POLITECNICO DI TORINO

Dipartimento di INGEGNERIA MECCANICA E AEROSPAZIALE

Corso di laurea magistrale

in Ingegneria Aerospaziale e Astronautica

Tesi di Laurea Magistrale

Thruster layout optimization for a space mission with demanding control requirements

Referente Accademico

Prof.ssa Battipede Manuela

Candidato

Pacentra Francesco

Referente Aziendale

Dott. Fasano Giorgio (Thales Alenia Space)

Anno Accademico 2018-2019

Alla mia famiglia,
per l'incommensurabile sostegno datomi

LIST OF FIGURES

List of Tables

Riepilogo

Lo scopo del seguente lavoro è quello di fornire strumenti di modellazione per complessi problemi matematici riguardanti l'orientamento di motori per il controllo di assetto a bordo di satelliti, nonché l'analisi di casi di studio di interesse dell'azienda ospitante (Thales Alenia Space Torino). Le metodologie sviluppate sono state applicate per lo studio in prima fase di progetto della missione LISA [6], [10]. Le configurazioni sviluppate dovranno garantire la stabilità del satellite durante la fase di missione scientifica e al contempo ottenere una riduzione di consumo di propellente per minimizzare la massa a bordo o, a parità di massa totale, consentire un'allocazione maggiore di massa per altri sistemi quali ad esempio il payload o infine ad una estensione della missione oltre il periodo previsto.
Partendo da una configurazione possibile fornita da ESA, le metodologie applicate in questo lavoro sono state impiegate per ridurre il consumo di propellente rispetto alla configurazione di partenza riducendo, al contempo, il numero totale di attuatori a bordo rispettando i requisiti di progetto.

La missione LISA (Laser Interferometer Space Antenna) si pone come successore della celebre missione LISA Pathfinder [3] iniziata nel 2015 e terminata nel 2017, la quale era un dimostratore per la successiva missione LISA di cui sopra. Lo scopo della missione LISA è quello di rilevare onde gravitazionali generate da sistemi di stelle binarie all'interno della Via Lattea, da buchi neri super-massicci in altre galassie oppure dalla fusione tra buchi neri super-massicci come ulteriore conferma alla teoria della relatività generale di Einstein. L'esperimento condotto all'interno di LISA Pathfinder consisteva nella misurazione della distanza presente tra due cubi posti a bordo del satellite. Se la distanza tra le due masse cambia, allora una perturbazione dovuta ad una forza gravitazionale deve aver agito sulle masse, producendo un effetto analogo a quello prodotto dalle onde gravitazionali osservate da Ligo e Virgo [3], i due laboratori per la ricerca di onde gravitazionali presenti sulla Terra. L'esperimento che verrà condotto nella missione LISA è concettualmente identico a quello presente sul LISA Pathfinder con la differenza che i satelliti in orbita saranno tre, ognuno con al proprio interno una massa di prova e la distanza relativa tra due masse sarà misurata da un raggio laser. I satelliti verranno posti ad una distanza di 2.5 milioni di chilometri l'uno dall'altro.

Come è facile intuire, il controllo di assetto è una parte fondamentale della missione che potrebbe comportare il suo successo o fallimento e dunque prevede uno studio oltremodo approfondito ed accurato. È stato previsto l'uso a bordo di attuatori con spinta completamente regolabile da un valore nullo ad uno massimo e quindi eventualmente spegnibili. Da un punto di vista ingegneristico l'uso di una tale tipologia di attuatori comporta il grande vantaggio di non dover compensare la spinta minima che uno di questi può esercitare con la spinta di un altro attuatore andando, dunque, a diminuire potenzialmente i consumi.

Il problema matematico che è stato affrontato in questo lavoro è di complessa struttura. La presenza, difatti, di variabili come l'orientamento e la spinta di ogni singolo attuatore interdipendenti tra loro, rende il problema fortemente non lineare e dunque di difficile soluzione anche per gli ottimizzatori più evoluti. Un ulteriore elemento di complessità riguarda il numero di step di controllo (istanti) considerato. Dal momento che la legge di controllo deve essere implementata ad ogni istante, un aumento del numero di questi porta ad un aggravio di costo computazionale e dunque di difficoltà. Pertanto, per affrontare il problema ed arrivare ad

ottenere delle configurazioni soddisfacenti si è deciso di scorporare quest' ultimo in due sotto-problemi. Nella prima parte il numero di istanti considerato viene ridotto tramite un algoritmo che seleziona gli istanti più rappresentativi della missione in base a quanti se ne desiderano analizzare [9]. Così facendo le dimensioni del problema diminuiscono significativamente e può essere affrontato più agevolmente. Per andare ulteriormente a semplificare il problema gli orientamenti vengono discretizzati con una spaziatura decisa dall'utente. Quest'ultima non deve essere né troppo fine, altrimenti si rischierebbe di andare aumentare troppo il costo computazionale, né troppo larga per avere una maggiore copertura angolare. Con tali assunzioni è stato possibile implementare il modello da qui in avanti denominato *Layout*, che consente di trovare delle prime soluzioni di partenza.

Le configurazioni così ottenute non è detto che siano ammissibili anche considerando l'intero set di istanti. Per verificarne la bontà si utilizza allora un ulteriore modello, denominato *Continuous*, in cui è coperto l'intero arco temporale. In questa fase gli orientamenti sono fissati, secondo i risultati derivanti dal modello *Layout*. Il costo computazionale di questo modello è relativamente basso in quanto è un modello lineare in cui sono presenti solamente le variabili di spinta e non quelle relative agli orientamenti.

Per migliorare ulteriormente le configurazioni trovate, che corrispondono a minimi locali, si è utilizzata una rifinitura locale della soluzione. Partendo dagli orientamenti del modello *Layout*, infatti, si è usata una discretizzazione molto fine nell'intorno della soluzione di partenza considerando un numero di orientamenti possibili relativamente limitato. Con tale accorgimento è possibile costruire una mesh di dimensioni ridotte **FIGURE 12** ed utilizzare dunque l'intero set di istanti. Il processo viene iterato più volte utilizzando come soluzione di partenza quella ottenuta dalla iterazione precedente e, man mano che il numero di iterazioni aumenta, la discretizzazione si fa sempre più fine.

I solutori usati finora lavorano cercando una soluzione globale all'interno del dominio considerato. Un tale metodo di ricerca delle soluzioni ha il vantaggio di considerare tutte le soluzioni possibili e, potenzialmente, la migliore ma, al contempo, richiede un costo computazionale talmente elevato da non poter essere preso in considerazione come unico strumento di analisi. Si è quindi optato per l'utilizzo di un ottimizzatore locale in ambiente MATLAB [11] per la rifinitura locale delle soluzioni, dopo aver trovato delle configurazioni sub-ottime. In tale ambiente è possibile implementare vari algoritmi risolutivi del problema di minimizzazione in esame che si basano su formulazioni matematiche differenti dello stesso problema.

Lo scopo di questa tesi è, come precedentemente indicato, in primo luogo quello di minimizzare il consumo di propellente durante la fase dei rilevamenti scientifici cercando di ridurre, al contempo, il numero di attuatori a bordo. La missione d'altro canto è costituita da svariate fasi e non solo da quella dei rilevamenti scientifici e quindi sul satellite agiranno forze e coppie differenti da quelle agenti nella suddetta fase per cui è stata condotta l'ottimizzazione. Le configurazioni ottenute sono dunque state testate anche in differenti scenari per valutare la relativa fattibilità del controllo di assetto anche nei suddetti.

Gli attuatori presenti a bordo appartengono essenzialmente a due rami: l'AOCS (Attitude and Orbit Control System) e DFACS (Drag-Free Attitude Control System). I primi sono necessari per il controllo di assetto dell'intero satellite e quindi agiscono in varie fasi della missione come la separazione dal lanciatore (detumbling) e il raggiungimento dell'orbita finale, mentre i

secondi agiscono per garantire la perfetta stabilità del satellite durante la fase di rilevamenti scientifici in modo da non compromettere l'esperimento.

L'ottimizzazione svolta in questo lavoro riguarda fondamentalmente gli attuatori del DFACS ma anche quelli dell'AOCS sono stati presi in considerazione. La complessità del posizionamento ed orientamento degli attuatori dell'AOCS deriva dalla tipologia di attuatori considerata. Difatti, gli attuatori a bordo sono del tipo on/off e quindi la spinta che possono esercitare è zero oppure un valore massimo. Nei modelli utilizzati anche la spinta dovrà essere considerata come variabile logica, aggravando ulteriormente la complessità del problema. Dal momento che considerare come variabili logiche le spinte e gli orientamenti avrebbe richiesto un costo computazionale troppo elevato, si è deciso di scorporare il problema. In una prima fase si utilizza il modello *Layout*, considerando cioè la spinta continua e gli orientamenti discretizzati, e successivamente si riduce il dominio angolare in un intorno della soluzione così ottenuta, contemplando però spinte discretizzate.

Introduction

The research work discussed in this thesis has been carried out at the Thales Alenia Space, Turin premises, Domain Exploration and Science Italy (DESI), in support of the LISA (Laser Interferometer Space Antenna) program [10], funded by the European Space Agency (ESA), currently under study.
Following the successful LISA Pathfinder mission (ESA, 2015-2007, [3]), devoted to the gravitational wave detection in flight, the LISA program is aimed at realizing the first space-based observatory to investigate this very intriguing aspect of the general theory of relativity by A. Einstein. LISA, whose launch is expected in 2034, will consist of three identical spacecraft separated by 2.5 million km in a triangular formation, following Earth in its orbit around the Sun.
Among the numerous demanding issues relevant to this very challenging space program, one concerns the layout of the thrusters on-board each spacecraft, made available to provide the requested attitude control, in the different phases of the whole mission. At each control step, the entire action exerted by the thrusters has to satisfy the demand from the on-board controller, expressed as overall force and torque that has to act on the spacecraft (with respect to an assigned system-based reference frame).
As is gathered, different positions and orientations of the actuators can yield even a significantly diverse overall performance, in terms of fuel consumption. The number of thrusters adopted, moreover, gives rise to a further non-negligible concern. As a matter of fact, although a rather large number of these might advantageously contribute to a reduction of the overall fuel consumption (leveraging on an extended distribution), the more thrusters are installed, the heavier and the more complex the system becomes. This aspect entails an additional non-trivial issue, and, consequently, an adequate trade-off between the fuel consumption minimization and the limitation of the number of actuators represents the basic framework of any dedicated systems engineering analysis.
In recent years, Thales Alenia Space has been looking into a similar problem, in the context of the Next Generation Gravity Mission (NGGM), a candidate Earth observation program, promoted by ESA, currently at a preliminary study phase [2]. As is understood, for this kind of mission, due to the strong atmospheric drag effect, a very strict attitude control strategy has to be envisaged. In order to tackle the relevant thruster layout optimization problem effectively, an ad hoc optimization methodology has been introduced [7].
A dedicated controller has the task of determining, at a predefined frequency, the overall control action, aimed at achieving (step by step) the desired system attitude. A number of thrusters are available to exert the overall force and torque, as required. The system engineer in charge of the control-actuator layout is therefore presented with the not-at-all-easy task of positioning and orienting the thrusters on the external surface of the spacecraft. Their primary objective, noticeably, consists in minimizing the overall fuel consumption, during the whole mission, while keeping the total number of actuators below an assigned threshold.
The resulting optimization problem (even when simplified by focusing exclusively on the thruster orientation task) relates to a non-convex quadratically constrained structure, well known for being NP-hard. This intrinsic difficulty, from a practical point of view, becomes even more evident, when dealing with real-world large-scale instances, as in the specific NGGM case. To this purpose, an overall heuristic methodology, aimed at providing satisfactory (albeit sub-optimal) solutions has been thought up, by adopting a mathematical programming approach [12], in particular linear, nonlinear and mixed-integer-linear programming (LP, NLP, MILP, see e.g. [8]).
The fundamental concept is to divide the thruster layout problem into much easier sub-problems and to solve these applying an overall iterative process until a global valid solution to the

original problem is found. The approach explained above exploits the very specific structure of this problem.
The scenario considered in this thesis is a reduced scenario where the thrusters application-points were assigned a priori and thus only two discrete set of variables are involved, i.e. those representing the relevant orientations and those associated with the forces exerted by actuators at each control step considered.
With the abovementioned assumption, the majority of the non-linear constraints of the problem, expressed in the equations concerning the forces and torques request, are bilinear. The remaining constraints, i.e. those regarding the orientations of thrusters, are instead of quadratic type. Once the orientation variables are fixed, the resulting problem becomes linear (since the bilinearity were given by orientations variable and thrust variable), and all the quadratic constraints can be dropped. Having said that, two separate sub-problem can be generated: the first has the scope to find some suitable set of values for the orientation variables in order to make the original problem linear and thus easier to solve; the second consists in thrust minimization.
To be more specific, the first sub-problem (that is per se quadratic and non-convex) is structured to find the orientation of the actuators. To this purpose is taken into account just a limited subset of control steps representing the whole time span. On the other hand, the second sub-problem (that is useful to remind is linear) aims to optimize the overall original problem contemplating the whole time span and fixing the orientation variables to the values obtained by solving the first sub-problem.
The outcome thus obtained (if necessary by including a certain tolerance level on the equations concerning the forces and torques request) is in general a sub-optimal solution of the original problem. Therefore, if the solution isn't judged satisfactory, then the first sub-problem is run again in order to generate a new set of values for the orientation variables to introduce in the second sub-problem for the overall optimization. The entire process can continue until the solutions are deemed satisfactory. Through the applications of NLP or SLP programming, refinements of the current or final solutions can be carried out. The overall search process applies, albeit heuristically, a global optimization logic. Specific MILP models (to be utilized at different levels of approximation) have been conceived to solve (globally) the thruster orientation sub-problem.
From the thruster layout perspective, all of the three spacecraft of the LISA mission, in the last layout developed by the program [10] are equipped with three clusters of three thrusters each, to support the attitude control during the entire scientific phase.
As a first significant step, this thesis focused on the fitting of the methodology depicted above, referring to NGGM context, to the specific and not easier framework regarding the LISA mission. In particular, the MILP models have been adapted to respond to LISA mission demands. Afterwards, the process implemented for NGGM to refine the orientation sub-problem solutions iteratively, has been significantly revised to take into account the specificities relevant to the LISA context. It was necessary to develop a dedicated (local) NLP model to enhance the MILP solutions (that are based on a discretization introduced to eliminate the non-linearity of the problem) that are tainted by the approximation adopted in the resolution process.
To carry out the aforementioned modelling and algorithm-development activities, IBM-CPLEX [4] and Matlab [11] have been utilized as optimizers (CPLEX and Matlab), as well as the programming environment (Matlab).

Once the necessary computational tools had been adequately built up, an extensive and in-depth experimental analysis, addressing the current LISA study phase, followed. Two specific scenarios were investigated, i.e.

- The DFACS system: it is a system, consisting of sensors and actuators, whose scope is to control the spacecraft dynamics in such a way that the main requirement on the residual acceleration is satisfied.
- The AOCS system: it is a system that has to maintain the stability of the spacecraft when this isn't in the final orbit.

The results derived from the whole experimental analysis performed have provided a significant contribution both to the present and upcoming phases of the LISA study.

The remainder of this thesis is structured as follows. Chapter 1 provides an overall insight on the LISA mission and the specific features relevant to the thruster layout optimization problem. Chapter 2 depicts the problem from a mathematical perspective and deepen the model developed. Chapter 3 focuses on the resolution methods for the problem in question and on the computational environment adopted. Chapter 4 explains in detail the solution strategies applied to tackle the problem. Chapter 5 outlines the results obtained thanks to tools illustrated in the previous chapter.

As requested by the host company, a number of technical details have been omitted or appropriately "encrypted", for confidentiality reasons. When this precaution is taken, it will be indicated, throughout the text.

1. Lisa Mission overview

LISA will be the first ever mission to survey the entire Universe with Gravitational Waves. It will allow scientific community to investigate the formation of binary systems in the Milky Way, detect the guaranteed signals from the verification binaries, study the history of the Universe when it was less than 200 million years old, test gravity in the dynamical sector and strong-field regime with unprecedented precision, and probe the early Universe at TeV energy scales. LISA will play a unique and prominent role in the scientific landscape of the 2030s.

The ground-breaking discovery of Gravitational Waves by ground-based laser interferometric Gravitational Wave observatories (LIGO) in 2015 is changing astronomy, giving us access to the high-frequency regime of Gravitational Wave astronomy. This is the realm of stellar mass objects at low redshift. Over the coming years, as the sensitivity of ground-based detectors improves, we will see the growth of a rich and productive Gravitational Wave astronomy. New sources with small mass will be discovered in the low redshift Universe. Already the first observation of Gravitational Waves brought a surprise, because the existence of such heavy stellar origin binary black holes was not widely expected. But the low-frequency window below one Hertz will probably never be accessible from the ground. It is in this window that we expect to observe the heaviest and most diverse objects. Opening a gravitational window on the Universe in the low-frequency regime with the space-based detector LISA will let us go further than any alternative. These low-frequency waves let us peer deep into the formation of the first seed black holes.

Exquisite and unprecedented measurements of black hole masses and spins will make it possible to trace the history of black holes across all stages of galaxy evolution, and at the same time test the General-Relativistic nature of black holes through detailed study of the amplitude and phase of the waveforms of Gravitational Wave strain.

LISA is an all-sky monitor and will offer a wide view of a dynamic cosmos using Gravitational Waves as new and unique messengers to unveil The Gravitational Universe. It provides the closest ever view of the infant Universe at TeV energy scales, has known sources in the form of verification binaries in the Milky Way, and can probe the entire Universe, from its smallest scales near the horizons of black holes, all the way to cosmological scales. The LISA mission will scan the entire sky as it follows behind the Earth in its orbit, obtaining both polarisations of the Gravitational Waves simultaneously, and will measure source parameters with astrophysically relevant sensitivity in a band from below 10−4 Hz to above 10−1 Hz.

The LISA mission will be based on laser interferometry between free flying test masses inside drag-free spacecraft. These test masses, contained within the Gravitational Reference Sensors and effectively identical to the ones flown on LISA Pathfinder, will follow their geodesic trajectories with sub femto $g/\sqrt{\mathrm{Hz}}$ spurious acceleration due to the change in local mass distribution. Two test masses free-fall inside each spacecraft, with each one serving as a geodesic reference end mirror for a single arm of the interferometer. The spacecraft is forced to follow the two test masses along each of the two interferometry axes they define, based on local interferometric position readouts. The test masses are then electrostatically suspended to the spacecraft along the other degrees of freedom, controlled by a combination of interferometric and capacitive position readouts. This system was successfully tested in the LISA Pathfinder mission, and this provides the confident basis for the acceleration performance of the mission.

The observatory will be based on three arms with six active laser links, between three identical spacecraft in a triangular formation separated by 2.5 million km. Continuously operating heterodyne laser interferometers measure with pm/$\sqrt{\text{Hz}}$ sensitivity in both directions along each arm, using well-stabilized lasers at 1064 nm delivering 2 W of power to the optical system.

Again, using technology proven in LISA Pathfinder, the Interferometry Measurement System is using optical benches in each spacecraft. They will be constructed from an ultra-low expansion glass-ceramic to minimize optical pathlength changes due to temperature fluctuations. 30 cm telescopes transmit and receive the laser light to and from the other spacecraft.

The mission configuration is shown in **FIGURE 1**.

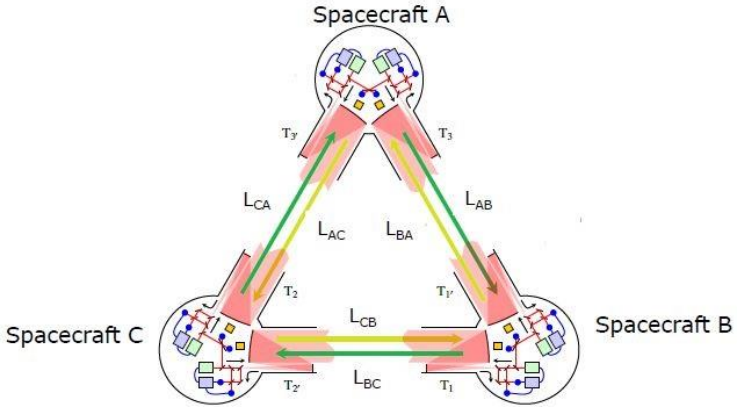

FIGURE 1 LISA MISSION CONCEPT

The proposed orbit for LISA is an Earth-trailing heliocentric orbit between 50 and 65 million km from Earth, with a mean inter-spacecraft separation distance of 2.5 million km. A reference orbit has been produced, optimised to minimise the key variable parameters of inter S/C breathing angles (fluctuations of vertex angles) and the range rate of the S/C, as both of these drive the complexity of the payload design, while at the same time ensuring the range to the constellation is sufficiently close for communication purposes. The orbital configuration is depicted in **FIGURE 2**.

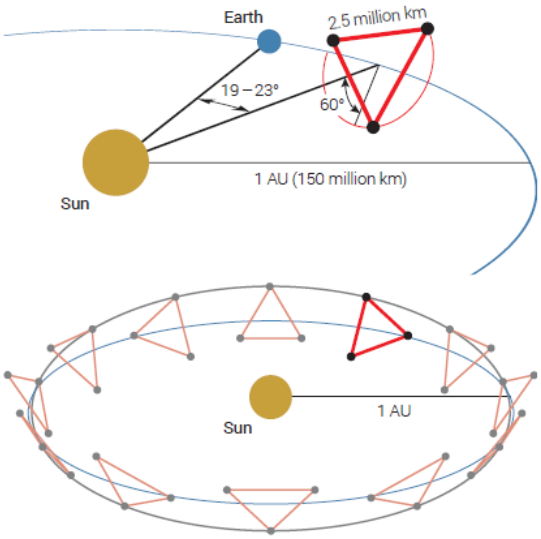

FIGURE 2 LISA ORBIT

2. Mathematical Problem

This section discusses the mathematical problem relevant to this dissertation form an overall point of view, referring to a general dynamic system [1], [2]. Considering a rigid body S over a given timeframe $[0,T]$, is defined an appropriate orthogonal reference frame $O(x,y,z)$ centred in the centre of gravity of S body. This reference frame is the only considered throughout this work, so all the vectors are referred w.r.t. it. The timeframe $[0,T]$ may be equipartitioned into a set of sub-intervals, each of which of Δ duration, since the controller on board works on a discrete time step. The timeframe is therefore delimited by $N_I + 1$ instants with $i = 0, 1, \dots, I$. The timeframe is divided in 365 instants, corresponding to a single day over a single year. The spacecraft should be placed in an orbit near the L2 point, so the forces and torques acting on it are yearly periodic. It is assumed that, at each instant i, a force $\boldsymbol{F_i} = (F_{xi}, F_{yi}, F_{zi})$ and a torque $\boldsymbol{T_i} = (T_{xi}, T_{yi}, T_{zi})$, representing the overall control request, have to be exerted by the system through a set of N_A actuators with $r \in \{1, \dots, N_A\}$ at any instant.

Before going deep in the problem, the following notation is introduced:

$\|w\|$ is the Euclidean norm of vector w;

I is the set of time instants, starting from 0, N_i is the last instant;

N_A is the number of actuators;

$\boldsymbol{F_i} = (F_{xi}, F_{yi}, F_{zi})$ is the overall force requested by the controller from the actuators at instant i;

$\boldsymbol{T_i} = (T_{xi}, T_{yi}, T_{zi})$ is the overall torque requested by the controller from the actuators at instant i;

$\boldsymbol{v_r} = (v_{xr}, v_{yr}, v_{zr})$ is the unit vector representing the orientation of the r actuator.

$\boldsymbol{v} = \left((v_{1x}, v_{1y}, v_{1z})^T, \dots, (v_{rx}, v_{ry}, v_{rz})^T, \dots, (v_{N_Ax}, v_{N_Ay}, v_{N_Az})^T\right)$ is the matrix whose columns are column vectors associated which each $\boldsymbol{v_r}$.

$f_{ri} = (f_{rx}.f_{ry}, f_{rz})$ is the force exerted by the actuator r at instant i.

u_{ri} are, for each actuator r, the Euclidean norm $\|\boldsymbol{f_{ri}}\|$ of the force exerted at instant i.

$p_r = (p_{rx}, p_{ry}, p_{rz})$ is, for each actuator r, the application-point vector of f_{ri}.

$p \times v$ is the matrix whose columns are the column vectors associated with each cross product $p_r \times v_r$.

$\overline{U}_r, \underline{U}_r$, are respectively, for each actuator r, the upper and the lower bounds imposed on u_{ri}. It is assumed that both are time-independent.

$D_{vr} \subset \boldsymbol{R}^3$ is a compact domain delimited by specific conditions on actuator r orientation.

Once the notation was introduced the problem can be stated as follow:

$$\forall\, i \in I \qquad \begin{pmatrix} v \\ p \times v \end{pmatrix} \begin{pmatrix} u_{1i} \\ \cdots \\ u_{N_A i} \end{pmatrix} = \begin{pmatrix} F_i \\ T_i \end{pmatrix} \tag{1}$$

$$\forall r \in A \qquad \|\boldsymbol{v_r}\| = 1 \tag{2}$$

$$\forall r \in A, \forall i \in I \qquad u_{ri} \in \left[\underline{U}_r, \overline{U}_r\right] \tag{3}$$

$$\forall r \in A \qquad \boldsymbol{v_r} \in D_{vr} \tag{4}$$

Equations (1) express the assigned system control law that, in a more explicit vector formulation, reads as follow:

$$\forall i \in I \sum_{r \in A} u_{ri}\, \boldsymbol{v_r} = \boldsymbol{F_i}$$

$$\forall i \in I \sum_{r \in A} \boldsymbol{p_r} \times (u_{ri} \boldsymbol{v_r}) = \boldsymbol{T_i}$$

Equations (2) are a normalization conditions, in order to obtain a unit vectors for the orientations of forces exerted by the corresponding actuator. The final optimization problem can be worded as follow:

Chose the value for parameters v, as well as for variables u, minimizing a given cos function $f(p, v, u)$ and subject to conditions (1), (2), (3) and (4).

The application-points of each actuators r are given as input and don't figure as a problem's parameter. Furthermore, for each actuator r, D_{vr} is assumed to be a three-dimensional domain in which the orientation's vector must be and it is different from one thruster to other in order to consider some limitations that will be discussed in the following chapters. $\underline{V}_r$ and $\overline{V}_r$ represent, respectively, the lower and the upper bounds of the allowed orientations.

After using this simplifications and specifications, the model stated above can be read as follow:

$$\min \sum_{\substack{r \in A \\ i \in I}} f_r(u_{ri}) \tag{5}$$

$$Subject\ to$$

$$\forall i \in I \qquad \begin{pmatrix} v \\ P \times v \end{pmatrix} \begin{pmatrix} u_{1i} \\ \cdots \\ u_{ri} \\ \cdots \\ u_{N_A i} \end{pmatrix} = \begin{pmatrix} F_i \\ T_i \end{pmatrix} \tag{6}$$

$$\forall r \in A \qquad v_{rx}^2 + v_{ry}^2 + v_{rz}^2 = 1 \tag{7}$$

$$\forall r \in A\ \forall i \in I \qquad u_{ri} \in \left[\underline{U}_r, \overline{U}_r\right], \boldsymbol{v_r} \in D_{vr} \tag{8}$$

Equations (2) have been substituted with the more convenient expression (7). Furthermore, it's assumed that the cost associated with each actuator is time-independent. The last expedient that must be noticed is the convention denoting the constants with capital letters and the variables with lower-case characters ($p \times v$ has been substituted with $P \times v$).

In order to obtain an acceptable approximation, the objective function appearing in (5) is reduced to a linear one. (5) assumes the explicit expression shown below

$$\min \sum_{\substack{r \in A \\ i \in I}} K_r u_{ri} \tag{9}$$

Where K_r are a positive constant, representing for each actuator r the associated cost per unit force.

An important assumption must be made regarding the application-points. It is assumed that the orientations of each thrusters and the application-points are time independent and remain the same during all the mission life.

It's always possible to identify a fixed orientation for the actuators to satisfy the control demand, instant by instant, over the entire span and in compliance with all operational restrictions. Error variables can be added in (6), defined within given tolerance ranges and readjusting, if necessary, the objective function introducing the total error as a term to be minimized. Equations (6) can be substituted with the following:

$$\forall i \in I \qquad \begin{pmatrix} v \\ P \times v \end{pmatrix} \begin{pmatrix} u_{1i} \\ \dots \\ u_{ri} \\ \dots \\ u_{N_A i} \end{pmatrix} = \begin{pmatrix} F_i + \varepsilon_{Fi} \\ T_i + \varepsilon_{Ti} \end{pmatrix} \tag{10}$$

$$\forall i \in I \qquad -E_F \leq \varepsilon_{Fi} \leq E_F, -E_T \leq \varepsilon_{Ti} \leq E_T \tag{11}$$

$\varepsilon_{Fi} = \left(\varepsilon_{Fxi}, \varepsilon_{Fyi}, \varepsilon_{Fzi}\right)^T$, $\varepsilon_{Ti} = \left(\varepsilon_{Txi}, \varepsilon_{Tyi}, \varepsilon_{Tzi}\right)^T$, $E_F > 0$ and $E_T > 0$ are the chosen admissible levels of tolerance (expressed as column vectors).

The difficulty of the overall mathematical model under the above consideration is essentially generated by constraints (6) and (7) since are non-convex constraints. Mainly due to the state-of-the-art algorithms, the problem structured as far is very tricky to be solved.

The applications-point were assumed as constants, so equations (6) became bilinear. This suggest splitting the problem in two sub-problem, involving two different models. The first, known as *Layout*, focuses on the thrusters' orientations considering just a representative set of instants and trying to reduce the total consumption. The second, the *Continuous* model, utilizes the whole set of instants and the orientation given by the *Layout* model in order to confirm the feasibility of the solutions found thanks to the *Layout* model.

In the *Layout* model all the variables of the general problem are treated as such (i.e. v and u) resulting in a quadratic constraint problem, while the *Continuous* model, having as variables just u, become linear.

The *Continuous* model could be seen as a tool to verify the feasibility of solutions found in the *Layout* model, covering the whole set of instants.

The two models will be better explained in the following paragraphs.

2.1. Layout Problem

The general formulation of the *Layout* model can be expressed as follows:

$$\min \sum_{r \in A} \sum_{i \in \underline{I}} K_r u_{ri}$$

$$Subject\ to$$

$$\forall i \in \underline{I} \qquad \begin{pmatrix} \boldsymbol{v} \\ P \times \boldsymbol{v} \end{pmatrix} \begin{pmatrix} u_{1i} \\ \dots \\ u_{ri} \\ \dots \\ u_{N_A i} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F_i} \\ \boldsymbol{T_i} \end{pmatrix} \tag{12}$$

$$\forall r \in A \qquad v_{rx}^2 + v_{ry}^2 + v_{rz}^2 = 1 \tag{13}$$

$$\forall i \in \underline{I} \qquad u_{ri} \in \left[\underline{U}_r, \overline{U}_r\right] \tag{14}$$

In this model the available orientations $\boldsymbol{v}$ are discretized. This approach is quite advantageous because there are only three orientation variables for each thruster. The aforementioned discretization requires the introduction of logical variables that, therefore, transforms the original nonlinear model into a mixed integer linear programming (MILP). The set of instants considered are a subset of the initial timeframe, so i.e. $\underline{I} \subset I$. All the admissible orientations are associated to a semi-sphere of unit radius. All the vectors are centred in the applications-point and directed externally, w.r.t. the corresponding satellite surface. A local reference frame is defined for each semi-sphere, centred in applications point and with axis parallel to the reference frame centred in a neighbourhood of geometrical centre of spacecraft. Each orientation can be identified through two spherical coordinates. The angle α $(0 \leq \alpha \leq 2\pi)$ identifies the polar coordinate with $\alpha = 0°$ corresponding to y-axis while the β angle $\left(0 \leq \beta \leq \frac{\pi}{2}\right)$ represents the azimuthal coordinate with $\beta = 0$ corresponding to the z-axis.

Both α and β must be partitioned into two finite subsets, dividing the corresponding intervals by a pre-selected number.

2.2. Linear Problem

The *Continuous* model can be expressed as

$$\min \sum_{r \in A} \sum_{i \in I} K_r u_{ri} \tag{15}$$

$$Subject\ to$$

$$\forall i \in I \quad \forall r \in A \qquad \begin{pmatrix} \boldsymbol{V} \\ P \times \boldsymbol{V} \end{pmatrix} \begin{pmatrix} u_{1i} \\ \dots \\ u_{ri} \\ \dots \\ u_{N_A i} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F_i} \\ \boldsymbol{T_i} \end{pmatrix} \tag{16}$$

$$\forall i \in I \qquad u_{ri} \in \left[\underline{U}_r, \overline{U}_r\right] \tag{17}$$

In this model all the variables $\boldsymbol{V}$ are fixed on the solution obtained from the *Layout* model. The following general considerations accounts for the instance's model generation and determines its size:

- Number of thrust variables = (number of thrust) x (number of instants);
- Number of orientation variables = (number of thruster) x (3 direction cosines per thruster);
- Number of lower bounds (for the thrust variables) = number of thrust variables;
- Number of upper bounds (for the thrust variables) = number of thrust variables;
- Number of equations (corresponding to (16)) = (6 matrix rows) x (number of instants);

For each thruster r, each sub-angle h of α and each sub-angle k of β, a binary variable is introduced with the meaning:

$\delta_{rhk} = 1$	If thruster r takes the discretized orientation (h, k)
$\delta_{rhk} = 0$	Otherwise.

The total number of binary variables results in being:

Number of binary variables = [(number of sub-angles α) x (number of sub-angles β)] x (number of thrusters)

The total number of the discretization constraints is therefore the following:

number of bound constraints = (number of thruster) x (number of sub-angles of α) x

x (number of sub-angles of β) x (6 matrix rows corresponding to (16)) x

x (number of instants) x (2 bounds associated with each thruster).

3. Overall Optimization Process and Computational Environment

In many practical problems, such the problem studied in this work, some variables make sense only if they have integer value. If requiring integer values is the only way in which a problem deviates from linear programming formulation, then it is an integer programming (IP) problem.

If only some of the variables are required to have integer values, this model is referred to as Mixed Integer Linear Programming (MILP).

It may seem that IP problems should be relatively easy to solve. After all, linear programming problems can be solved extremely efficiently, and the only difference is that IP problems have far fewer solutions to be considered. In fact, pure IP problems with a bounded feasible region are guaranteed to have just a finite number of feasible solutions. Unfortunately, there are two fallacies in this line of reasoning. One is that having a finite number of feasible solutions ensures that the problem is readily solvable but finite numbers can be astronomically large.

