POLITECNICO DI TORINO

Dipartimento di INGEGNERIA MECCANICA E AEROSPAZIALE

Corso di laurea magistrale

in Ingegneria Aerospaziale e Astronautica

Tesi di Laurea Magistrale

Advanced optimization for the thruster layout problem in a challenging space mission

Referente Accademico

Prof.ssa Manuela Battipede

Candidato

Federica Trudu

Referente Aziendale

Dott. Giorgio Fasano (Thales Alenia Space)

Anno Accademico 2018-2019

A mamma, papà

e Giorgia

TABLE OF CONTENTS

INDEX OF FIGURES

INDEX OF TABLES

RIEPILOGO

La presente tesi è stata svolta presso Thales Alenia Space, sede di Torino, in supporto alla missione LISA (Laser Interferometer Space Antenna). Questa missione, il cui lancio è previsto nel 2034, ha l'obiettivo di rilevare onde gravitazionali in un intervallo di frequenze più ampio rispetto a quello rilevabile con osservatori a terra. LISA prevede la presenza di masse di prova all'interno di tre satelliti identici, posti in formazione triangolare a distanza di 2,5 km l'uno dall'altro, che seguono la Terra nella sua orbita attorno al Sole. Interferometri al laser misurano eventuali variazioni di distanza tra le masse di prova provocate dalle onde gravitazionali. Per questo motivo il controllo d'assetto dei satelliti è di estrema importanza.

Lo scopo della presente tesi è trovare la configurazione dei propulsori preposti al controllo d'assetto che minimizzi il consumo di propellente. Ciò comporta una riduzione della massa di propellente a bordo o, a parità di suddetta massa, un'estensione della durata operativa della missione. A ogni step di controllo, la spinta dei propulsori deve compensare forze e coppie richieste dal sistema di controllo in quella determinata fase di missione. L'ingegnere sistemista deve decidere dove posizionare e come orientare i propulsori in modo da ottenere il minimo consumo di propellente. Per semplificare il problema, le posizioni dei propulsori sono considerate fissate a priori. Nonostante questo, il problema matematico risulta molto complesso e trovare soluzioni anche soltanto sub-ottime non è un compito semplice, quindi esso viene suddiviso in due sotto-problemi. Il primo prevede che il dominio degli orientamenti ammissibili per i propulsori venga opportunamente discretizzato. Inoltre, viene considerato soltanto un sottoinsieme di step di controllo, scelto in modo da essere rappresentativo dell'intero intervallo temporale che costituisce l'oggetto di analisi. Gli orientamenti ottenuti come soluzione del primo sotto-problema, che è fortemente non lineare, vengono fissati nel secondo, che diventa in tal modo lineare. Questa notevole semplificazione permette di ottenere una soluzione in termini di consumo di propellente considerando l'intero intervallo temporale. In seguito, il problema viene riproposto iterativamente considerando gli orientamenti possibili in una zona ristretta attorno alla soluzione precedente. La nuova discretizzazione del dominio viene fatta in modo che, nella peggiore delle ipotesi, possa essere ritrovata la soluzione precedente. Le soluzioni ottenute vengono successivamente processate con metodi di ottimizzazione non lineare locale.

Gli strumenti utilizzati per trovare una soluzione al problema dell'ottimizzazione della configurazione dei propulsori sono IBM ILOG CPLEX Optimization Studio e Matlab. CPLEX è un ottimizzatore che si basa sull'algoritmo branch and bound, ma prevede l'implementazione di variazioni dell'algoritmo diverse e molto complesse. Il pacchetto di Matlab Optimization Toolbox permette di utilizzare diversi algoritmi: interior-point, sequential quadratic programming e active-set. Da un breve test l'algoritmo migliore è risultato essere il sequential quadratic programming, per cui è stato utilizzato solamente questo per svolgere analisi con Matlab in questa tesi.

La prima parte delle analisi svolte riguarda i propulsori del DFACS (Drag Free Attitude Control System) durante la fase scientifica della missione. Sono stati considerati due scenari, uno che consiste in tre clusters di tre thrusters ciascuno, di cui uno rivolto verso l'alto per ogni cluster. Questo scenario coincide con quello proposto come soluzione di riferimento dall'ESA. Il secondo scenario considerato consiste in tre clusters di due thrusters ciascuno, tutti rivolti verso il basso.

In seguito, si è cercato di qualificare le soluzioni ricercando un lower bound per il consumo, ovvero un suo minorante. Per questa analisi si sono effettuati opportuni rilassamenti lineari dei modelli matematici.

Successivamente sono state considerate forze e coppie relative a condizioni diverse da quelle presenti nella fase scientifica della missione. Si è valutato se le migliori soluzioni ottenute in termini di consumo per i due scenari fossero in grado di operare anche in suddette condizioni.

L'ultima fase delle analisi riguarda i propulsori dell'AOCS (Attitude and Orbit Control System). Questi propulsori, a differenza di quelli del DFACS, non possono esercitare una spinta continua, ma possono essere attivi ed esercitare un valore costante di spinta oppure inattivi e non esercitare alcuna spinta. Questi thruster devono essere in grado di compensare tutte le combinazioni di un certo valore di coppia massima (positiva e negativa) simultaneamente sui tre assi. Sono stati considerati uno scenario con otto thrusters e due scenari con sei thrusters, uno con tutti i thrusters sulla superficie inferiore del satellite ed uno con tre thrusters sulla superficie inferiore e tre su quella superiore. In un primo momento si è considerata la spinta continua, in seguito si è ristretto il dominio attorno alla soluzione ottenuta e si è considerata la spinta come variabile discontinua, nel senso sopra indicato. Per lo scenario con sei thrusters tutti posti sulla stessa superficie del satellite, si è provato a ripetere le analisi per valori maggiori di spinta massima esercitabile dai thruster.

I risultati ottenuti dall'analisi sperimentale svolta hanno apportato un significativo contributo alle fasi sia presenti che prossime del programma LISA.

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 [1], funded by the European Space Agency (ESA), currently under study.

After the successful LISA Pathfinder mission (ESA, 2015-2017) [2], 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 [3]. LISA, whose launch is expected in 2034, will consist of three identical spacecraft separated by 2.5 million km in a triangular formation, which will follow Earth in its orbit around the Sun.

Among the great number of difficult 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 the overall force and torque that have to act on the spacecraft (with respect to an assigned system-based reference frame).

Different positions and orientations of the actuators can result in a significantly diverse overall performance in terms of fuel consumption. Moreover, the number of thrusters adopted gives rise to a further non-negligible concern: although a rather large number of thrusters might advantageously contribute to a reduction in 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 brings about an additional non-trivial issue and, consequently, an adequate trade-off between reducing fuel consumption and limiting 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 [4]. As is understood, for this kind of mission a very strict attitude control strategy has to be envisaged due to the strong atmospheric drag effect. In order to tackle effectively the relevant thruster layout optimization problem, an ad hoc optimization methodology has been introduced [5].

A dedicated controller determines, at a predefined frequency, the overall control action, aimed at achieving the desired system attitude step by step. 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 [6]. From a practical point of view, this intrinsic difficulty 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 [7], in particular linear, nonlinear and mixed-integer-linear programming (LP, NLP, MILP) [7][8][9].

The basic idea consists in partitioning the thruster layout task into much easier sub-problems and in solving these by following an overall iterative (or recursive, if necessary) process, until a valid (global) solution to the original problem is found. The approach proposed takes advantage of such very specific structure.

Considering, for the sake of simplicity, a reduced scenario where the thrusters have been assigned their locations a priori, only two discrete sets of variables are involved, i.e. those representing the relevant orientation and those associated with the forces exerted by the actuators at each control step. Moreover, most of the non-linear constraints of the problem (with the above mentioned assumption), i.e. those corresponding to the equations determined by the force and torque requests, are bilinear. The remaining constraints (of a comparatively very limited number) ruling the orientations of the thrusters are instead quadratic. This entails that if the orientation variables are fixed, then the resulting problem becomes linear (since all bilinear equations are reduced to linear), and all quadratic constraints can be dropped (being, as in this case, redundant). All that being stated, two separate sub-problems can be considered: the first aimed at finding a suitable set of values for the orientation variables, in order to make linear the original problem (i.e. easy to solve); the second consisting in this reduction.

More precisely, the first sub-problem (that is per se quadratic and non-convex) mainly addresses the orientation of the actuators. To this purpose, limited subsets of control steps assumed to be representative of the whole time span are taken into account [10]. On the other hand, the second sub-problem (that is linear) consists in optimizing the overall original problem, including the entire set of instants, once the orientation variables have been assigned the values obtained by solving the orientation sub-problem. The result thus obtained (if necessary by introducing a certain tolerance level with respect to the equations ruling the force and torque requests) is in general a sub-optimal solution of the original problem. If this solution is not deemed satisfactory, then a further set of values for all the orientation variables is generated and the search process continues until a satisfactory solution is found. Refinements of the current or final solutions obtained may be carried out by applying (local) NLP or sequential linear programming (SLP) [9]. The overall search process applies, albeit heuristically, a global optimization (GO) logic [11]. Specific MILP models (to be utilized at different levels of approximation) have been conceived to solve (globally) the thruster orientation sub-problem.

As a first significant step, this thesis focused on the tailoring of the methodology outlined above, with reference to the NGGM context, to the specific and not any easier framework concerning the LISA mission. In particular, an ad hoc adaptation of the (MILP) model for the orientation sub-problem has been performed. 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. A dedicated (local) NLP model has further been developed to enhance the MILP solutions that are biased by the approximations adopted (being based on suitable discretization, introduced to eliminate the problem intrinsic non-linearities).

To carry out the aforementioned modeling and algorithm-development activities, IBM-CPLEX [12] and Matlab [13] 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. DFACS (Drag Free Attitude Control System) and AOCS (Attitude and Orbit Control System). From the DFACS thruster layout perspective, the reference layout given by ESA provides for the presence of three clusters of three thrusters each to support the attitude control during the entire scientific phase. In this thesis, this layout and a layout with three clusters of two thruster each have been considered. The solutions found for the DFACS have been qualified by a lower-bound analysis. From the AOCS thruster layout perspective, the reference scenario provides for the presence of six thrusters to support the attitude and orbit control starting at the separation stage from the launch vehicle. In this thesis this layout, a different layout with six thruster and one with eight thrusters have been considered.

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 illustrates the thruster layout optimization problem from a mathematical point of view. Chapter 2 provides overall insight on the LISA mission and the specific features relevant to the thruster layout optimization problem. Chapter 3 introduces the algorithms used to solve the problem. Chapter 4 reports the analysis carried out in depth and Chapter 5 illustrates the analysis results.

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. MATHEMATICAL PROBLEM

The present chapter illustrates the thruster layout optimization problem from a mathematical point of view [4][5]. A number of actuators (thrusters) are available to exert the overall force and torque required by the control during the science mode. This force and torque profile is the input for the problem. The purpose is to find the thruster layout, in terms of location and orientation, which minimizes the fuel consumption. The mathematical approach proposed in [5] for the control dispatch in a general dynamic system is adopted and properly tailored to the specific case under study.

1.1 PROBLEM STATEMENT

We consider a general rigid body system S over a given timeframe [0,T]. An appropriate S-based main orthogonal reference frame is defined. Due to the discrete nature of the control action, the interval [0,T] may be partitioned into a set of time steps, of duration Δ each. The following notations are introduced:

$I = \{0,1,\dots,N_I\}$ is the set of time instants;

$A = \{1,\dots,N_A\}$ is the set of actuators;

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

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

$\boldsymbol{v}_r = (v_{rx}, v_{ry}, v_{rz})$ are the unit vectors representing the orientation of each actuator r;

$\boldsymbol{f}_{ri} = (f_{rxi}, f_{ryi}, f_{rzi})$ is the force exerted by actuator r at instant i;

$u_{1i},\dots,u_{N_A i}$ are the thrusts associated with each actuator respectively, at each instant considered;

$\boldsymbol{p}_r = (p_{rx}, p_{ry}, p_{rz})$ is the position vector of the actuators (the force application points);

$\underline{U_r}, \overline{U_r}$ are, for each actuator r, the lower and upper bounds imposed on u;

$D_{vr} \subset \boldsymbol{R}^3$ is a compact domain delimited by specific conditions on the actuator orientations;

$D_{pr} \subset \boldsymbol{R}^3$ is a compact domain delimited by specific conditions on the actuator positions.

The given control law is expressed in terms of overall force and torque demand by the following equations:

$$\forall i \in I \quad \begin{pmatrix} v \\ p \times v \end{pmatrix} \begin{pmatrix} u_{1i} \\ \dots \\ u_{ri} \\ \dots \\ u_{N_A i} \end{pmatrix} = \begin{pmatrix} F_i \\ T_i \end{pmatrix} \tag{1}$$

Equations (1) can also be expressed in a more explicit formulation as follows:

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

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

The following normalization condition has to be set for the direction cosines:

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

Each actuator has given limitations on the minimum and maximum force that it can exert, therefore the lower and upper bounds are set as a basic condition:

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

It is understood that the lower bound $\underline{U_r}$ is always non-negative.

The following conditions express the admissible positions and orientations for each actuator:

$$\forall r \in A \quad \boldsymbol{v}_r \in D_{vr}, \boldsymbol{p}_r \in D_{pr} \tag{4}$$

The actuator positions have been assumed to be constant and are given as input: D_{pr} is reduced to a single point for each actuator, reducing conditions (1) to a set of bilinear equations. The domain D_{vr} takes into account some constraints and it will be specified later in this thesis.

The optimization problem in question features the following objective function:

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

In the current study the objective function has been assumed to be linear. Expression (5) can thus be replaced by the following:

$$min \sum_{\substack{r \in A \\ i \in I}} K_r u_{ri} \tag{6}$$

where the constants K_r represent the fuel consumption per force unit associated with each actuator (supposed to be time independent).

All that being stated, the resulting problem could be infeasible. This means there is no thruster accommodation that meets the control request at any instant for the whole time span. To avoid this inconvenience, a possible relaxation of the problem could be taken into account. It consists of adding error variables, defined within given tolerance ranges and changing the objective function by introducing the total error as the term to be minimized. Equations (1) may be replaced by the following:

$$\forall i \in I \quad \begin{pmatrix} v \\ p \times v \end{pmatrix} \begin{pmatrix} u_{1i} \\ \cdots \\ u_{ri} \\ \cdots \\ u_{N_A i} \end{pmatrix} = \begin{pmatrix} F_i + \varepsilon_{Fi} \\ T_i + \varepsilon_{Ti} \end{pmatrix} \tag{7}$$

$$\forall i \in I \quad -E_F \leq \varepsilon_{Fi} \leq E_F, -E_F \leq \varepsilon_{Fi} \leq E_F \tag{8}$$

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

1.2 BREAKDOWN INTO TWO SUB-PROBLEMS

The optimization problem under discussion belongs to the NP-hard class of problems [6] and even finding sub-optimal solutions to this class of problems can be extremely challenging. In order to simplify the search for solutions, the problem is partitioned into two sub-problems. This involves the implementation of two dedicated mathematical models. The first one focuses on the thruster layout, in particular on their orientation, since the thruster position is assumed to be set a priori. This is referred to as the discretized model. It takes into account a limited sub-set of instants, supposed to be representative of the whole mission. The second sub-problem, denoted as the continuous model, focuses on the total fuel consumption minimization, considering the whole operational scenario. This model intends to verify the feasibility of the solution found by the discretized model, taking into account the entire set of instants.

