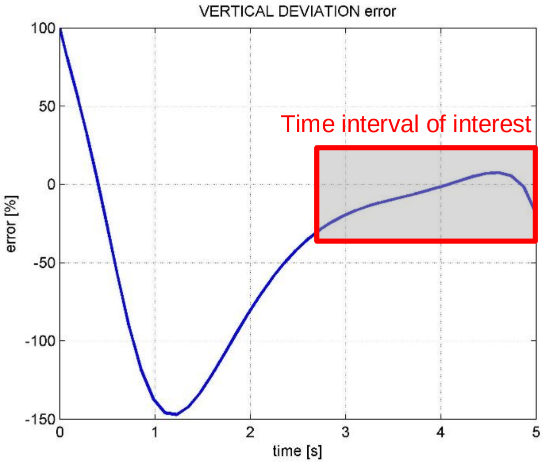


Figure A.12: percentage error on the vertical deviation of the simplified model.

Analyzing the error on the predicted vertical deviation (Fig. A.12), it can be noted that, neglecting the initial time instants, the error at the capture time is 2.5e-3 m (5%), almost negligible.

A.7 Conclusions and Remarks

We analyzed the parabolic flight dynamics and its implications in the modelling of the external accelerations. We also proved the possibility to simplify the overall dynamics in order to perform a parametric sensitivity analysis, essential for the preliminary mockup of the entire series of experiments. Due to its simplicity, this model also provides a useful tool for a rapid estimation of the ideal Target position and attitude according to all the possible combination of shooting parameters and Net configurations (that may be need to change in short time between the tests).

It is important to remark that the analysis presented so far, refers only to the ideal *zero-gravity* environment. Further perturbing accelerations, coming from the flight, are stochastic and cannot be foreseen, so the analysis reported is to be considered a baseline indicative behavior, with respect to which any external acceleration sums up. It means that all the reported results and analyses are useful for a first guess solution of the dynamics and Target positioning (in particular the sensitivity to the shooting parameters). Calibration shots are required before starting the campaign.

ACRONYMS AND SYMBOLS

NeST	= Net Simulation Tool
SoA	= State of the Art
ESA	= European Space Agency
CDF	= Concurrent Design Facility
CAM	= Collision Avoidance Maneuver
LEO	= Low Earth Orbit
MEO	= Medium Earth Orbit
GEO	= Geostationary Earth Orbit
SSO	= Sun-Synchronous Orbit
RAAN	= Right Ascension of the Ascending Node
ADR	= Active Debris Removal
ECI	= Earth Centered Inertial frame
LVLH	= Local Vertical Local Horizontal frame
PI	= Principal Inertia axes frame
CoM	= Center of Mass
BR	= Bullet Ratio
PDE	= Partial Differential Equation
ODE	= Ordinary Differential Equation
AMM	= Assumed Modes Method
FEM	= Finite Elements Method
LPM	= Lumped Parameters Model
2D - 3D	= 2 - 3 dimensions
DOF	= Degree Of Freedom
SBV	= Spherical Bounding Volume
SMBV	= Spherical Minimum Bounding Volume

AABB	= Axis Aligned Bounding Box
OCP	= Oriented Convex Polyhedra
LOD	= Levels of Detail
BV	= Bounding Volume
RSV	= Representing Spherical Volume
SPM	= System Physical Model
IS	= Integration Solver
CD	= Collision Detection
EBC	= Essential Boundary Conditions
NBC	= Natural Boundary Conditions
ACS	= Attitude Control Subsystem