The second fallacy is that removing some feasible solutions (the non-integer ones) from a linear programming problem through some cuts will make it easier to solve. To the contrary, it is only because all these feasible solutions are there that the guarantee can be given that there will be a corner-point feasible that is optimal for the overall problem. This guarantee is the key to the remarkable efficiency of the simplex method.

Consequently, most successful algorithms for IP problem involve the simplex method as much as they can by relating portions of the IP problem under consideration to the corresponding linear programming problem. This corresponding linear programming problem is commonly known as LP relaxation.

An algorithm that solves MILP problems through an LP relaxation is the *Branch-And-Bound* technique that will be well explained in the following section [8], [12].

3.1. A Branch-And-Bound algorithm for Mixed Integer Programming

Because any bounded MIP problem has only a finite number of feasible solutions it's natural to consider using some kind of enumeration procedure for finding an optimal solution. But a finite number can be, and usually is very large. Therefore, it's imperative formulate an enumeration procedure that cleverly examines only a tiny fraction of feasible solution in order to obtain the best one. A procedure that works in this way is the *Branch-And-Bound* technique.

The basic concept underlying the branch-and-bound technique is to divide and conquer. Since the original problem is too 'large' to be solved directly, it's divided into smaller and smaller sub-problems until the sub-problem can be conquered. The dividing (*branching*) is done by partitioning the entire set of feasible solutions into smaller and smaller subsets. The conquering (*fathoming*) is done partially by *bounding* how good the best solution in the subset can be and then discarding the subset if its bound indicates that it cannot possibly contain an optimal solution for the original problem.

The three basic steps – branching, bounding and fathoming – will now be discussed.

In a pure IP problem, the first step consists in partitioning the set of all feasible solutions into subsets, fixing the value of one integer-restricted variables at 0 for one subset and 1 for the other n-1 subsets, with n number of integer-restricted variables. In this way the original

problem, after the LP relaxation, is divided into subproblems by a tree with branches from the LP relaxation solution (corresponding to the whole problem having all feasible solutions) to the n nodes corresponding to the n sub-problems.

This tree, that will continue generates branches iteration by iteration, is referred to as the solution tree for the algorithm. The variables used to do this branching t any iteration by assigning values to the variables is called the branching variable. In a pure IP problem, the most basic choose for assigning the branching variable is in their natural order. In a MIP problem only the integer-restricted variables that have a non-integer value in the optimal solution for the LP relaxation of the current sub-problem.

The general integer-restricted variable could have a large number of possible integer values and it would be inefficient to create and analyse many subproblems by fixing the variable at its integer values. Therefore, what is done instead is to create just two new subproblems (as a pure IP problem) by specifying two ranges of values for the variable.

Let x_j be the branching variable and let x_j^* be its non-integer value in the optimal solution for the LP relaxation of the current sub-problem. Let's denote

$$[x_j^*] = \text{greatest integer} \leq x_j^*$$

We have for the range of values for the two new subproblems

$$x_j \leq [x_j^*] \quad \text{and} \quad x_j \geq [x_j^*] + 1$$

Each inequality became an additional constraint for that new subproblem. This process is made at each iteration until the branching variable, now became a recurring branching variable, can assume only an integer value. **FIGURE 3** shows an example of the process here discussed.

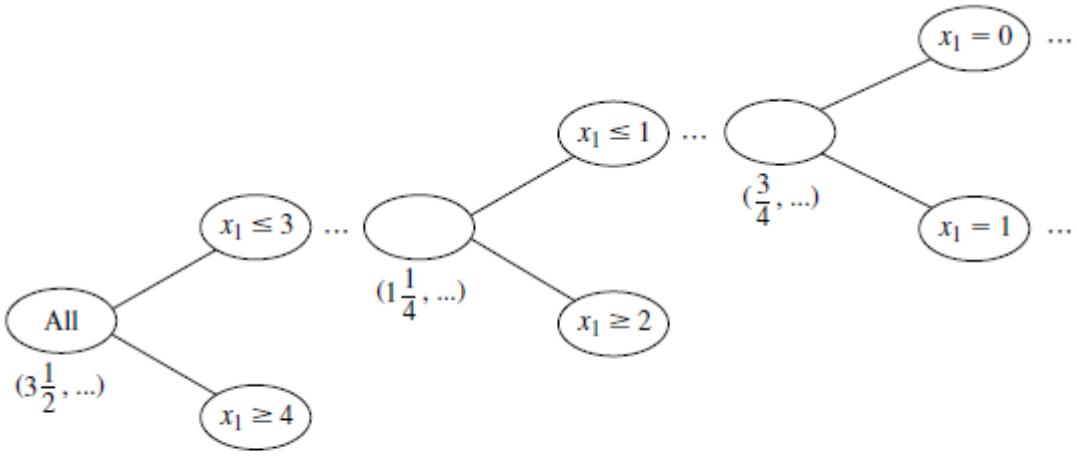

FIGURE 3 A BRENCH-AND-BOUND EXAMPLE

For each of these subproblems, now it's necessary to obtain a bound on how good its best feasible solution van be. The standard way of doing this is to quickly solve a simpler relaxation of the subproblem. A relaxation of the problem can be obtained simply by deleting one set of constraints that made the problem difficult to solve. Since the most troublesome constraints for

a MIP problem are those requiring integer variables to be integer, the most widely used relaxation is the LP relaxations that deletes this set of constraints.

The solution obtained by the LP subproblem establishes a bound for the problem of which is the linearization. For a minimization problem, for example, if the LP solution is Z then all the feasible solutions of the subproblem have a solution $Z^* \geq Z$.

A problem can be fathomed and then dismissed, in three ways described below.

A subproblem can be fathomed if its integer solution has a value lower than the best solution found so far, called as incumbent solution Z^*. If this is true, then the solution found become the new incumbent solution.

The above result suggests a second key fathoming test. There is no reason to consider further any subproblem whose bound $\geq Z^*$ since such a subproblem can't have a feasible solution better than the incumbent.

The third way of fathoming is straightforward. If the simplex method finds that a subproblem's LP relaxation has no feasible solutions then the subproblem itself must have no feasible solutions, so it can be fathomed.

Finally, the search is conducted for an optimal solution by retaining for further investigation only those problem that could possibly have a feasible solution better than the current incumbent.

3.2. CPLEX utilization for solving MILP problem

IBM-ILOG CPLEX [4] is the solver used to tackle the problem. It is an optimization tool developed by IBM to solve, first of all linear optimization problems. To solve linear programming problems CPLEX implements optimizers based on the simplex algorithms. CPLEX is also a tool for solving MILP problem. The CPLEX mixed integer optimizer solves MILP models using a very general and robust algorithm based on branch & bound technique explained in 3.1. While MILP models have the potential to be much more difficult than their LP counterparts, it is also the case that large MILP models are routinely solved in many applications. A great deal of algorithmic development effort has been devoted to establish default CPLEX parameters settings that achieve good performance on a wide range of MILP problems. However, there is a wide set of parameters from which the user can choose and that can better suit the problem in analysis.

In 2 has been illustrated and explained the mathematical model and has been emphasized the necessity to limit the dimensions of the problem in order to achieve an affordable computational cost. The IBM CPLEX parameters [5] may play a crucial role in reducing the problem size and make it solvable, in particular they have a strong influence on runtimes and on memory used by the solver.

These parameters, in fact, influencing the way finding solutions, allow to concentrate efforts in exploring solutions in some directions as the search of multiple feasible solutions or the best ones without focusing the feasibility.

An experimental phase has been carried out to evaluate possible parameter settings combinations and their performance. It's worth noting that the influence of parameters is strongly dependent on the model, so the results hereafter reported can't be transposed to the full thruster optimization problem. Since the model with the whole timeframe is too large to be

studied, it has been decided to consider a reduced timeframe (32 instants) and a not very thin angles' discretization.

Another important aspect that must be considered is that the cases studied in this work are six and not just one. It's clear that each case is different from the others and a combination of parameters that give a good response in one case could fails in another one. However, especially thanks to application-points, the problem presents at least a geometrical symmetry that leaded to consider just one case and not all the possible, in the hypothesis that the results would be the same.

The experiments have been conducted on the simplified thruster orientation problem shown in TABLE 3. 1:

Simplified Problem		
PROBLEM SIZE	Number of instants	32
	Number of thrusters	8
MESH SIZE	Number of alpha points	35
	Number of beta points	10
	Alpha range	$[0,2\pi)$
	Beta range	$\left[0,\frac{\pi}{2}\right]$
PROBLEM BOUNDS AND COEFFICIENTS	K	1
	Thrust lower bound	1e-5 μN
	Thrust upper bound	100 μN
	Error lower bound	0
	Error upper bound	0
FINAL MODEL SIZE	Number of rows	3326
	Number of columns	50512
	Number of nonzero	375012
	Number of integer variables	272

TABLE 3. 1 SIMPLIFIED PROBLEM

The parameters considered are the following:

- **NODE SELECTION STRATEGY**

Sets the rule for selecting the next node to process when the search is backtracking. The depth-first strategy (NODESEL 0) chooses the most recently created node and seek to set all binary variables to integer values as soon as possible. The best-bound strategy (NODESEL 1), that is the default strategy, choose as node to be explored that with the best objective function for the associated LP relaxation. The best-estimating strategy (NODESEL 2) selects the node with the best estimate of the integer objective value that would be obtained from a node once all integer infeasibilities are removed, thus choosing the node with a greater number of integer variables. There's also an alternative best-estimate search (NODESEL 3).

- **MIP STRATEGY**

Controls trade-offs between speed, feasibility, optimality and moving bounds in the MIP problems. The default setting (MIP 0) enables finding a rapid proof of an optimal solutions balancing efforts in order to find high quality feasible solutions early in optimization. When the parameter is set to 1 (MIP 1), the solver frequently will generate more feasible solutions as it optimizes the problem sacrificing speed. The specular parameter setting (MIP 2) apply less efforts to finding feasible solutions early increasing the speed. A greater emphasis is placed on proving optimality through moving the best bound value when the parameter is set on 3 (MIP 3), in this way the detection of feasible solutions became incidental. The last possibility is setting the parameter on 4 (MIP 4). In this case the MIP optimizer works hard to find high quality feasible solutions that otherwise are very difficult to find. It's a good alternative when setting the parameter 1 (focusing on feasibility) doesn't find solutions of acceptable quality.

- **FEASIBILITY PUMP**

At the default setting (PUMP 0), CPLEX automatically chooses whether or not to apply the feasibility pump heuristic on the basis of characteristics of the model. The feasibility pump heuristic can be forced to be off (PUMP -1) or on. If the parameter is set to 1 (PUMP 1), the feasibility pump tries to find a feasible solution without taking the objective function into account. If the parameter is set on 2 (PUMP 2), instead, the heuristic usually finds solutions of better objective value sacrificing the feasibility.

The outputs of analysis take in account the following quantities:

- Output Status

The cause for which CPLEX terminated the run. There are many output status messages that CPLEX could return, the ones encountered in the analysis are mainly two:

  - 107: time limit exceeded, and integer solutions exist.
  - 109: terminated because an error (frequently an Out of memory message) but integer solutions exist.

A 107-output message is an encouraging one because it would indicate that setting used allow for a correct use of the computational resources and could give good solutions over the time limit.

- Number of solutions found

The number of feasible solutions evaluated. A greater number of solutions furnishes a good sample on which evaluate the optimality.

- Tree size

The maximum size reached by the tree during the run. It represents how many branches are made by the solver and therefore is a good indicator on how well the computational resources are used.

- Objective function value

The lowest this value is, better is the solution found. In the problem in exam, in fact, a lower function value indicates a lower fuel consumption.

TABLE 3. 2 summarize the analysis conducted.

Strategy	Acronym	Output	Tree size [Mb]	Objective Function Value	Number of solutions	Time [s]
Node Selection	NODESEL (0)	109	9.47	5111.58	9	830.83
	NODESEL (1)	109	14.68	4947.91	1	2027.24
	NODESEL (2)	109	13.56	4497.91	3	1251.89
	NODESEL (3)	109	13.59	4619.74	2	997.2
MIP Emphasize	MIP (0)	107	12.9	4729.31	2	10800
	MIP (1)	107	476.14	3963.41	54	10800
	MIP (2)	107	30.94	4109.58	14	10800
	MIP (3)	109	112.35	4229.08	2	3742.66
	MIP (4)	109	163.15	4150.89	16	7053.06
Heuristic Pump	PUMP (-1)	107	10.35	4437.76	7	10800
	PUMP (0)	107	12.9	4729.31	2	10800
	PUMP (1)	107	44.46	4350.98	11	10800
	PUMP (2)	107	9.79	4275.89	6	10800

TABLE 3. 2 CPLEX PARAMETERS TEST OUTCOMES

As can be seen, each 'Node Selection' strategy return as output status an out of memory message even the three size is very small, 14.68 Mb at most. This behaviour is predictable since the 'Node Selection' strategy focus on cutting many branches in order to maintain a small size tree. The runtime never exceeds the hour because the solver fills the memory very soon because CPLEX default working memory (2048 Mb) is saturated and can't swap to disk files.

The 'MIP Emphasize' strategy uses generally more memory than the 'Node Selection', especially the MIP (1) that focus on finding feasible solution while the solver runs. The number of solutions found thanks to this parameter, in fact, is very large compared to the other strategies at the cost of a lower speed that is reflected in a maximum runtime. It's interesting to notice that even in the MIP (4) strategy, as in the MIP (1), focus on finding feasible solutions, the working memory is overtaken.

The 'Heuristic Pump' strategy doesn't allocate a large amount of memory on the tree size in all the four cases analysed and ever exploits all the runtime so it's a parameter that should be taken

into account for the study. A last consideration can be made regarding the choice between finding feasible solutions rather than optimal ones.

All the strategies involving the search of feasible solutions as many as possible (MIP (1) MIP (4) and PUMP (2)), even if allocate more memory than the other strategies, they find the lower solutions.

In conclusion of this brief analysis it was chosen to use the MIP (1) strategy with an allocation memory halved, 1024 Mb, in order to swap earlier to disk files.

3.3. Matlab optimization environment

A non-linear problem solver in MATLAB operational environment has been utilized as a further analysis' tool. In this environment, in fact, is present a function known as FMINCON that includes the utilization of various algorithm in order to minimize the object function [11].

These algorithms are briefly explained in the following section:

- **Interior Point**

This algorithm considers the resolution of a series of approximate minimization problems.

$$\min_x f(x), \text{ subject to } h(x) = 0 \text{ and } g(x) \leq 0$$

The slack variables were introduced in order to transform the constraints' inequalities into equalities in order to state the problem as follow:

$$\min_{x,s} f_\mu(x,s) = \min_x f(x) - \mu \sum_i \ln(s_i), \text{ subject to } h(x) = 0 \text{ and } g(x) + s = 0$$

The number of slack introduced is equal to the constraints g and each s_i is strictly positive, as the parameter μ that, decreasing, permits at minimum of f_μ to get close to the minimum of f. Through the introduction of slack variables and the logarithmic function, known as *barrier function*, the problem was simplified into an approximated one and this is solved through the resolution of the following function, known as *merit function*:

$$f_\mu(x,s) + v\|h(x), g(x) + s\|$$

The parameter v increases as the number of iterations raises, in order to force the solution to feasibility. The approximate problem can be solved through two ways:

- A direct step $(\Delta x, \Delta s)$ obtained from the resolution of the system

$$\begin{bmatrix} H & 0 & J_h^T & J_g^T \\ 0 & S\Lambda & 0 & -S \\ J_h & 0 & I & 0 \\ J_g & -S & 0 & I \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ -\Delta y \\ -\Delta \lambda \end{pmatrix} = -\begin{bmatrix} \nabla f - J_h^T y - J_g^T \lambda \\ S\lambda - \mu e \\ h \\ g + s \end{bmatrix}$$

  The matrix H is the Hessian of the Lagrangian of f_μ

$$H = \nabla^2 f(x) + \sum_i \lambda_i \nabla^2 g_i(x) + \sum_j \lambda_j \nabla^2 h_j(x)$$

- J_h^T and J_g^T stand for the Jacobian of respectively, constraint's functions h and g.
- $S = diag(s)$.

- λ stand for the Lagrangian multiplier vector associated with the constraints h.
- $\Lambda = diag(\lambda)$.
- y stands for the Lagrangian multiplier vector associated with the constraints g.
- e stands for a unitarian vector of the same size of g.

- A conjugate gradient step, within a trust region of radius R, calculated through the minimization of a quadratic approximation of the approximated problem, subject to the constraints:

$$\min_{\Delta x, \Delta s} \nabla f^T(x) + \frac{1}{2}\Delta x^T \nabla^2_{xx} L \Delta x + \mu e^T S^{-1} \Delta S + \frac{1}{2}\Delta S^T S^{-1} \Lambda \Delta S$$
$$\text{subject to } g(x) + J_g \Delta x + \Delta s = 0, h(x) + J_h \Delta x = 0$$

where

$$\nabla_x L = \nabla_x f(x) + \sum_i \lambda_i \nabla g_i(x) + \sum_j \lambda_j \nabla y_j(x) = 0$$

- **Sequential Quadratic Programming**

This algorithm works on a quadratic approximation of the following Lagrangian function:

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i \cdot g_i(x)$$

The lagrangian function is utilized to formulate the following quadratic sub-problem:

$$\min_{d \in \Re^n} \frac{1}{2} d^T H_k d + \nabla f(x_k)^T d$$

$$\nabla g_i(x_k)^T d + g_i(x_k) = 0, i = 1, \dots, m_e$$

$$\nabla g_i(x_k)^T d + g_i(x_k) \leq 0, i = m_e + 1, \dots, m$$

Where d is a search direction. The sub-problem is solved iteratively in three steps.

In the first step is updated the Hessian matrix:

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k s_k s_k^T H_k^T}{s_k^T H_k s_k}$$

Where

$$s_k = x_{k+1} - x_k$$

$$q_k = \left(\nabla f(x_{k+1}) + \sum_{i=1}^{m} \lambda_i \cdot \nabla g_i(x_{k+1}) \right) - \left(\nabla f(x_k) + \sum_{i=1}^{m} \lambda_i \cdot \nabla g_i(x_k) \right)$$

The Hessian matrix must be positive defined, to do this the q_k is modified at each iteration if necessary.

The second phase envisages the calculation of a feasible point, if exists, and, in the last phase, an iterative sequence of feasible points that converges to s_k is generated. This term is used to update the solution at the following iterate as

$$x_{k+1} = x_k + \alpha_k d_k$$

Where α_k is a step length parameter that allows a reduction of a merit function thus leading to the solution of the problem.

- **Active Set**

The active set algorithm is similar to the SQP because both involve the solution of a quadratic programming problem. The SQP algorithm discussed above, in fact, differs from active set in the way the feasibility is obtained because every iterative step may only be taken within region constrained by bounds, providing better performances, w.r.t. SQP, for the specific case.

4. Solution Strategies

As mentioned in 2, the problem studied is intrinsically complex due to the nature and the number of decisional variables generated and the constraints that must be satisfied by the variables themselves in order to comply with the design requirements. During years the computational power that optimization tools can offer allow tackling more challenging problems, thanks to the use of techniques that reduce the problem size and complexity generating solutions confirmable in the general problem. Of course, the solutions of the reduced problem can provide a starting point for a global search. The number of time instants is a variable that strongly influence the problem size, especially the number of constraints generated, thus influencing the time with which a solution can be found. Therefore, the first action to do is to elaborate an algorithm that reduces the number of instants considered, generating a set of instants of a desiderated size representative of the whole set of instants.
The algorithm used in this work derives from a previous study [9].
The algorithm's logic and the main structure are presented in the following paragraph.

4.1. Instance selection

The developed algorithm implements different heuristics approaches to select representative sub-sets of instants from supplied control laws. Such control law is implemented in order to contrast the external forces and torques that act on spacecraft. The main strength acting on spacecraft is along its z-axis due to solar pression radiation. It's worth notice that the torques' and forces' trends are very regular thanks to the specific spacecraft's orbit, an orbit around a Lagrangian point that, by definition, is quite regular. The forces history is shown in **FIGURE 4**, while the torques history in **FIGURE 5**.

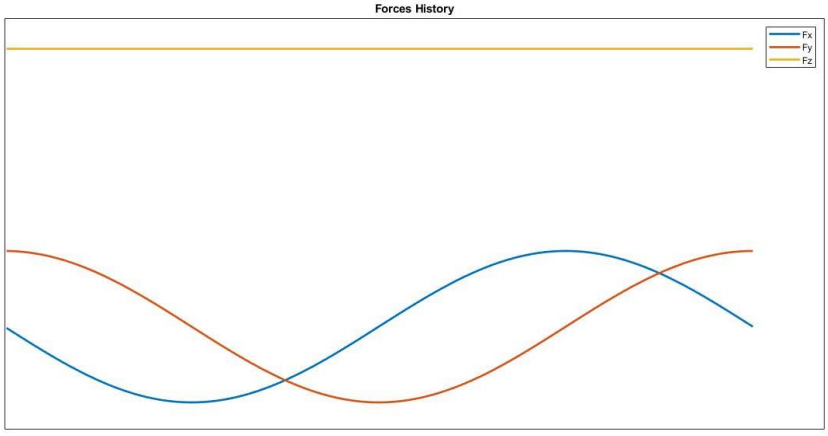

FIGURE 4 SCIENCE PHASE FORCES

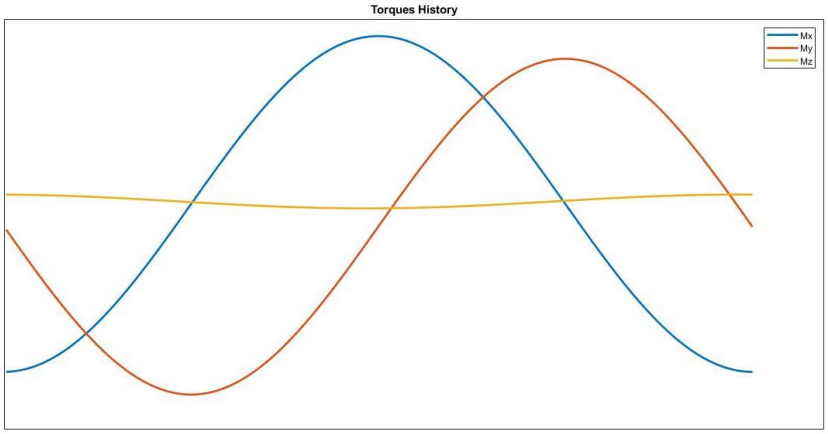

FIGURE 5 SCIENCE PHASE TORQUES

The forces and the torques given as input for the optimization have a strong influence on the solution of the problem. Therefore, varying the forces and the torques in input the solutions change and a configuration feasible for a set of forces and torques could not be feasible for another one. The input for the analysis comes from the science mission phase but there are other phases in which the DFCAS could acts that will discussed in 5.2.
The global problem is tackled through sequential steps:

- Input Elaboration

To construct a selection model, it's essential to define different selection criteria relevant from an engineering point of view. In order to do this, two criteria have been implemented: the first is a feasible criterion aiming to include in the final set of chosen instants all critical conditions encountered in the control law. Sixteen critical points are selected, a maximum and a minimum for every component of force and torque to be exerted (on a spacecraft centred coordinate system) and a maximum and minimum for the magnitude of the force and torque vectors. The second selection criterion aids in the research of optimality in the thruster model by providing representative sets whose distribution of command difficulty resembles that of the original control law. The magnitude of the force and torque vectors is calculated and assigned to a load level. Both force and torque vectors show individual distribution of instants within each load level. A number of load classes is generated from the combined load level of force and torque, each with its relative distribution value within said control law. The magnitude of the force and torque vectors is calculated and assigned to a load level. To produce a fully representative set the control law is evaluated and assigned with regards to both load levels for force and torque magnitudes simultaneously. The final representative set will have to reproduce as closely as possible this distribution. This is obtained by rounding to the closest integer the product of the load class distribution percentage with the final set size selected by the user. This criterion has been chosen with regards to the final use of the set of instances generated: it is desirable to orient the thruster to best suit the most frequent conditions. It is important to note that this criterion is second in importance to the inclusion of critical conditions: if a certain load class has more critical instants to be selected than the distribution allows, the feasibility requirement overrides the distribution criteria leading to a less representative final set, but that complies with critical conditions constraint. Finally, in order to reduce the size of the problem tackled during the selection, the original control law is very simply skimmed: given the continuous

nature of these timelines, it is reasonable to reduce the choosing domain by selecting conditions every x instants.

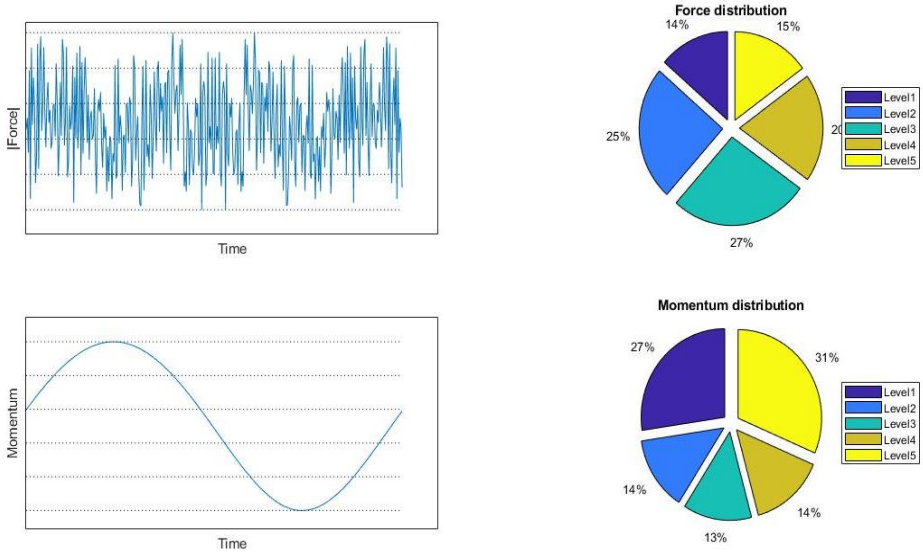

FIGURE 6 MAGNITUDE CLASSIFICATION

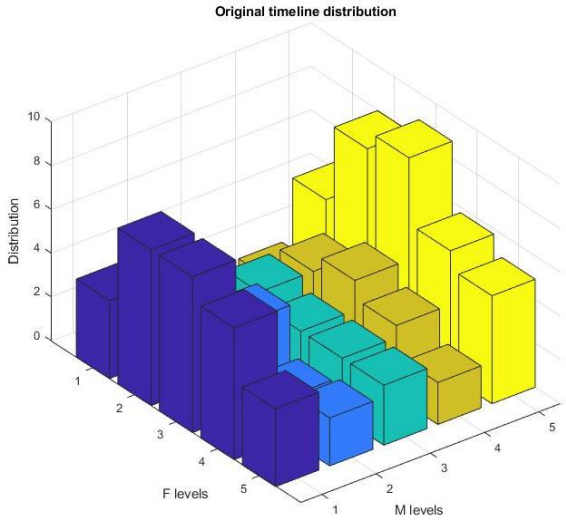

FIGURE 7 LOAD CLASS DISTRIBUTION

- Per-load-class selection

Having sorted each instant in a load class, each subset of instants may be optimized independently to output a smaller yet optimized set of instants representative of each load class. During the current solving step, the final sample size for each load class must be defined by the user in such a way that the problem is greatly reduced in size to ease the load of integer variables on the following step. The feasibility constraint, on the other hand, is still treated in this step, with the only difference being that for each load class only the pertinent critical conditions must be maintained.

- Final Selection

Once a local optimization has been executed for each individual load class, the output subsets may be merged into a single timeline. To further reduce the sample size and to apply the optimality selection criteria, a second pass of the optimization engine is run on the merged timeline. Furthermore, the final application of the distribution constraint allows the optimization engine to output a final representative set of instants whose size matches as closely as possible the reduced size desired by the user in the first place, thus completing the selection process.

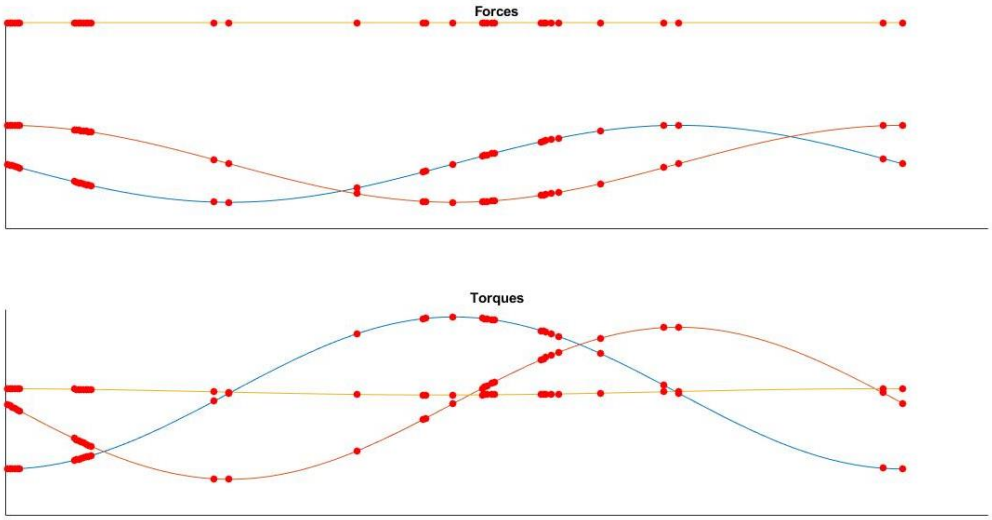

FIGURE 8 32 INSTANTS SOLUTION

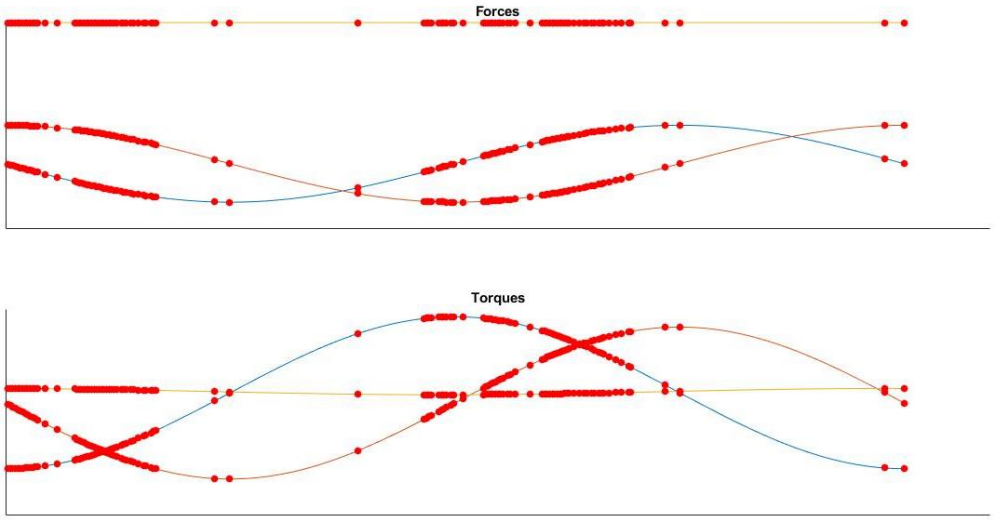

FIGURE 9 101 INSTANTS SOLUTION

4.2. Reduced Time Domain Layout Model

The thruster layout problem, as mentioned in 2, consists of orientation relative to a spacecraft body centred coordinate system. Even on a small instance scale as one produced in previous paragraph, the problem remains very difficult to tackle. In order to produce some solutions, is necessary to discretize the orientation domain in addition to reduce the time domain. A certain orientation can be individuated by two spherical coordinates, α and β. The alpha angle is the angle measured in the x-y plane, starting from the y-axis and counter-clockwise, while the beta angle is the azimuthal angle measured from the x-y plane and considered positive in the -z direction. **FIGURE 10** well illustrates these angles.

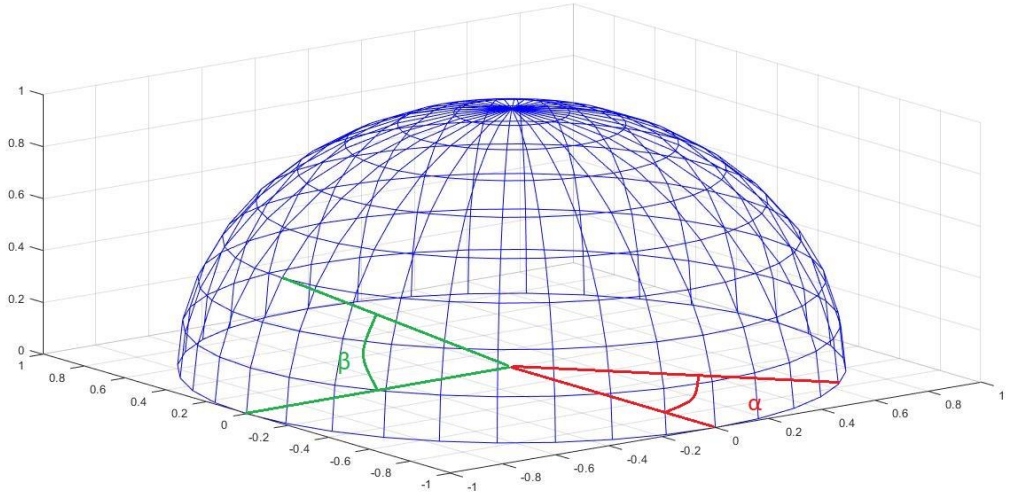

FIGURE 10 ALPHA AND BETA ANGLES DEFINITION

Using the *Layout* Model presented in 2.1., a first run is conducted to find some initial solutions.

The discretization utilized in this first run consists of an equal-spaced partitioning of both alpha and beta angles. There are some limitations in the possible choice of these angles due to the presence of the spacecraft body itself and of others spacecraft components such as the solar panels or the communication antenna. The angles that must be excluded for each actuator are reported in two files: *NO_A_MESH_FL* and *NO_B_MESH_*FL that respectively represent the limitations in α and β.

These limitations, as said before, exclude some orientations and thus the domain instead being a semi-sphere is just a portion of it. **FIGURE 11** shows a possible domain for an actuator.