Both models include equations (1), bounds (3) and objective function (6), while equations (2) are considered only in the discretized model. In the first model, all the variables of the general problem are treated as such; while in the second one, the variables relative to the thruster orientation are fixed on the basis of the results obtained by the discretized model. As a consequence, the continuous model becomes linear. That is why the continuous model can consider a large-scale instance, contemplating the full set of instants.

The discretized model is based on the discretization of the variables corresponding to the thruster orientations. This way, the quadratic equations (1) become linear and the normalization conditions (2) are dropped. The discretization, however, involves the introduction of 0-1 variables that make the original nonlinear model become a mixed-integer-linear-programming (MIP) one. It also belongs to the NP-hard class of problems.

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

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

subject to

$$\forall i \in \bar{I} \quad \begin{pmatrix} v \\ p \times v \end{pmatrix} \begin{pmatrix} u_{1i} \\ \dots \\ u_{ri} \\ \dots \\ u_{N_A i} \end{pmatrix} = \begin{pmatrix} F_i \\ T_i \end{pmatrix}$$

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

$$\forall r \in A, \forall i \in \bar{I} \quad u_{ri} \in \left[\underline{U_r}, \overline{U_r}\right], \boldsymbol{v}_r \in D_{vr}$$

This sub-problem is derived from the original one by replacing the set of instants I by a sub-set $\bar{I} \subset I$ as mentioned before. For each actuator the set of all admissible orientations are associated with a unit semi-sphere. This is described by unit vectors centered in the thruster position and directed externally, with respect to the corresponding satellite surface. A local reference frame is defined for each semi-sphere with the axis (x,y,z) parallel to the corresponding (X,Y,Z) of the global reference frame. Each unit vector can be identified by two spherical coordinates α and β. The angle α represents the polar coordinate, while the angle β represents the azimuthal coordinate. The condition $\alpha = 0$ corresponds to the y axis and $\beta = 0$ corresponds to the x axis. Both angles α and β are partitioned into sub-sets, dividing the corresponding intervals $\alpha \in [0,2\pi]$ and $\beta \in \left[0, \frac{\pi}{2}\right]$ by a pre-selected number. In this way the variables v_{rx}, v_{ry} and v_{rz} corresponding to all possible orientations of each actuator are no longer continuous, but they can only take a finite number of values.

The continuous linear model can be expressed as follows:

$$\min \sum_{\substack{r \in A \\ i \in I}} K_r u_{ri} \tag{10}$$

subject to

$$\forall i \in I \quad \begin{pmatrix} v^* \\ p \times v^* \end{pmatrix} \begin{pmatrix} u_{1i} \\ \dots \\ u_{ri} \\ \dots \\ u_{N_A i} \end{pmatrix} = \begin{pmatrix} F_i \\ T_i \end{pmatrix}$$

$$\forall r \in A, \forall i \in I \quad u_{ri} \in \left[\underline{U_r}, \overline{U_r}\right]$$

In this model the whole set of instants is considered and the terms v^* are given by the values obtained as solutions of the sub- problem by means of the discretized model.

2. A POSSIBLE APPLICATION: LISA MISSION

The mathematical problem introduced in the previous chapter can be applied to a wide range of real world scenarios, including applications in automation and robotics. The methodology presented has been already adopted in the context of NGGM (Next Generation Gravity Mission) studies, as described in [4]. In this thesis, the thruster layout optimization problem is applied to the LISA (Laser Interferometer Space Antenna) mission. Therefore, this chapter will focus on such mission.

2.1 LISA OVERVIEW

LISA is a space-based gravitational wave observatory. Studying gravitational waves gives the opportunity to discover the aspects of the universe that are invisible by other means, such as black holes, the Big Bang effects, and other, as yet unknown, objects. LISA will increase our knowledge about the beginning, evolution and structure of the universe [14].

Compared to the Earth-bound gravitational wave observatories like LIGO and VIRGO, LISA addresses the much richer frequency range between 0.1 mHz and 1 Hz, which is inaccessible on Earth due to arm-length limitations and terrestrial gravity gradient noise. The gravitational wave sources that LISA would discover include ultra-compact binaries in our Galaxy, supermassive black hole mergers, and extreme mass ratio in spirals [15].

The LISA mission will be based on laser interferometry between free flying test masses inside drag-free spacecraft.

Gravitational waves change the light travel time or the optical path-length between free falling test masses. The test masses will be undisturbed by forces other than gravitation. These test masses and the surrounding Gravitational Reference Sensor (GRS) hardware has been tested successfully on LISA Pathfinder [2]. They will be located inside three identical spacecrafts in a triangular formation separated by 2.5 million km. Laser interferometers will measure the pm to nm path-length variations caused by gravitational waves. The interferometers neither require nor allow for any pointing towards specific sources. As a result, they are all-sky monitors of gravitational waves.

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. The centre of the formation is in the ecliptic plane at 1 Astronomical Unit (AU) from the Sun and 20° behind the Earth. The plane of the triangle is inclined by 60° with respect to the ecliptic. The orbital configuration is shown in Figure 1.

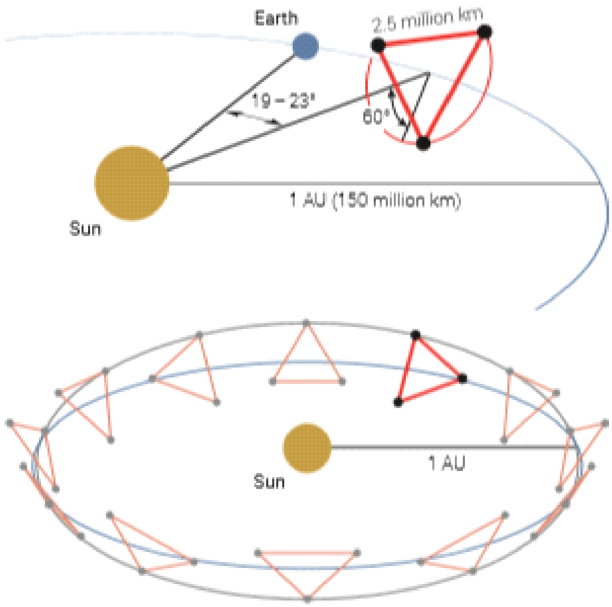

Figure 1: LISA orbit.

These particular heliocentric orbits for the three spacecraft were chosen so that the triangular formation is maintained throughout the year, with the triangle apparently rotating around the centre of the formation. The orbit is optimized to minimize the key variable parameters of the so-called "arm breathing" and range change between the spacecraft, as both of these drive the complexity of the payload design. At the same time it ensures that the distance between LISA and Earth is sufficiently small for communication purposes [1]. A launch might be feasible around 2030. A mission lifetime of 4 years in science mode extendable to 10 years for LISA is proposed.

2.2 LISA CONSTRAINTS

In this thesis, the science phase of LISA will represent the main concern. The forces and torques requested are therefore mainly considered with respect to this phase. Three clusters of three thrusters each are considered in order to satisfy the force and torque demand. All nine thrusters are located on the lower surface of the spacecraft. The position of each cluster is given and all the thrusters of the same cluster have the same position. At least one thruster has to be oriented upward with a 30° inclination from the horizontal plane. This is to ensure that disturbing forces and torques in any direction can be balanced.

The configuration of the spacecraft imposes some restrictions on the mathematical problem. First of all, a rotation of the reference frame is necessary in order to use the convention on polar and azimuthal angles introduced in chapter 1. Then the thruster orientation domain D_{vr} must be restricted taking into account that the thrusters have to remain outside of the spacecraft sides and the thruster plume must not affect the solar panel.

Details on the S/C configuration are not given in this thesis due to confidentiality restrictions.

3. SOLUTION METHODS

Two different tools have been used to find a solution for the thruster layout optimization problem. The first one is the optimization software package IBM ILOG CPLEX Optimization Studio, the second one is MATLAB. The search for the solution with CPLEX is based on branch and bound algorithm. The MATLAB Optimization Toolbox provides functions to solve constrained optimization problems. In particular, the function used is *fmincon*. Different algorithms can be utilized: interior-point, sequential quadratic programming and active-set.

3.1 BRANCH & BOUND

The branch and bound technique [8] comes from the idea of using some kind of enumeration procedure to find an optimal solution for IP problems. It is essential that this enumeration procedure is structured so that only a small part of the feasible solutions needs to be examined. Hereafter the specific application to MILP (Mixed Integer Linear Programming) problems of the branch and bound technique will be discussed.

The branch and bound technique is based on the concept of dividing and conquering. The original problem is assumed to be too difficult to be solved directly, therefore it is divided into smaller sub-problems which are simpler to solve. The dividing, better known as branching, is done by partitioning the whole set of feasible solutions into smaller subsets. The conquering, better known as fathoming, is done by evaluating how good the best solution in the subset can be and then dismissing the subset if it cannot contain an optimal solution for the original problem.

We consider an MIP problem, where some of the variables are restricted to integer values (in the specific case 0 or 1) and the remaining are continuous.

The first step is the branching. The way to partition the set of feasible solutions into subsets consists in fixing the value of one of the integer-restricted variables at 0 for one subset and at 1 for the other one. The variable used to do this branching at any iteration is called the branching variable. At the first iteration, the original problem is divided into two sub-problems. This branching can be represented by a tree, referred to as the solution tree, with branches from the “all node” (which corresponds to the original problem) to the nodes corresponding to the two sub-problems (Figure 2).

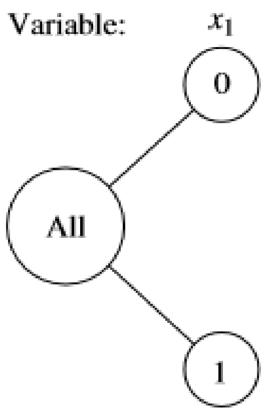

Figure 2: first branch of the solution tree.

This tree will generate as many branches as the iterations increase (Figure 3).

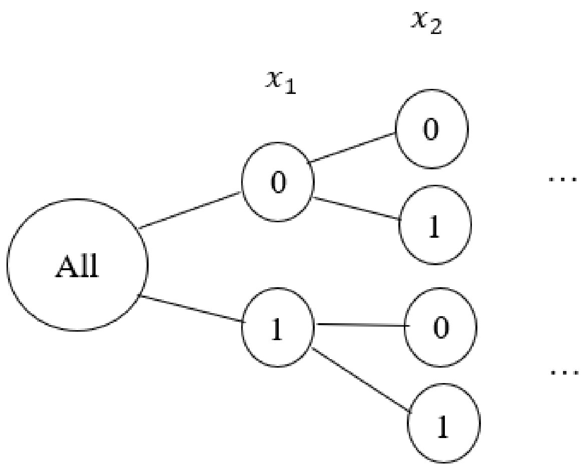

Figure 3: branches of the solution tree.

Branching entails the selection of the pending sub-problem to be further partitioned into smaller sub-problems. A common practice consists in selecting the sub-problem on the basis of the best bound rule, because the corresponding sub-problem would be the most promising one, as it tends to find quickly better incumbents and implies more fathoming.

The second step of the whole process is the bounding: for each sub-problem a bound on how good its best feasible solution can be is needed. To obtain this bound, a relaxation of the sub-problem is usually done. The most widely used relaxation is the LP relaxation, which consists in deleting the constraints requiring the variables to be integer. Solving with the simplex method the relaxation of the sub-problems provides the bounds for the sub-problems.

The third step is the fathoming. A sub-problem can be fathomed and then dismissed from further consideration in three ways. One way occurs when the optimal solution for the LP relaxation of the sub-problem has integer values for the integer restricted variables. It is sufficient to guarantee that the solution is feasible and optimal for the sub-problem, thus there is no reason to consider this sub-problem any further for branching. This solution must be stored as the first incumbent, which is the best feasible solution found so far for the whole problem. The second way provides for the dismissal of the sub-problem whose bound is greater than or equal to the value of the objective function of the incumbent, because such a sub-problem cannot have a feasible solution better than the incumbent. The third way of fathoming is straightforward: if the LP algorithm finds that the sub-problem has no feasible solutions, then it has to be fathomed.

When there are no remaining sub-problems, the current incumbent is optimal. If there is no incumbent, the conclusion is that the problem has no feasible solution.

3.1 INTERIOR POINT ALGORITHM

A brief overview of the interior point algorithm is given hereinafter. For details of the algorithm and the derivation, see [16]. The original problem is

$$\min f(x) \text{ subject to } h(x) = 0 \text{ and } g(x) \leq 0 \tag{11}$$

For each $\mu > 0$, the following approximate problem is considered:

$$\min f_\mu(x, s) = \min f(x) - \mu \sum_i \ln(s_i)$$
$$\text{subject to } h(x) = 0 \text{ and } g(x) + s = 0 \tag{12}$$

The added logarithmic term is named barrier function. As μ decreases to zero, the minimum of f_μ approaches the minimum of f. The approximate problem is easier to solve because there are only equations as constraints. At each iteration the algorithm attempts to use a direct step to solve the approximate problem, but if it cannot, it uses a conjugate gradient step.

The direct step, also called Newton step, involves trying to solve the approximate problem with a linear approximation. The following variables are used:

H is the Hessian of the Lagrangian of f_μ

J_g is the Jacobian of the constraint function g

J_h is the Jacobian of the constraint function h

$S = diag(s)$

λ is the Lagrange multiplier vector associated with constraints g

$\Lambda = diag(\lambda)$

y is the Lagrange multiplier vector associated with constraints f

e is the vector of ones with the same size as g

The direct step is defined by

$$\begin{bmatrix} H & 0 & J_h^T & J_g^T \\ 0 & S\Lambda & 0 & -S \\ J_h & 0 & I & 0 \\ J_g & -S & 0 & I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ -\Delta y \\ -\Delta \lambda \end{bmatrix} = - \begin{bmatrix} \nabla f - J_h^T y - J_g^T \lambda \\ S\lambda - \mu e \\ h \\ g + s \end{bmatrix} \tag{13}$$

The most expensive step from a computational point of view is the factorization of the matrix. This factorization determines if the projected Hessian is positive definite. If it's not, the algorithm uses the conjugate gradient step.

The conjugate gradient step consists of minimizing a quadratic approximation to the approximate problem subject to linearized constraints in a thrust region with radius R. The linearized constraints are the following:

$$g(x) + J_g \Delta x + \Delta s = 0, h(x) + J_h \Delta x \tag{14}$$

Then the algorithm takes a step to solve

$$\min \nabla f^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2_{xx} L \Delta x + \mu e^T S^{-1} \Delta s + \frac{1}{2} \Delta s^T S^{-1} \Lambda \Delta s \quad (15)$$

3.2 SEQUENTIAL QUADRATIC PROGRAMMING ALGORITHM

Sequential Quadratic Programming is a consolidated optimization method for the numerical solution of constrained nonlinear optimization problems. A preliminary introduction to this non-trivial subject is set out hereinafter, a more detailed description can be found in [9].