X,Y,Z	= spatial coordinates of the inertial frame [m]
x,y,z	= spatial coordinates of the local frame [m]
u, v, w	= local displacement coordinates [m]
s,t	= space [m], time [s]
$\boldsymbol{h}_{orb}$	= specific orbital angular momentum [N m s kg^{-1}]
i	= orbit inclination [rad];
ω	= orbit anomaly of pericenter [rad]
θ	= orbit true anomaly [rad]
m_i	= mass [kg]
$\boldsymbol{R}_i$	= position vector in the inertial frame [m]
$\boldsymbol{V}_i$	= velocity vector in the inertial frame [m/s]
$\boldsymbol{r}_i$	= position vector in the local frame [m]
$\boldsymbol{F}_i$	= force vector in the inertial frame [N]
$\boldsymbol{T}_i$	= torque vector in the local frame [N m]
$\boldsymbol{I}_i$	= inertia matrix in local frame [kg m^2]
$\boldsymbol{M}$	= mass matrix [kg] and [kg m^2]
$\boldsymbol{\omega}_i$	= angular velocity vector in local frame [rad/s]
$\boldsymbol{q}$	= quaternion
c_i	= direction cosine

A_i	= area [m^2]
L_i	= length [m]
d_i	= diameter [m]
ρ	= density [kg/m^3]
P_i, v_i	= pressure [Pa] and specific volume [m^3 kg^{-1}]
γ	= specific heat ratio of the gas
c_D	= drag coefficient
$\boldsymbol{N_i}$	**=** normal surface exiting unit vector in inertial frame
$\boldsymbol{n_i}$	**=** normal surface exiting unit vector in local frame
$\boldsymbol{t_i}$	**=** tangential unit vector in local frame
$\boldsymbol{e_i}$	= surface border normal unit vector
$\boldsymbol{b_i}$	= surface border
h_b, r_b	= Bullet height and radius
$\boldsymbol{K}$	= stiffness matrix
E_i, G_i	= Young’s modulus and shear stiffness [Pa]
EA	= Axial Stiffness
ε_{max}	= ultimate strain
ν_i	= Poisson’s ratio
ζ	**=** damping ratio
λ	= hysteresis damping factor
e_{eff}	= effective restitution coefficient
μ_s	= Coulomb’s friction coefficient
k, c	= stiffness [N/m] and damping coefficient [N s/m]
J	= cross section inertia [m^4]
J_p	= cross section polar inertia [m^4]
φ	= incidence angle [rad]
α	= Bullet shooting divergence angle [rad]
W_i	= work [J]
E_k, U_i	= kinetic and potential energy [J]
$\boldsymbol{\Gamma}_i$	= angular momentum [N m]
Q	= deployment quality index

μ	= Earth gravitational constant: $3.986\text{e}14\ \text{m}^3/\text{s}^2$
J_2	= Earth first zonal harmonic coefficient: 1.08271e-3
c	= light speed: 3e8 m/s
W_S	= solar radiation specific power: 1358 W/m^2

BIBLIOGRAPHY

[1] B. Von Herzen, A. Barr, and H. Zatz. *Geometric collisions for time-dependent parametric surfaces*, volume 24, pages 39–48. ACM, August 1990.

[2] P.M. Moore and J. Wilhelms. *Collision detection and reponse for computer animation*, volume 22, pages 289–298. ACM, August 1988.

[3] J. Canny. Collision detection for moving polyhedra. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2), 1986.

[4] E.G. Gilbert and S.M. Hong. *A new algorithm for detecting the collision of moving objects*. International Conference on Robotics and Automation, pages 8–13. IEEE, 1989.

[5] W. Meyer, *Distance between boxes: Applications to collision detection and clipping*. International Conference on Robotics and Automation, pages 597–602. IEEE, 1986.

[6] P.A. Cundall. *Formulation of a three-dimensional distinct element model—Part I. A scheme to represent contacts in a system composed of many polyhedral blocks*. International Journal of Rock Mechanics, Mineral Science and Geomechanics, 25, 1988.

[7] F.P. Preparata and M.I. Shamos. *Computational Geometry*. Springer-Verlag, New York, 1985.

[8] F.P. Preparata and M.I. Shamos. *Computational Geometry*. Springer-Verlag, New York, 1985.