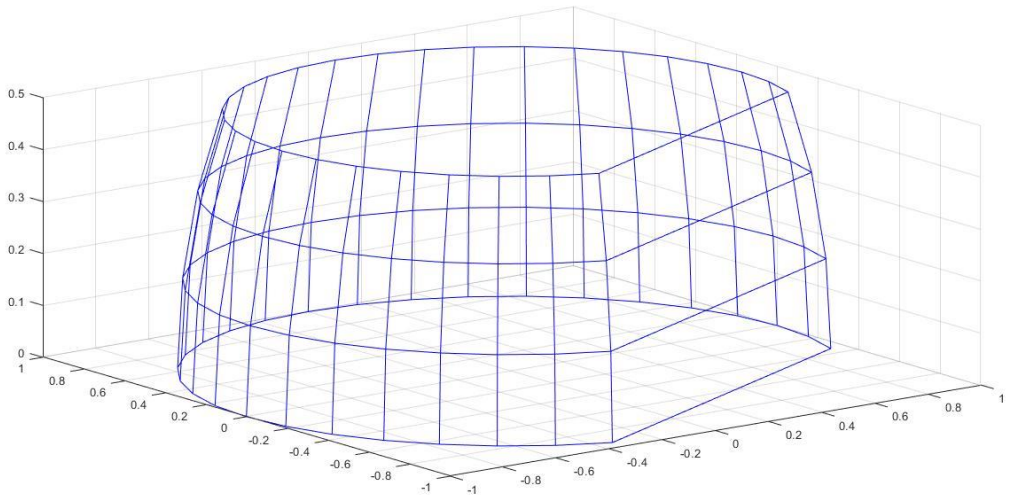

FIGURE 11 AN ORIENTATION DOMAIN EXAMPLE

The cases analysed in this work regard the possibility to reduce the number of thrusters obtaining, at the same time, the lowest consumption. The exclusion of some thruster must be considered in the model, therefore were introduced some data:

- *EXCLUDED_TH_FL* indicates the thruster or thrusters that want to be excluded;
- *TH_UP_NUMBER* indicates the number of total thrusters with a positive z-component;
- *THRUSTER_UP_FL* indicates all the possible thrusters that could have a positive z-component;

The orientations that a thruster can assume are expressed by three files, *VX, VY* and *VZ.*

$$v_x = -\sin\alpha * \cos\beta$$
$$v_y = \cos\alpha * \cos\beta$$
$$v_z = sign * \sin\beta$$

SIGN is a file of -1 or 1, respectively for the thruster with a negative z-component and for a positive z-component.
The application-points are contained in three files, *AX*, *AY* and *AZ* in which for each thruster are indicated the coordinates expressed in the reference frame.
In two other files, *MIN_FORCE* and *MAX_FORCE*, are expressed the lowest and the highest values of force that a thruster could achieve.
In order to consider the errors too, as seen in 1.1, are introduced two files:

- *MAX_ERR_TOT_FORCE* stands for E_f in (11);
- *MAX_ERR_TOT_TORQUE* stands for E_t in (11).

Finally, the object functions are represented by two flags:

- *MIN_TOT_FUEL_CONS_FL* set to 1 indicates that the object function is (9)
- *MIN_TOT_ERR_FL* set to 1 indicates that the solver would try to reduce the total error given by the sum of forces and torques errors.

An example of instance is reported in **TABLE 4. 1**. In the case of just the third thruster of the second cluster with a positive z-component:

Time_coordinates.dim	32
Thruster.dim	12
MIN_FORCE	0.00001 μN
MAX_FORCE	100 μN
TH_UP_NUMBER	1
TH_UP_FL	CLUSTER_1_TH_3_P 1 CLUSTER_2_TH_3_P 1 CLUSTER_3_TH_3_P 1
EXCLUDED_TH_FL	CLUSTER_1_TH_3_M 1 CLUSTER_1_TH_3_P 1 CLUSTER_3_TH_3_M 1 CLUSTER_3_TH_3_P 1

TABLE 4. 1 INSTANCE THRUSTER UP

The computational cost of this model is relatively high, since the discretization is enough thin, and the set of instants is not so large. A quantity representative of the problem's size is the number of integer variables. In fact, the number of possible solutions for a pure IP problem, for example, is 2^n with n the number of integer variables. It is evident that an exponential growth as that affects the solution finding procedure since a little increment of n produce a large increment of possible solutions that the solver must enumerate. An example of problem's size is reported in TABLE 4. 2 below:

Number of rows	3326
Number of columns	50512
Number of integer variables	272

TABLE 4. 2 PROBLEM MATRIX SIZE

The problem in exam, is useful to reminder, is a MILP problem, thus a problem in which the variables are both integer and continuous.

4.3. Linear Model

Once solutions in a time reduced domain have been obtained, an analysis must be carried out in order to evaluate their feasibility. As mentioned in 2.2., the model used to do this is the *Continuous* model. This model is a linear model since the binary variables $\boldsymbol{v}$ are fixed and the only variables present are the thrusts u. Since α and β are not variables yet but just parameters, the limitations on these are not necessary. The orientations used in this model came from the previous analyses conducted with the *Layout* model and, as in the *Layout*, two object function is available. The first tries to reduce the propellant consumption and is set by *MIN_TOT_FUEL_CONS_FL* and the latter reduces the total error, if there is, and is activated by *MIN_TOT_ERR_FL*. This model has a very low computational load since the only variables are the u and thus has a short runtime.

4.4. Discretized Refinement Model

The iterative process explained in this chapter dynamically narrows the domain of possible orientations to a smaller domain centred on the previous orientations to refine locally the solution. The starting orientations come from the solutions found thanks to the *Layout* model. The refinement is made defining a discretization refinement, i.e. deciding an odd number equal or greater than three that will constitute the number of possible α and β. It was decided to apply a discretization of five possible orientations per angle and the mesh resulting envisage twenty-

five possible orientations for the considered actuator. The choice of the refinement parameter has a strong influence on the computational load of the model, since the number of logical variables is directly correlated to the discretization. The discretization must consider the limitations discussed above and thus the possible orientations could be lesser than the twenty-five previous said. A possible discretization refinement is illustrated in **FIGURE 12**, in which the starting orientation is indicated as a red point.

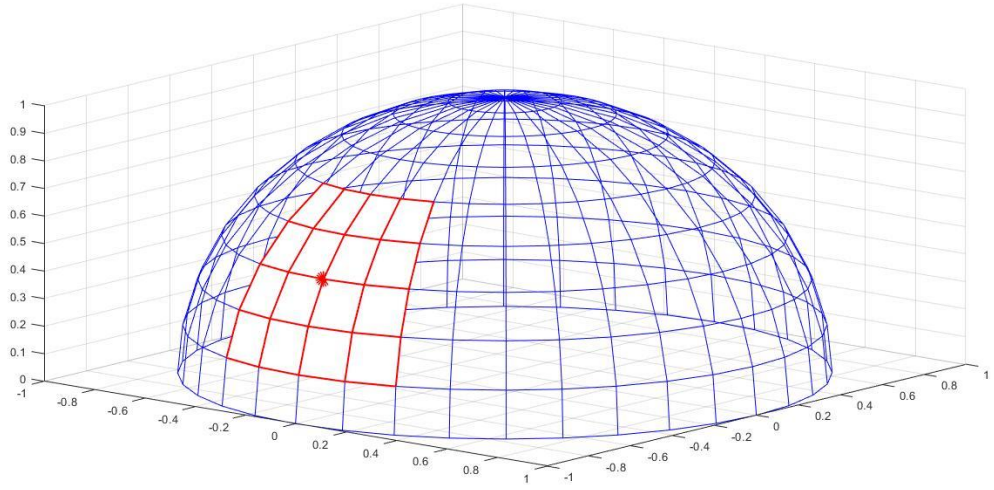

FIGURE 12 A REFINEMENT DOMAIN EXAMPLE

The step length varies as iterations proceed, halving at each iteration. It was considered an initial step length of five degrees both for α and β. Since the improvement decreases as the iterations continue, is not useful iterate many times, also because the time required for a representative analysis is large. Considering a runtime of thirty-six hours for iteration, the iterations envisaged for each of six cases are four and thus a total time employed for this analysis is of thirty-six days. Augmenting the number of iterations or the runtime would have required too time thus the compromise above was adopted.

Since the iterations considered are four, the step length is reduced from the initial 5° of the first iteration to 0.625° of the last one with the pattern shown in **FIGURE 13**.

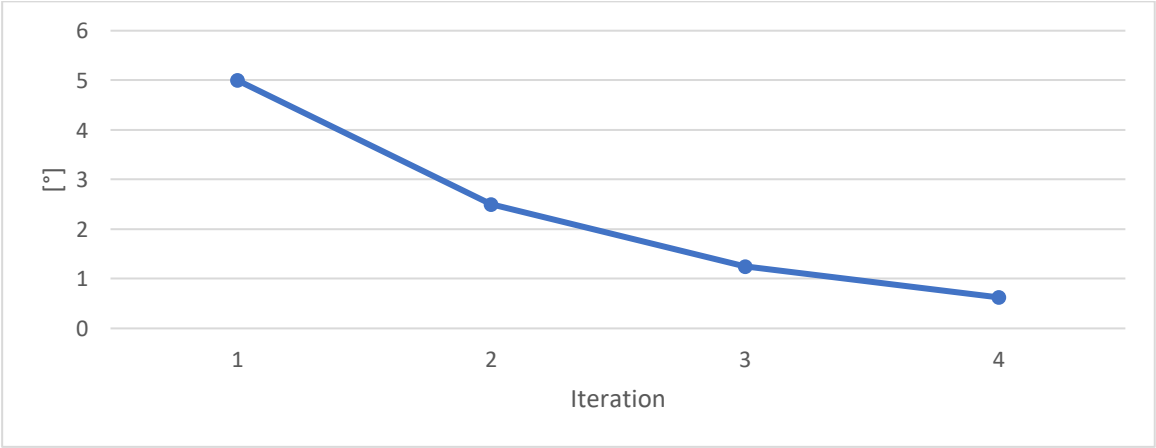

FIGURE 13 DYNAMIC DOMAIN SIZE

It was decided, moreover, to consider the total set of instances instead of the reduced one used in the *Layout* model. Using all the instants of course increase the complexity of the problem and the computational load but avoid the transfer of solution from the reduced set of instants to the total one that could result in an infeasibility or in an non optimum solution.

5. Solutions Description

The solutions obtained are compared to a solution, hereafter called Nominal, that establishes the term of comparison of the conducted analyses. The Nominal solution consists of nine thrusters, 3 of which z-positive oriented, grouped in three clusters. FIGURE 14 shows the nominal configuration that can also be found in [10]. The application-points and the external shape of the spacecraft illustrated in the following sections don't reflect the real ones considered by the program in order to maintain the confidentiality of the analyses made.

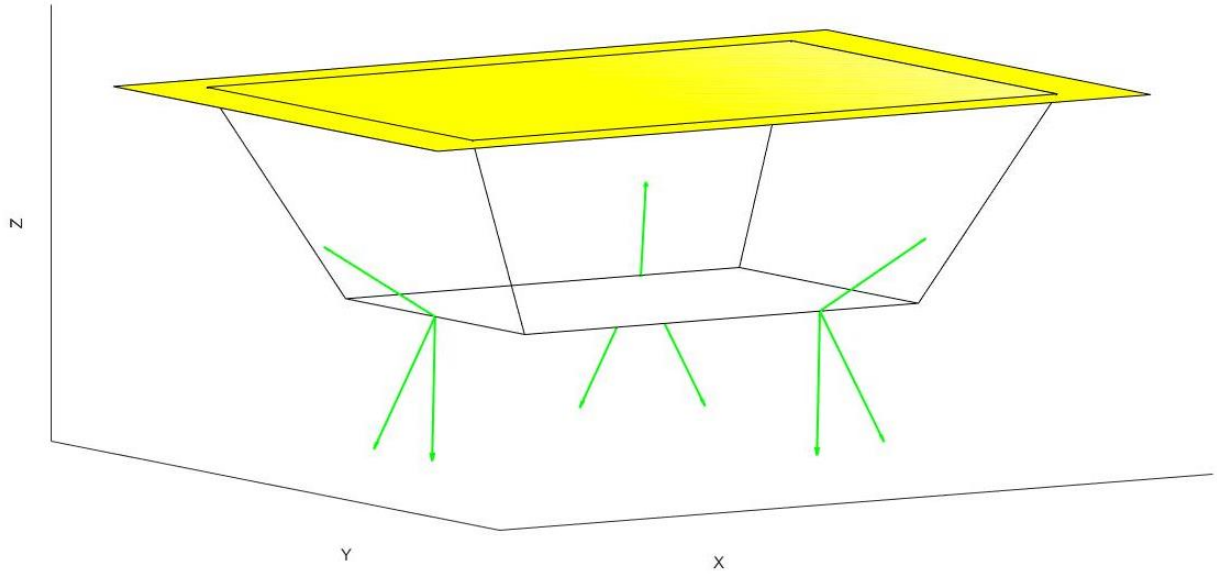

FIGURE 14 NOMINAL CONFIGURATION

The forces and torques acting on spacecraft varying during the lifespan since, during the mission, must be performed some manoeuvres that generate forces on it and that must be contrasted in order to maintain an enough pointing accuracy. As every mission, also LISA is subdivided into mission phases that occur in a precise sequence. The most relevant phase is the scientific one, that represent the aim for which the mission born. As mentioned in 1, the pointing accuracy is a critical mission requirement that must be satisfied to 29ulfil the mission objectives. In order to satisfy this requirement, the control law must act in a very precise way and the thrusters' layout has to be studied appropriately. This work it is proposed to find an optimum configuration able to contrast the external loads during the scientific phase minimizing the propellant consumption and, at the same time, considering a lower number of actuators compared to Nominal configuration. Once obtained an optimum configuration for the scientific phase, see 5.1, the same configuration is tested for other manoeuvres required during lifespan to verify if the configuration found is suitable to satisfy the control law in other scenarios, see 5.2.

5.1. Scientific phase optimization

The cases analysed are six in total, three considering one thruster upward and three considering two thrusters upward while the number of thrusters downward remains always six. The presence of at least an actuator upward oriented is preferable since a configuration with all actuators downward oriented couldn't respond to a force along -z axis, even if this configuration could be the best one. In order to maximize the +z component that the upward oriented actuators can

exert, it was decided to place them with the maximum β inclination allowed by the limitations explained in 4.2. Once defined the limitations on angles, it was run an analysis with the *Layout* model considering 32 instants and a maximum runtime of 36 hours. **FIGURE 15** shows the percentage consumption improvement compared to the nominal solution of each considered case.

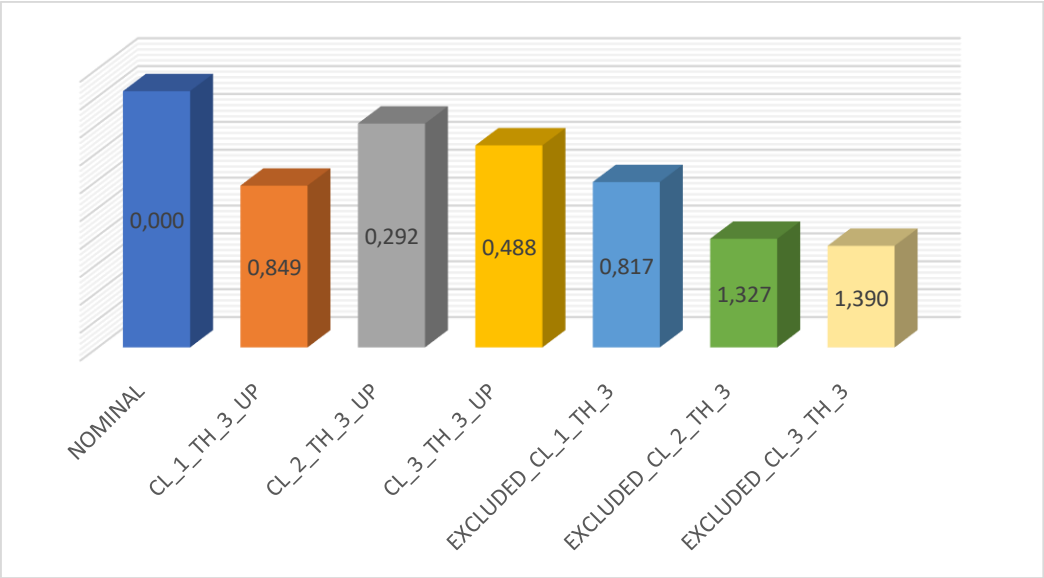

FIGURE 15 FUEL REDUCTION PERCENTAGE W.R.T. 32 INSTANTS

As can be seen the configurations with two thrusters upward give the higher gain compared to them with just one thruster up.

As said, the solutions here obtained regard a representative set of instants and not the whole one. The configurations that are feasible for that reduced time domain could be infeasible in the total set. In order to verify the feasibility of solutions found, it is needing the model called *Continuous*. In 2.2 was explained how this model works and briefly will be reminded its structure. In input it receives the orientations of thrusters found thanks to the *Layout* model as directors' cosine and, reducing the system (16) in a linear system in which just the thrusts u are the variables, it tries to minimize the total fuel consumption if *MIN_TOT_FUEL_CONS_FL* is set to 1 or the total error if *MIN_TOT_ERR_FL* is set to 1. The model built in this way requests a little computational cost, since the logical variables for the choice of α and β was deleted and the runtime reduces to a few seconds.

FIGURE 16 shows the percentage of gain compared to the nominal consumption obtained by the *Continuous* model. As can be seen all the configurations found by the *Layout* model are feasible also in the total time domain and the improvement's trend is the same in the reduced time domain and in the total set, as proof of goodness of the algorithm for the time selection.

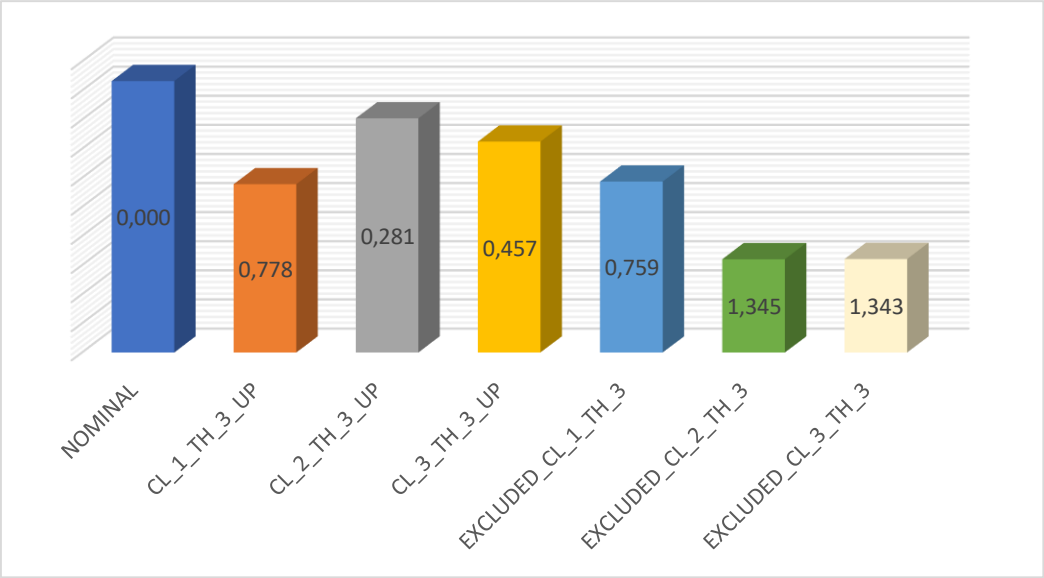

FIGURE 16 FUEL REDUCTION PERCENTAGE W.R.T. 365 INSTANTS

The solutions found so far are improved through the iteration procedure explained in 4.4. The total consumption gain compared to the nominal configuration is reported in **FIGURE 17**.

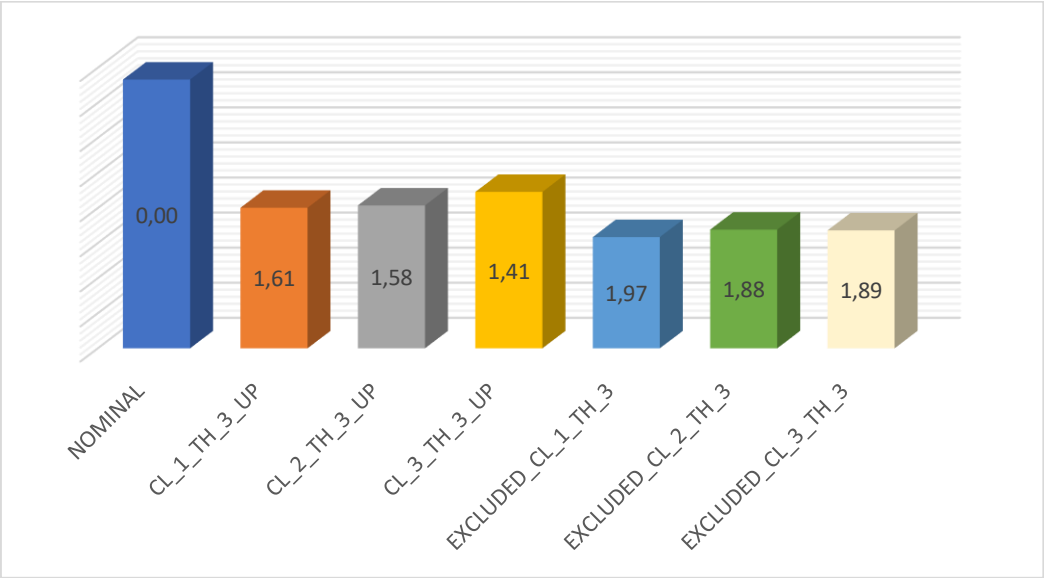

FIGURE 17 PERCENTAGE FUEL REDUCTION

After the iteration refinement the best configurations remain those with the higher number of actuators upward oriented even if the best configuration without iterations, *EXCLUDED_CL_2_TH_3*, is not more the best.
The space orientations of each configuration are illustrated in the next pages.