The original problem is expressed by Equation (11). Sequential Quadratic Programming is an iterative procedure which approximates the general nonlinear optimization problem, for any iterate x_k, by a quadratic optimization (Quadratic Programming, QP) sub-problem. Then, it solves the QP sub-problem and uses the solution to find a new iterate x_{k+1}. The overall procedure is aimed at generating a sequence (x_k) that converges to a local minimum x^* of the original problem, as $k \to \infty$.

The basic concept of Sequential Quadratic Programming is to express the original problem by adopting the following approximations (Taylor's second and first order expansions for objective function and constraints, respectively):

$$f(x) \approx f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2} (x - x_k)^T H_f(x)(x - x_k) \quad (16)$$

$$h(x) \approx h(x_k) + \nabla h(x_k)(x - x_k) \quad (17)$$

$$g(x) \approx g(x_k) + \nabla g(x_k)(x - x_k) \quad (18)$$

where $H_f(x)$ is the Hessian matrix of $f(x)$. This leads to the quadratic optimization sub-problem below:

$$\min_d \left\{ \nabla f(x_k)^T d + \frac{1}{2} d^T B_k d \right\} \quad (19)$$

subject to

$$h(x_k) + \nabla h(x_k)^T d = 0,$$

$$g(x_k) + \nabla g(x_k)^T d \leq 0,$$

where $d \in R^n$ and $B_k = H_f(x_k)$.

The QP sub-problem is solved at each step of the process by means of an NLP (Non-linear Programming) algorithm, among a number of possible choices (e.g. line-search, trust-region, Newton and quasi-Newton methods) [8].

Note that the algorithm requires a starting point whose choice can significantly affect the computational process as well as the final solution.

3.3 ACTIVE-SET ALGORITHM

Active-set algorithm is similar to sequential quadratic programming algorithm. The most important difference between active-set and SQP is about the linear algebra routines used to solve the quadratic programming sub-problem (19). The routines used by sequential quadratic programming algorithm are more efficient in both memory usage and speed than the active-set routines [17].

4. ANALYSIS WORKFLOW

In this chapter all the steps of the analysis for the LISA thruster layout optimization will be listed and described in detail. Most of the analysis concerns the optimization of the layout of the thrusters relevant to the DFACS (Drag Free Attitude Control System). At first, only the science phase is considered. A qualification of the solutions obtained in this phase has been achieved through a bound analysis. In order to value the handling of the spacecraft, other operative modes, with different requirements of forces and torque as compared to the science one, have been further considered. The last part of the analysis concerns the optimization of the layout of the thrusters relevant to the AOCS (Attitude and Orbit Control System).

4.1 SCENARIOS FOR DFACS

The first scenario considered for the DFACS thrusters provides for the presence of three clusters of three thrusters each. The thruster orientations can be properly selected to obtain the minimum fuel consumption, with the constraint that one thruster for each cluster has to be oriented upward with an inclination of 30° with respect to the z axis.

The second scenario considered provides for the presence of three clusters of two thrusters each. The thruster orientations are free to change in order to obtain the minimum fuel consumption, but all the thrusters are oriented downwards. This scenario stems from the fact that the forces and torques demand in science mode do not require thrusters oriented upward. The presence of an additional thruster oriented upward can be considered in order to balance an unexpected request of forces and torques, during the whole mission. Since this thruster does not need to work in science mode, it will not be considered in the analysis.

The reference scenario provides for the presence of three clusters of three thrusters each. The thruster orientation is fixed and the fuel consumption it is known. It will be used as a reference for all the subsequent analyses about DFACS.

4.2 SELECTION OF INSTANTS

The discretized model is adopted to find the thruster orientation, but a reduction of the model dimension is necessary to obtain a feasible solution with an acceptable computational effort. To this purpose, the set of time instants covered by the given control law (the forces and torques demand in science mode) is reduced. The given control law is shown in Figure 4.

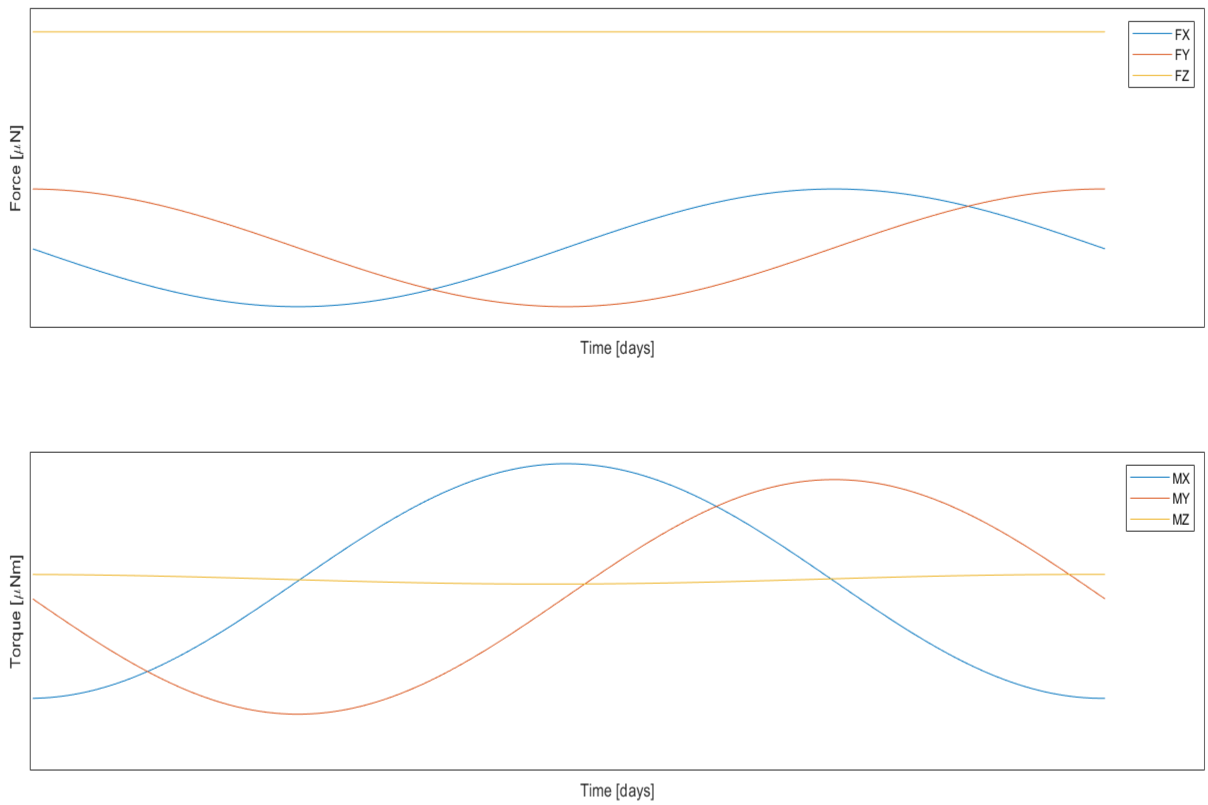

Figure 4: control law timeline for science phase.

Two methods are used to obtain this reduction.

The first one is straightforward: it entails the partitioning of the time range into a number of spaced intervals equal to the number of instants desired minus one. Then, the node values are approximated to the nearest integer obtaining the instants considered. Force and torque values at the subset of instants obtained with this method are shown in Figure 5.

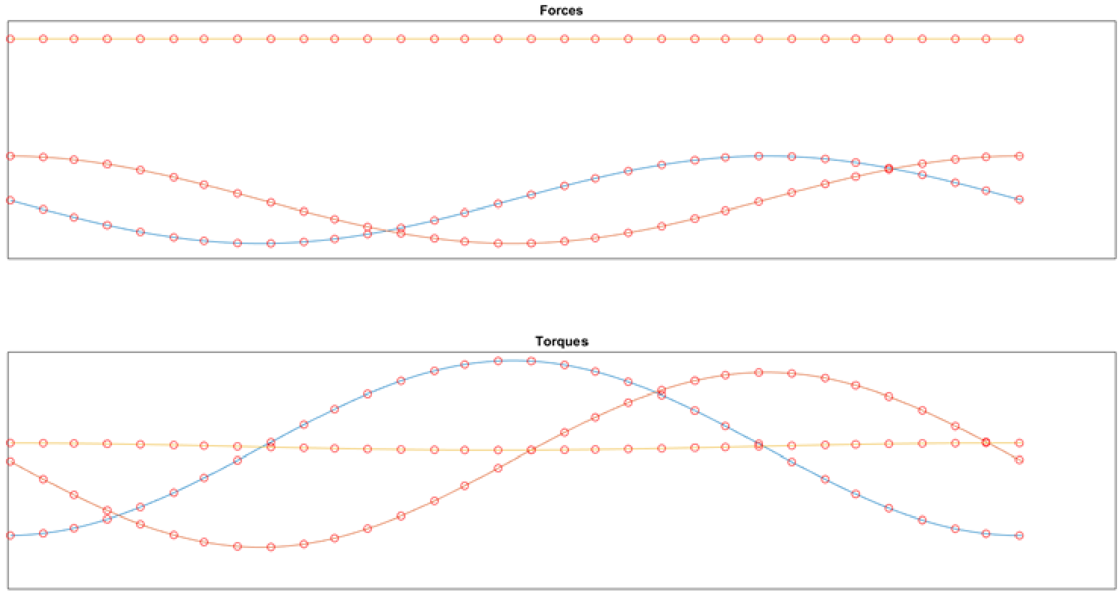

Figure 5: set of instants selected with the first method.

The second method is more sophisticated [10]. It is based on two criteria: the first one is a feasibility criterion, aimed at including the critical conditions of the control law in the final set of the instants chosen. The second criterion provides that the distribution of command of the representative set resembles the one of the original control law. The magnitude of the force and torque vectors is assigned to a load level (Figure 6). Force and torque vectors show individual distribution of instants within each load level. To produce a representative set, both load levels for force and torque magnitudes are considered simultaneously. A number of load classes is generated from the combined load level of force and torque (Figure 7).

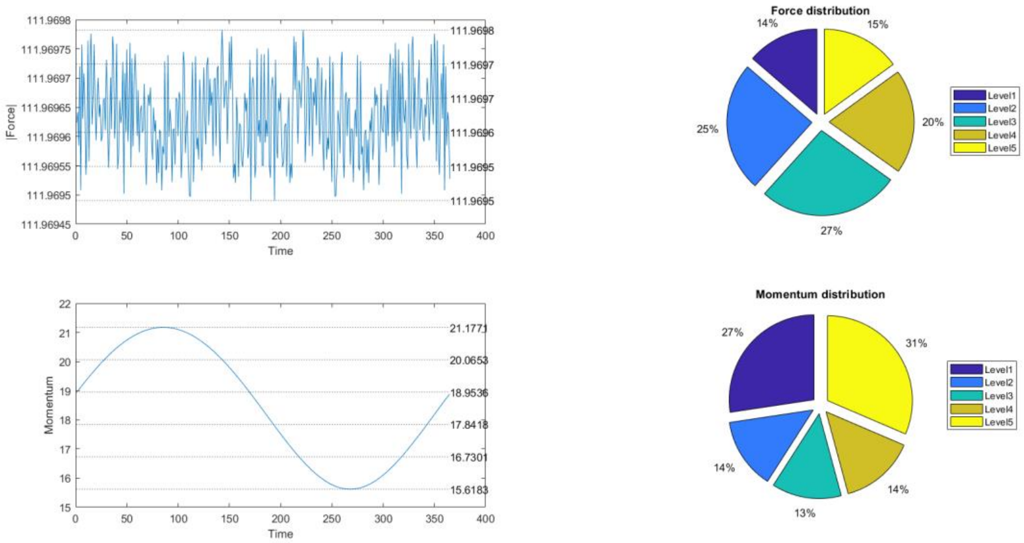

Figure 6: magnitude classification.

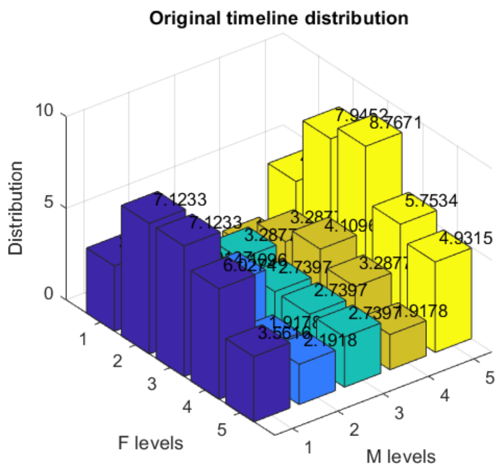

Figure 7: load class distribution.

The final representative set 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 because it is desirable to orient the thruster to best suit the most frequent conditions. Force and torque values at the subset of instants obtained with this method are shown in Figure 8.

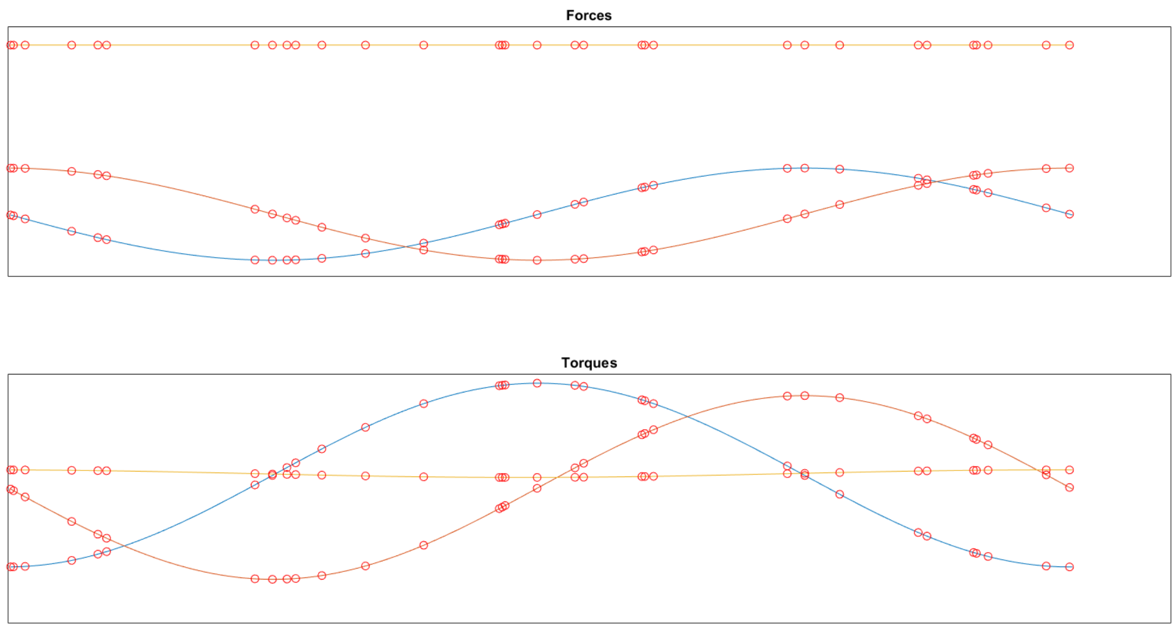

Figure 8: set of instants selected with the second method.

At this point a test is made to choose one of the two methods to use for all the analyses. The scenario with nine thrusters is considered. The discretized model is used to find the thrusters orientation: first with the set of instants obtained with the first method and subsequently with the set of instants obtained with the second method. Thereafter, the orientations found by the discretized model are used in the continuous model and the resulting values of fuel consumption are compared. The results are reported in Table 1.