[9] S. Pabst, A. Koch and W. Straßer, *Fast and Scalable CPU/GPU Collision Detection for Rigid and Deformable Surfaces*, Eurographics Symposium on Geometry Processing, 2010

[10] Min Tang, Dinesh Manocha, Jiang Lin, Ruofeng Tong, *Collision-Streams: Fast GPU-based Collision Detection for Deformable Models,* 2011

[11] http://www.sofa-framework.org/

[12] http://gamma.cs.unc.edu/SWIFT/

[13] http://gamma.cs.unc.edu/I-COLLIDE/

[14] http://bulletphysics.org/wordpress/

[15] http://gamma.cs.unc.edu/software/

[16] H. Samet. *Spatial Data Structures: Quadtrees, Octrees and Other Hierarchical Methods*. Addison-Wesley, Reading, MA, 1989.

[17] S. Gottschalk, M. C. Lin, and D. Manocha. *OBB-tree: A hierarchical structure for rapid interference detection*. Computer Graphics, 1996.

[18] G. van den Bergen. *Efficient collision detection of complex deformable models using ABB trees*. J. Graphics Tools, 1997.

[19] Chai-Cheng Huang, Hung-Jie Tang, Jin-Yuan Liu, *Dynamical analysis of net cage structures for marine aquaculture: Numerical simulation and model testing,* ELSEVIER 2006.

[20] Chun-Woo Lee, Ju-Hee Lee, Bong-Jin Cha, Hyun-Young Kim, Ji-Hoon Lee, *Physical modeling for underwater flexible systems dynamic simulation*, ELSEVIER 2005.

[21] G. Zachmann. *Minimal hierarchical collision detection. In ACM Virtual Reality Software and Technology*, Nov. 2002.

[22] K. Yamamoto, K. Nashimoto, *Validity and visualization of a numerical model used to determine dynamic configurations of fishing nets*. Fisheries Science 2003.

[23] McKenna H.A., Hearle J.W.S, O'Hear N.,*Handbook of fibers rope technology*, The Textile Institute, Woodhead Publishing Limited, Cambridge, England.

[24] Astrium, *RObotic GEostationary orbit Restorer (ROGER) Phase A Final Report,* ROGSIBRE- FP Issue 1, 26 June 2003.

[25] R. Biesbroek, *The e.Deorbit study in the Concurrent Design Facility*, Presentation handouts, Workshop on Active Space Debris Removal, 17 September 2012.

[26] P. M. Hubbard. *Approximating polyhedra with spheres for time-critical collision detection.* ACM Trans. on Graphics, 1996.

[27] P. Jimnez, F. Thomas, and C. Torras. *3D Collision detection: A survey*. Computers and Graphics, 2001.

[28] J. Klosowski, M. Held, J. S. B. Mitchell, K. Zikan, and H. Sowizral. *Efficient collision detection using bounding volume hierarchies of k-DOPs*. IEEE Trans. on Visualizat. Comput. Graph., 1998.

[29] P. Kumar, J. S. B. Mitchell, and A. Yıldırım. *Approximate minimum enclosing balls in high dimensions using core-sets*. The ACM J. of Experimental Algorithmics, 8, 2003.

[30] Canudas de Wit, C., Olson, H., Astrom, K. J. and Lischinsky, *A new model for control of systems with friction*, IEEE Transaction on Automatic Control 1995.

[31] I. J. Palmer and R. L. Grimsdale. *Collision detection for animation using sphere-trees.* Computer Graphics Forum, June 1995.

[32] J. Canny. *Collision detection for moving polyhedra*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986.

[33] P.A. Cundall. *Formulation of a three-dimensional distinct element model— Part I. A scheme to represent contacts in a system composed of many polyhedral blocks*. International Journal of Rock Mechanics, Mineral Science and Geomechanics, 25, 1988.