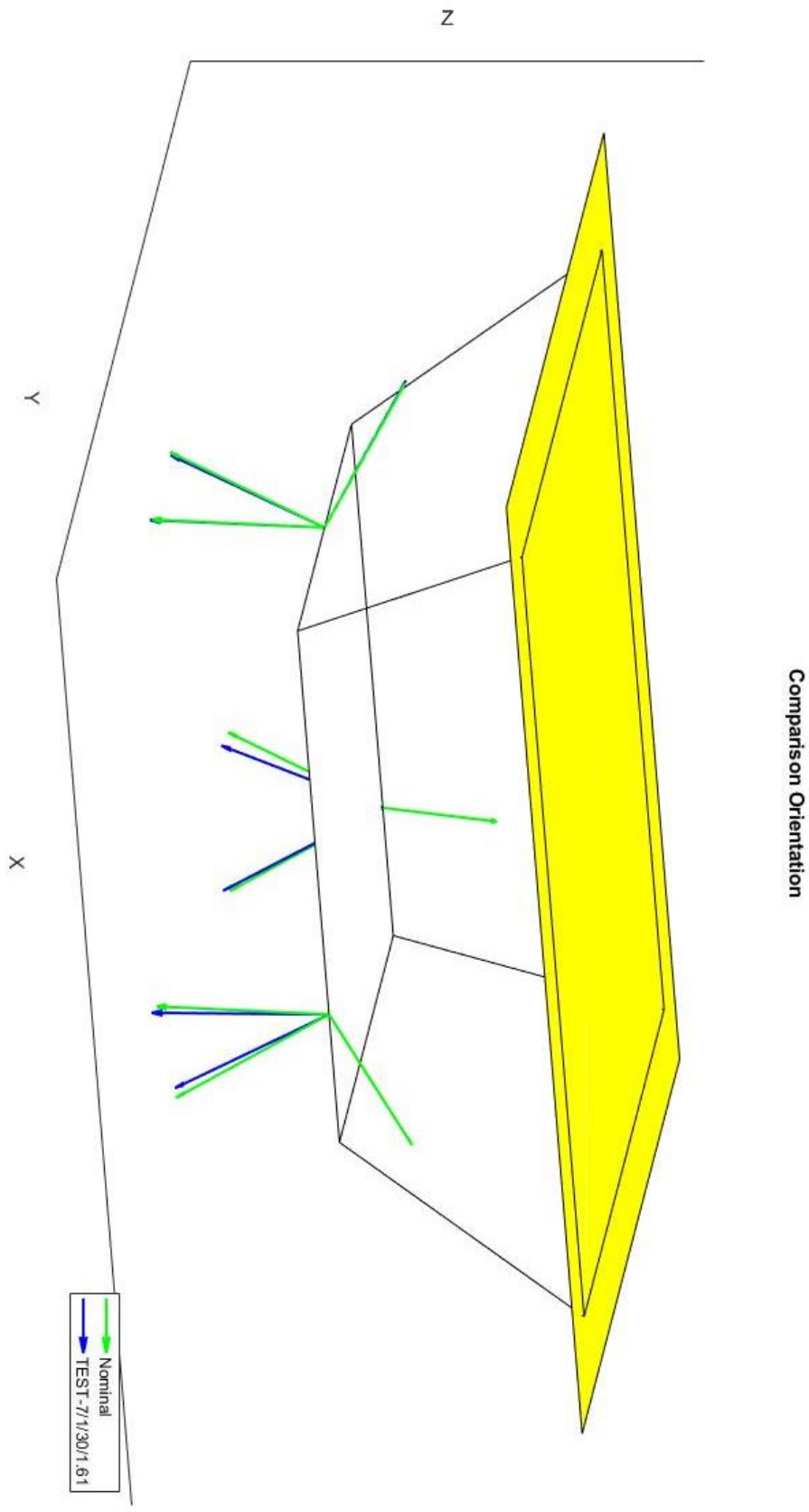

FIGURE 18 CL_1_TH_3 ORIENTATION

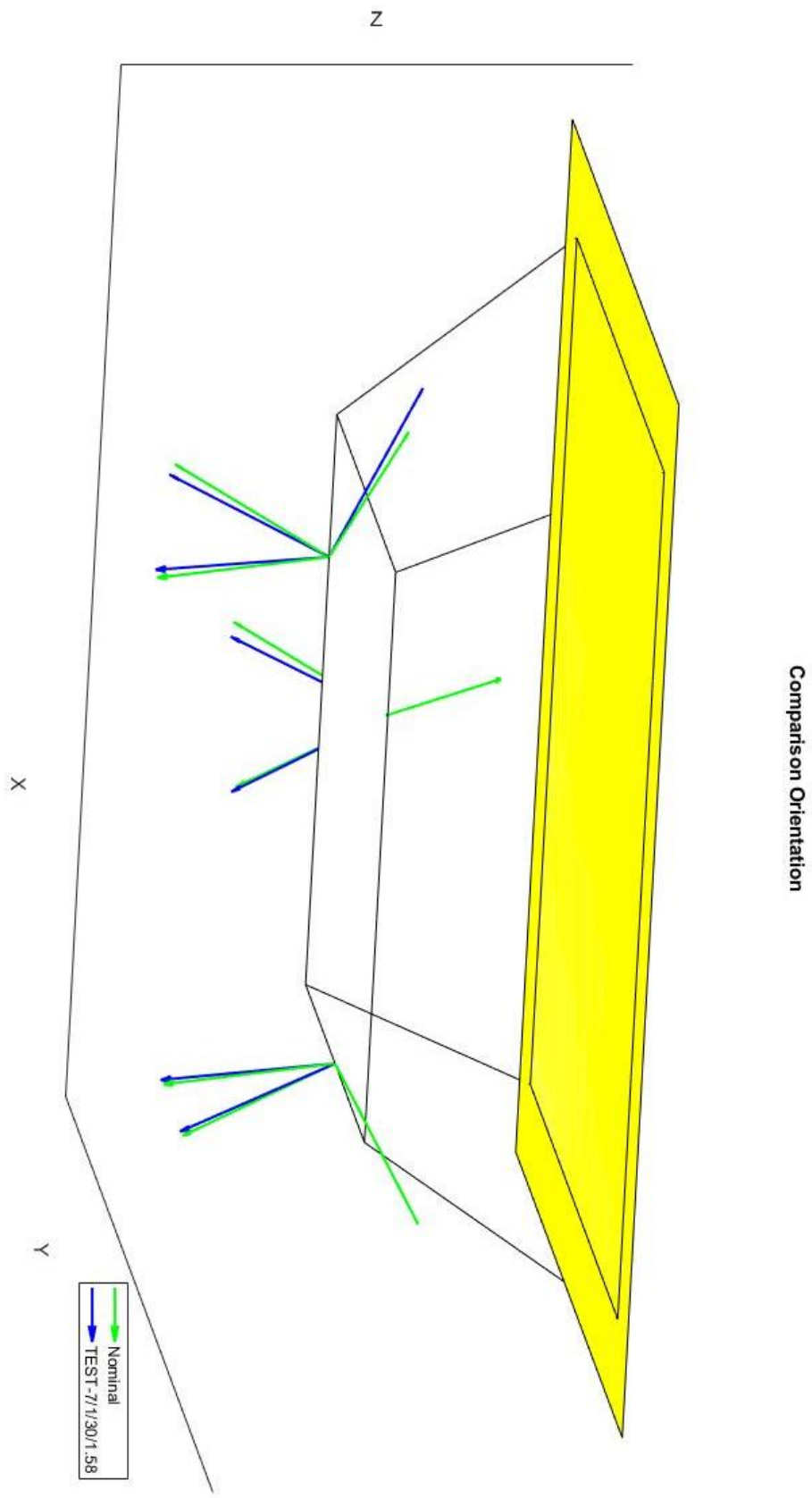

FIGURE 19 CL_2_TH_3 ORIENTATION

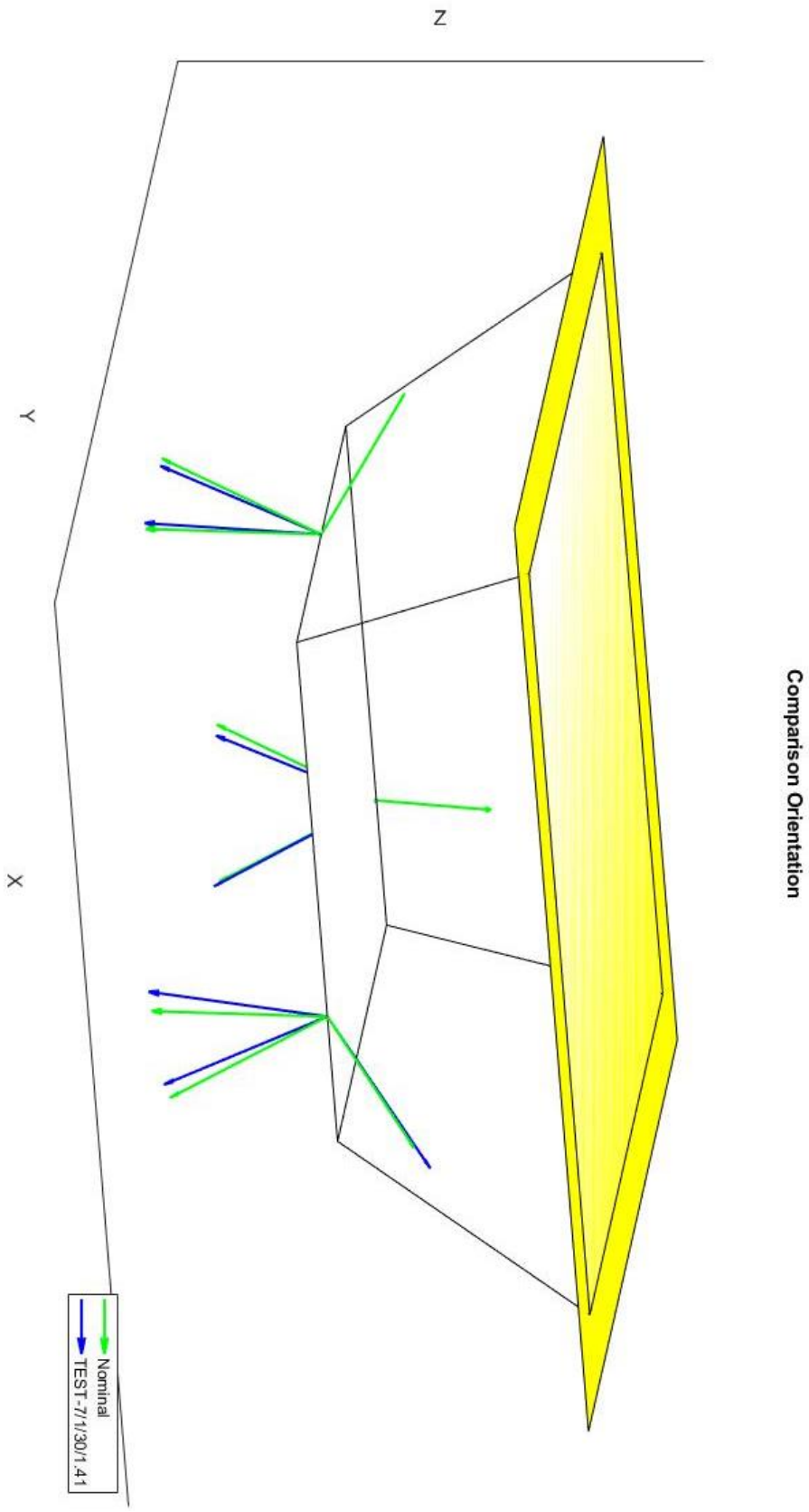

FIGURE 20 CL_3_TH_3 ORIENTATION

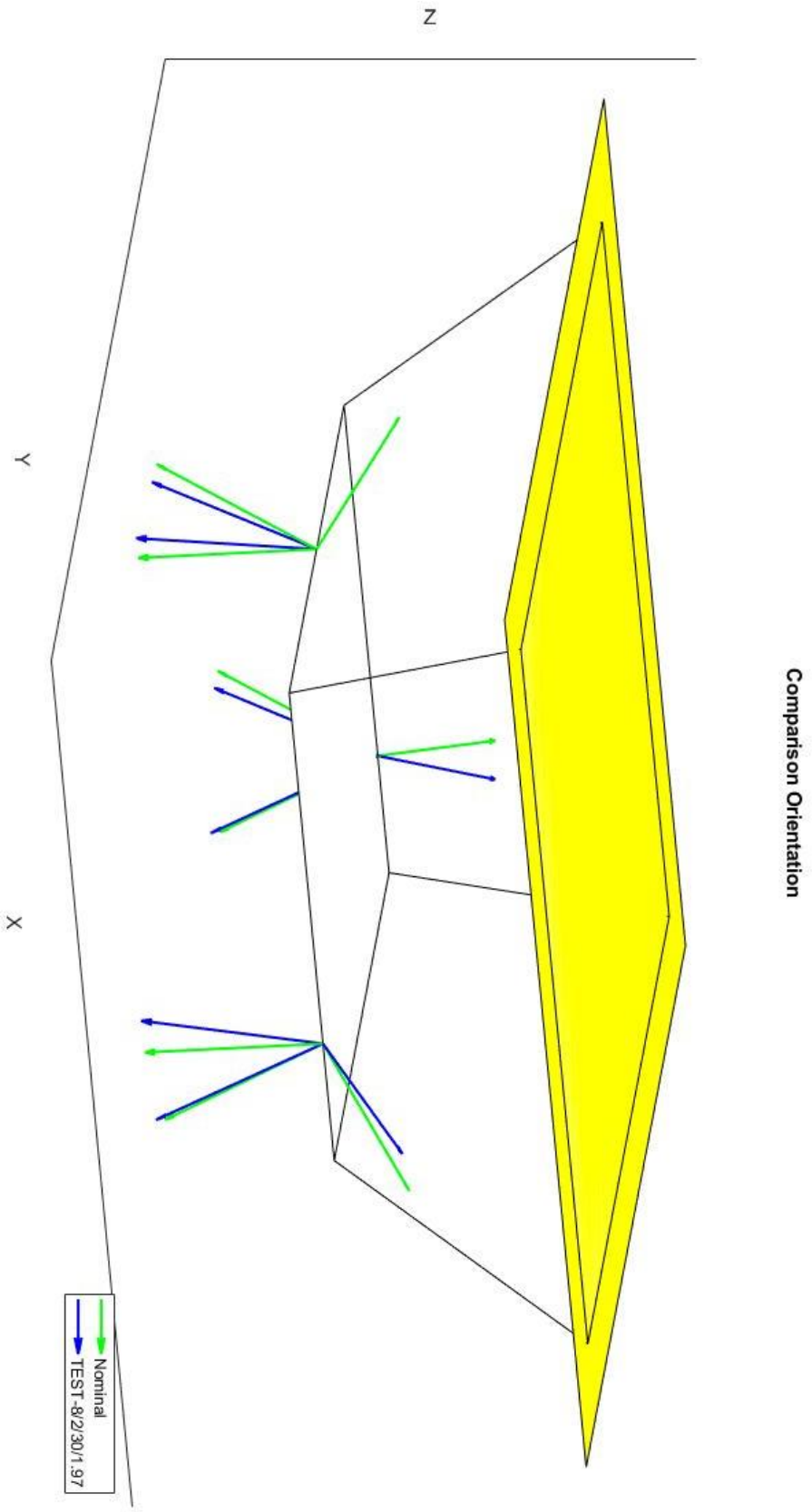

FIGURE 21 EX_CL_1_TH_3 ORIENTATION

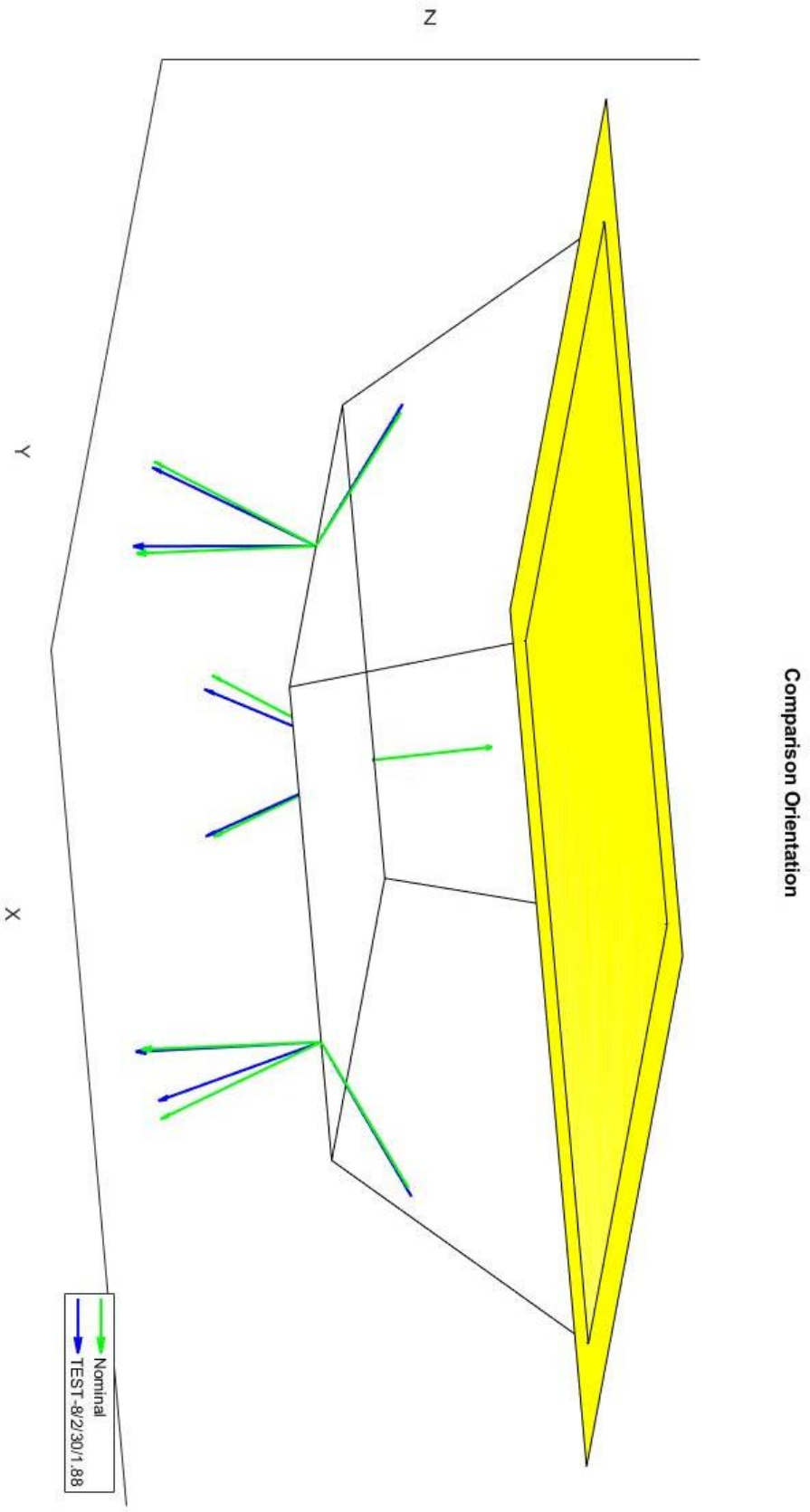

FIGURE 22 EX_CL_2_TH_3 ORIENTATION

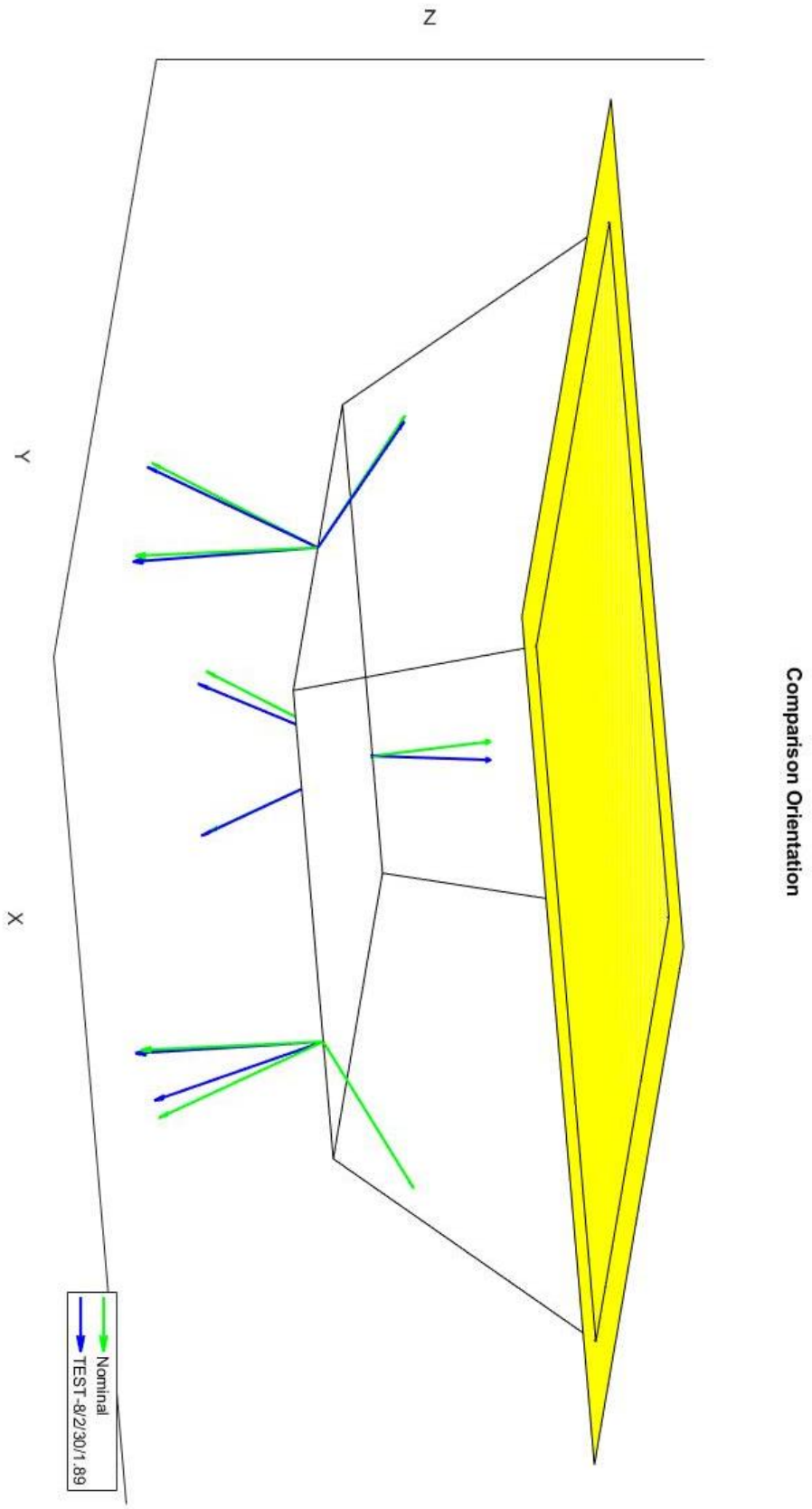

FIGURE 23 EX_CL_3_TH_3 ORIENTATION

As the figures in the pages above illustrate, the configurations found are close to the nominal one. In fact, as the TABLE 5. 1 shows, the difference between the α and β angles of nominal configuration and the cases in analysis never exceed the 50°. The higher differences are in the cases *CL_2_TH_3* and *CL_3_TH_3* that are cases far from the nominal configuration since they have two actuators fewer. In β angles, instead, the maximum difference is 5° that is a very low angle. It is also true that the upward oriented actuators are imposed to have the same inclination in β of the nominal configuration and so there are less actuator to consider.

Thruster	α	β
CL_1_TH_3	12.5	4.375
CL_2_TH_3	40.625	3.4375
CL_3_TH_3	48.75	3.75
EX_CL_1_TH_3	22.5	3.75
EX_CL_2_TH_3	13.125	5
EX_CL_3_TH_3	13.125	4.6875

TABLE 5. 1 MAXIMUM ANGLES DIFFERENCE FROM NOMINAL

Since the orientations found are almost the same of nominal configuration, it is predictable imagine that the thrust exerted by each thruster are similar. In the figures in the following pages is shown the trend assumed by the thrust of each actuator during the examined time.

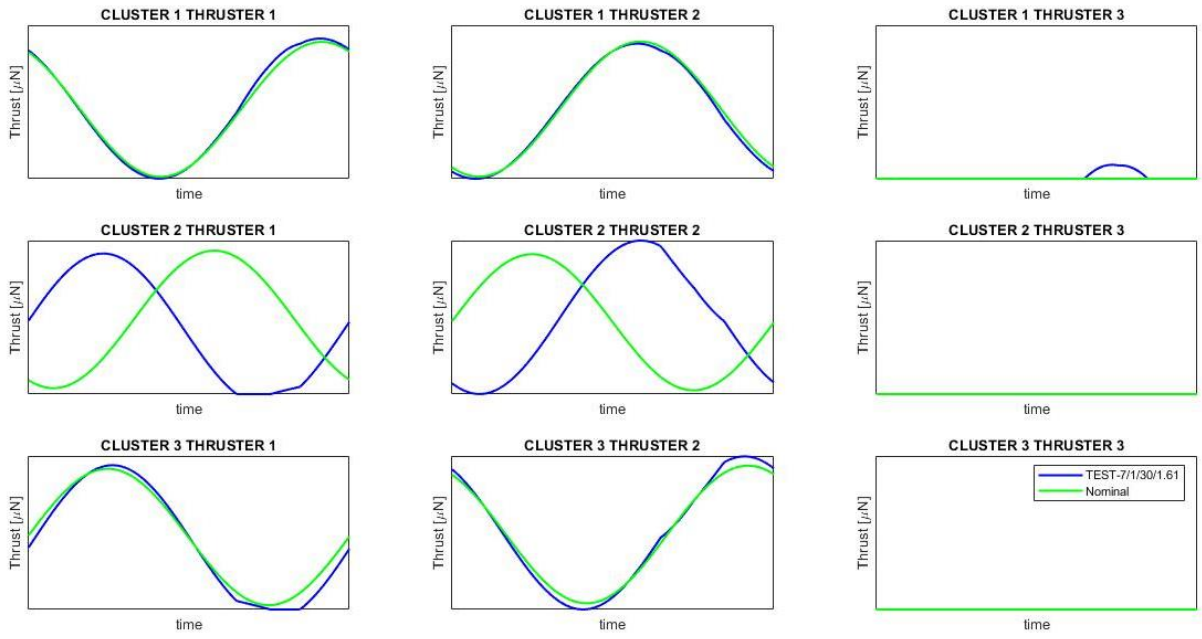

FIGURE 24 CL_1_TH_3 THRUST VS TIME

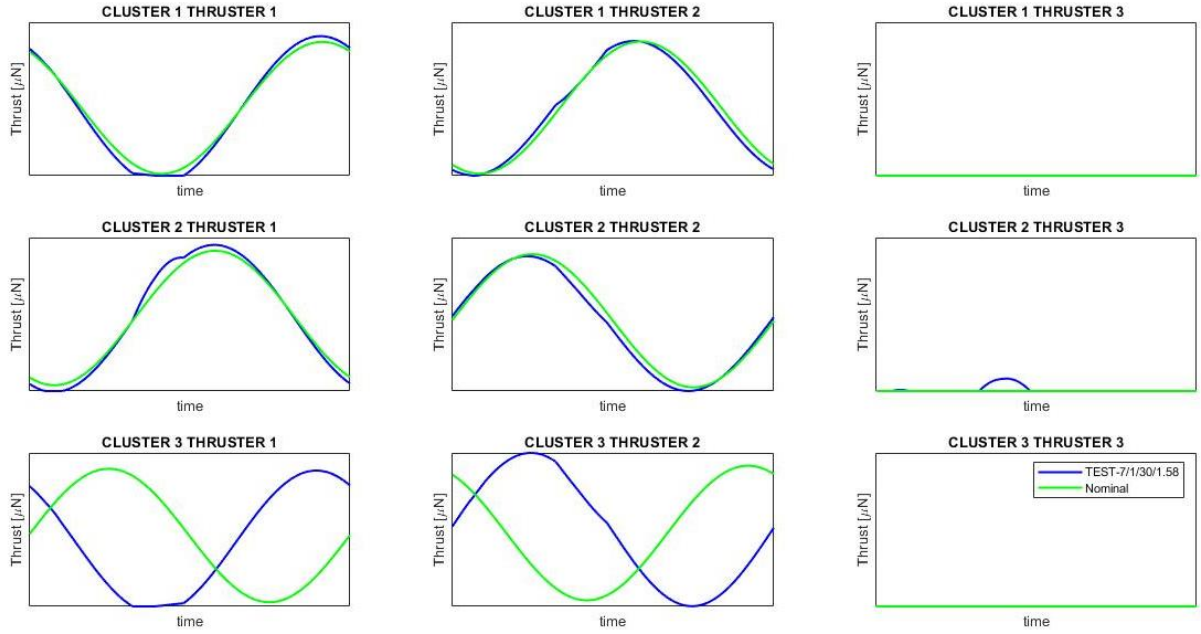

FIGURE 25 CL_2_TH_3 THRUST VS TIME

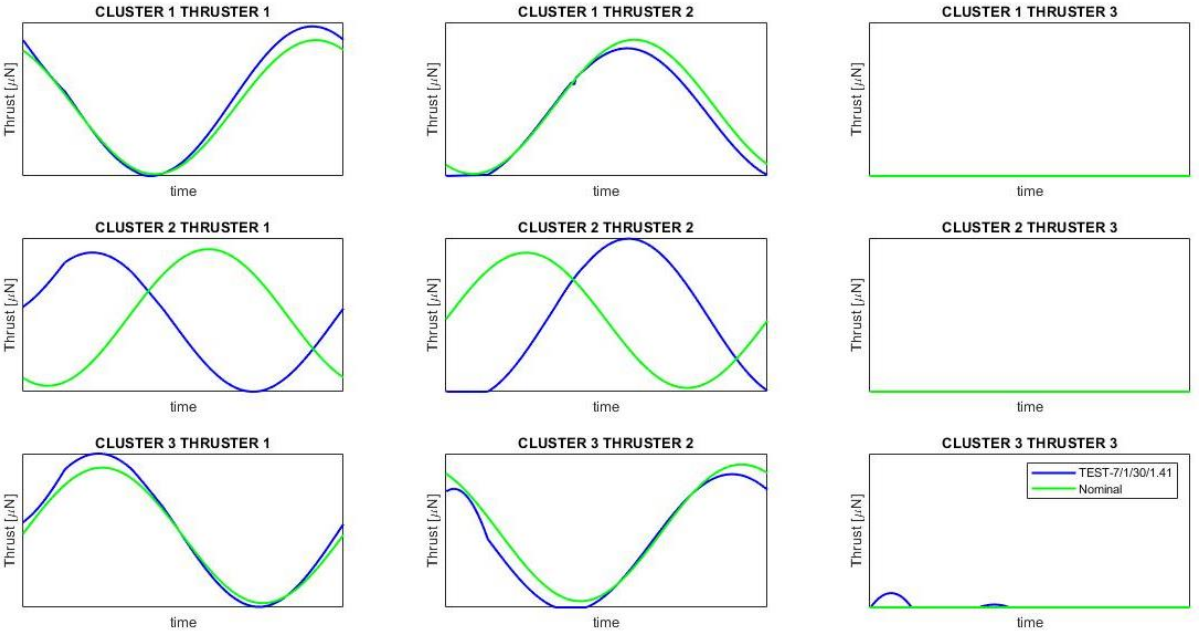

FIGURE 26 CL_3_TH_3 THRUST VS TIME

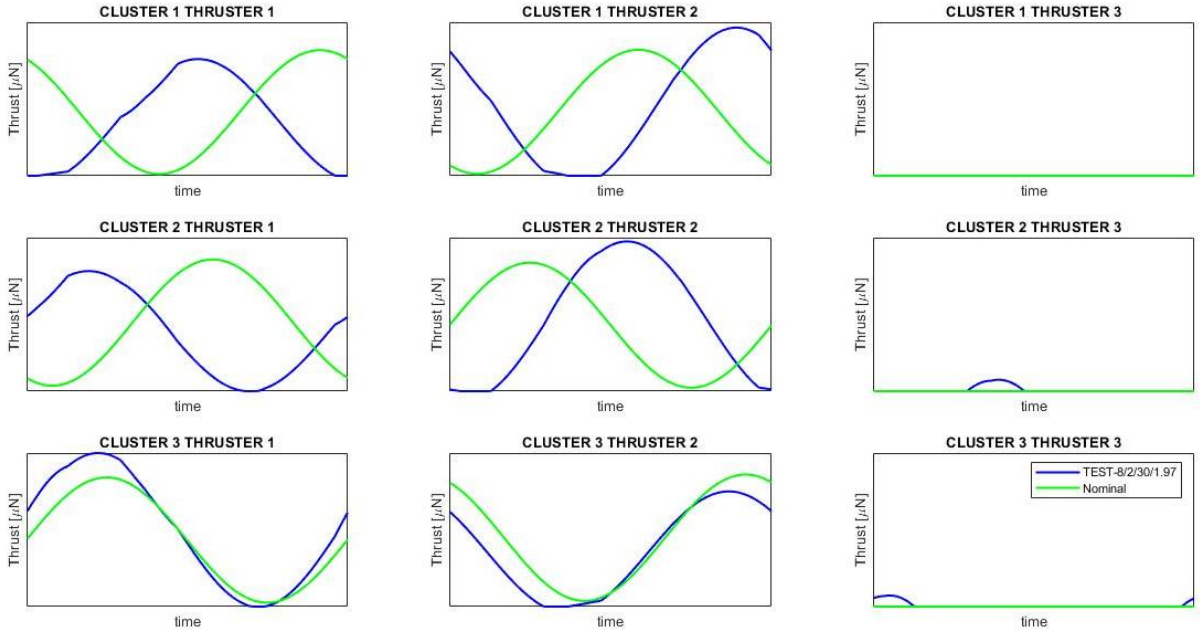

FIGURE 27 EX_CL_1_TH_3 THRUST VS TIME

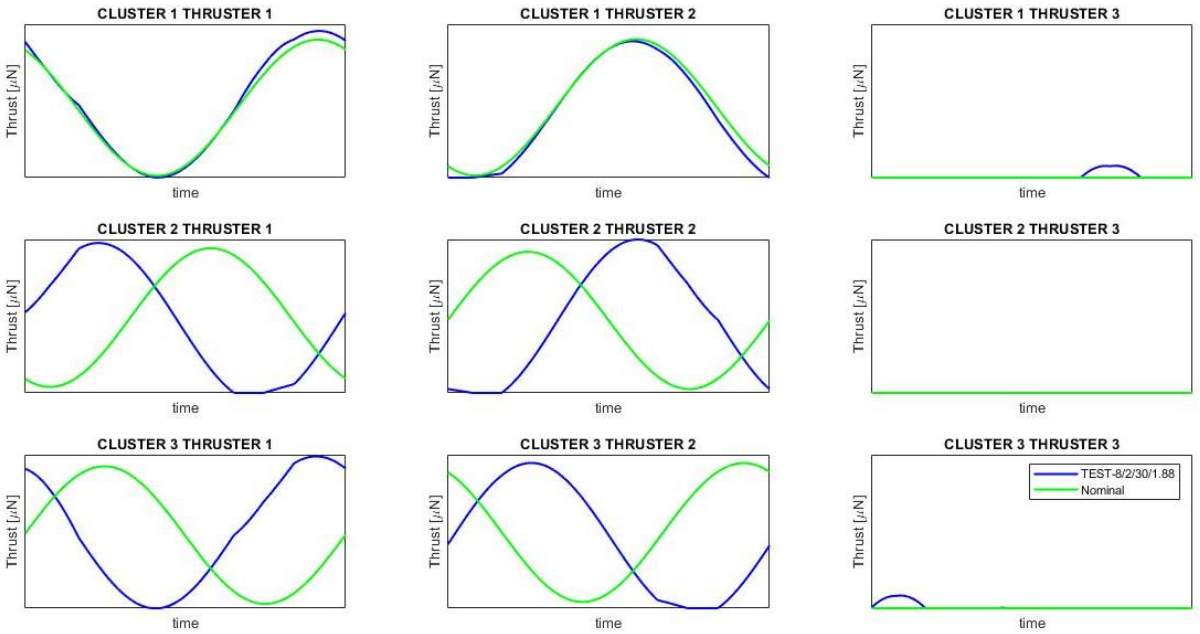

FIGURE 28 EX_CL_2_TH_3 THRUST VS TIME

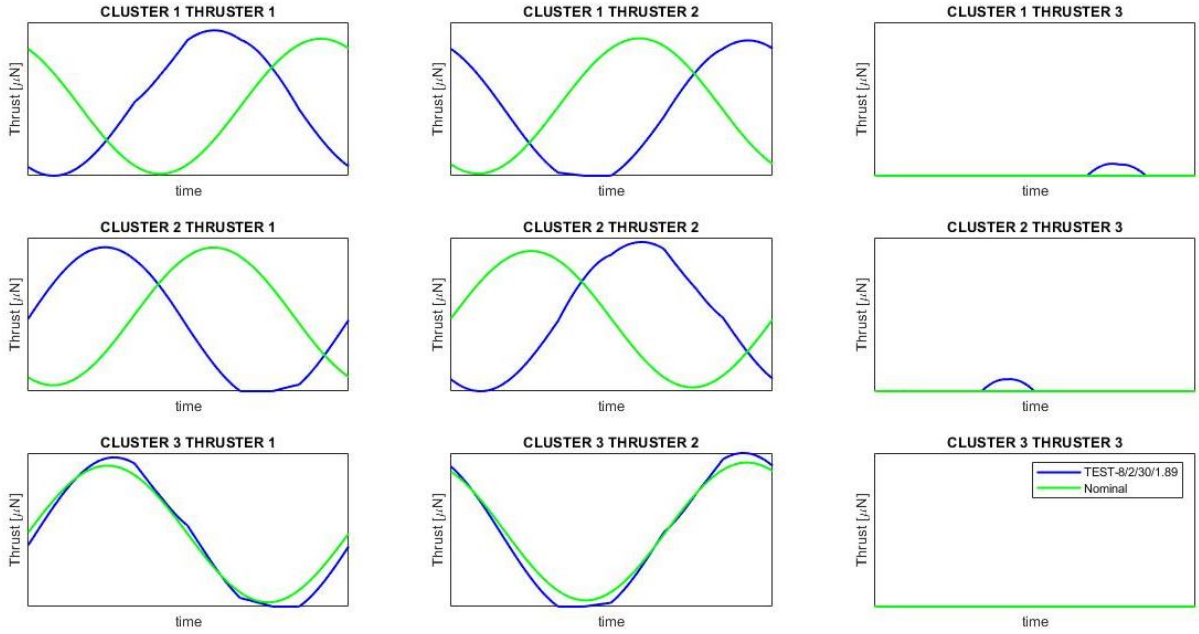

FIGURE 29 EX_CL_3_TH_3 THRUST VS TIME

It is worth noticing that the actuators upward oriented, in each examined case, exert a very low thrust just for few instants, but their functioning is crucial for reducing the total consumption over the entire time domain. Furthermore, the thrust's trends are very similar to the nominal configuration but in some cases they are in counter-phase. A possible explanation of this weird behaviour could be found in the geometrical symmetry of the problem. The application-points position has a sort of geometrical symmetry and the orientations of the configurations studied are quite similar to the nominal one. These considerations could explain the counter-phase trends because for a torque, for example, it is not important the forces themselves, but the product of the forces times the arm. If the arm is the same (geometrical symmetry) the sum of forces is the key parameter and not how they are distributed.

A further graph that can help understanding the actuators' behaviour is that representing the total thrust exerted by an actuator during the whole set of instants. This kind of graph is illustrated in the below pages.