	Fuel consumption	
	Subset of instants	Whole set of instants
Method 1	98.4%	98.5%
Method 2	98,6%	99.0%

Table 1: comparison of the fuel consumption obtained with the two different subset of instants.

It should be noted that using the first method to reduce the number of instants leads to a better solution. This was predictable in virtue of the regularity of the control law. Therefore, the subset of instants obtained with the first method has been used for all the analyses carried out in this thesis.

4.3 DISCRETIZED MODEL

As anticipated in the previous section, the discretized model is used to find the approximated orientations of the thrusters, suitable for minimizing the overall fuel consumption. For the relevant mathematical description see Chapter 2. CPLEX is the tool used in this phase to solve the problem. At the first run the whole domain, according to the constraints, is considered. After the optimization, the thrusters' orientations are given in terms of angles α and β. At this point an iterative process starts which considers a smaller domain and tries to refine locally the solution. Note that the first run is the only one that deals with the problem globally, therefore an appropriate mesh has to be chosen. After the first run, for every thruster the domain is reduced from a portion of semi sphere to a region limited by the nodes adjacent to the solution. If $\alpha_{opt} = \alpha_n$ and $\beta_{opt} = \beta_m$ represent the optimal orientation for a given thruster (Figure 9) with n number of alpha node and m number of beta node, the new domain will be bounded as follows:

$$[\alpha_{n-1} \dots \alpha_{opt} \dots \alpha_{n+1}]$$

$$[\beta_{m-1} \dots \beta_{opt} \dots \beta_{m+1}]$$

The new domain is shown in Figure 10.

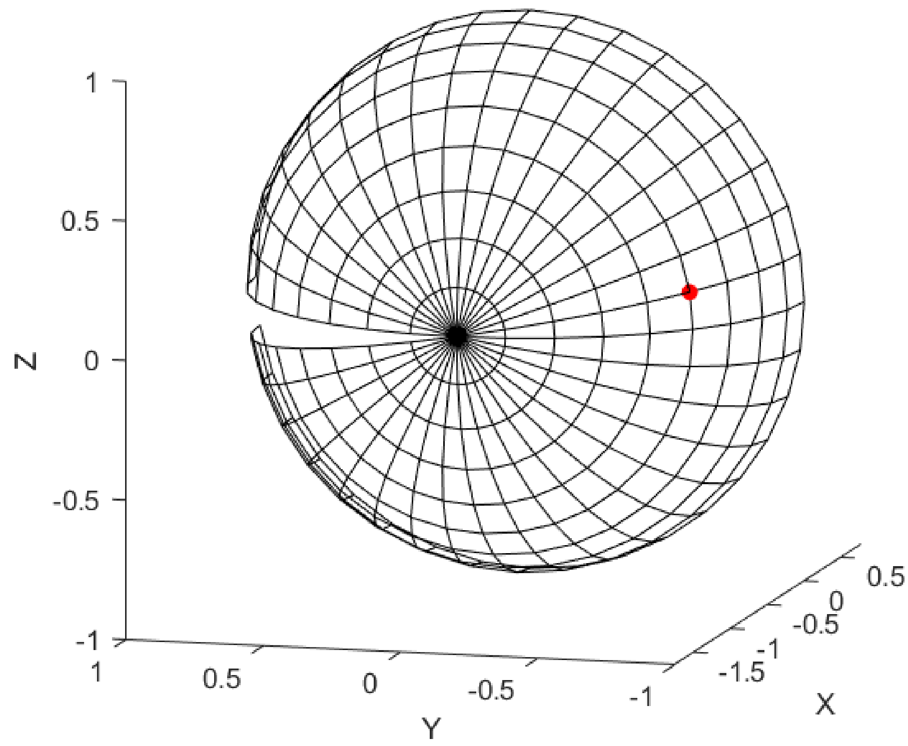

Figure 9: example of sub-optimal solution in the discretized domain.

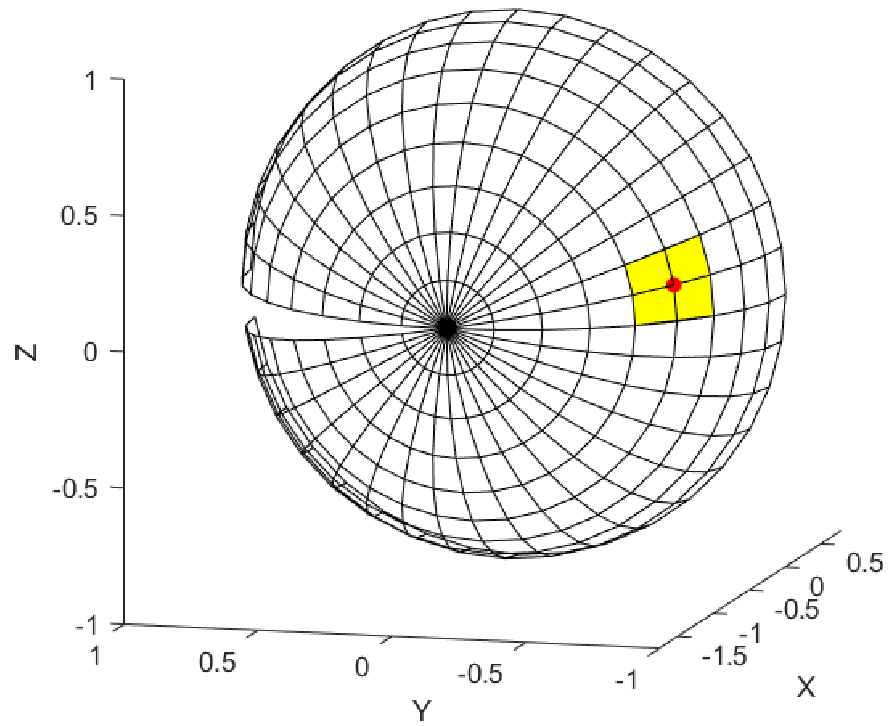

Figure 10: new domain.

A new mesh of the new domain is created. The new mesh is centered correspondently to the previous sub-optimal solution, in order to allow the solver to find a better solution or, in the worst case, the previous one. An example is shown in Figure 11. If the previous solution is located on the edge of the domain, the new mesh cannot be centered in this solution because, in such a way, a forbidden region would be included in the mesh. In this case, only one half of the domain will be considered.

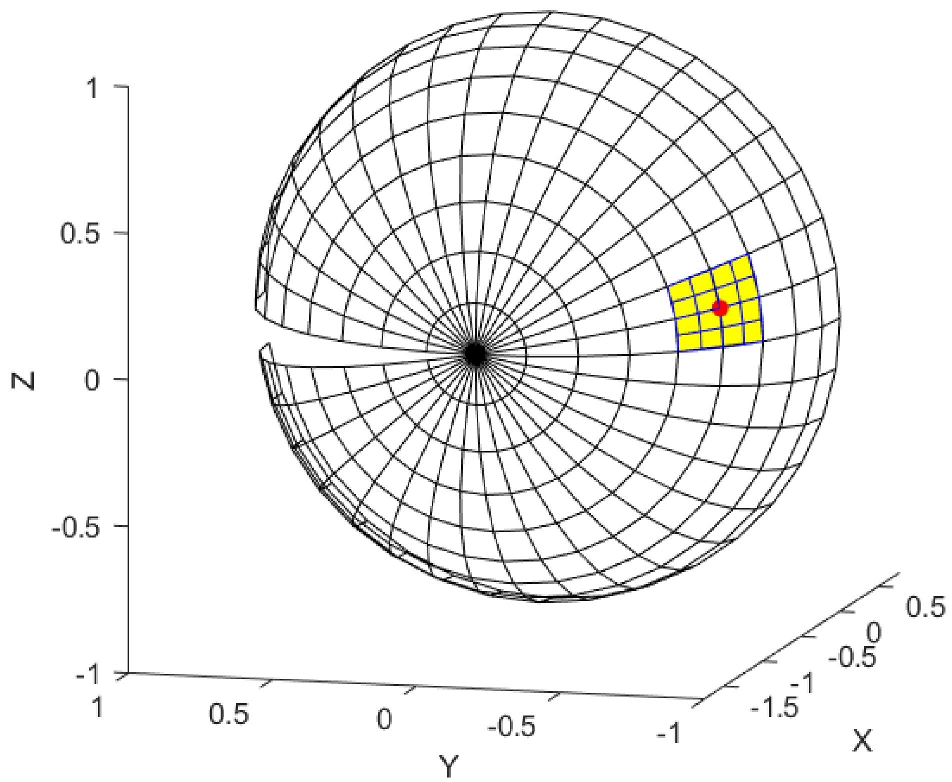

Figure 11: new mesh.

The number of angles α and β considered in the iterations is reduced with reference to the global run, therefore a larger set of instants can be considered.

4.4 CONTINUOUS MODEL

At the end of every run of the discretized model, a run with the continuous model is performed to evaluate the fuel consumption for the whole set of instants. The thruster orientations in terms of angles α and β must be converted in direction cosines with the following equations:

$$v_x = sign(th) \cdot \sin\beta$$

$$v_y = \cos\alpha \cdot cos\beta$$

$$v_z = \sin\alpha \cdot cos\beta$$

where $sign(th)$ is -1 if the thruster is oriented on the positive x axis and +1 otherwise. Note that the orientation of the thruster is considered as the orientation of the force it can exert: a thruster oriented upward although emitting particles on the positive axis, generates a thrust of negative sign. Once the direction cosines are calculated, they are fixed in the continuous model and the test can start.

4.5 MATLAB ANALYSIS

The other tool used to find the thruster orientation adopted to minimize the overall fuel consumption is the MATLAB non-linear solver *fmincon.* All the algorithms used (see Chapter 3) need an initial solution, therefore the MATLAB analyses are executed after the analysis carried out by CPLEX. The initial solution is a column vector with the alpha and beta angles of the optimal solution and the thrust exerted at every instant. In order to choose which algorithm is better to use for all the analyses, a test has been performed considering the scenario with nine thrusters. The initial solution is derived from that obtained after the first run of the discretized and the continuous model. The whole set of instants is taken into account. The nonlinear run carried out with the interior point algorithm is reported in Figure 12. Figure 13 and Figure 14 show the nonlinear runs carried out with the sequential quadratic programming and active-set algorithm, respectively.

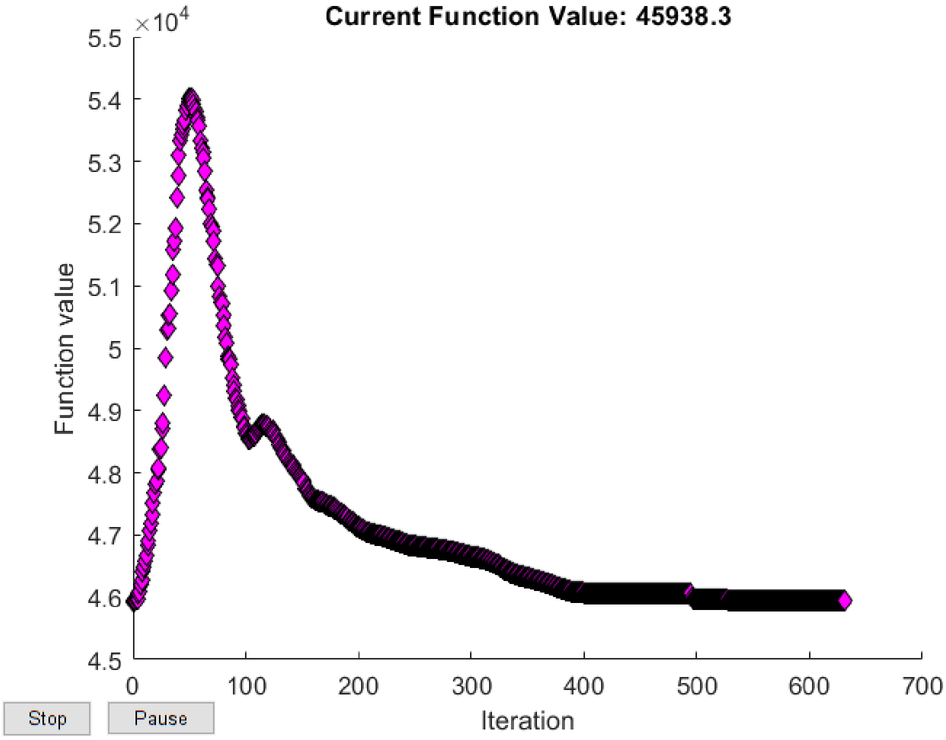

Figure 12: interior point run.

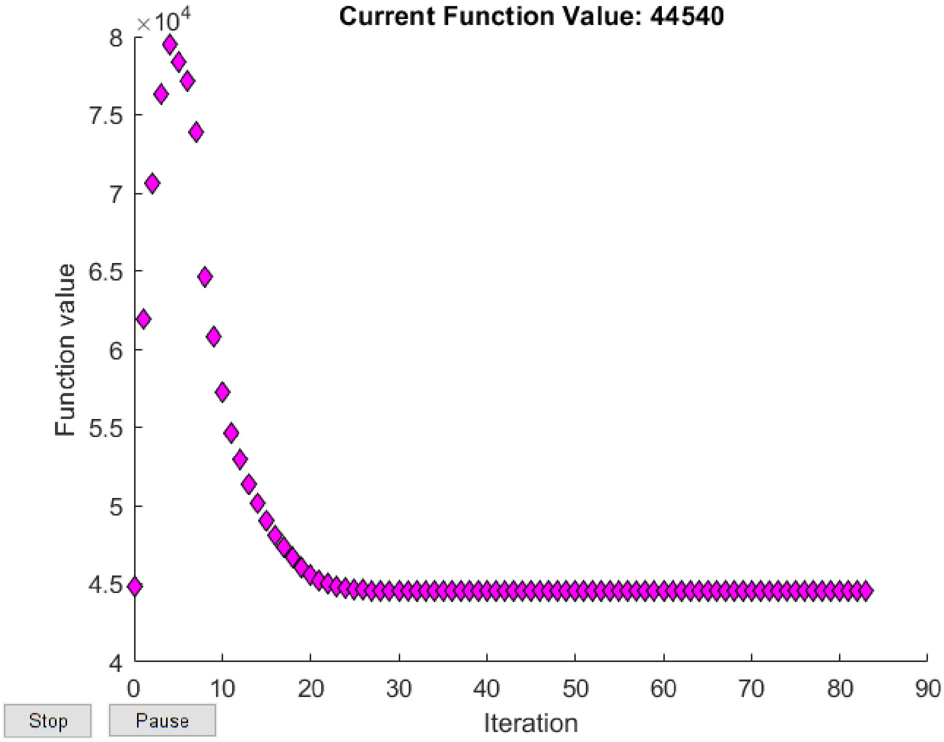

Figure 13: sequential quadratic programming run.