[34] Jonathan D. Cohen, Ming C. Lin, Dinesh Manocha, Madhav K. Ponamgi, *I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scale Environments,* University of North Carolina, 1998

[35] H. Mazhar, T. Heyn, A. Pazouki, D. Melanz, A. Seidl, A. Bartholomew, A. Tasora, and D. Negrut, *Chrono: a parallel multi-physics library for rigid-body, flexible-body, and fluid dynamics,* Mech. Sci., 2013

[36] Han, I. and Gilmore, *Multi-body impact motion with friction—analysis, simulation, and experimental validation,* ASME Journal of Mechanical Design 1993.

[37] Glocker, C. and Pfeiffer, *Dynamical systems with unilateral contacts, Nonlinear Dynamics,* 1992

[38] Stronge, *Impact Mechanics*, Cambridge University Press 2000

[39] Brogliato, *Nonsmooth impact mechanics,* Springer, London 1996.

[40] Gonthier, McPhee, J., Lange, C. and Piedboeuf, *A regularized contact model with asymmetric damping and dwell-time dependent friction, Multibody System Dynamics*, 2004

[41] Johnson, *Contact Mechanics,* Cambridge University Press, London 1985.

[42] Hertz, *Miscellaneous Papers by H. Hertz,* Eds. Johns abd Schott, London 1896.

[43] Hippmann, *An algorithm for compliant contact between complexly shaped bodies,* Multibody System Dynamics 2004

[44] Hasegawa, S. and Sato, *Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects,* EUROGRAPHICS 2004.

[45] Goldsmith, *Impact: The Theory and Physical Behavior of Colliding Solids,* Edward Arnold Ltd., London 1960.

[46] Timoshenko, S. P. and Goodier, *Theory of Elasticity*, McGraw-Hill, New-York 1970.

[47] Hunt, K. H. and Crossley, *Coefficient of restitution interpreted as damping in vibroimpact*, Transactions of the ASME Journal of Applied Mechanics 1975

[48] Khulief, Y. A. and Shabana, *A continuous force model for the impact analysis of flexible multibody systems,* Mechanism and Machine Theory 1987

[49] Lankarani, H. M. and Nikravesh, *Application of the canonical equations of motion in problems of constrained multibody systems with intermittent motion*, ASME Design Automation Conference, 1988 Orland (Florida).

[50] Johnson, *Contact Mechanics*, Cambridge University Press, London 1985.

[51] Goldsmith, *Impact: The Theory and Physical Behavior of Colliding Solids*, Edward Arnold Ltd., London 1960.

[52] Armstrong, *Control machine friction*, Kluwer Academic 1991

[53] Rabinowicz, *Stick and slip*, Scientific American 1956.

[54] G.Hippmann, *An algorithm for compliant contact between complexly shaped surfaces in multibody dynamics*, MULTIBODY DYNAMICS 2003, Jorge A.C. Ambrosio (Ed.)

[55] J. Wojewoda, A. Stefański, M. Wiercigroch and T. Kapitaniak, *Hysteretic effects of dry friction: modelling and experimental studies*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 2008.

[56] V. S. Aslanov, A S Ledkov, *Dynamics of Tethered Satellite Systems*, Woodhead Publishing, 2012.

[57] Krasnov N F , Koshevoy V N , Danilov A N and Zakharchenko, *Rocket Aerodynamics* . Moscow, Vysshaya Shkola Press, 1968.

[58] Jumkins J. and Youdan K., *Introduction to dynamics and control of flexible structures*. AIAA, 1998.

[59] Kim, E. Vadali, *Modelling issues related to retrieval of flexible tethered satellite systems*, J. Guidance Control Dyn. 18(5). 1169-1176

[60] Dr.-Ing. Carsten Wiedemann, Prof. Dr.-Ing. Peter Vörsmann, *Space Debris – Current Situation*, space debris session, IAC Toronto 2014

[61] www.esa.int/Our_Activities/Space_Engineering/Clean_Space