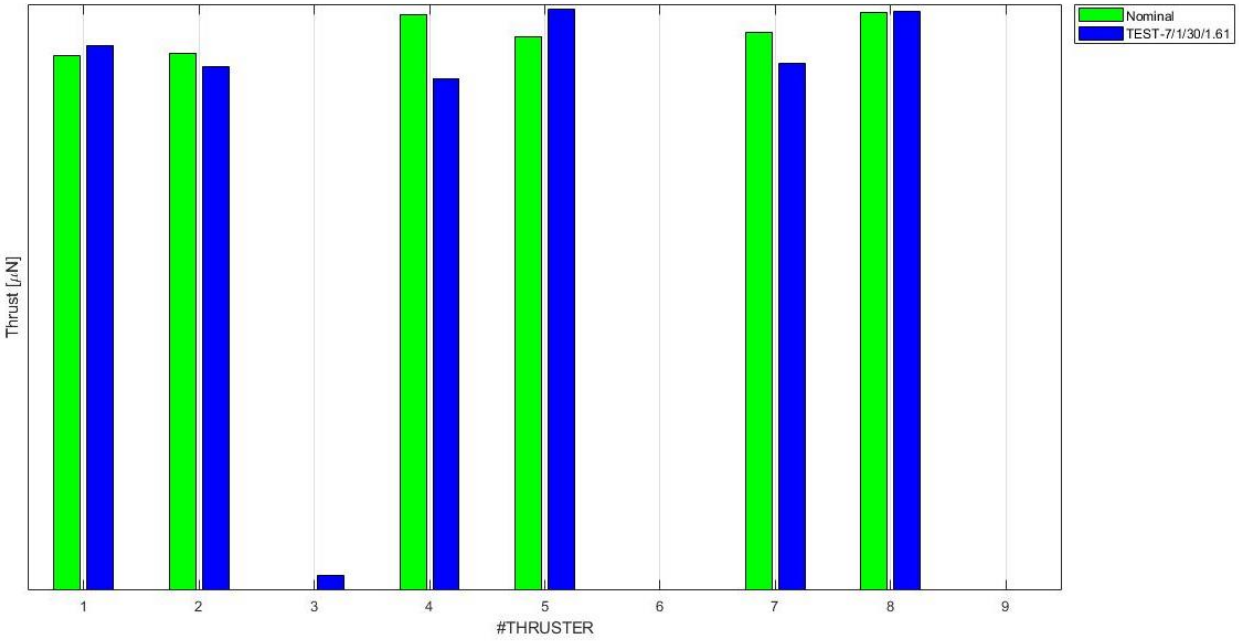

FIGURE 30 CL_1_TH_3 TOTAL IMPULSE

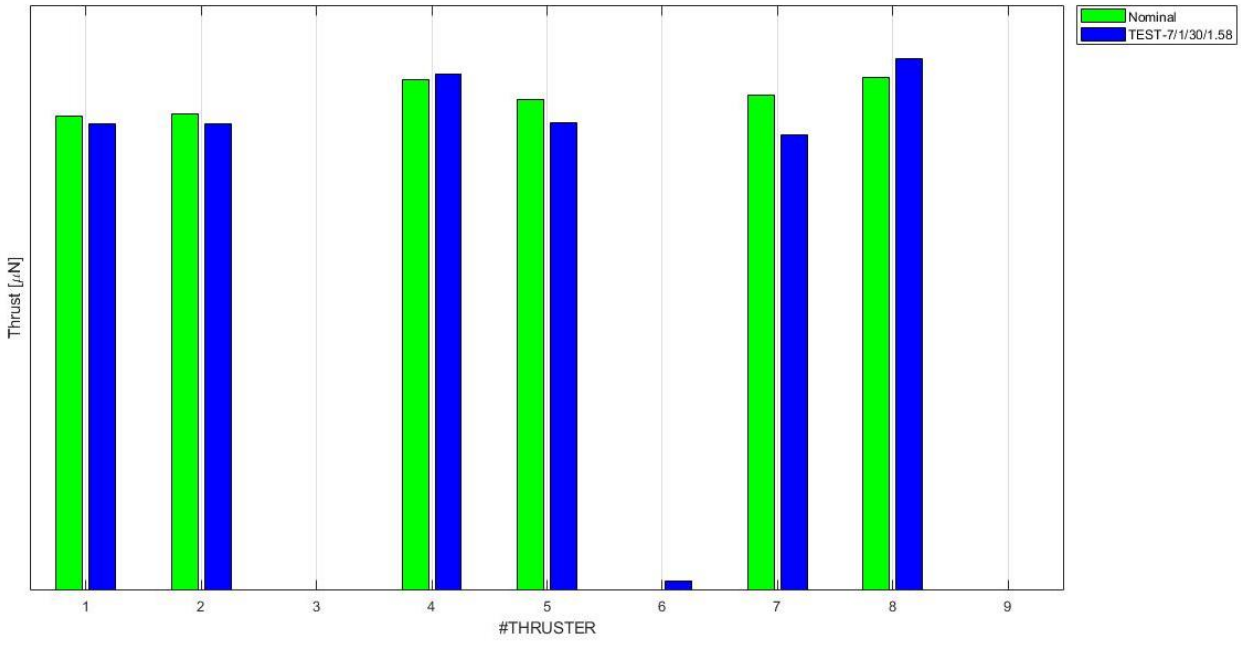

FIGURE 31 CL_2_TH_3 TOTAL IMPULSE

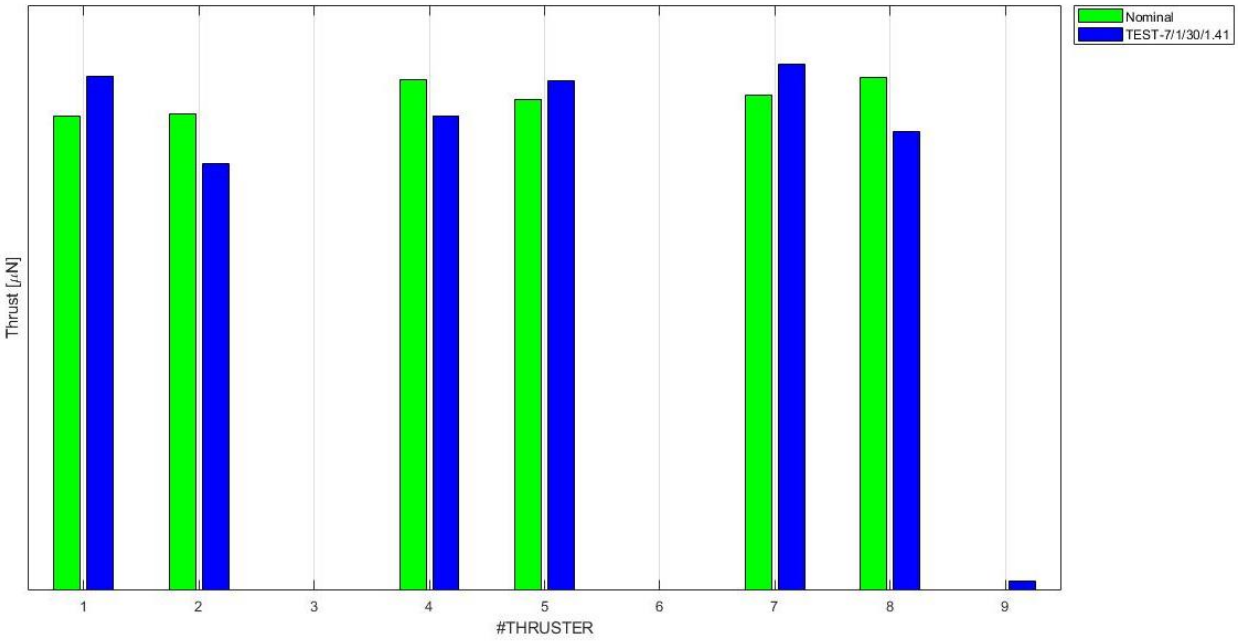

FIGURE 32 CL_3_TH_3 TOTAL IMPULSE

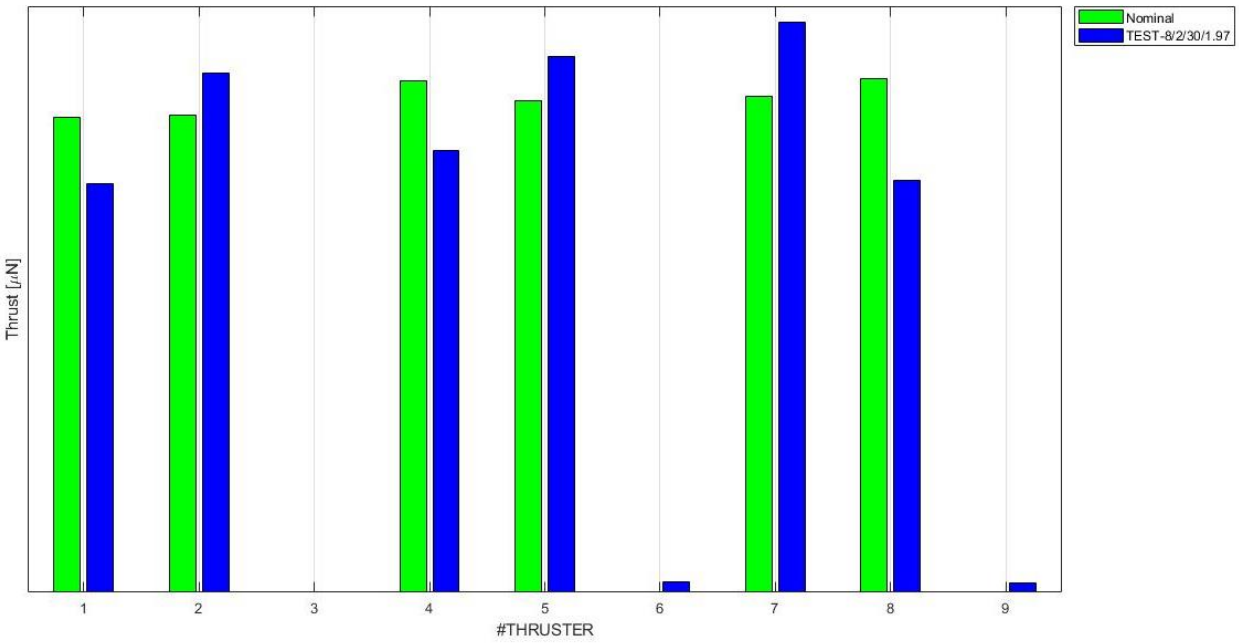

FIGURE 33 EX_CL_1_TH_3 TOTAL IMPULSE

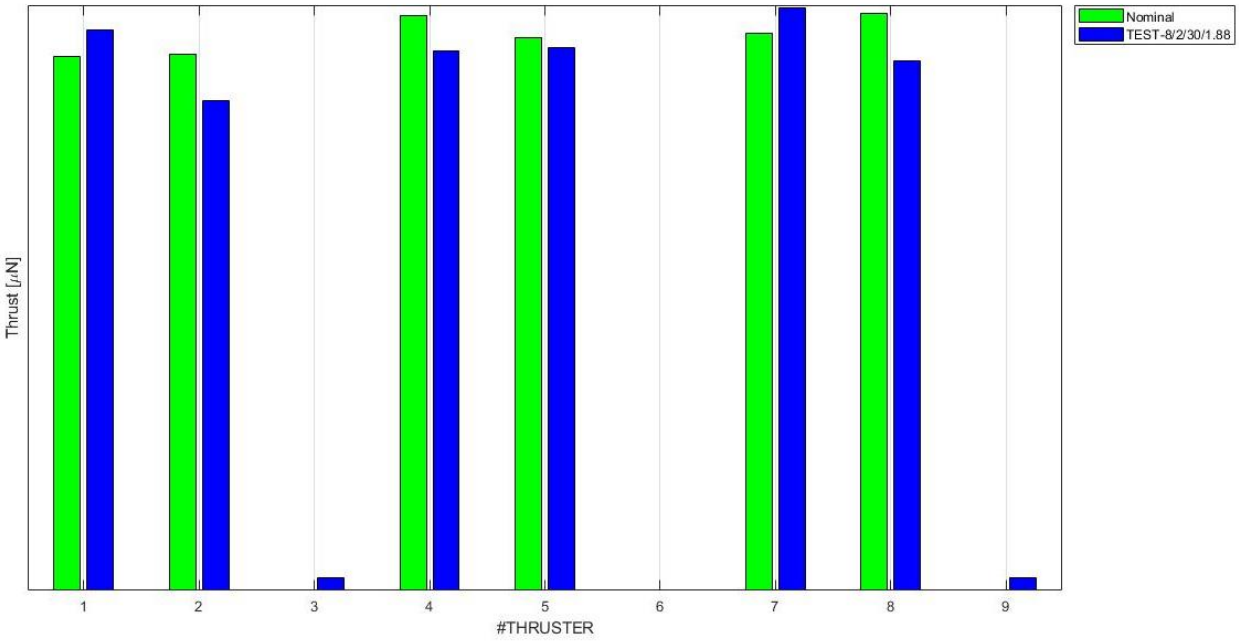

FIGURE 34 EX_CL_2_TH_3 TOTAL IMPULSE

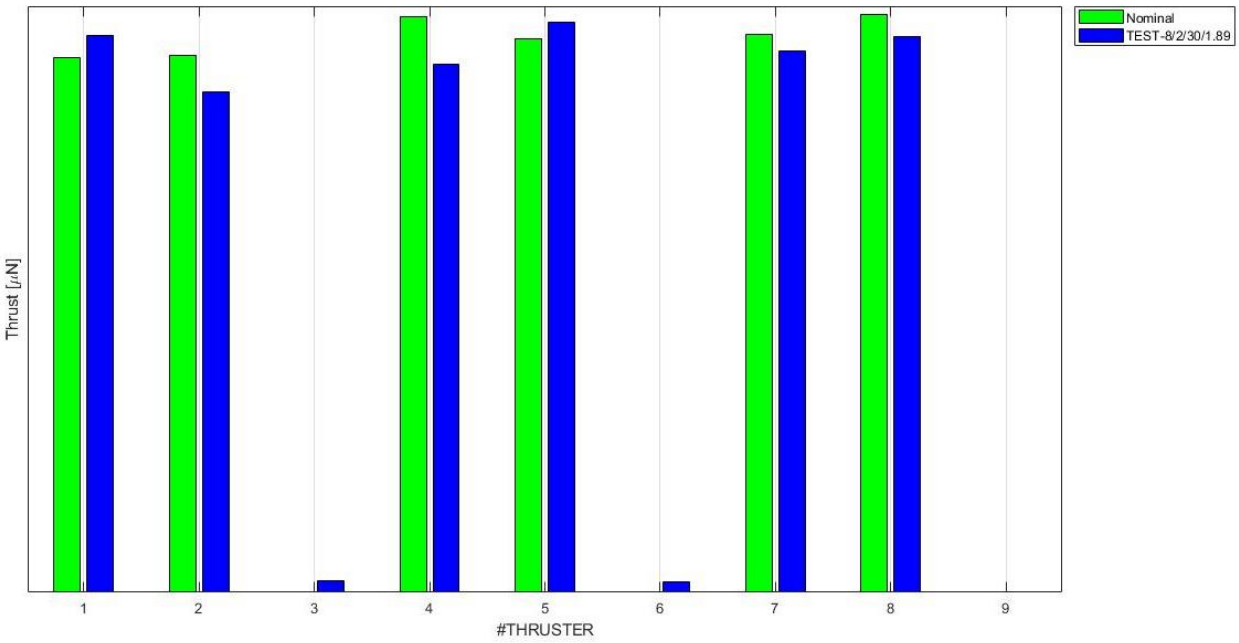

FIGURE 35 EX_CL_3_TH_3 TOTAL IMPULSE

These graphs validate the possible explanation of counter-phase behaviour explained in the previous rows. In fact the integral of the forces over time, the total impulse that an actuator exert, is similar to the nominal configuration thus, always regarding the same arm's length, the integral of the torque over time (the subtended area in **FIGURE 5**) is equal the arm times the integral of force over time that is similar in all the configurations studied.

The solutions found were generated on the whole timeframe but, in order to reduce the runtime, must be considered a reduced timeframe. As a trade-off between the computational cost and the speed was decided to consider a set of 101 instants.

The last effort to reduce the total consumption is utilizing the function FMINCON in the Matlab environment. As explained in 3.3, the tool needs an initial point to start the analysis. This starting point must contain the orientations of each actuator and the thrust exerted in every instant by each actuator. The dimension of this file, therefore, is $n_{insta} \times n_{actuator} + 2 \times n_{actuator}$. The analysis was conducted considering a set of 101 instants shown in **FIGURE 9** and the orientations of the best solution found so far that is always resulting from the fourth iteration. In order to find the best algorithm for the problem in exam, there are conducted the following analyses for *CL_1_TH_3* case.

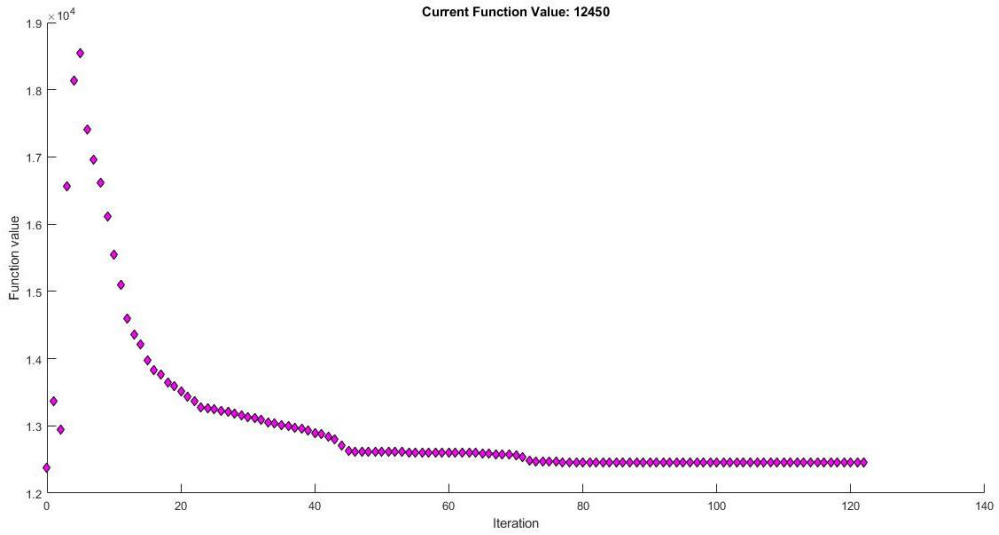

FIGURE 36 SQP ALGORITHM TEST

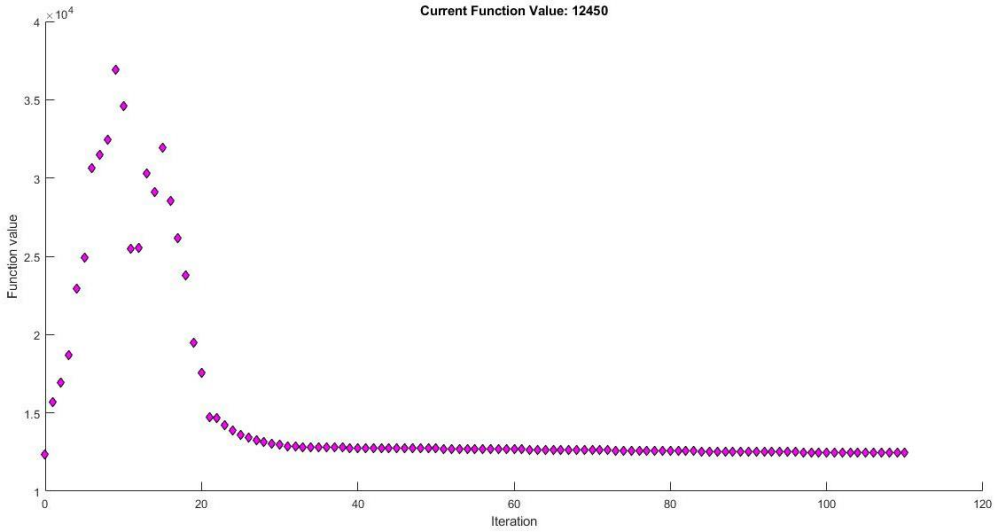

FIGURE 37 ACTIVE-SET ALGORITHM TEST

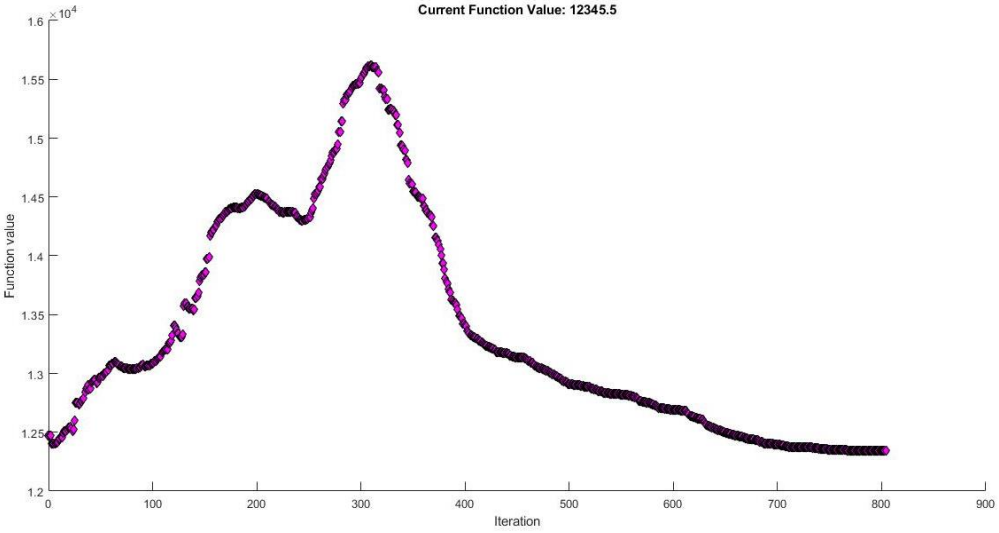

FIGURE 38 INTERIOR-POINT ALGORITHM TEST

In TABLE 5. 2 are reported the key parameters for evaluating the goodness of the three algorithms:

Algorithms	Time [s]	Function Value [Consumption Unit]	Iterations
SQP	382.45	12450	122
Active-Set	35039.75	12449.97	110
Interior-Point	1472.81	12345.52	804

TABLE 5. 2 ALGORITHMS TEST

As can be seen, the best solutions resulted from SQP and Interior-Point. The first shows a very fast runtime but a higher function value, instead the latter presents a function value lower than the former but a worst runtime. Given the results above, both the algorithms were used.

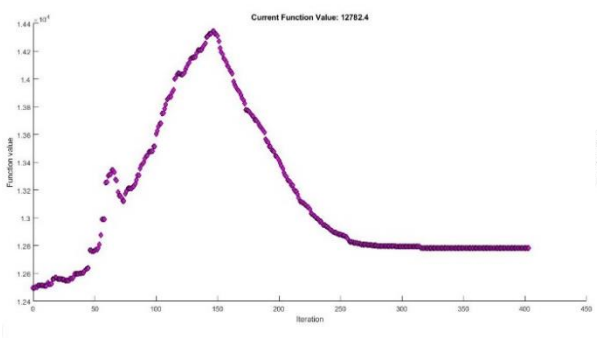

FIGURE 39 CL_2_TH_3 INTERIOR-POINT

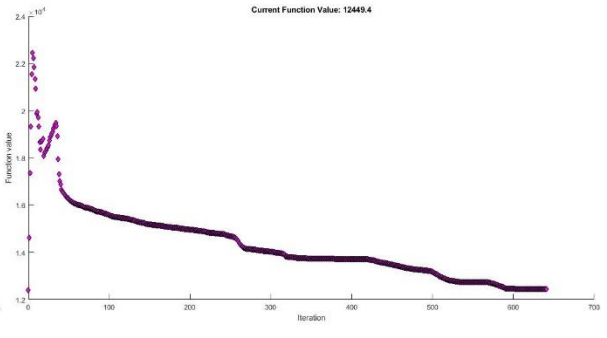

FIGURE 40 CL_2_TH_3 SQP

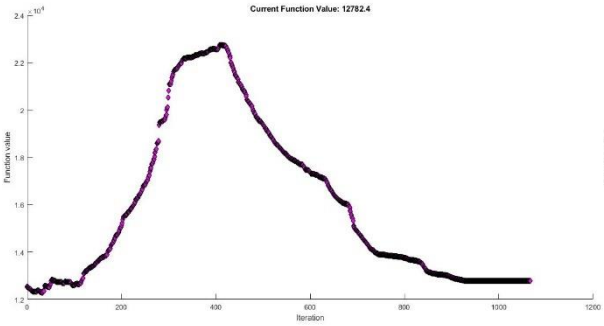

FIGURE 41 CL_3_TH_3 INTERIOR-POINT

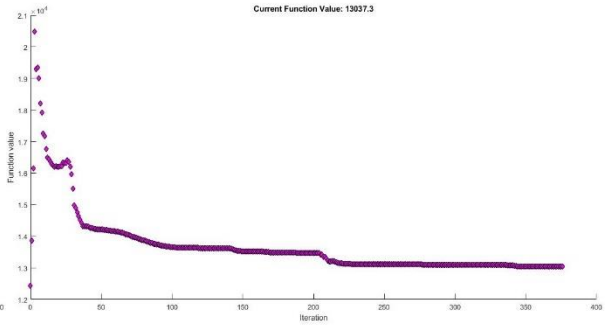

FIGURE 42 CL_3_TH_3 SQP

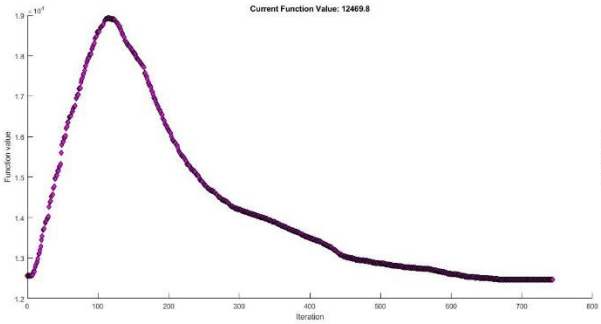

FIGURE 43 EX_CL_1_TH_3 INTERIOR-POINT

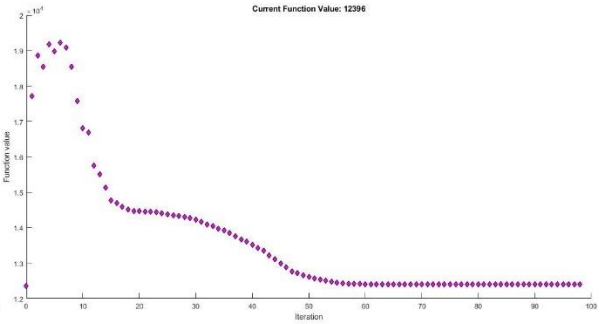

FIGURE 44 EX_CL_1_TH_3 SQP

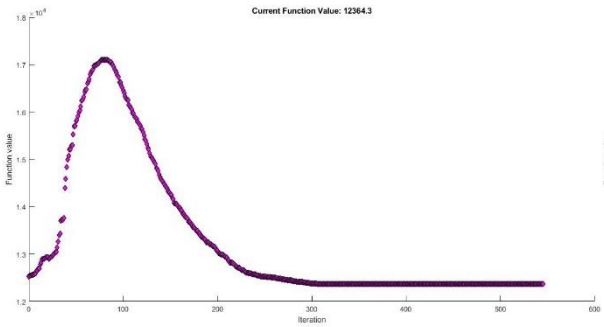

FIGURE 45 EX_CL_2_TH_3 INTERIOR-POINT

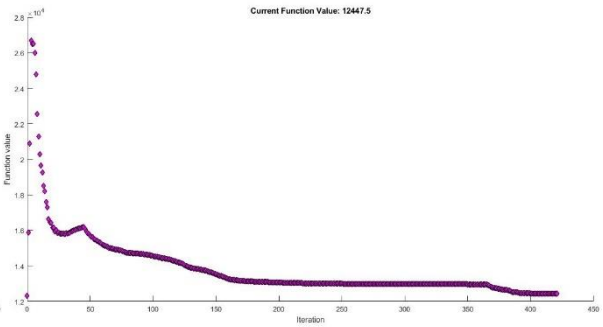

FIGURE 46 EX_CL_2_TH_3 SQP

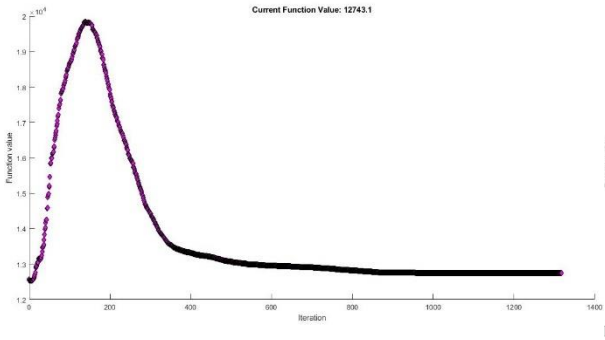

FIGURE 47 EX_CL_3_TH_3 INTERIOR-POINT

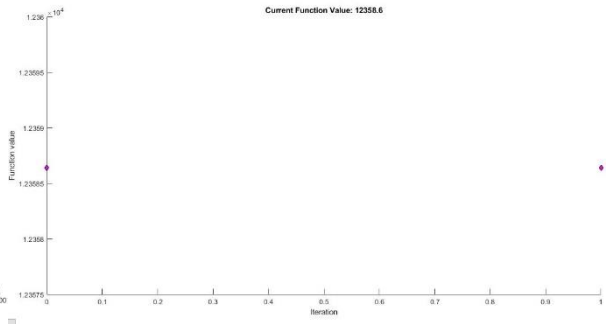

FIGURE 48 EX_CL_3_TH_3 SQP

	Interior Point		SQP		
	Gain [%]	Time [s]	Gain [%]	Time [s]	Starting Function Value [Consumption Unit]
CL_1_TH_3	0.2129	1472.81	-0.6316	382.45	12371.86
CL_3_TH_3	-3.1284	588.11	-0.4464	1328.64	12394.61
CL_3_TH_3	-2.7869	987.32	-4.8366	664.85	12435.83
EX_CL_1_TH_3	-0.9052	1510.19	-0.3077	281.81	12357.98
EX_CL_2_TH_3	-0.262	1174.21	-0.9367	1624.66	12331.99
EX_CL_3_TH_3	-0.31109	2992.27	0	55	12358.64
AVERAGE	-1.6634	1454.15	-1.1931	722.9	

TABLE 5. 3 MATLAB OPTIMIZATION SOLUTION

As can be seen from **TABLE 5. 3**, the use of SQP is preferable to Interior-Point since the former employs fewer time than the latter and a gain lower negative than Interior-Point. Even if the SQP is the best algorithm utilizable for the problem in exam, it does not allow a reduction of consumption respect to the initial function value that, is useful reminder, is the best solution found so far and thus the best found in this work with the tools used.

5.2. Scientific phase sensibility analysis

As explained at the beginning of this chapter, the β inclination of the upward oriented actuators is fixed in the analysis for maximizing the +z thrust's component exercisable in the case of a non-nominal scenario. The trend illustrated in the figures regarding the total impulse of each actuator shows clearly that the best configuration could be that with all actuators downward oriented since the total impulse of the upward actuators is extremely low compared to the downward. In order to prove that, there were tested configurations with a lower inclination in β for the upward actuators, always considering the downward free to be oriented within the limitation imposed. It was decided to conduct the analyses just utilizing the *Layout* model and the *Continuous* model thus excluding the solutions' refinement and MATLAB optimizer. This choice was inspired by the fact that the aim of these analyses is not to find some solutions substituting those found in 5.1 but to demonstrate, if the outcomes will agree, that the lower the inclination of the upward actuator/s is, the lower is the total consumption. In the follow are

reported the results of the analyses using the same graphs utilized in the optimization phase.