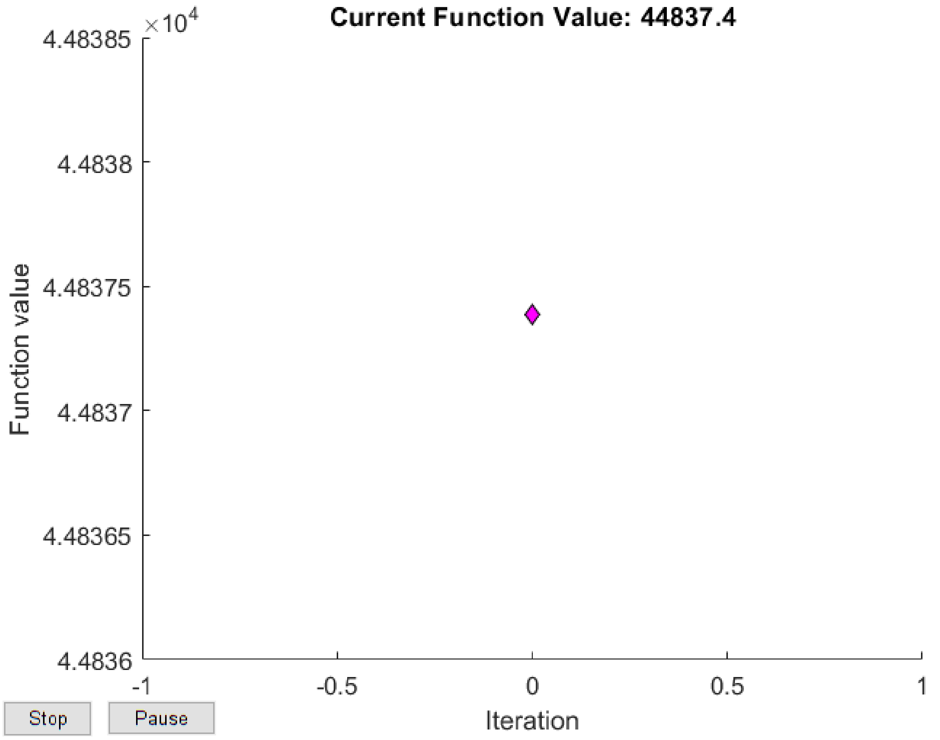

Figure 14: active set run.

As seen from the figures 12,13 and 14, the active set algorithm cannot find a solution to the problem probably because of the large number of variables. The other two algorithms, on the contrary, find out a solution. The sequential quadratic programming algorithm finds a better solution than the interior point one. Furthermore, the sequential quadratic programming takes less iterations to find the solution. For these reasons the sequential quadratic programming algorithm has been chosen for all the remaining analyses.

4.6 QUALIFICATION OF SOLUTIONS

When NP-hard problems [6] relevant to real-world applications are involved, especially if at large scale, as in our case, the proof of optimality is hardly ever guaranteed. The solutions obtained so far are not optimal but sub-optimal because the CPLEX analysis always ends because of an out of memory error or because the time limit of two days is reached, but not because all the possibilities are explored. In order to allow an evaluation of the quality of the solution obtained, a search for a reference (lower) bound is made. CPLEX provides the LP-relaxed lower bound of the objective function. The most representative bound would be the one obtained by the discretized model considering a very fine mesh of alpha and beta angles and considering all the time instants. This is however unrealistic because of the huge dimension of the problem. In order to allow the solver to find a bound with the maximum possible number of instants and the finest possible mesh of angles, a relaxation of the problem is implemented. It consists in considering no longer the variables alpha and beta as binary. This simplification allows the solver to find the LP-relaxed lower bound of the objective function for an angle discretization of three degrees and for 122 instants. The instants are derived from the original control law simply selecting one instant every three. Two sets of 122 instants are taken into account: the only difference is that for the first set the selection starts from the first instant while for the second set it starts from the second one.

4.7 HANDLING OF ADDITIONAL PERTURBANCES

The thrusters of DFACS are involved not only during the scientific measurements but also in other phases of the mission. These situations differ from the science mode in terms of force and torque demand. The thrusters have to compensate the disturbances of the antenna during the scientific measurements, the variation of the antenna orientation while in science orbit and while in transfer orbit and further compensate the main engine de-pointing while in transfer orbit. The control laws for these situations are not perfectly defined at this phase of the study, therefore random distributions and preliminary estimates have been considered. In case of the compensation of the disturbances of the antenna during the scientific measurements, random torques in a range of $[-10\ \mu Nm, 10\ \mu Nm]$ and random forces in a range of $[-10\ \mu N, 10\ \mu N]$ are added to torques and forces of the science mode (Figure 15).

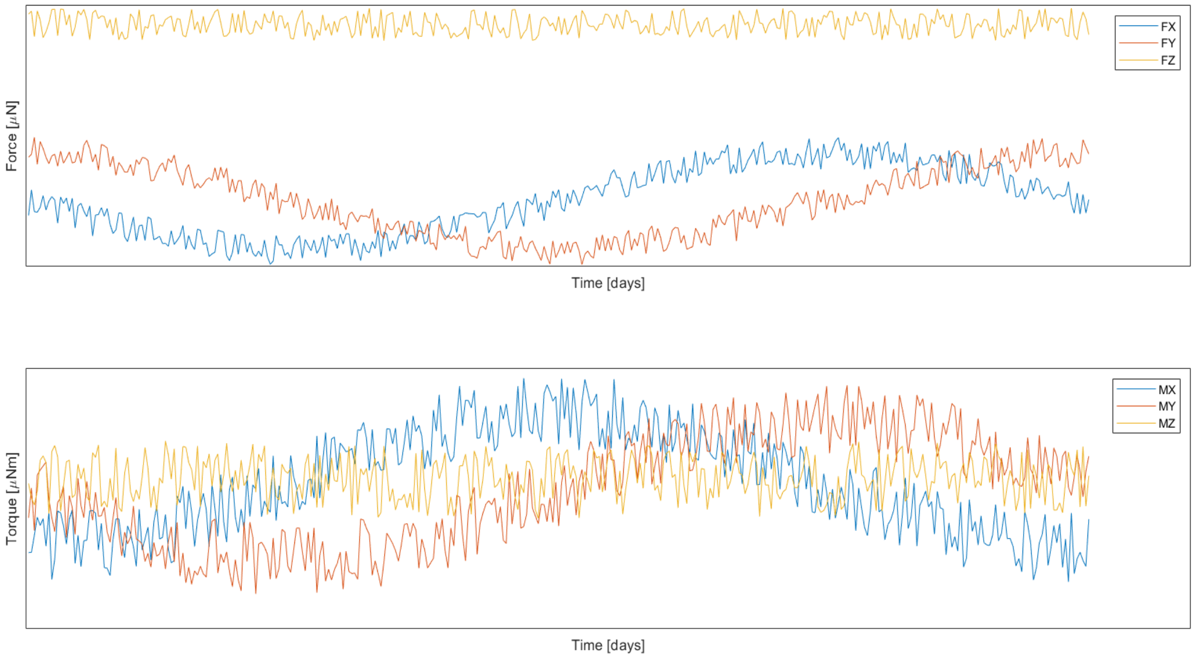

Figure 15: control law timeline In case of the compensation of the disturbances of the antenna during the scientific measurements.

In order to consider the compensation of the variation of the antenna orientation while in science orbit, random torques in a range of $[-200\mu Nm, 200\mu Nm]$ and random forces in a range of $[-200\mu N, 200\mu N]$ are added to the torques and forces in science mode (Figure 16).

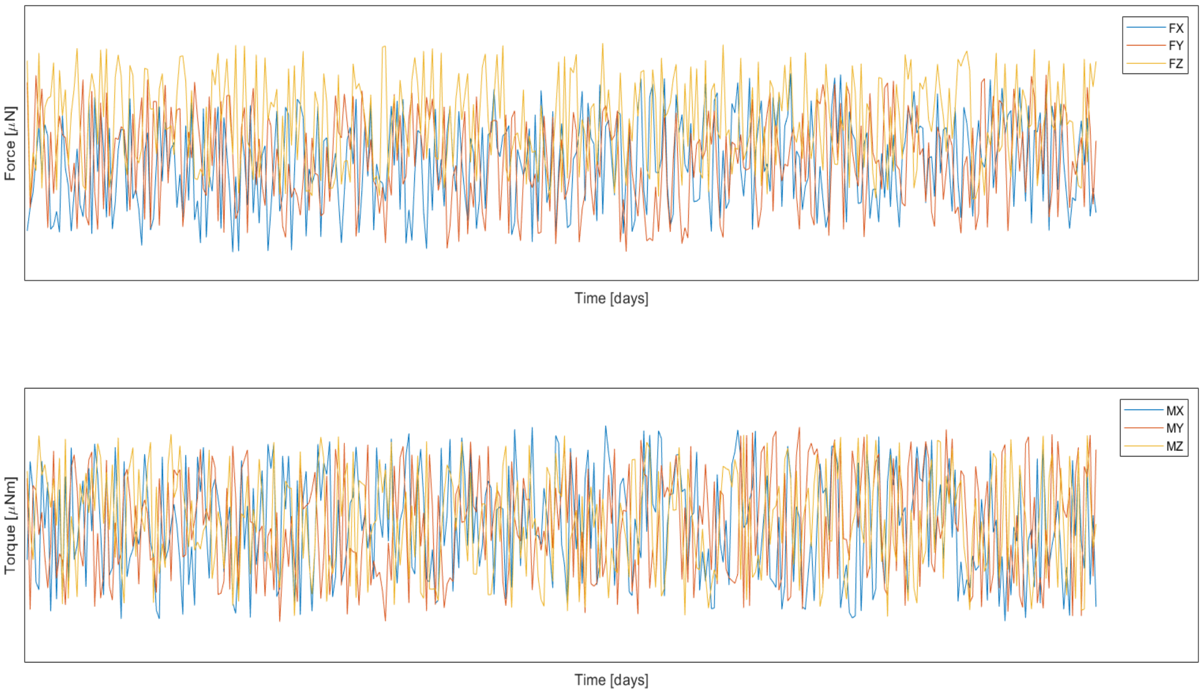

Figure 16: control law timeline in case of the compensation of the variation of the antenna orientation while in science orbit.

In the case of the compensation of the variation of the antenna orientation while in transfer orbit, torques and forces of science mode are considered, except for the torque around the x axis, which is considered constant and of 1,5 mNm (Figure 17).

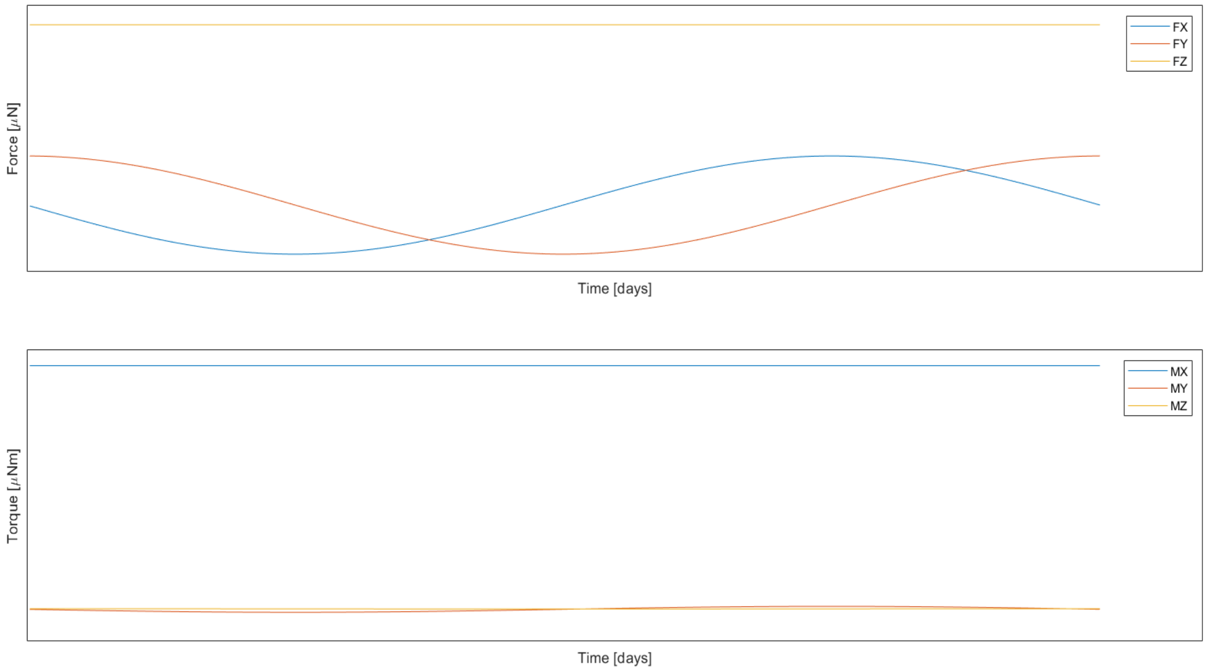

Figure 17: control law timeline in case of the compensation of the variation of the antenna orientation while in transfer orbit.

In the case of the compensation of the main engine de-pointing while in transfer orbit, torques and forces of science mode are considered with the exception of the torque around the y axis, which is considered at first with a constant value of 40 µNm (Figure 18) and later with a constant of -40 µNm (Figure 19).

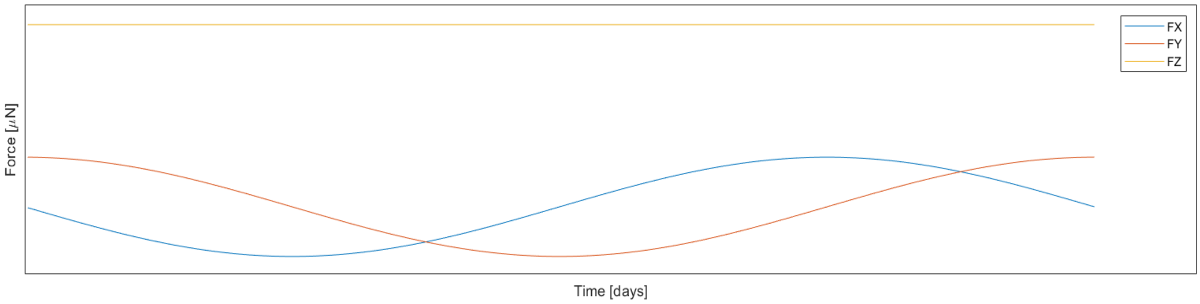

Figure 18: control law timeline in case of the compensation of the main engine de-pointing while in transfer orbit (with the torque around y axis of 40 µNm).

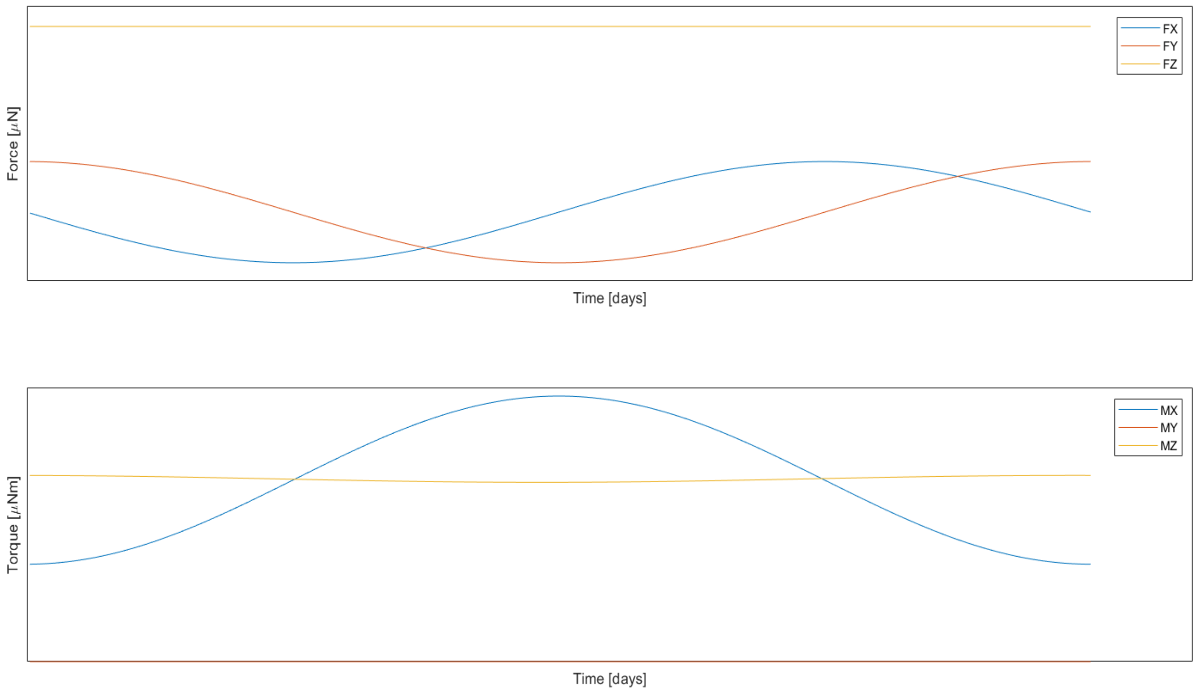

Figure 19: control law timeline in case of the compensation of the main engine de-pointing while in transfer orbit (with the torque around y axis of -40 µNm).

4.8 SCENARIOS FOR AOCS

The first scenario considered for the DFACS thrusters provides for the presence of two clusters of three thrusters each. All the thrusters are positioned on the lower surface of the spacecraft. It allows for cables saving with respect to others scenarios which provide for the thrusters on the lower and upper surface of the spacecraft.

Another scenario provides for the presence of six thrusters, of which three are positioned on the lower surface and three on the upper one. The thruster positions are chosen considering some symmetries of the resulting configuration.

The last scenario considered provides for the presence of eight thrusters, of which three are positioned on the lower surface and five on the upper one.

The reference scenario provides for the presence of two clusters of three thrusters each. All the thrusters are positioned on the lower surface of the spacecraft and the thruster orientation is fixed. This scenario is the one considered so far by LISA program.

4.9 AOCS CONTROL LAW

The thrusters of the AOCS are requested to balance a torque of ± 50 mNm simultaneously around the three axes x,y and z. All the combinations of maximum positive and negative torque around the three axes are considered. Every instant of the control law considers one of these combinations (Figure 20).

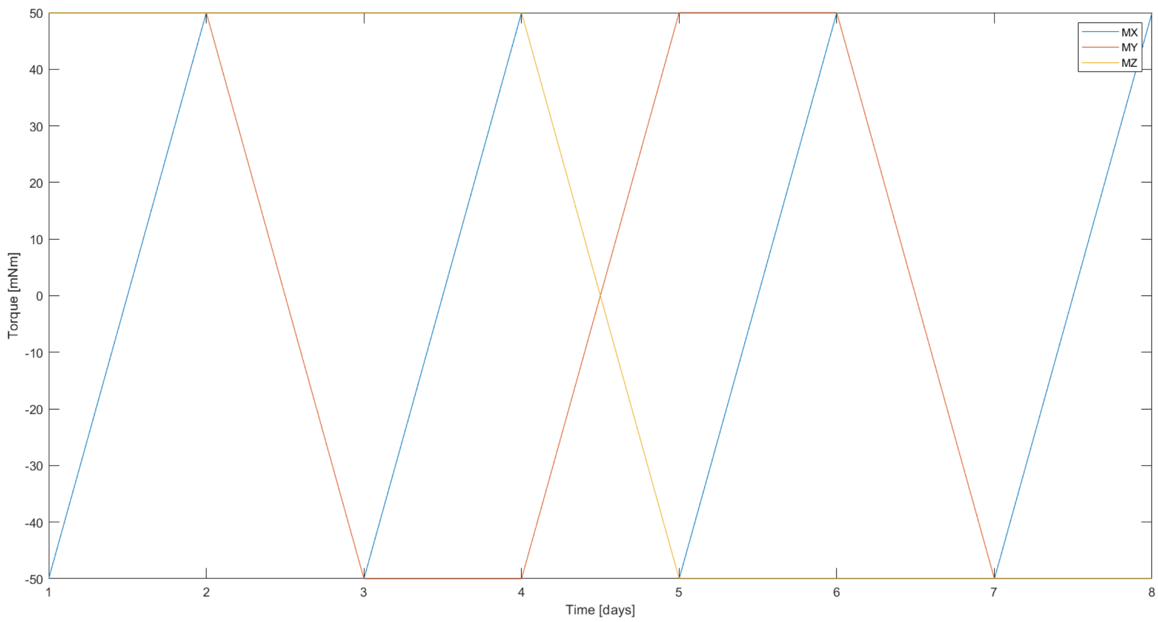

Figure 20: control law timeline for AOCS.

4.10 CHANGES TO THE MODEL

Differently from the DFACS thrusters, the AOCS thrusters cannot exert a continuous thrust, but they are on-off thrusters: they either exert a fixed thrust or zero thrust. This makes it necessary to apply some changes in the discretized model. The thrust variables must be discontinuous, but this implies an increase of the binary variables of the model, therefore the analyses begin with a run of the discretized model by considering the thrust as continuous. Afterwards, a refinement of the domain is considered as described in 4.3 of this chapter. As this allows for a significant reduction of the (binary) variables of the relevant model, it is possible to consider the thrust variables as actually discontinuous in the following run of the model.

5. RESULTS

In this chapter all the results of the analysis for the LISA thruster layout optimization will be discussed. The first results concern the optimization of the layout of the thrusters relevant to the DFACS (Drag Free Attitude Control System). Then the results of the bound analysis carried out to qualify the solutions obtained are shown. Afterwards the results of the analysis of the handling of the spacecraft subject to different disturbances are presented. Finally, the last part of the results concerns the optimization of the layout of the thrusters relevant to the AOCS (Attitude and Orbit Control System).

5.1 NINE-THRUSTERS SCENARIO FOR DFACS

In the following table (Table 2) the results of the analyses performed with CPLEX for the thrusters of the DFACS for the scenario with nine thrusters are summarized.

CPLEX ANALYSES			
Solution	Fuel consumption		Gain
	Subset of instants	Whole set of instants	Whole set of instants
Global	98.42%	98.47%	1.53%
Iteration 1	97.97%	98.03%	1.97%
Iteration 2	97.92%	97.99%	2.01%
Iteration 3	-	97.97%	2.03%

Table 2: fuel consumption for the DFACS scenarios with nine thrusters.

It has to be noted that the value of fuel consumption is expressed as a percentage of the reference scenario value. Note that all the runs of the iterations are done with the discretized model considering both the subset of instants and the whole set of instants. For iteration 1 and iteration 2, the best solution has been obtained from the solution found by the discretized model considering the subset of instants and then put in the continuous model. For iteration 3, the best solution has been obtained directly from the discretized model, considering the whole set of instants.

The thruster orientation for the global solution is shown in Figure 21. Figure 22 shows the comparison between the thrust trend of each thruster of the reference solution and of the global solution over time. The comparison between the overall thrust exerted by each thruster of the reference solution and the global solution is reported in Figure 23.

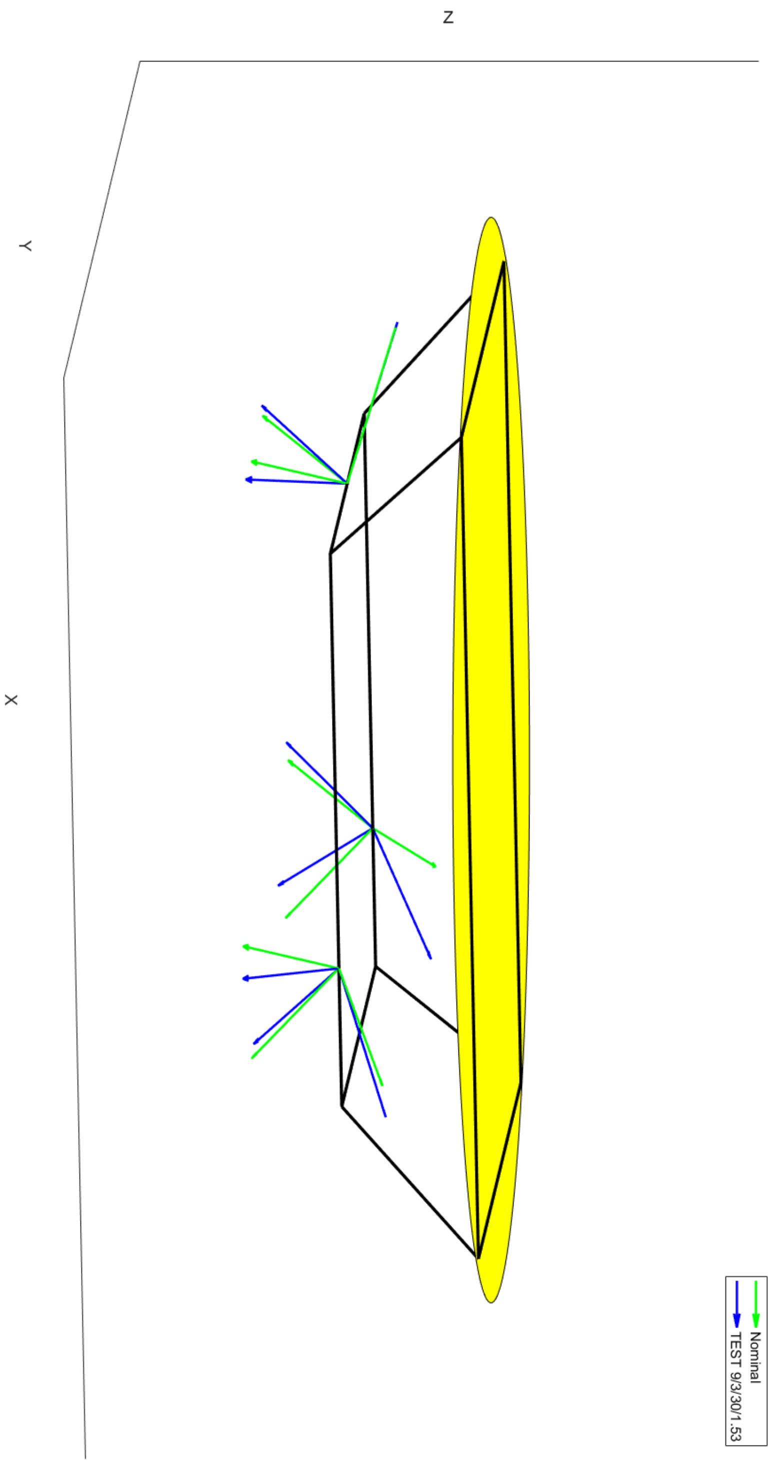

Figure 21: thruster orientation – global solution.

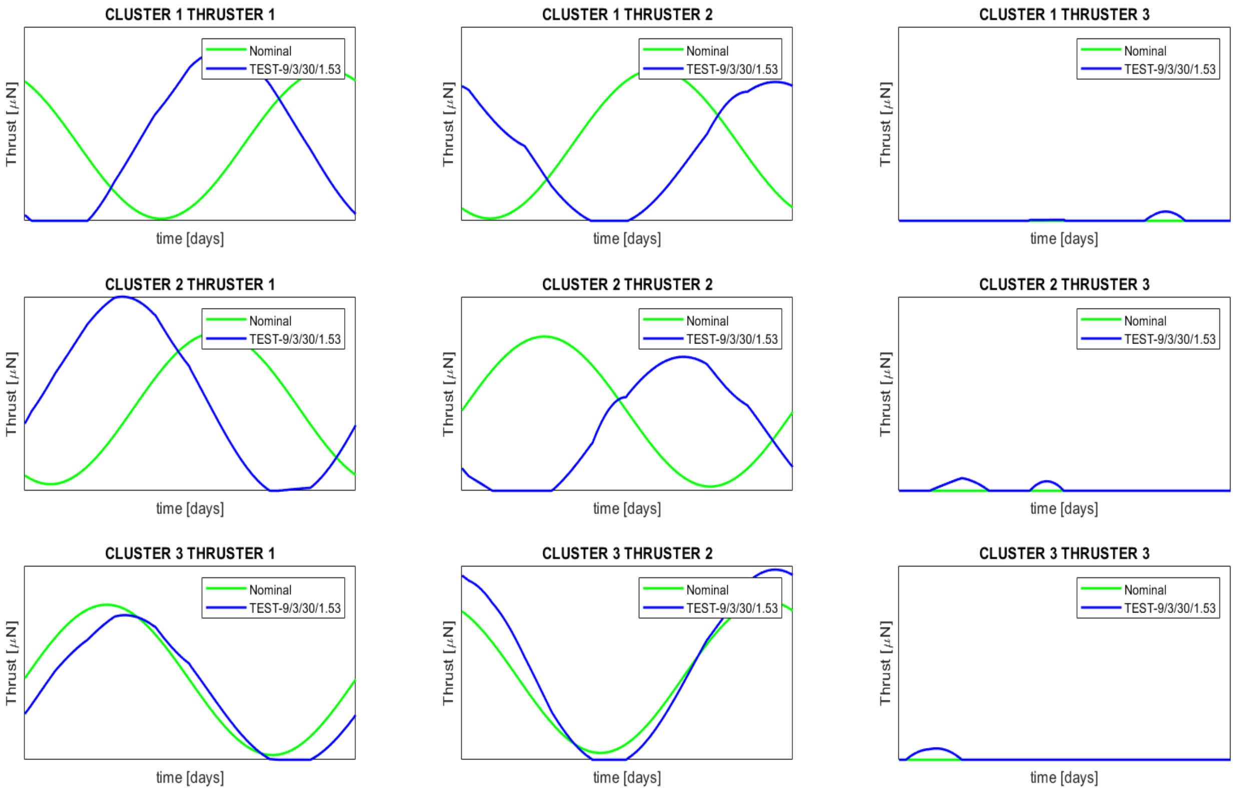

Figure 22: thrust over time – global solution.