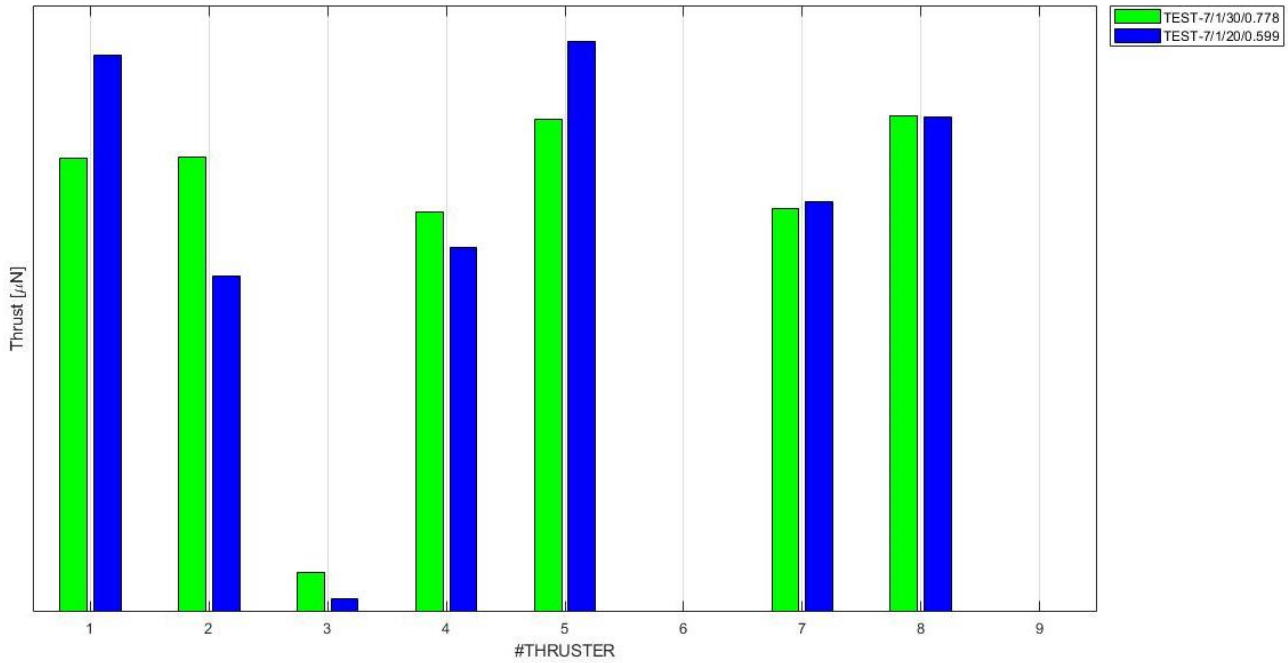

FIGURE 49 CL_1_TH_3 WITH LOWER INCLINATION TOTAL IMPULSE

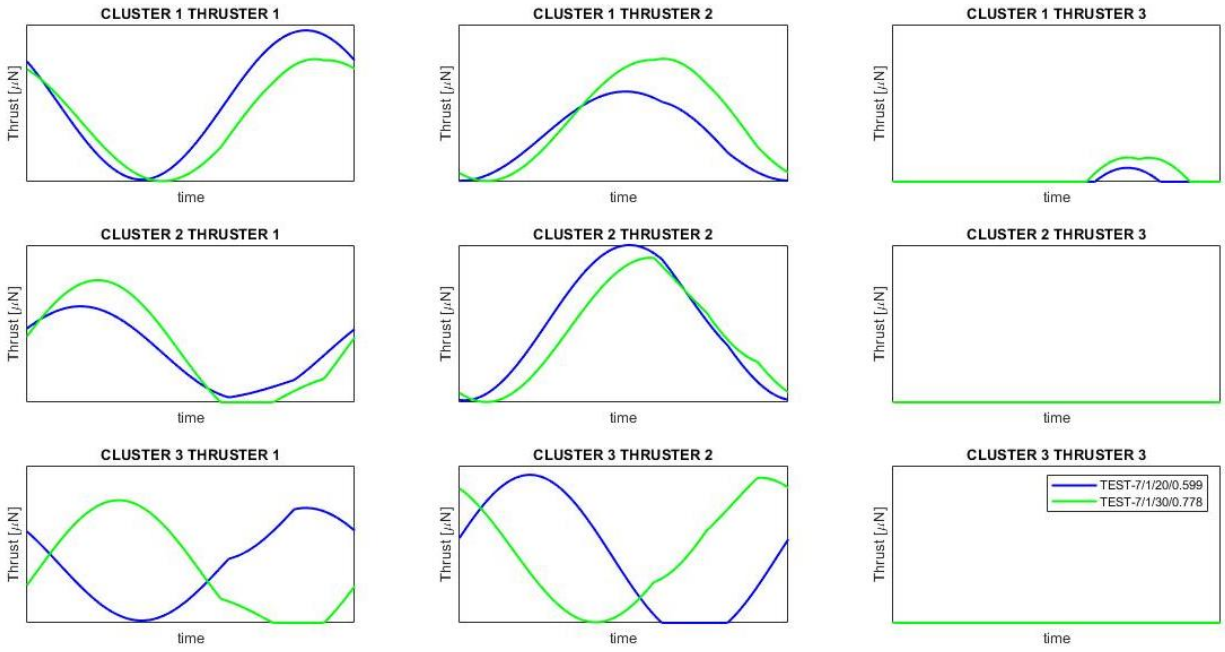

FIGURE 50 CL_1_TH_3 WITH LOWER INCLINATION THRUST VS TIME

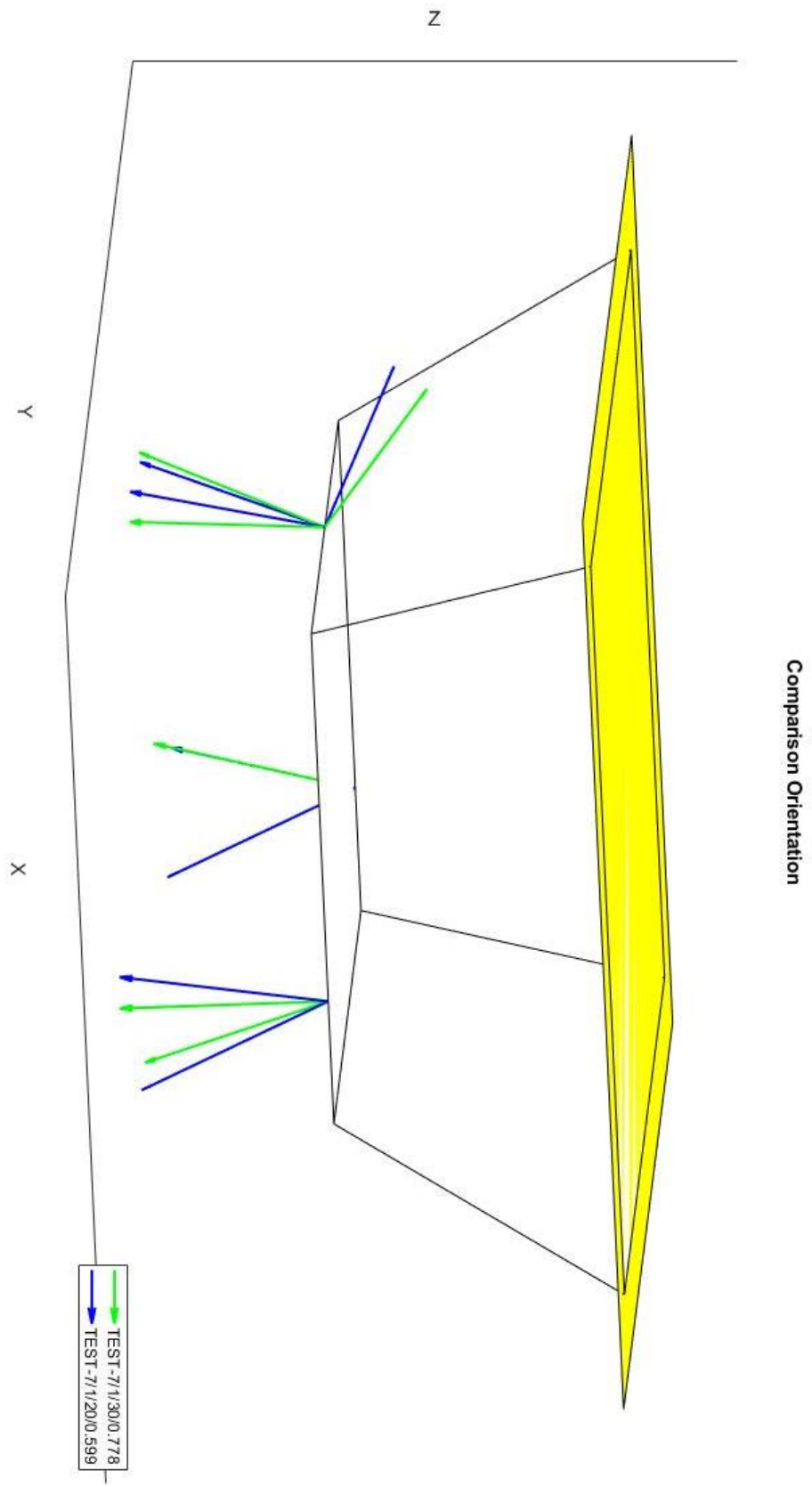

FIGURE 51 CL_1_TH_3 WITH LOWER INCLINATION ORIENTATION

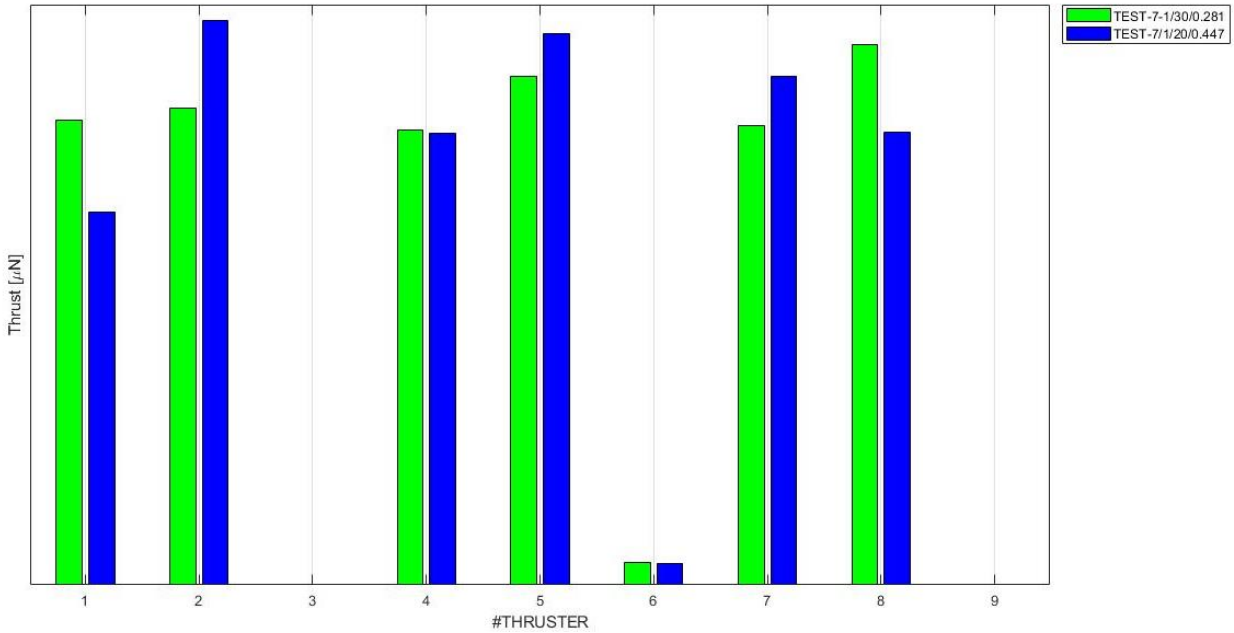

FIGURE 52 CL_2_TH_3 WITH LOWER INCLINATION TOTAL IMPULSE

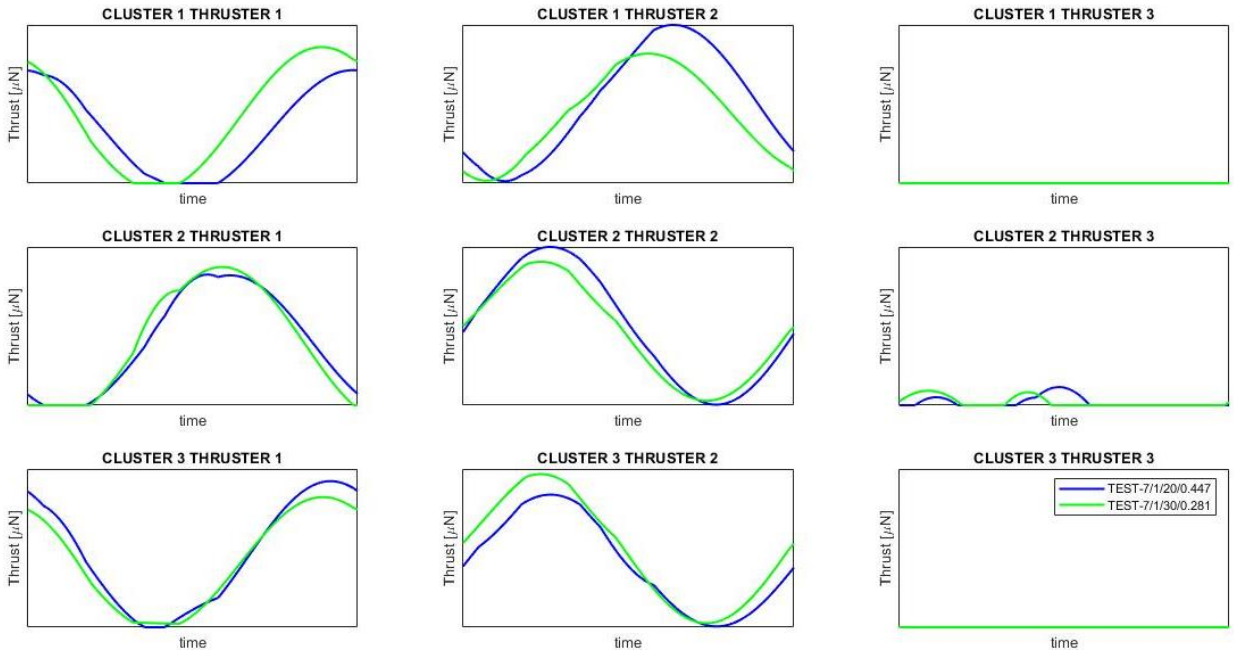

FIGURE 53 CL_2_TH_3 WITH LOWER INCLINATION THRUST VS TIME

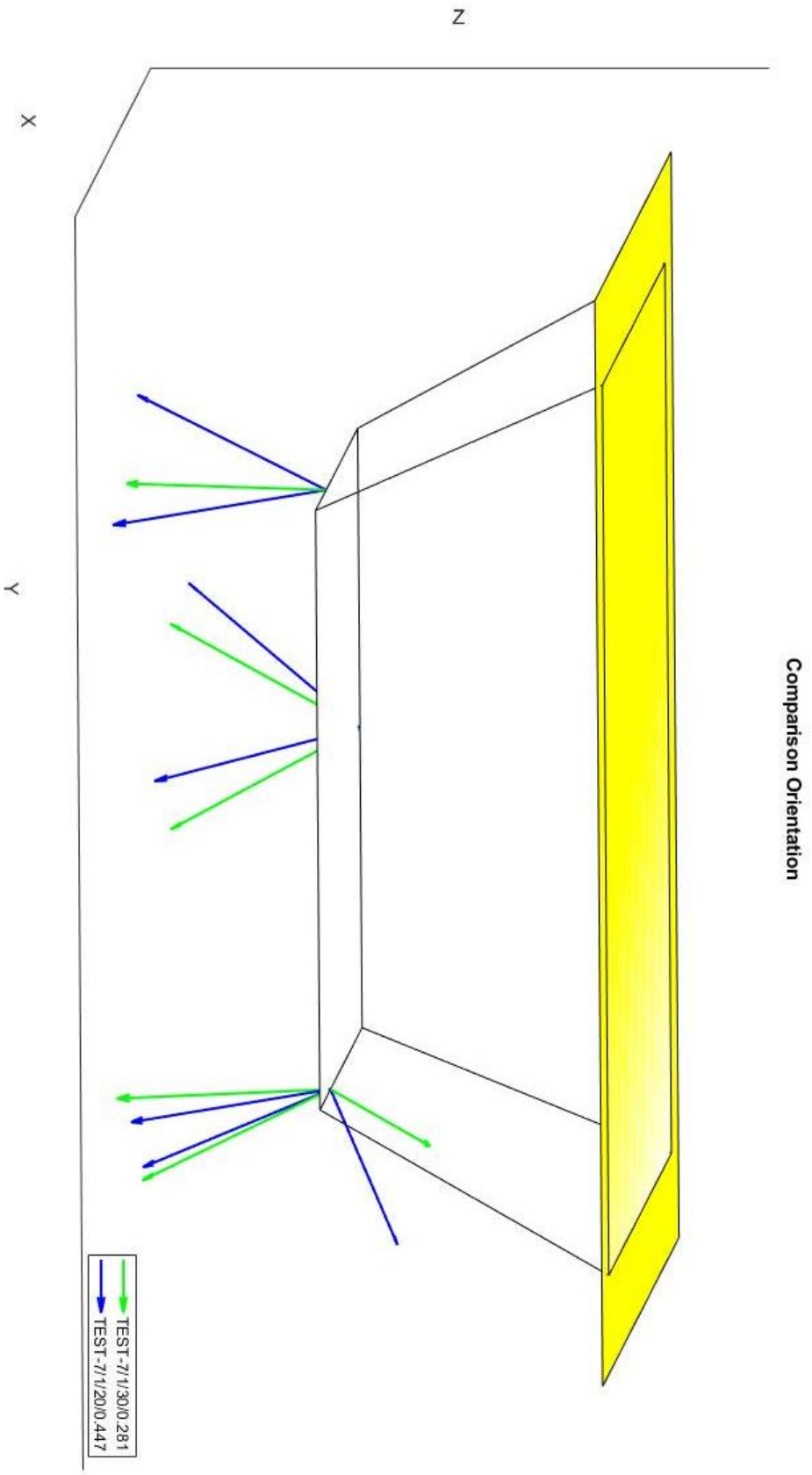

FIGURE 54 CL_2_TH_3 WITH LOWER INCLINATION ORIENTATION

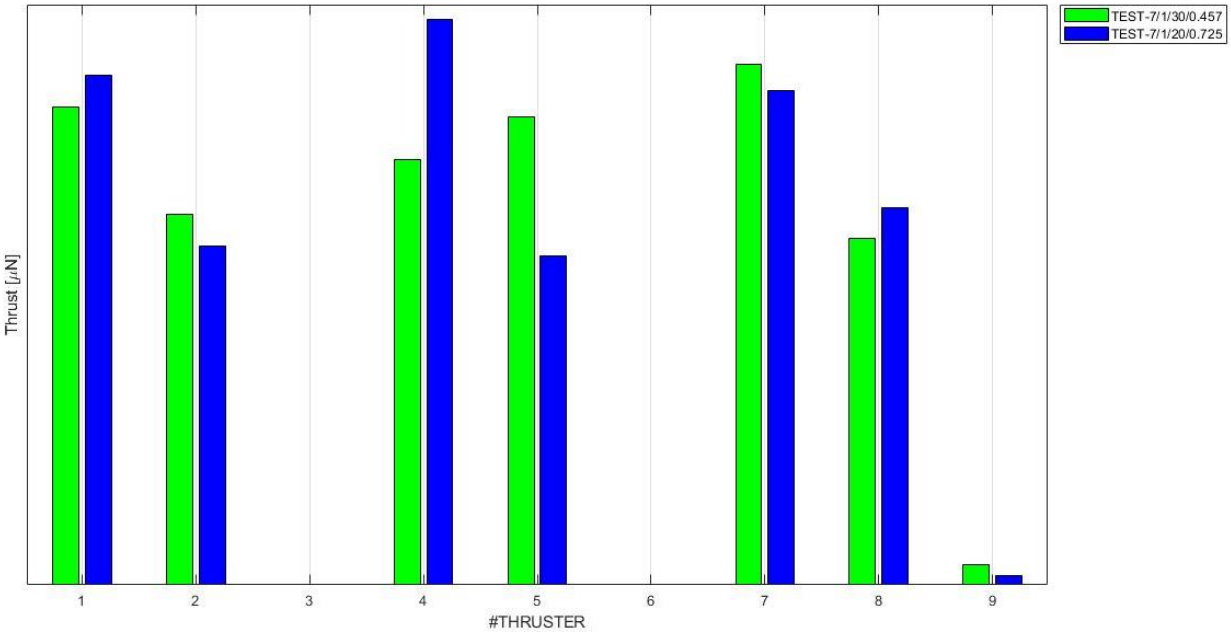

FIGURE 55 CL_3_TH_3 WITH LOWER INCLINATION TOTAL IMPULSE

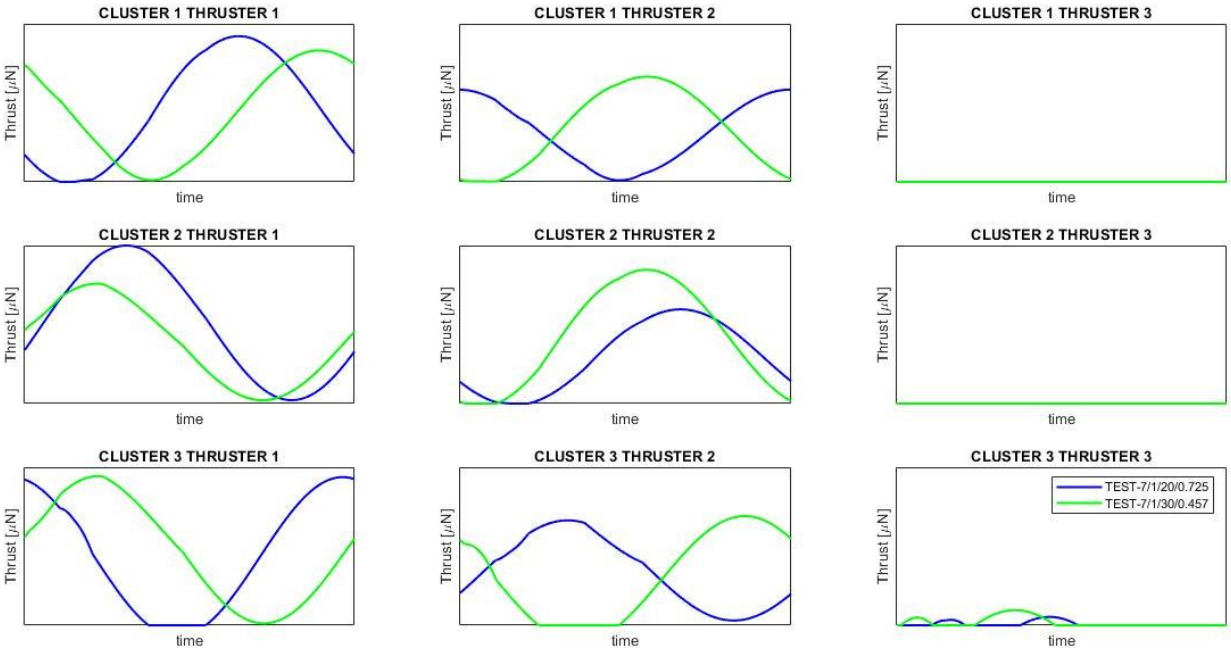

FIGURE 56 CL_3_TH_3 WITH LOWER INCLINATION THRUST VS TIME

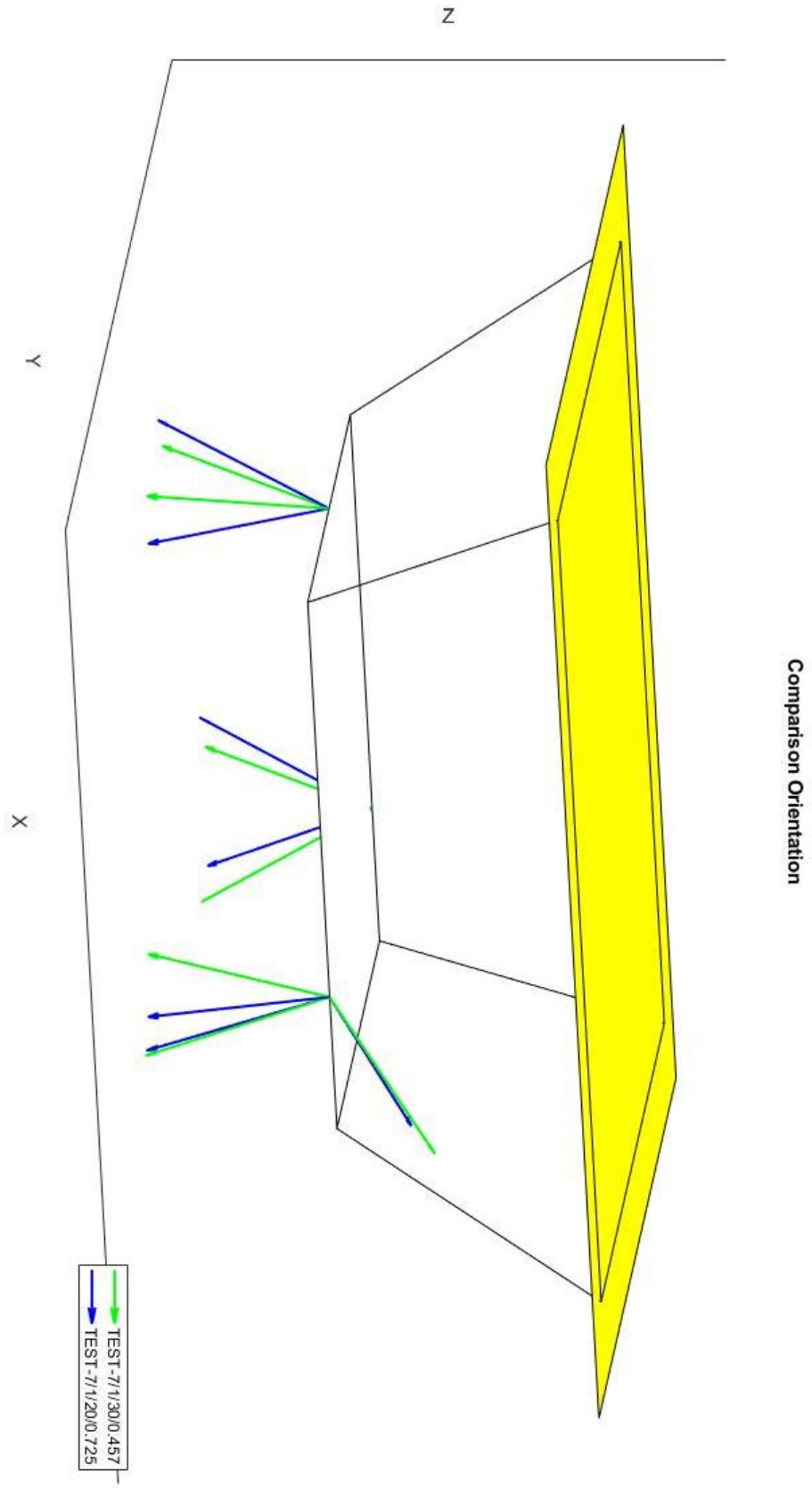

FIGURE 57 CL_3_TH_3 WITH LOWER INCLINATION ORIENTATION

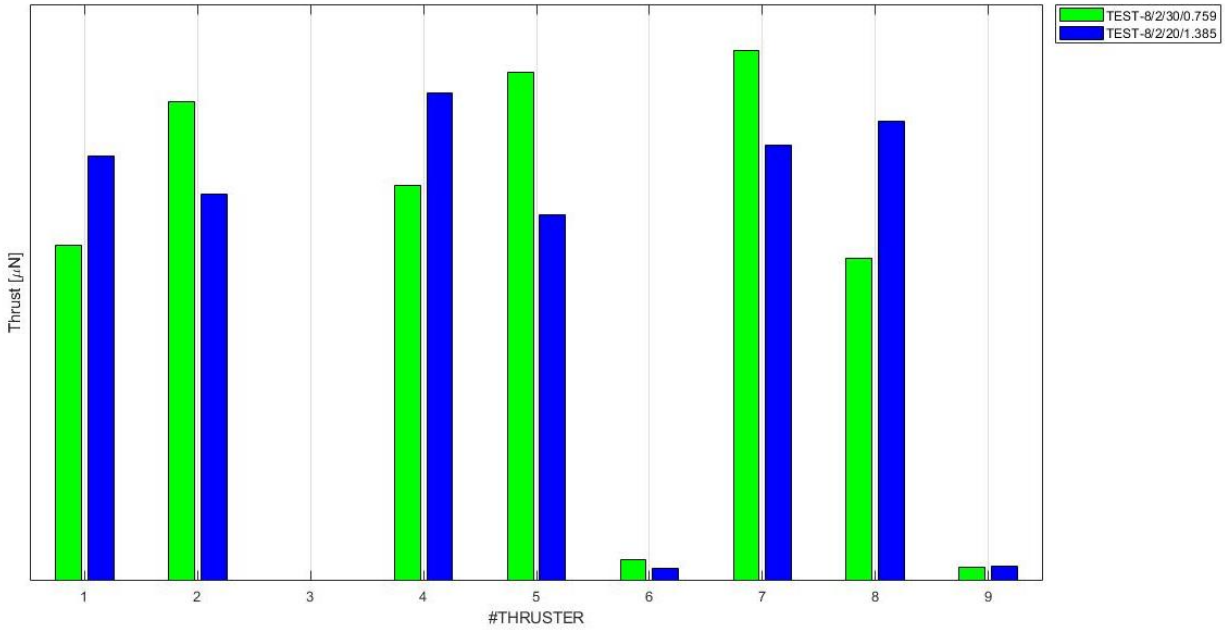

FIGURE 58 EX_CL_1_TH_3 WITH LOWER INCLINATION TOTAL IMPULSE

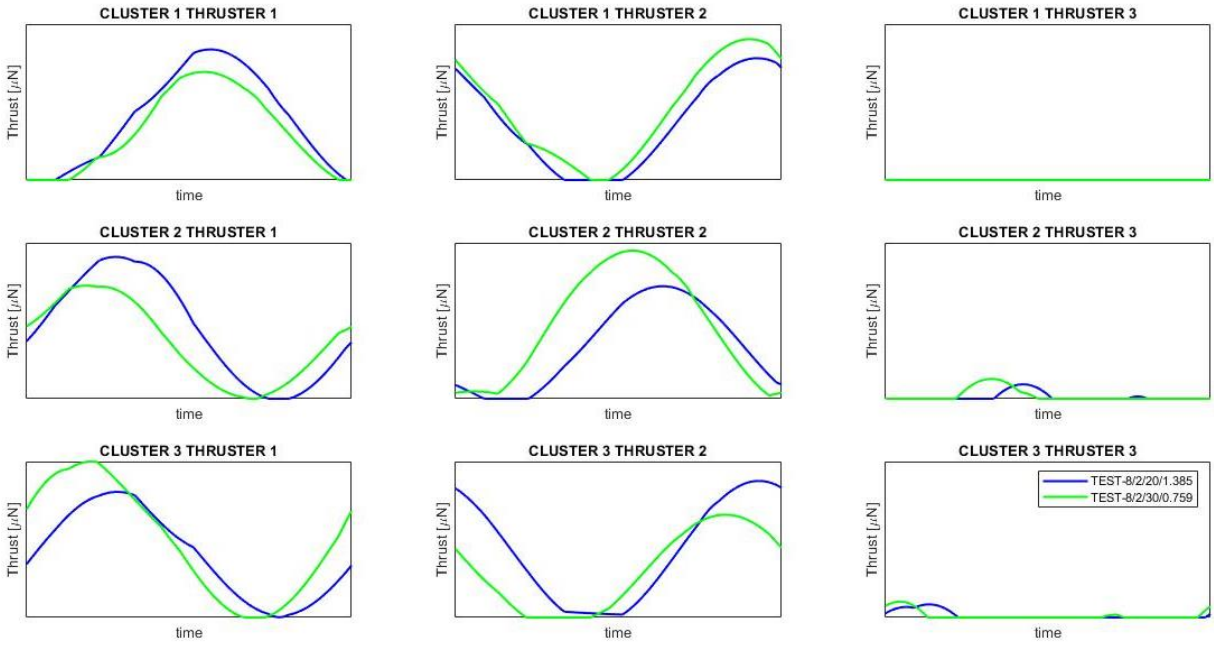

FIGURE 59 EX_CL_1_TH_3 WITH LOWER INCLINATION THRUST VS TIME

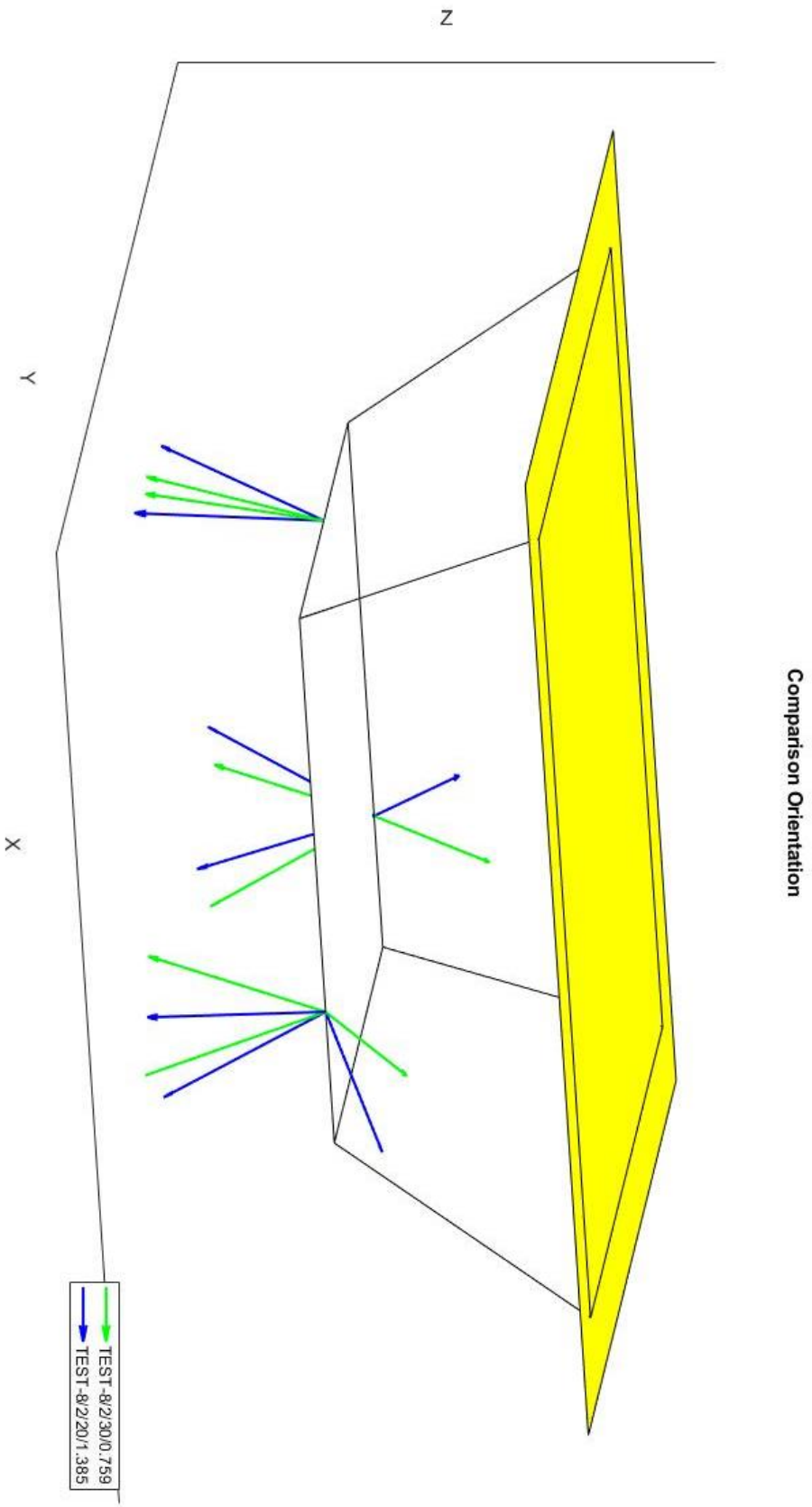

FIGURE 60 EX_CL_1_TH_3 WITH LOWER INCLINATION ***ORIENTATION***

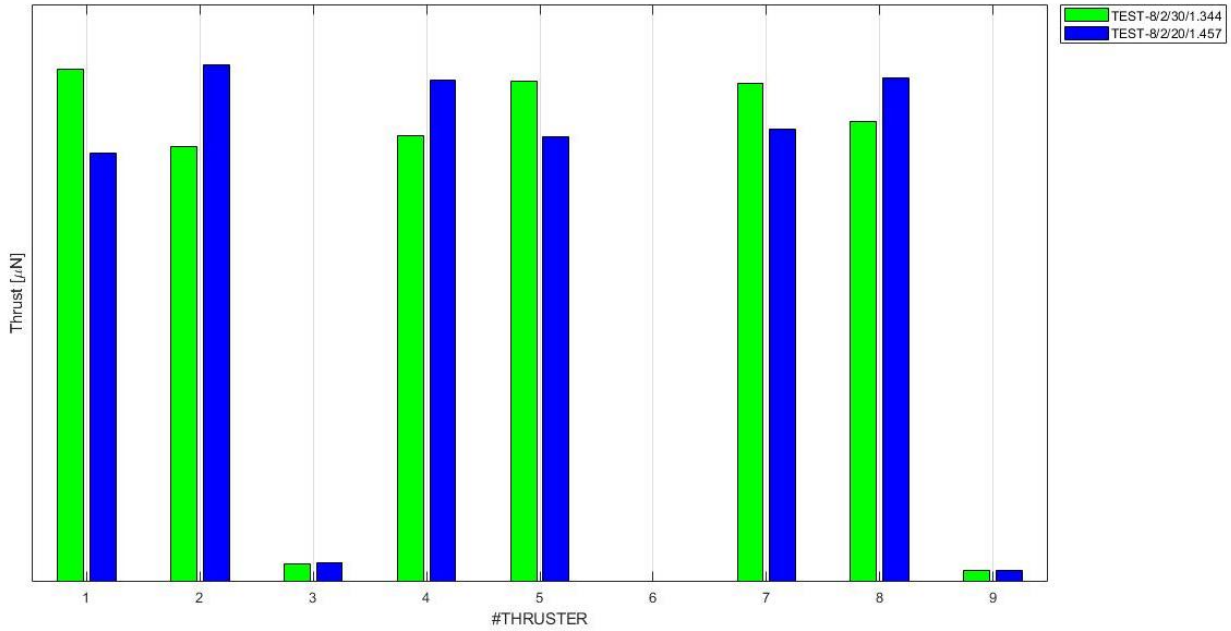

FIGURE 61 EX_CL_2_TH_3 WITH LOWER INCLINATION ***TOTAL IMPULSE***

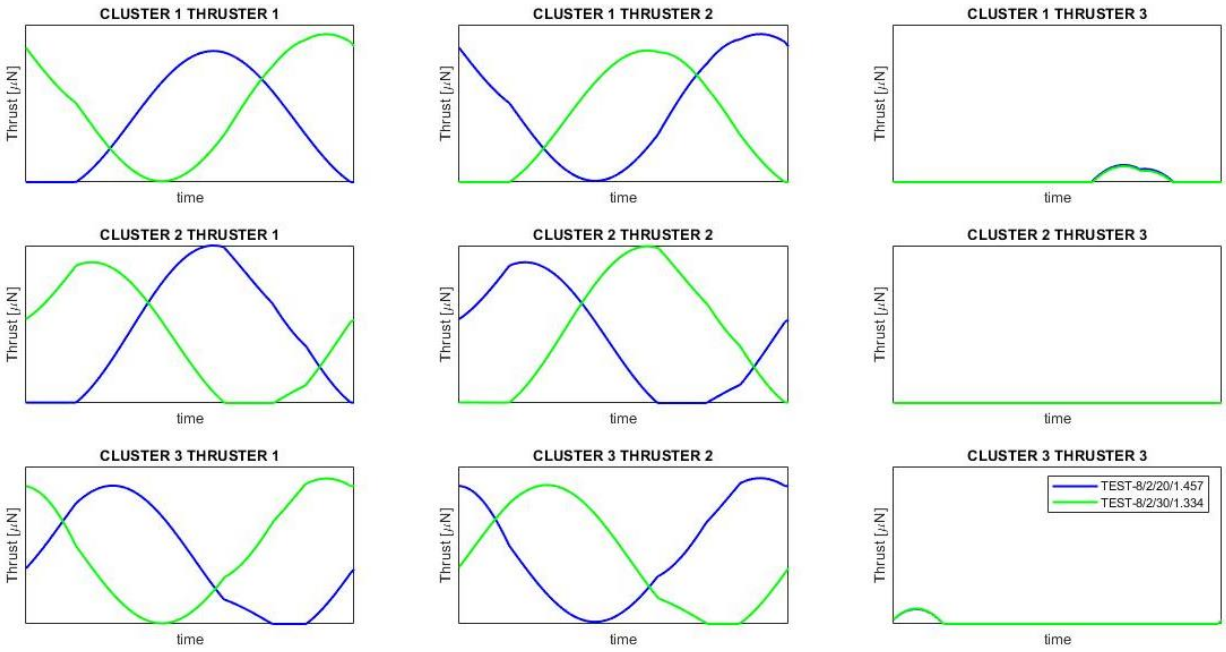

FIGURE 62 EX_CL_2_TH_3 WITH LOWER INCLINATION THRUST VS TIME

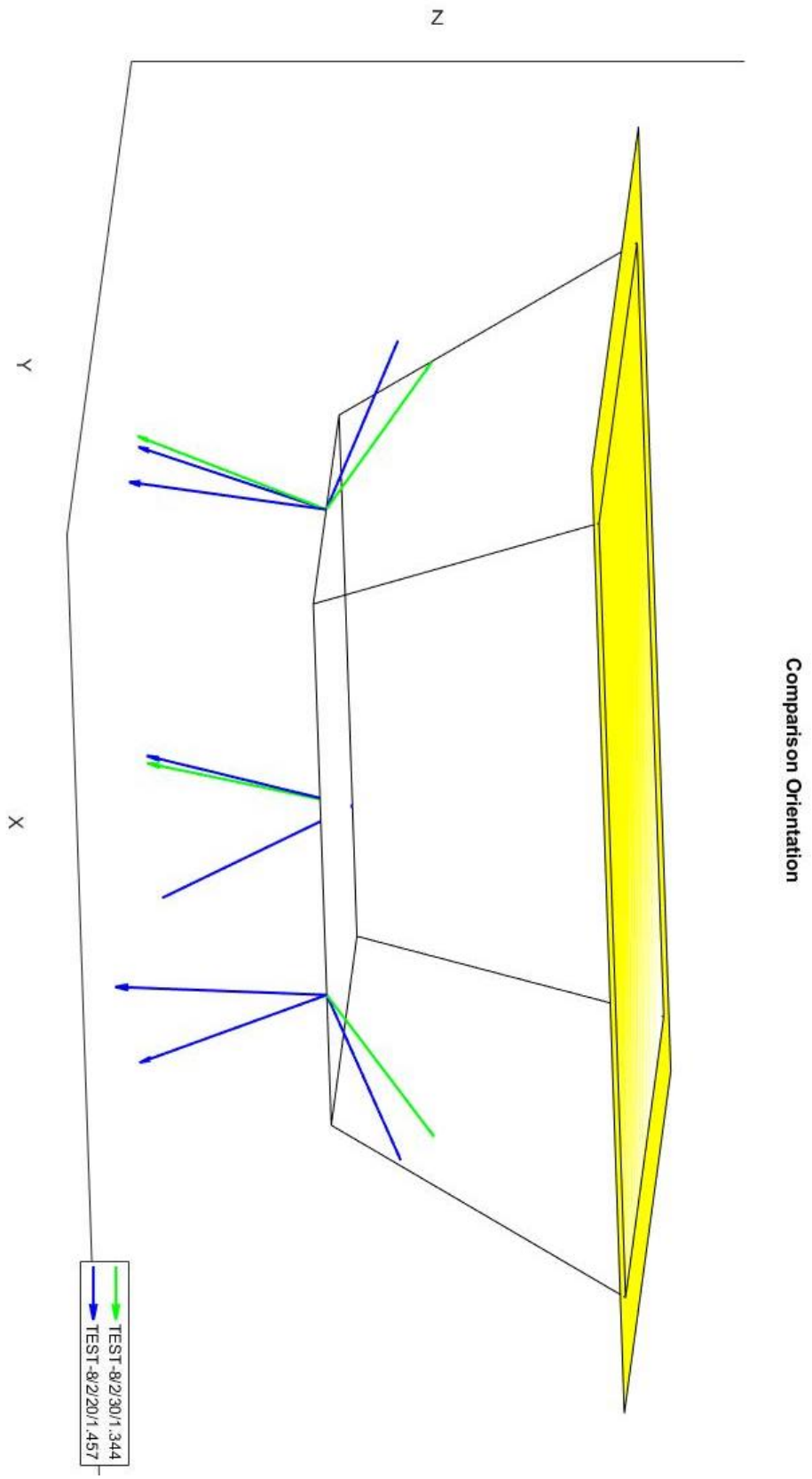

*FIGURE 63 EX_CL_2_TH_3 WITH LOWER INCLINATION **ORIENTATION***

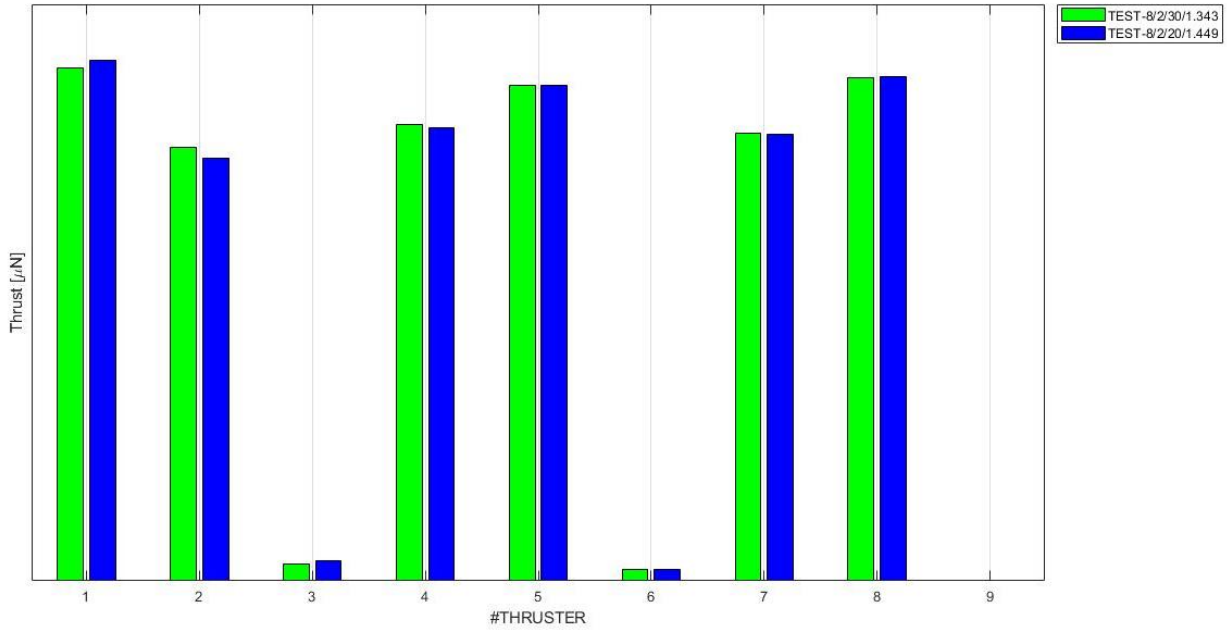

FIGURE 64 EX_CL_3_TH_3 WITH LOWER INCLINATION TOTAL IMPULSE

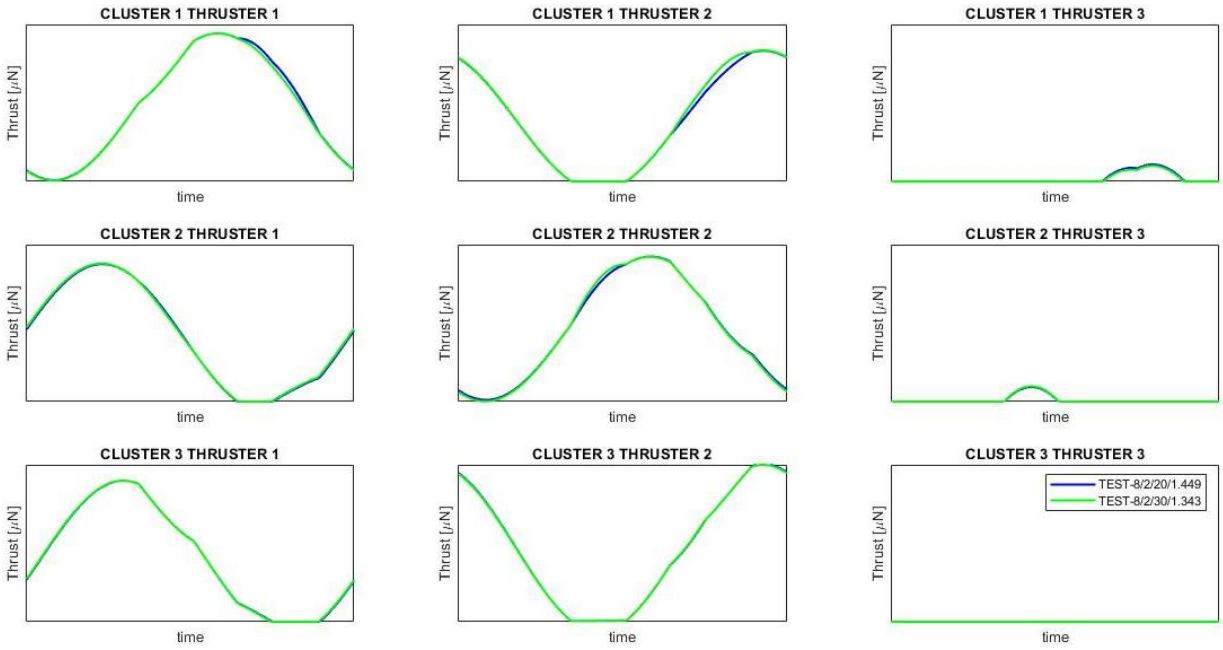

FIGURE 65 EX_CL_3_TH_3 WITH LOWER INCLINATION THRUST VS TIME

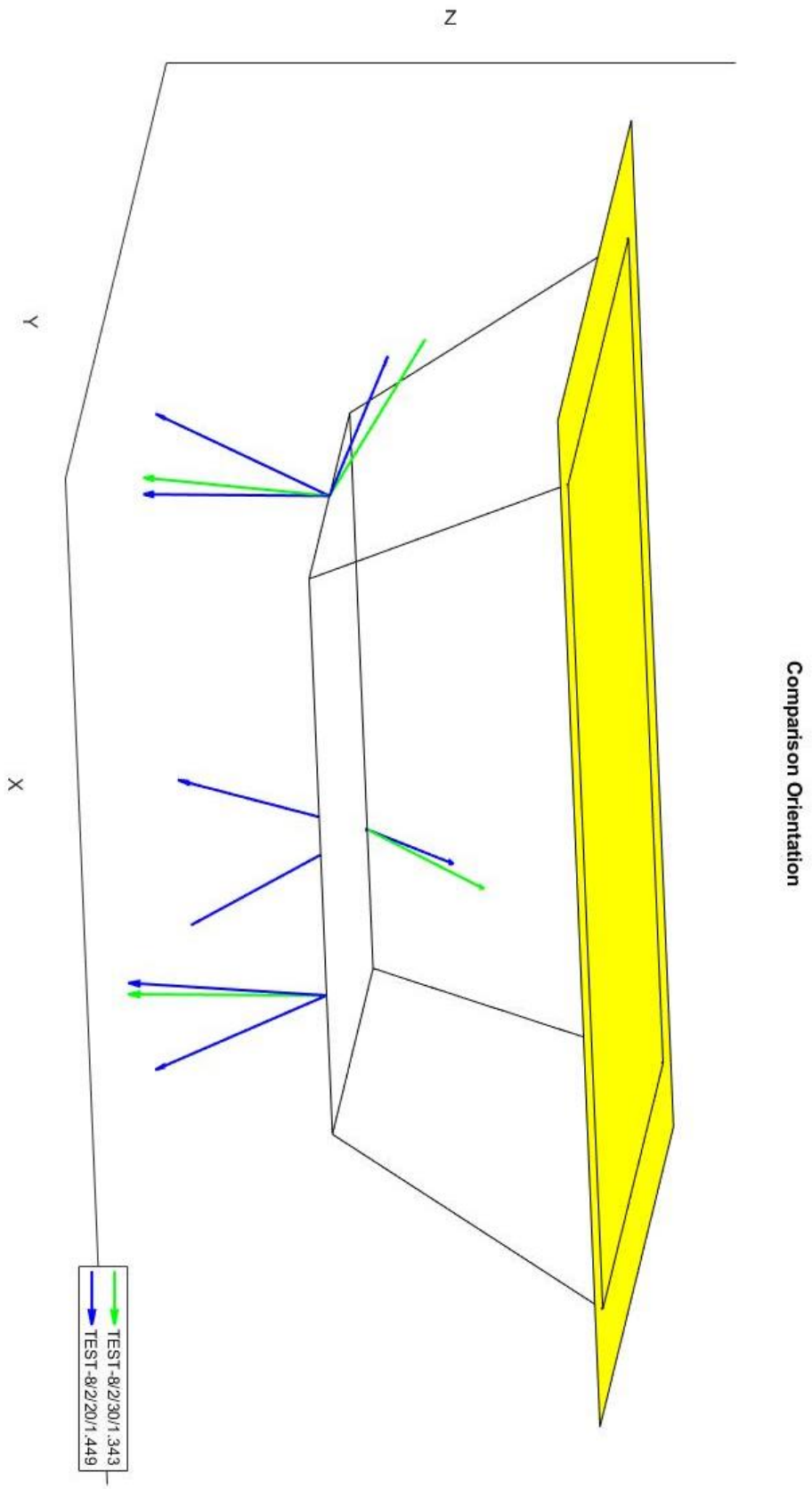

FIGURE 66 EX_CL_3_TH_3 WITH LOWER INCLINATION TOTAL ORIENTATION

The more interesting graphs are those regarding the total impulse. In fact, can be noticed a reduced total thrust exerted by upward actuators compared to the case with the maximum inclination allowed. This consideration validates the supposition made in the beginning of the chapter because in all the case except one the total consumption is reduced too. Is not useful continue along this path further reducing the inclination of the upward actuator/s since the results obtained clearly validate the beginning supposition and the analyses' outcomes can't be utilized by the hosting company because the maximum β inclination for the upward actuator/s was a programme constraint.

TABLE 5. 4 summarizes the results obtained focusing on the total fuel consumption improvement.

	Maximum β inclination [Consumption Unit]	Gain [%]	Reduced β inclination [Consumption Unit]	Gain [%]	Δ gain
CL_1_TH_3	45178.3	0.778	45260.21	0.598	-0.18
CL_2_TH_3	45404.57	0.281	45329.13	0.447	0.166
CL_3_TH_3	45324.61	0.457	45202.37	0.725	0.268
EX_CL_1_TH_3	45187.3	0.758	44902.23	1.384	0.626
EX_CL_2_TH_3	44920.51	1.344	44869.44	1.457	0.112
EX_CL_3_TH_3	44921.2	1.343	44872.99	1.449	0.106

TABLE 5. 4 GAIN COMPARISON IN SENSIBILITY ANALYSIS

5.3. Layout validation for different mission phases