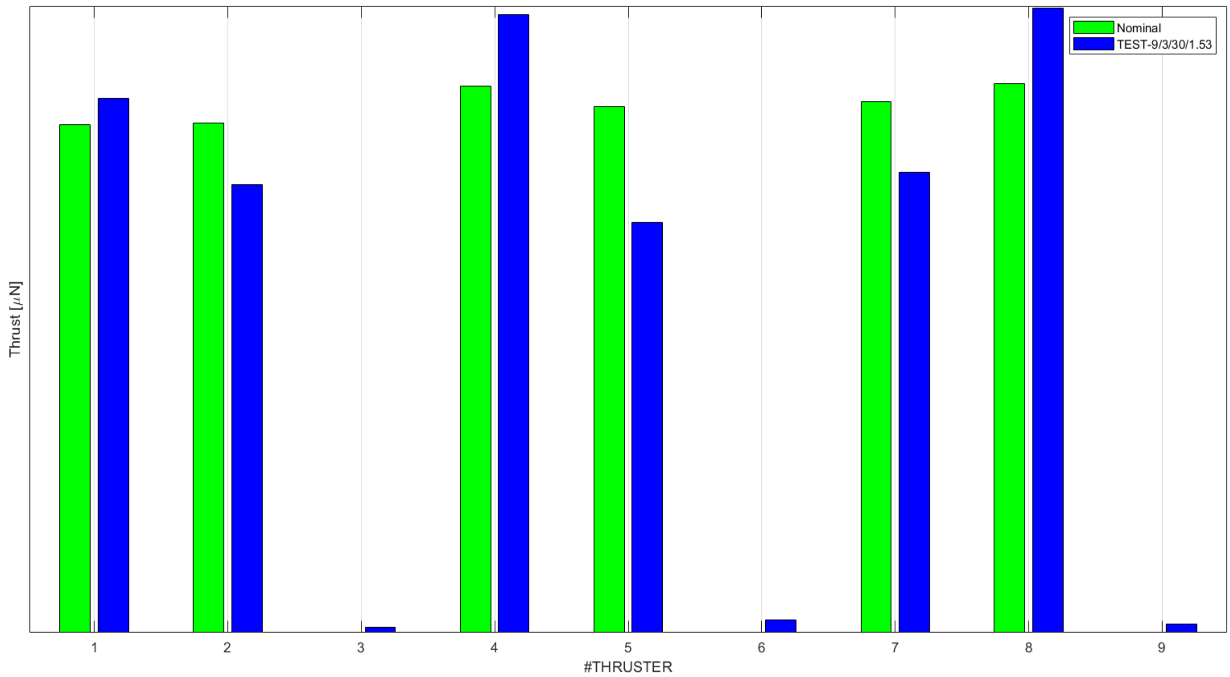

Figure 23: overall thrust for thruster – global solution.

It should be noted that the shape of the spacecraft in Figure 21 and in all the analogous figures reported later in this thesis is not the one envisaged in the program, due to confidentiality restrictions.

The thruster orientation for the iteration 1 solution is shown in Figure 24. Figure 25 shows the comparison between the thrust trend of each thruster of the reference solution and of the iteration 1 solution over time. The comparison between the overall thrust exerted by each thruster of the reference solution and the iteration 1 solution is illustrated in Figure 26.

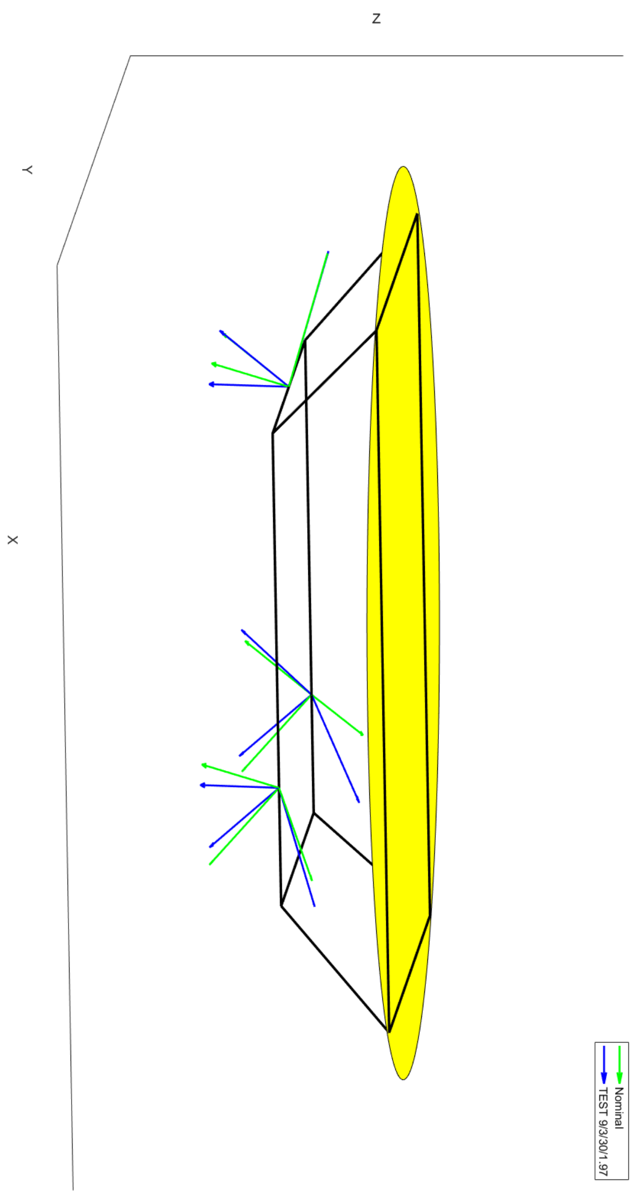

Figure 24: thruster orientation – iteration 1 solution.

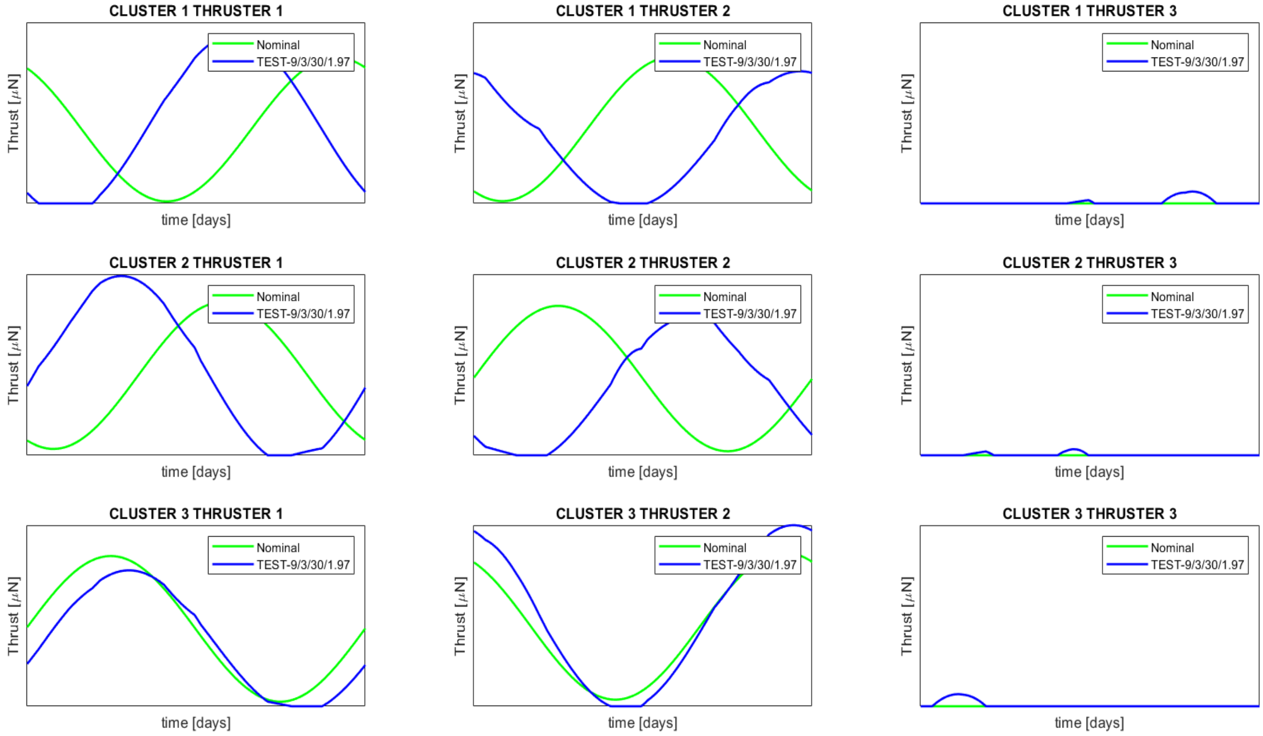

Figure 25: thrust over time – iteration 1 solution.

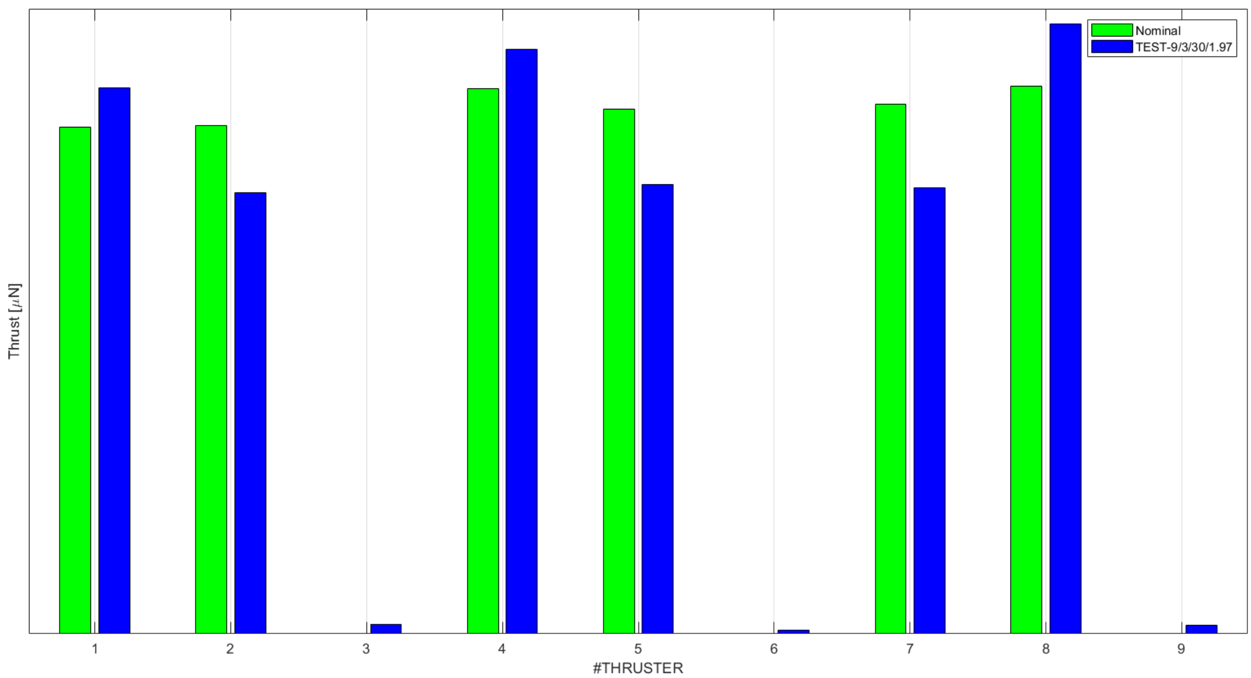

Figure 26: overall thrust for thruster – iteration 1 solution.

The thruster orientation for the iteration 2 solution is shown in Figure 27. Figure 28 shows the comparison between the thrust trend of each thruster of the reference solution and of the iteration 2 solution over time. The comparison between the overall thrust exerted by each thruster of the reference solution and the iteration 2 solution is illustrated in Figure 29.

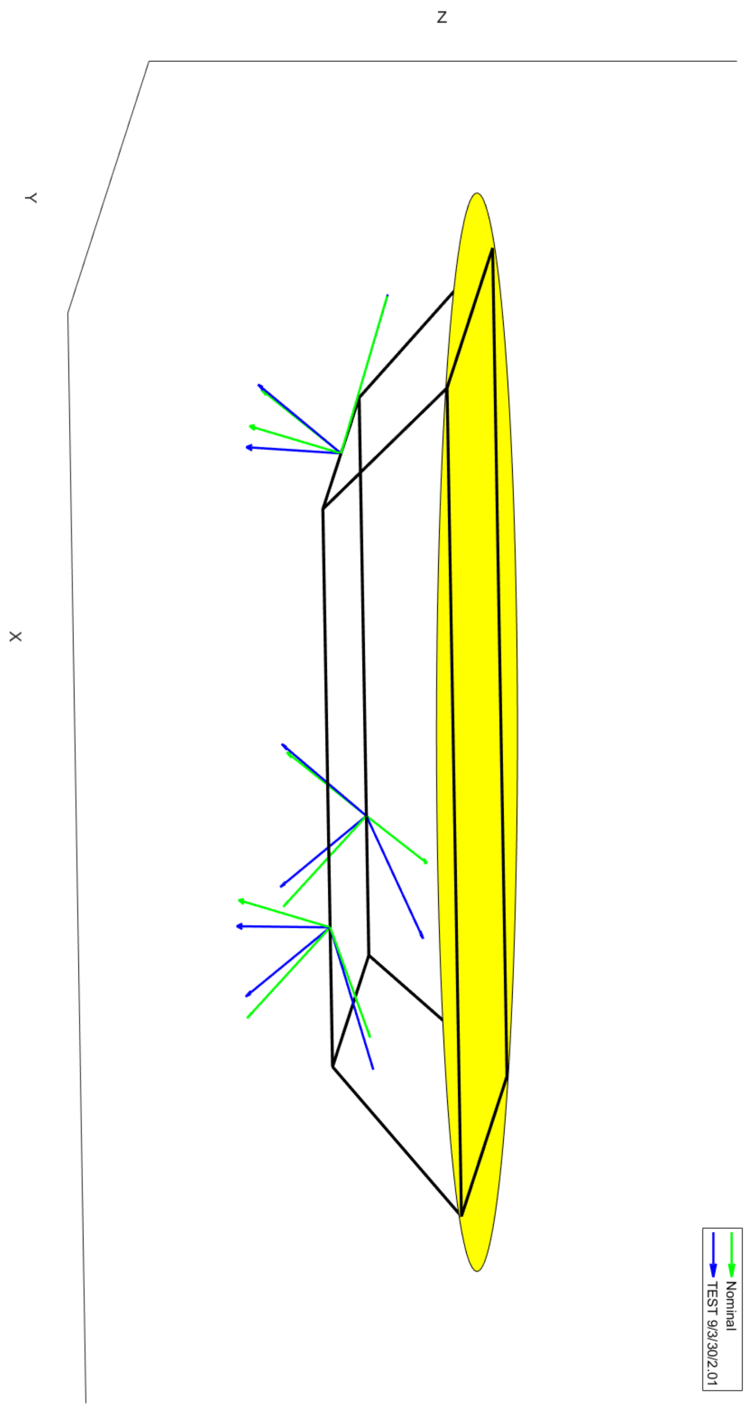

Figure 27: thruster orientation – iteration 2 solution.