The optimization conducted so far focused on the Science operational scenario that determines the forces and the torques given as input for the analysis and illustrated in **FIGURE 4** and **FIGURE 5**. In the lifespan of the spacecraft obviously there are other phases that must be faced. These phases generate forces and torques on the spacecraft that can vary in a wide range of values. There was considered a wide records for the analyses, in particular it was decided to analyse scenarios in some way linked to scientific phase and others completely unlinked to that. The scenarios that occur during the Science phase are two: the former regards the compensation of on-board instruments during the measurements (**FIGURE 67** and **FIGURE 68**) and the latter regard the orientation of an instrument (**FIGURE 69** and **FIGURE 70**).

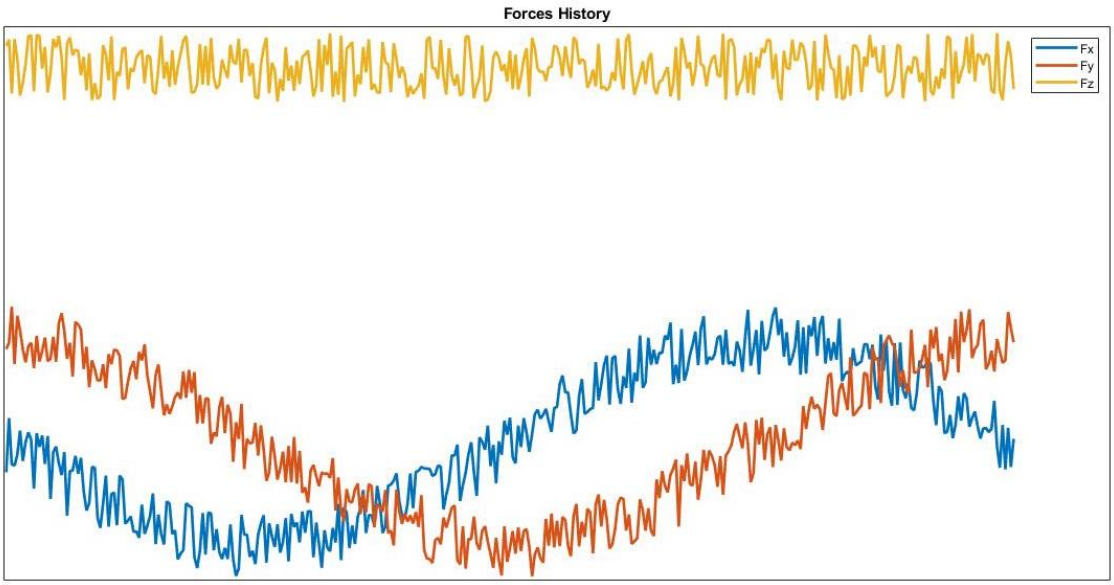

FIGURE 67 ON-BOARD INSTRUMENTS COMPENSATION FORCES

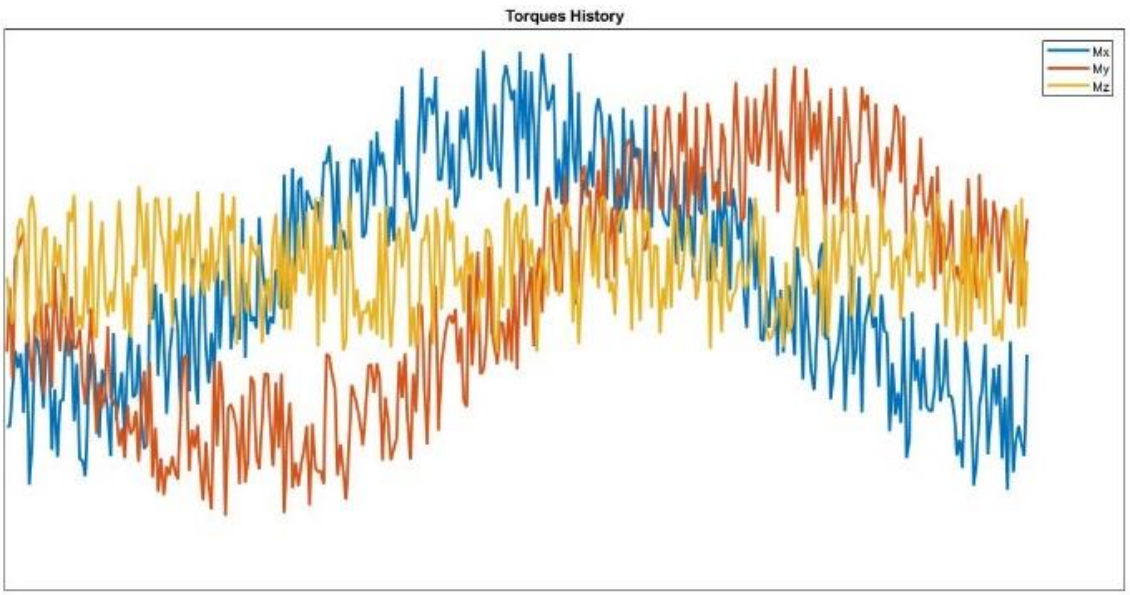

FIGURE 68 ON-BOARD INSTRUMENTS COMPENSATION TORQUES

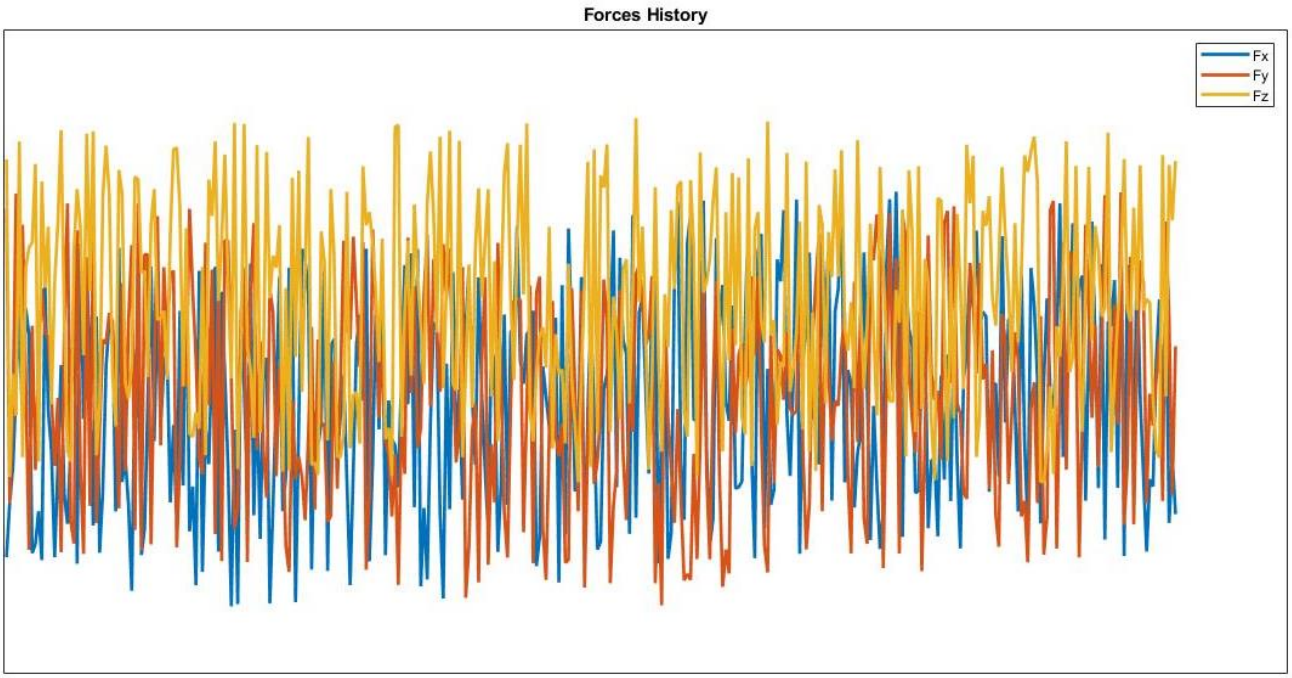

FIGURE 69 ON-BOARD INSTRUMENT ORIENTATION FORCES

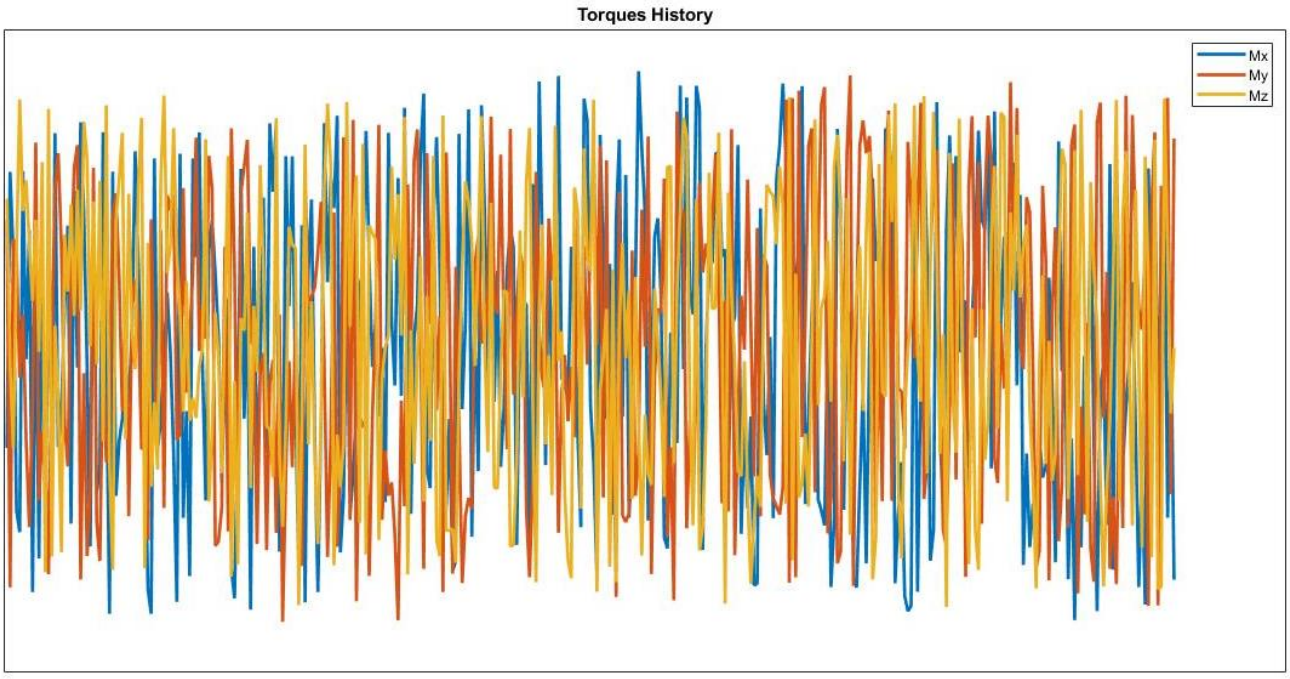

FIGURE 70 ON-BOARD INSTRUMENT ORIENTATION TORQUES

The other three scenarios are called ‘Scenario A’, ‘Scenario B’ and ‘Scenario A’ for confidentiality reason and their forces and torques history are reported in the following figures.

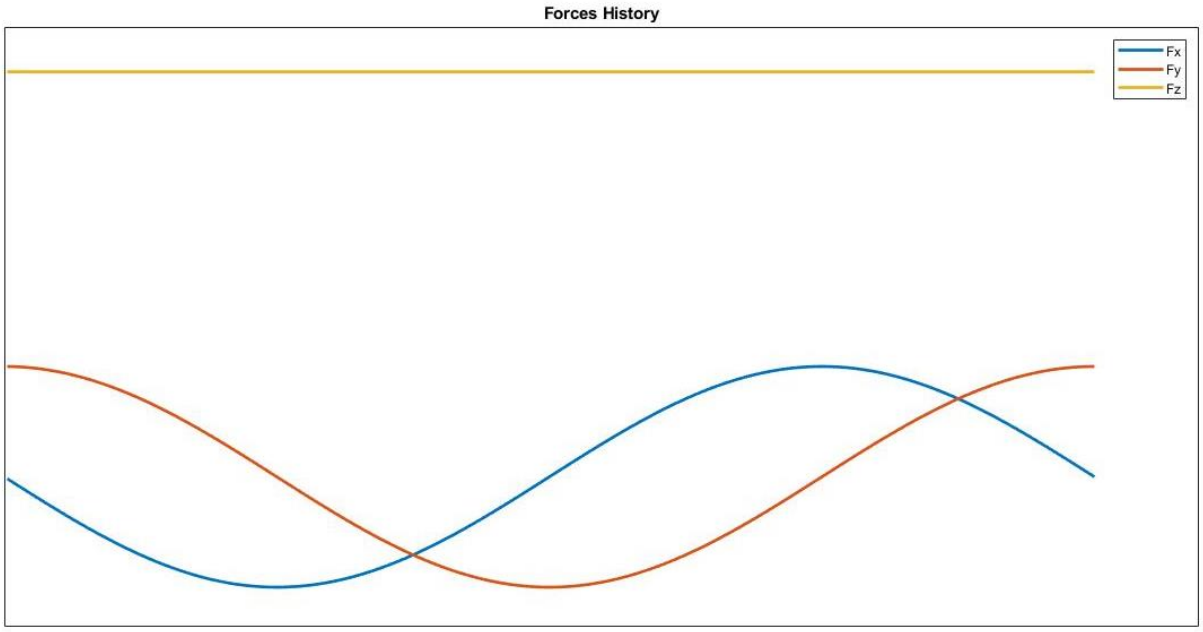

FIGURE 71 SCENARIO A FORCES

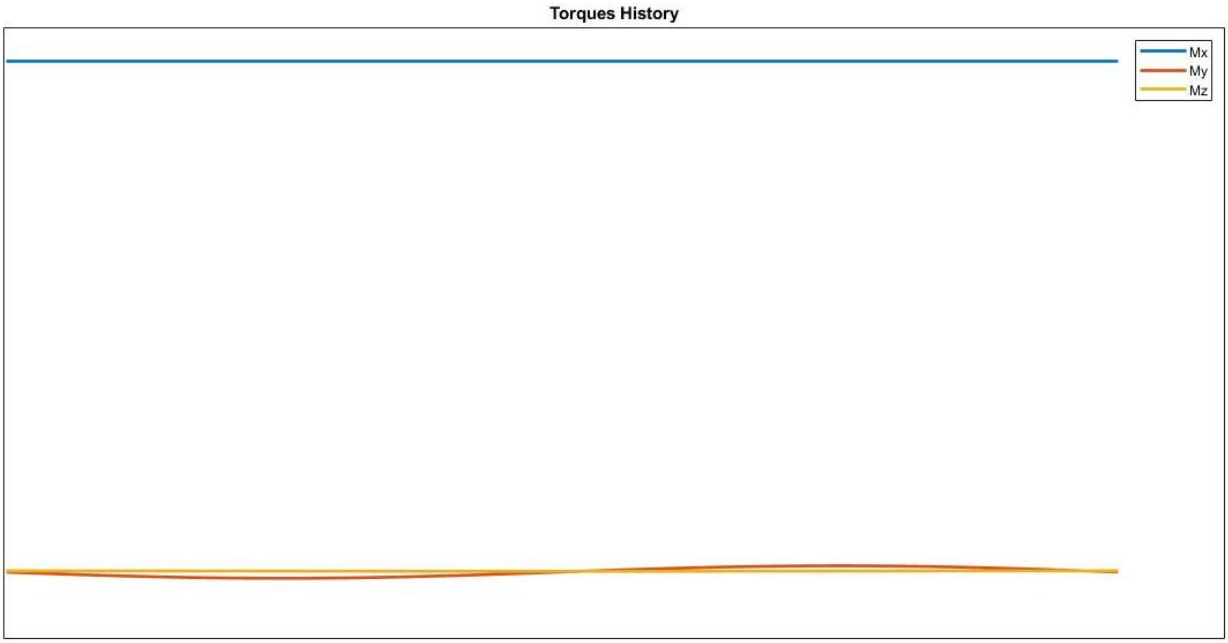

FIGURE 72 SCENARIO A TORQUES

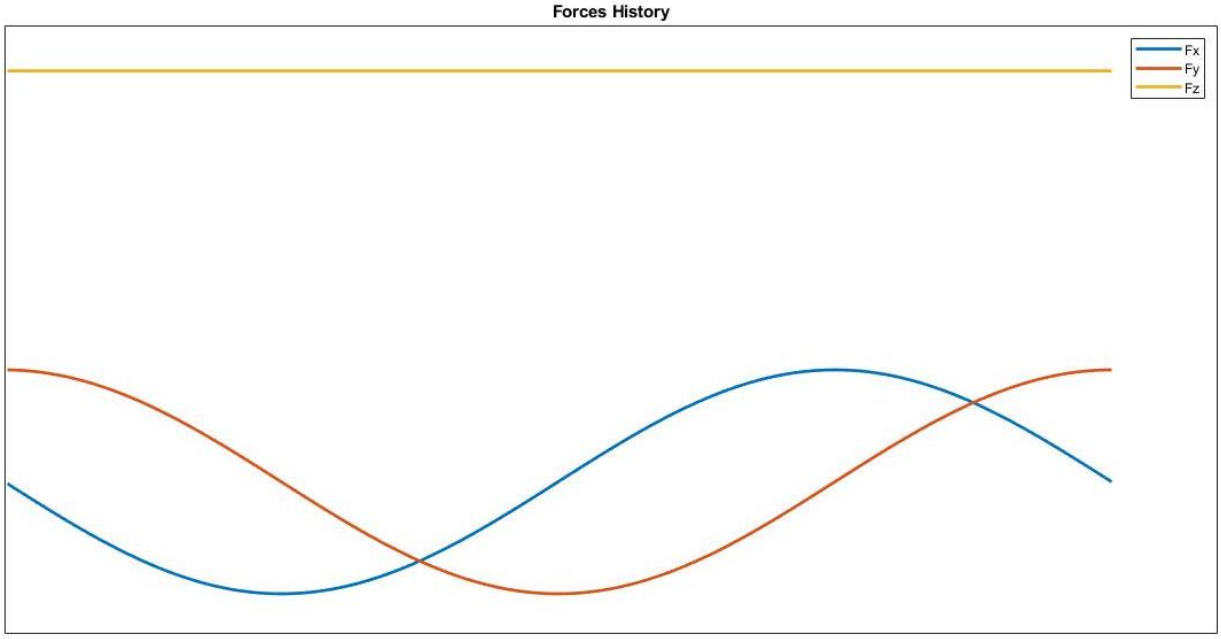

FIGURE 73 SCENARIO B FORCES

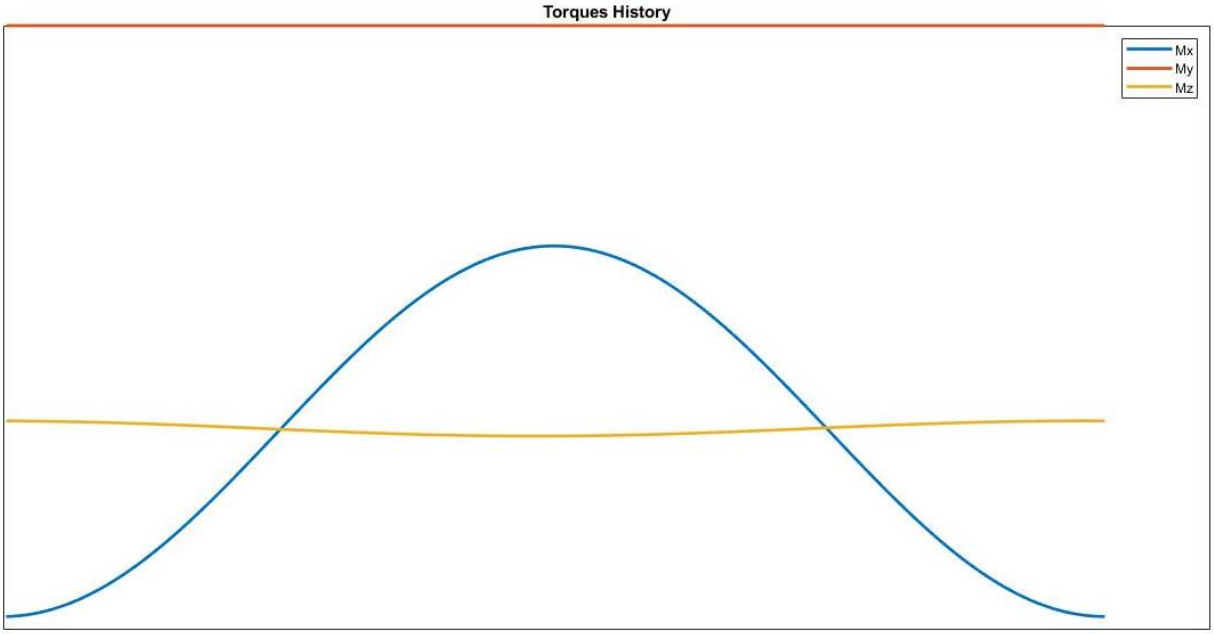

FIGURE 74 SCENARIO B TORQUES

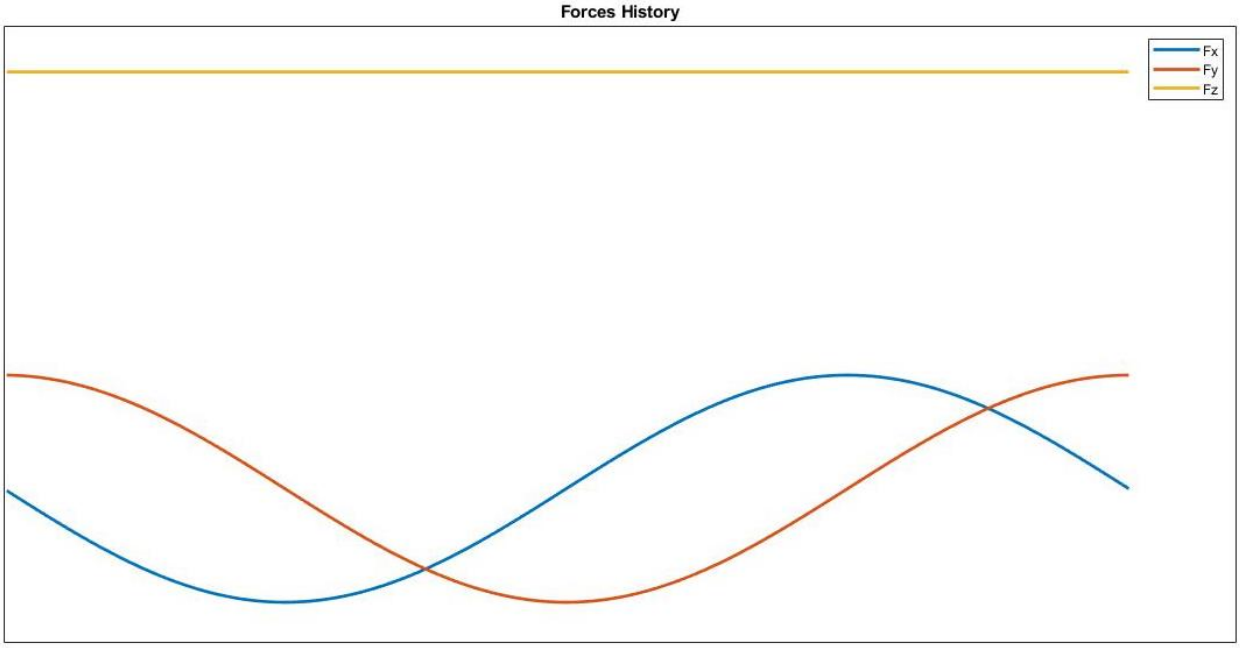

FIGURE 75 SCENARIO C FORCES

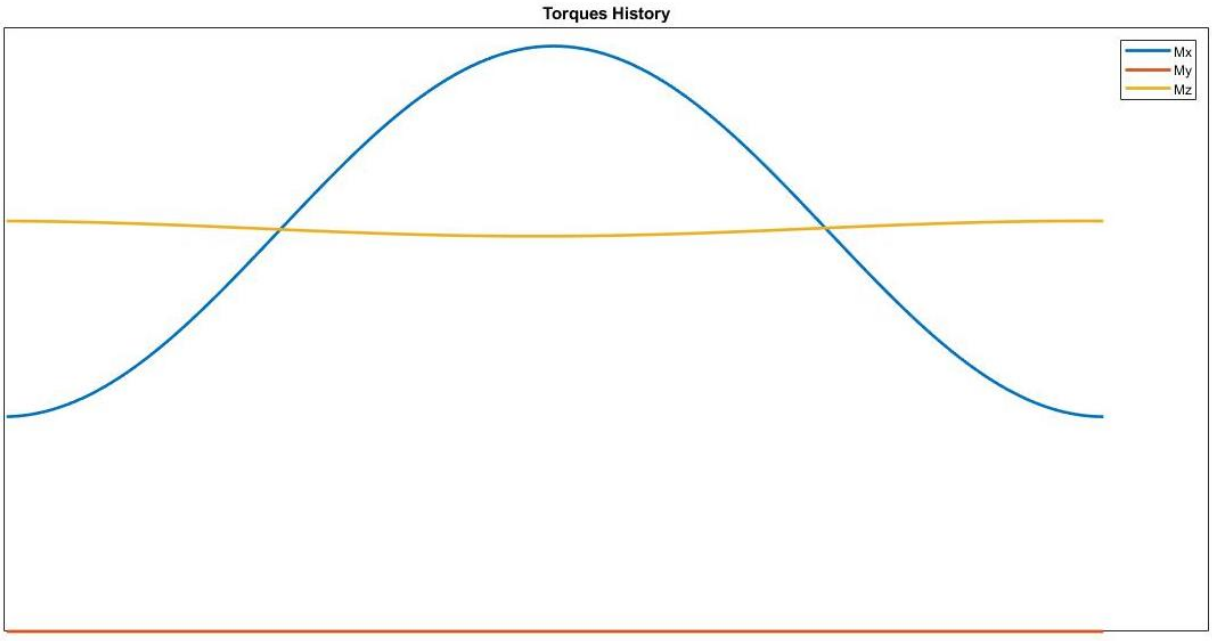

FIGURE 76 SCENARIO C TORQUES

The configurations found, as said before, are optimized for the science scenario and it is not granted that are feasible for the other possible scenarios. From an engineering point of view, it would be desirable that the configurations found will be feasible for other scenarios too. This result would reduce the complexity of the whole spacecraft from a system engineering point of view. In order to test this, it was conducted an analysis using the *Continuous* model over the whole set of instants. It was used this model because the orientations are fixed, since they must be tested. The analysing results are shown in **TABLE 5. 5**. In this table are reported the error occurred during the analysis and therefore the infeasibilities. Just one case presents a feasibility for the input forces and torques and is not in Science phase. The results show that in 'on-board instruments compensation' the total error is not too large, thus the layout considered could act in synergy with other actuators to stabilize the spacecraft. The same speech could be made for 'Scenario A' and 'Scenario B' layouts but not for 'Instrument Orientation' and 'Scenario C' layouts that have such a big total error that would be improper the DFCAS actuators' utilization in collaboration with other control system to stabilize the spacecraft.

	Science Phase		Non-Science Phases		
	On-board instruments compensation	**Instrument orientation**	**Scenario A**	**Scenario B**	**Scenario C**
CL_1_TH_3	384	89368	519	4475	519948
CL_2_TH_3	434	92803	1023	10	566815
CL_3_TH_3	340	92799	644	0.43	516984
EX_CL_1_TH_3	171	68144	0	5	365926
EX_CL_2_TH_3	106	52008	150	85	130022
EX_CL_3_TH_3	129	56307	187	120	517748

TABLE 5. 5 TOTAL ERROR FOR DIFFERENT MISSION SCENARIOS

The configurations analysed represent the best layout found in the case of maximum β inclination. Could be useful to compare these configurations to those found in 5.2. The outcomes of the analysis for the configurations with lower β inclinations can be found in **TABLE 5. 6** in which is represented the sum of errors on forces and torques derived from (10). The comparison between the two types of configurations is shown in **TABLE 5. 7**. In this table are compared the total error derived from and **TABLE 5. 5** and **TABLE 5. 6**. The value shown in **TABLE 5. 7** derive from the following formula:

$$value = \frac{error_{\beta lower} - error_{\beta max}}{error_{\beta max}} \times 100$$

	Science Phase		Non-Science Phases		
	On-board instruments compensation	**Instrument orientation**	**Scenario A**	**Scenario B**	**Scenario C**
CL_1_TH_3	294.889	93166	457.238	4176	521817
CL_2_TH_3	423.952	89145	1110	56.0309	579886
CL_3_TH_3	420.548	103353	1418.04	1068.18	616941
EX_CL_1_TH_3	175.212	56742	0	25.5336	514369
EX_CL_2_TH_3	80.6076	53124	144.47	0	257420
EX_CL_3_TH_3	114.834	58471	28.1785	492.026	516719

TABLE 5. 6 TOTAL ERROR FOR DIFFERENT MISSION SCENARIOS AND LOWER INCLINATION

	Science Phase		Non-Science Phases		
	On-board instruments compensation [%]	**Instrument orientation [%]**	**Scenario A [%]**	**Scenario B [%]**	**Scenario C [%]**
CL_1_TH_3	-23.21	4.25	-11.9	-6.69	0.36
CL_2_TH_3	-2.32	-3.94	8.52	460.31	2.3
CL_3_TH_3	23.69	11.37	120.19	248449.22	19.33
EX_CL_1_TH_3	2.46	-8.15	0	410.67	40.57
EX_CL_2_TH_3	-23.96	-2.67	-3.69	-85	97.98
EX_CL_3_TH_3	-10.98	3.84	-84.93	310.02	-0.2
AVERAGE	-34.3	4.69	28.19	249623.53	160.35

TABLE 5. 7 COMPARISON BETWEEN CONFIGURATIONS

As can be seen there are no significant improvement using actuators with a lower β inclination, in fact can be noticed a worsening in almost all the scenarios considered. This outcome could be explained through a geometrical consideration. To a parity of thrust and application-point, an actuator with a lower β inclination could exert a lower torque compared to an actuator with a higher β inclination and therefore, in some cases the control law couldn't be satisfied.

These results could be interpreted in another way. The two types of configurations are, in some way in contrast. In fact, a total fuel consumption could be obtained decreasing the β inclination for the upward actuator/s (as shows **TABLE 5. 4**) but the science optimized configurations have a worse behaviour considering different mission scenarios (see **TABLE 5. 7**).
Given this contrast between configurations, belong to the hosting company to decide which configuration better respond to the programme's demands.

5.4. AOCS layout study

The methodologies used so far for the DFACS system will be applied to the AOCS's actuators in order to find some feasible configurations concerning the appropriate scenario. The actuators of the AOCS system are of on/off type, i.e. the thrust that they can exert is zero or a fixed value. The model utilized so far could not represent this new scenario since the thrust variable u now is no more a continue variable but a discrete one, thus the model must be modified to provide some reliable results. In order to don't bring a lot of changes in the models utilized, it was decided to proceed in the following way.
Firstly, it was used the *Layout* model with an appropriate discretization and time domain with the u set as continue variables. In this way some orientations are found and, although the thrusts are not discretized, they can be used in the iteration module. Reducing the angles' discretization (5 points for α and β), the number of variables decreases and the thrust can be considered as a logical variable. It was decided to proceed with just one iteration per case because these runs require a significant amount of time to be completed and the analyses to conduct are many. The previous procedure was adopted because if in the *Layout* model u had set as binary, as well α and β, the computational load would have been too heavy and the memory would have been saturated soon.
A further complication derives from the application-points. There are many application-points on the spacecraft surface that could be chosen, therefore there are many possible combinations of actuators' positioning. Modify the model to comply with choice of application-points and the optimization of them would be too expensive. Taking this observation into account, the application-points were chosen a priori, considering some symmetries of them w.r.t. the reference frame.

The AOCS system will be used during many mission's phases including the separation from launcher. The AOCS system must be able to prevent rotation around any of the axes and, in case this happens, resettle the spacecraft in order to maintain the planned orientation in space. Therefore, the input for the analysis regards only the torques and not the forces. Considering this observation, the model can be further simplified removing the constraints concerning the forces and leaving just those on torques. The first run of the analysis considers that the maximum torque is requested along each axis in all the instants. Since the axes are three and the torque can have a positive or a negative value, there are 2^3 possible combinations.

As for the DFACS system there are limitations on the allowed orientations and these must be taken into account in the *Layout* model. These limitations derive from the presence of walls for α and from the presence of the solar panel and instrumentations for β and differ on the basis of application-points.
The α angle of the actuators mounted on a panel corner can vary in a range of ±90° w.r.t. the line joining the application-point and the origin of the reference frame, while the actuators positioned in the middle of a panel can have an orientation of ±60° respect to that line.
The limitations on the β angle, instead, are different for the actuators mounted above the origin of the reference frame (+z) and below (-z).

There have been considered the cases with six, eight or ten actuators positioned in a clever way and in the figures in the next pages are illustrated the actuators' orientations mounted on the spacecraft.

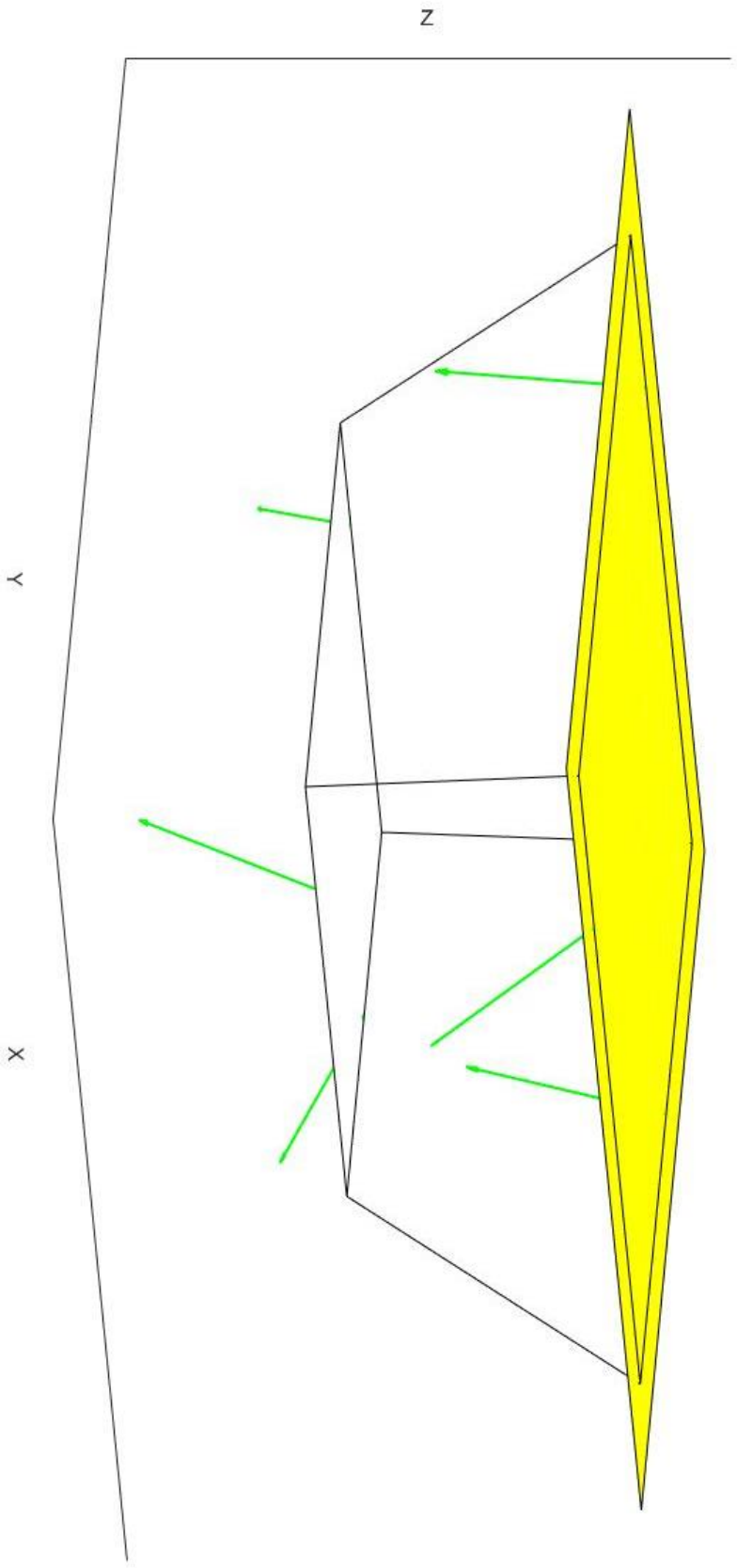

FIGURE 77 AOCS 6 THRUSTERS

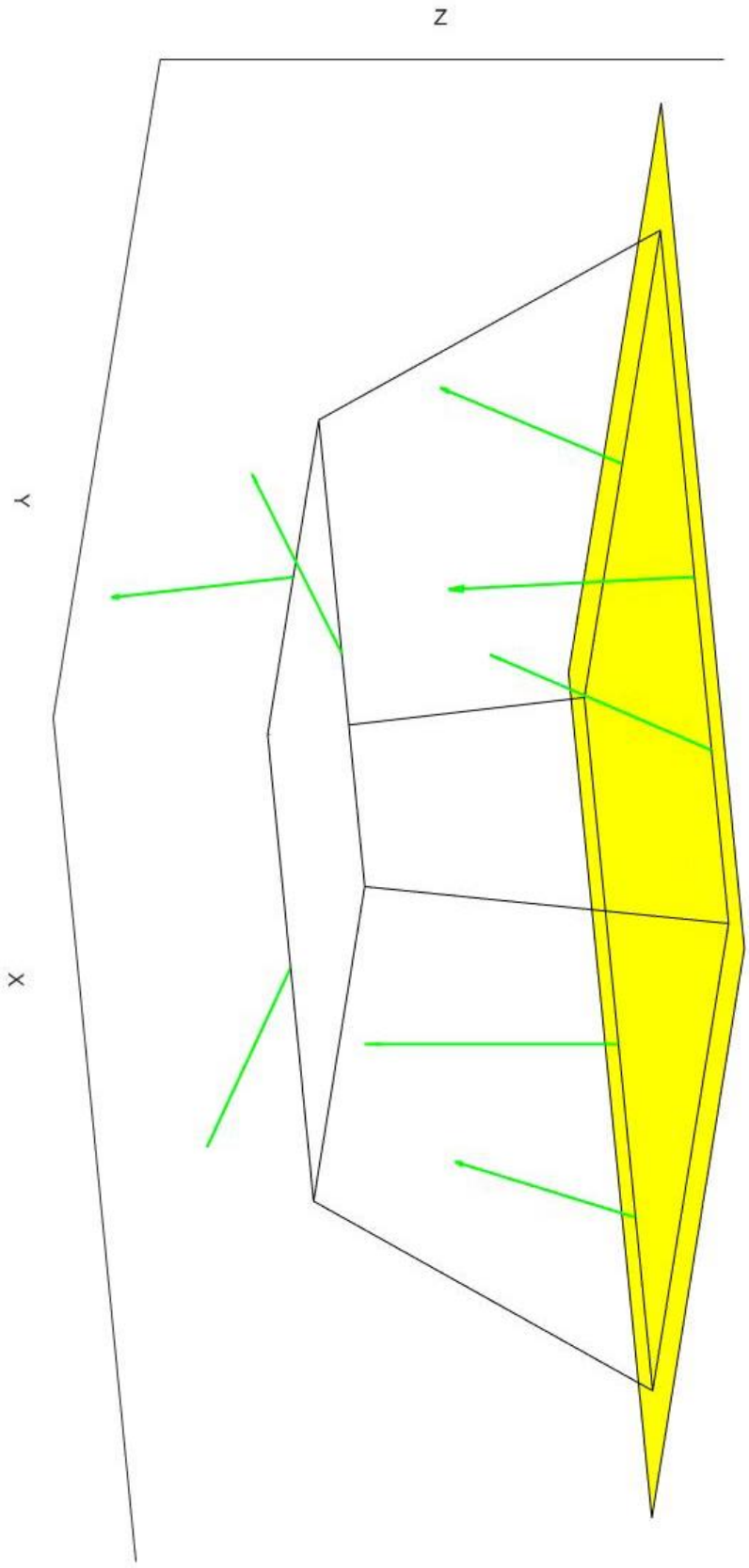

FIGURE 78 AOCS 8 THRUSTERS

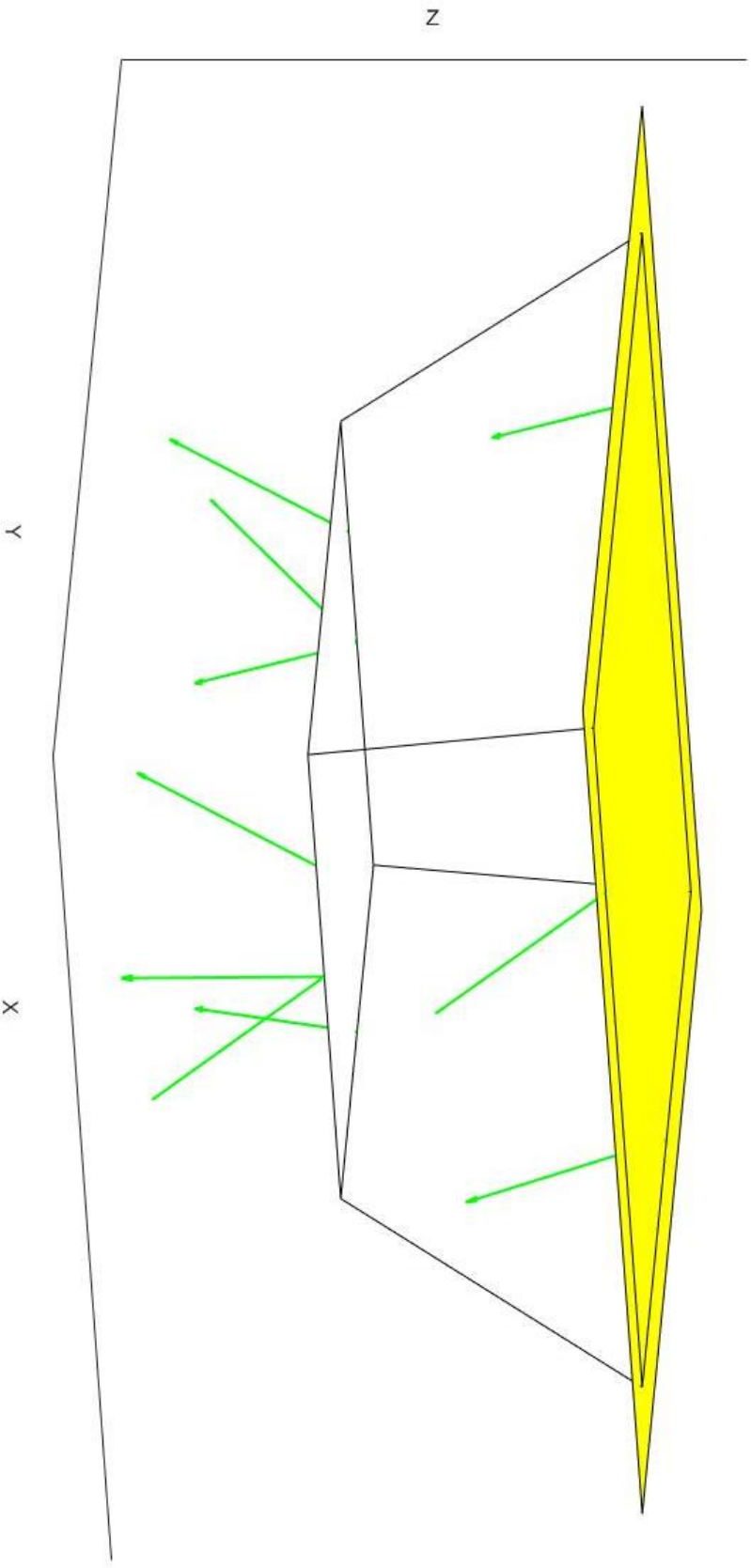

FIGURE 79 AOCS 10 THRUSTERS

As briefly mentioned in the previous pages, the torques in input derives from the request of spacecraft's controllability after the separation from the launcher, in the phase called detumbling. In order to guarantee the spacecraft's stability even in the worst case, it was assumed the worst scenario in which over all the axes is requested the maximum torque. This input is very tricky to be managed by the system, especially because the thrust is discretized and therefore can't be a fine regulation and a mutual adjustment by the actuators.

In fact, in all the cases analysed there is always a residual error on torques and can't be guaranteed the perfect controllability. Since the input concerns the worst case, it was decided to try the configuration found to guarantee the spacecraft's stability when the maximum torque is requested over one axis or two axes simultaneously. In TABLE 5. 8 are reported the percentage of the total error over the request in all these cases and FIGURE 80 shows the results in a more intuitive way.

	X	Y	Z	XY	XZ	YZ	XYZ
6 Actuators	1.8%	2.92%	1.5%	12.34%	13.48%	13.05%	20.26%
8 Actuators	1.5%	0.74%	1.6%	7.18%	8.29%	9.91%	10.04%
10 Actuators	0.2%	0.45%	0.8%	7.58%	5.47%	5.55%	7.04%

TABLE 5. 8 AOCS PERCENTAGE ERROR

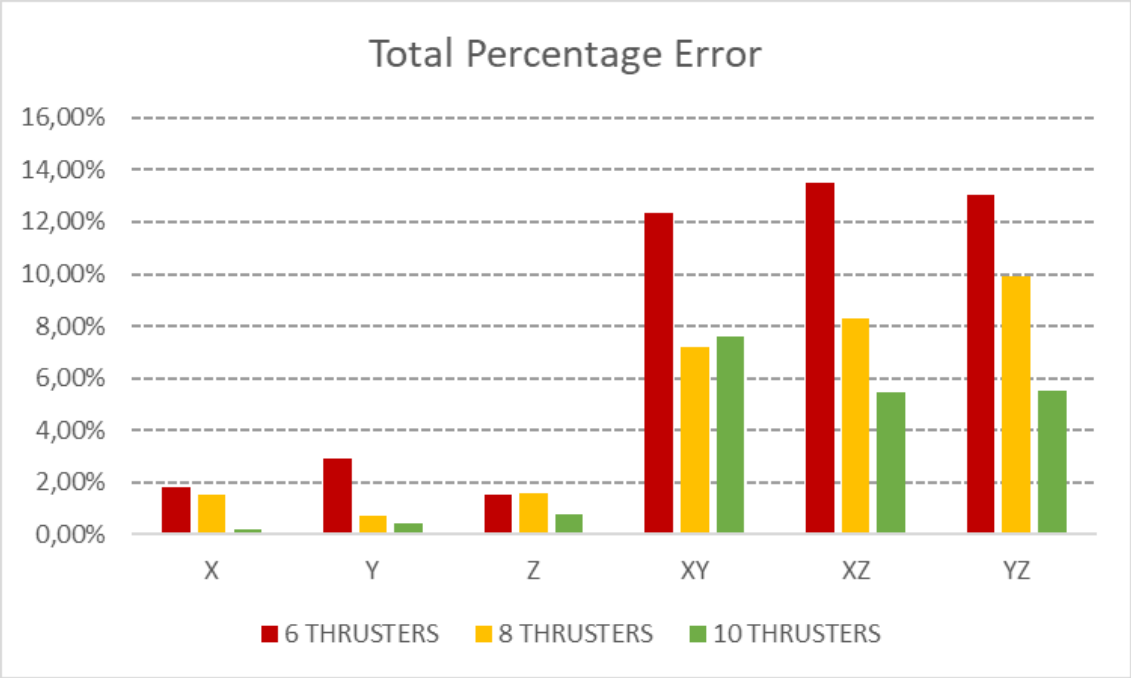

FIGURE 80 AOCS PERCENTAGE ERROR

As can be seen, increasing the number of actuators the total error decreases although this is never zero. Even considering the constraints regarding just one axis, i.e. the first three column in TABLE 5. 8, the total controllability isn't guaranteed although the percentage error is near to zero.

The total error may be reduced considering different maximum thrust level. The results found so far are obtained with a maximum thrust level of 40mN. This value is reduced or increased by 10mN in further analyses. It was decided to conduct the analyses in the case in which the total error is higher, i.e. considering six thrusters, because the room for improvement is higher. The procedure followed is the same used so far, i.e. initially found the orientations through the

Layout model with the thrust as a continue variable and after set the thrust as a binary variable and reducing the domain in a neighbourhood of the solution derived from the *Layout* model.

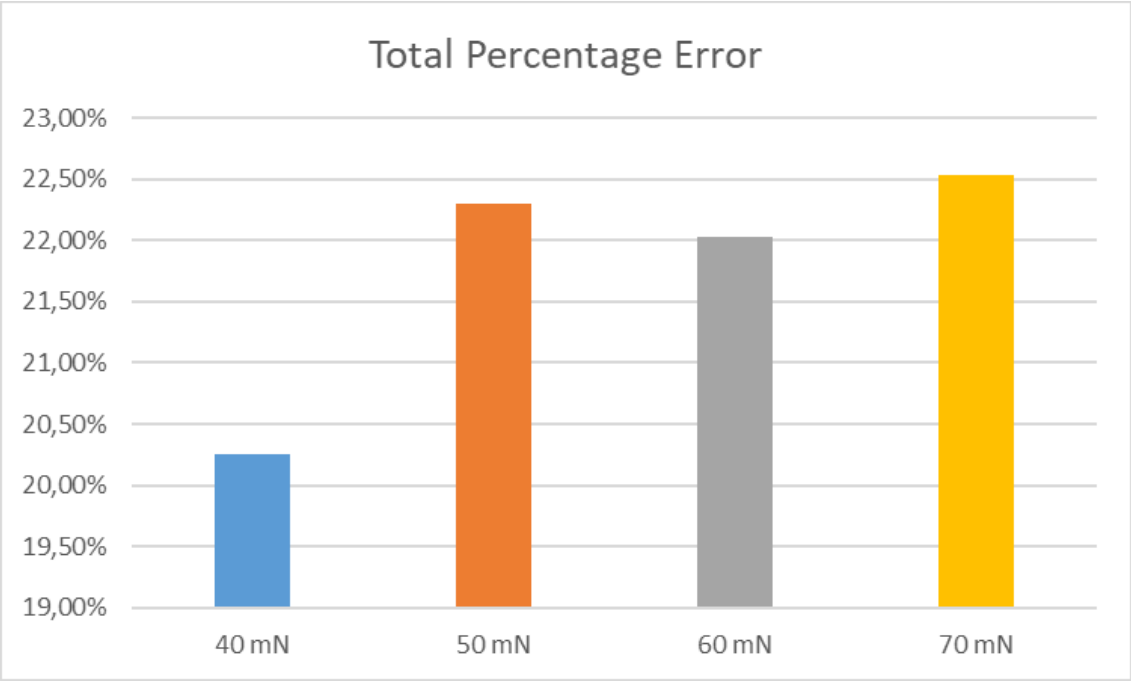

FIGURE 81 TOTAL ERROR FOR DIFFERENT THRUSTS

As **FIGURE 81** clearly shows, this isn't the right way to proceed. The biggest difficult of this problem is that the thrust can assume just two values, i.e. zero or a fixed value. Even considering the torques in input as a constant value and not as a complicated function, find some actuators' orientations that can accommodate the control law in each instant is very difficult. No solutions with a thrust lower than 40mN have been found, either with a continue thrust or a discontinuous value. This behaviour could derive from the value of the torques in input, greater than 30mNm, and therefore difficult to be counteracted using actuators with a low exercisable thrust.

Conclusions and future developments

This thesis focuses on the thruster layout problem and, more specifically, on the positioning and orientation of them on-board the spacecraft. The algorithms and methods developed are applied in a real-world mission project, i.e. the LISA mission, concerning the DFACS (Drag Free Attitude Control System) and AOCS (Attitude and Orbit Control System) systems. The problem dealt with are of a very large dimension, in terms of number of variables and constraints. The problem subdivision into sub-problems demonstrated to be an efficient approach. Specifically, the first sub-problem is aimed at finding a suitable set of values for the orientation variables, in order to make the original problem easier to solve, while the second consists in solving the original model by fixing the orientations previously found.

Regarding the DFACS system, the analyses were conducted from an initial configuration provided by ESA and used as reference during the whole work.
The main scope of this thesis was that of reducing, in the phase of scientific measurements, the total fuel consumption and, at the same time, to reduce the number of actuators mounted on-board w.r.t. the initial configuration. The configuration provided by ESA and used as a reference was declared already close to optimal and the fuel consumption percentage improvement w.r.t. this reached 1.97%. This value was obtained considering limitations on the allowed actuator orientations, derived from the project requirements. By relaxing these limitations led to better results concerning the fuel consumption. This insight could be considered for alternative design solutions.
The configurations found were optimised for the scientific measurements phase and they were further tested in others mission scenarios in order to prove their feasibility in an extended context. The relevant tests showed that each configuration was unfeasible for all the non-nominal scenarios considered. To guarantee the feasibility a future study could be conducted considering the simultaneous action of the DFACS and AOCS actuators.

Once the analyses for the DFACS system had been completed, the focus moved on the AOCS system. The models and algorithms used for the DFACS analyses were transferred to the AOCS, applying some modifications. The actuators of the AOCS system were of an on/off type. On the contrary, those considered for the DFACS could exert a continuous thrust. This entailed a number of changed in the model utilized.
Further investigation on this aspect could represent the object of a future research. A first step in this direction could significantly extend the space of solutions by considering the applications-points as variables of the problem and not chosen a priori from the user, clearly demanding more computational load but ensuring a more thorough process.
.

References

1. **Allasio A., Anselmi A., Cesare S., Dionisio S., Mottini S., Parisch M., Massotti L., Silvestrin P.** *From GOCE to the Next Generation Gravity Mission-5th International GOCE User Workshop*. [Online] 28 November 2014.
2. **Anselmi A., Cesare S., Dionisio S., Fasano G., Massotti L.,** *Control propellant minimization for the Next Generation Gravity Mission,* (2018).
3. **Clark S., Mignone C., O'Flaherty K.,** *First steps to observing gravitational waves from space.* [Online] 27 November 2015. *https://sci.esa.int/web/lisa-pathfinder/-/56869-lisa-pathfinder-media-kit*
4. **Corp., IBM,** *IBM ILOG CPLEX Optimization Studio.* IBM Corp., 2015.
5. **Corp., IBM,** *IBM ILOG CPLEX Optimization Studio CPLEX Parameters Reference* https://www.ibm.com/support/knowledgecenter/SSSA5P_12.8.0/ilog.odms.cplex.help/CPLEX/Parameters/topics/introListAlpha.html
6. **Danzmann K.,** *A proposal in response to the ESA call for L3 mission concept.* [Online] 20 January 2017.
7. **Fasano, G.** *Dynamic system control dispatch: a global optimization approach. [eds.] G.Pintér, J.D. Fasano. Modeling and Optimization in Space Engineering – State of the Art and New Challenges*. New York: Springer (2019).
8. **Hiller F., Lieberman G.,** *Introduction to operation resource-7th Edition,* MacGraw-Hill (2001).
9. **Lanzillotti A.**, *Optimization Methods for Thruster Layout*.
10. *Lisa phase 0 cdf study – Internal final presentation.* [Online] 6 May 2017. https://sci.esa.int/web/lisa/-/59336-lisa-internal-phase-0-cdf-study-final-presentation
11. Mathworks. Mathworks Optimization Toolbox Documentation. [Online] 2018. https://www.mathworks.com/help/optim/.
12. **Williams P.,** *Model Building in Mathematical Programming-5th Editon.* Wiley (2013).

RINGRAZIAMENTI

Ci sono alcune persone che vorrei ringraziare, non solo per questo lavoro di tesi ma anche e soprattutto per la vicinanza mostratami in questo percorso universitario. Vorrei ringraziare il Dott. Giorgio Fasano per aver reso meno pesante la mia prima esperienza in un ambito aziendale, dal quale ero un po' intimorito, e da cui ho avuto la possibilità di apprendere molto; il Dott. Stefano Cesare, in qualità di rappresentante di Thales Alenia Space, per avermi concesso l'opportunità di dare il mio contributo, seppur piccolo, ad un progetto così importante; la Prof.ssa Manuela Battipede per la disponibilità mostrata in questi mesi ad ogni dubbio avanzato.
Vorrei ringraziare anche Federica, prima solo collega ed adesso anche amica, per la pazienza mostrata nello stare dietro ad ogni mio cambio di idea e per la leggerezza data ad ogni giorno in azienda.
Ringrazio tutti i miei amici 'di su' per avermi dato la fortuna di chiamare casa due posti diversi nel mondo; in particolare Simone, per essere stato un insostituibile compagno di avventura durante questi 5 anni universitari, condividendo sia risate che impegni accademici senza farli minimamente pesare.
Ad Antonio, con cui condivido un'amicizia di lunghissima data e che nonostante la distanza è rimasto come un fratello per me; a Paola, per la gioia di vivere che riesce a trasmettermi ogni qualvolta ci vediamo; a Chiara, per essere sempre stata una spalla su cui piangere quando qualcosa va male ma anche la prima con cui gioire per un lieto evento.
Ed infine vorrei ringraziare e dedicare questo lavoro alla mia famiglia, senza cui non sarei arrivato da nessuna parte e che mi ha reso l'uomo che sono; in particolare vorrei ringraziare mio padre, per la serietà e la professionalità che mi trasmesso ma anche e soprattutto per i valori di onestà ed integrità che mi ha insegnato fin da quando ero piccolo; a mia madre, per tutto l'amore che mi ha sempre dimostrato e per avermi sempre sostenuto in qualsiasi scelta io abbia fatto; a mia sorella, perché, anche se indirettamente ed inconsciamente, mi ha sempre trasmesso la forza di credere in sé stessi e non lasciarsi abbattere da niente e nessuno.

Grazie a tutti