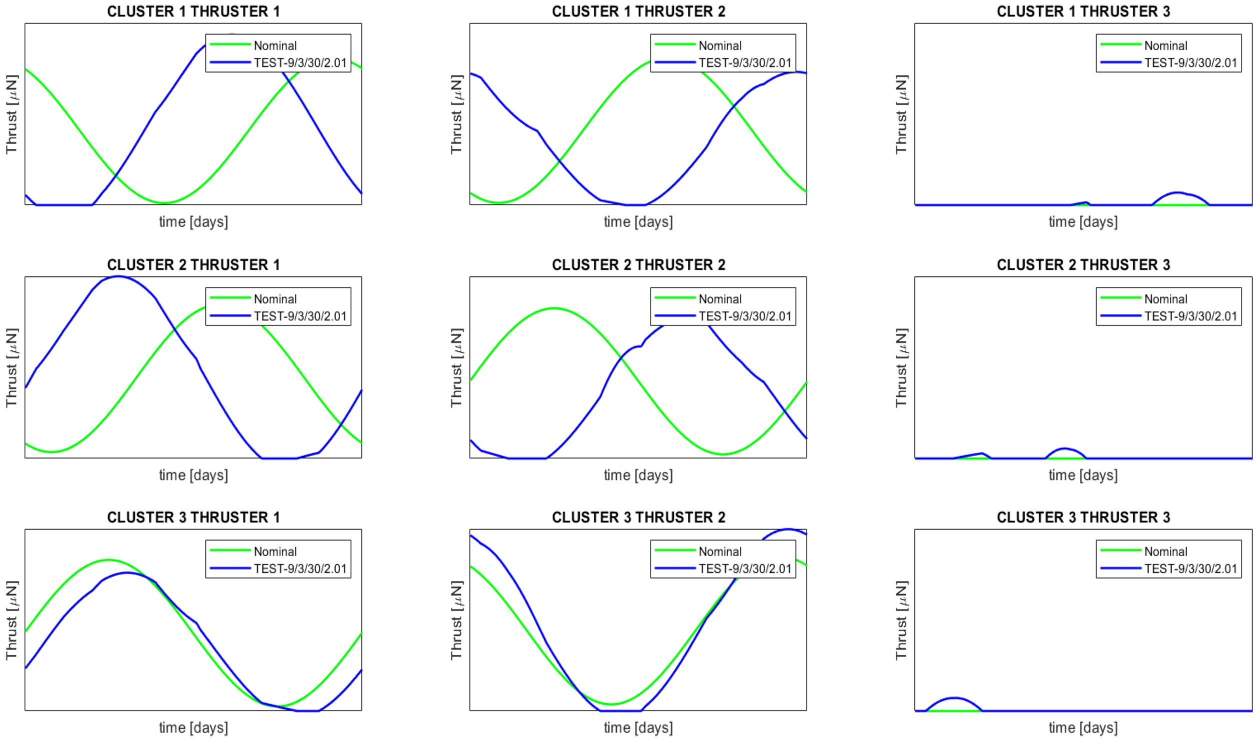

Figure 28: thrust over time – iteration 2 solution.

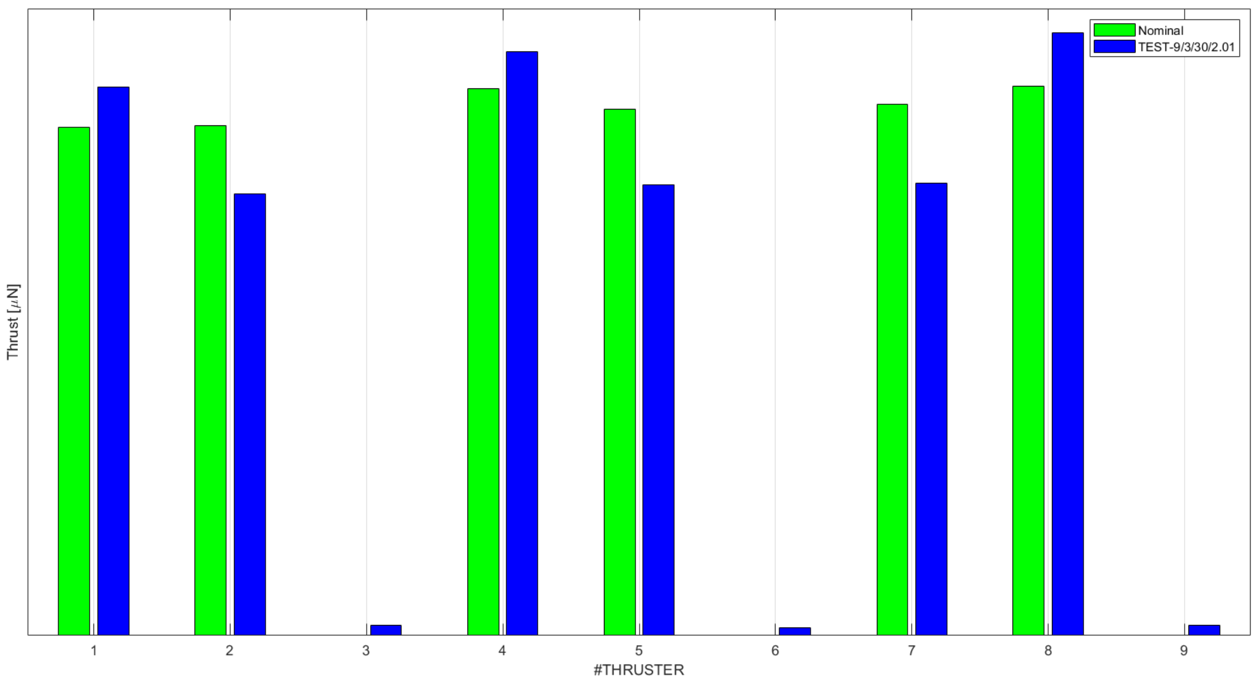

Figure 29: overall thrust for thruster – iteration 2 solution.

The thruster orientation for the iteration 3 solution is shown in Figure 30. Figure 31 shows the comparison between the thrust trend of each thruster of the reference solution and of the iteration 3 solution over time. The comparison between the overall thrust exerted by each thruster of the reference solution and the iteration 3 solution is illustrated in Figure 32.

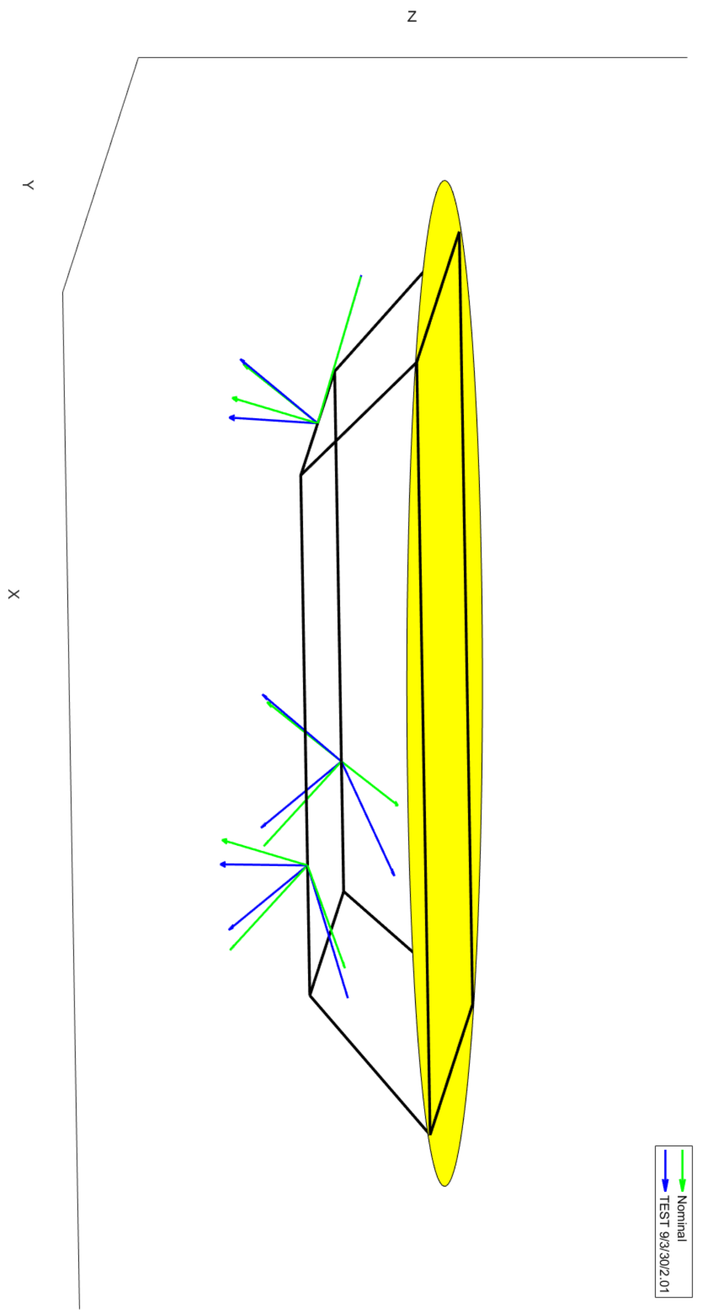

Figure 30: thruster orientation – iteration 3 solution.

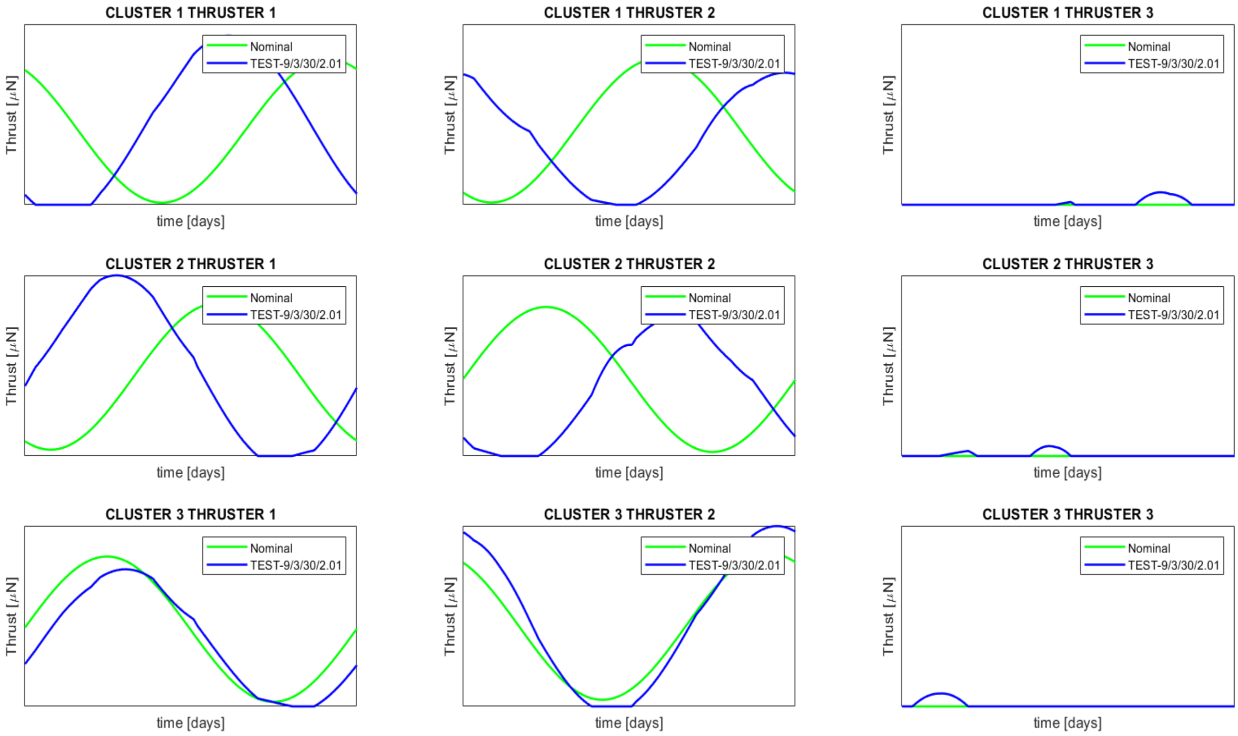

Figure 31: thrust over time – iteration 3 solution.

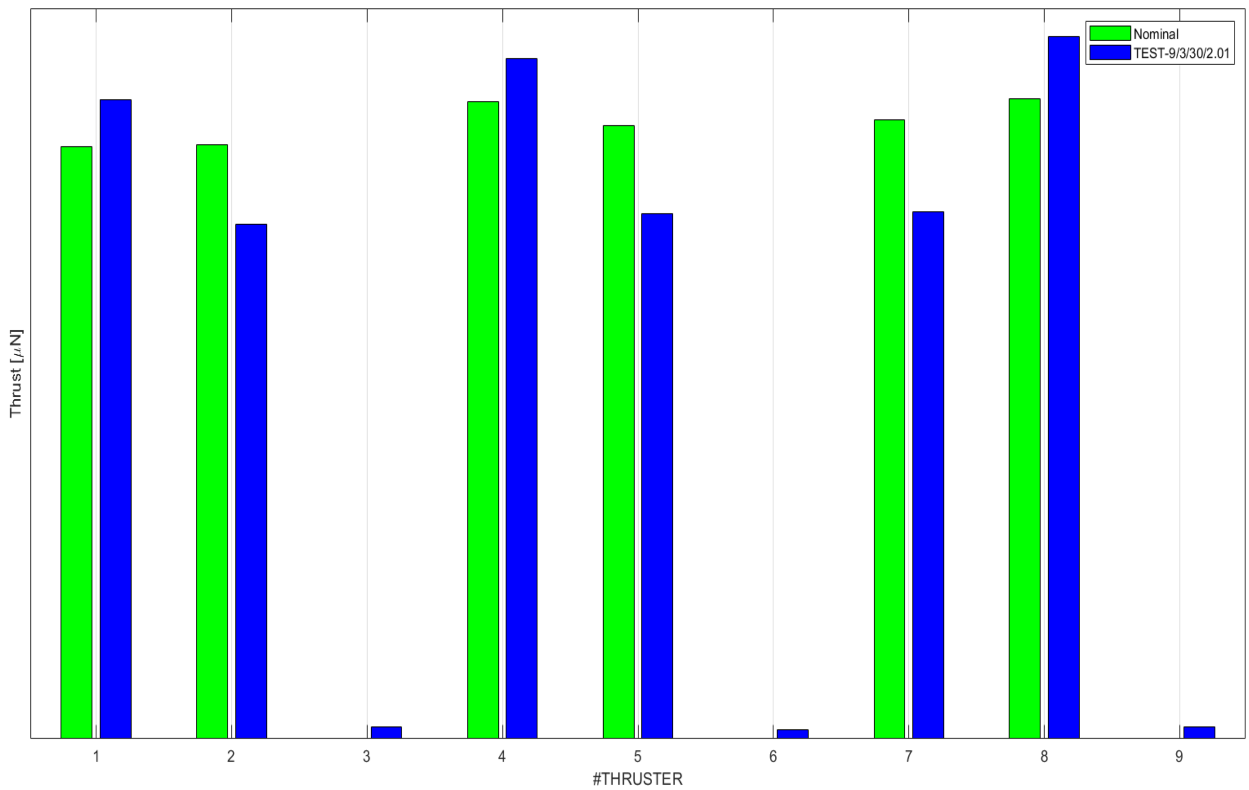

Figure 32: overall thrust for thruster – iteration 3 solution.

In the following table (Table 3) the results of the analyses performed with MATLAB for the thrusters of DFACS for the scenario with nine thrusters are summarized.

MATLAB ANALYSES		
Solution	Fuel consumption	Gain
	Whole set of instants	Whole set of instants
Global	97.82%	2.18%
Iteration 1	97.83%	2.17%
Iteration 2	97.83%	2.17%
Iteration 3	97.82%	2.18%

Table 3: fuel consumption for the DFACS scenarios with nine thrusters obtained with Matlab.

The Matlab analysis for the global solution is reported in Figure 33.

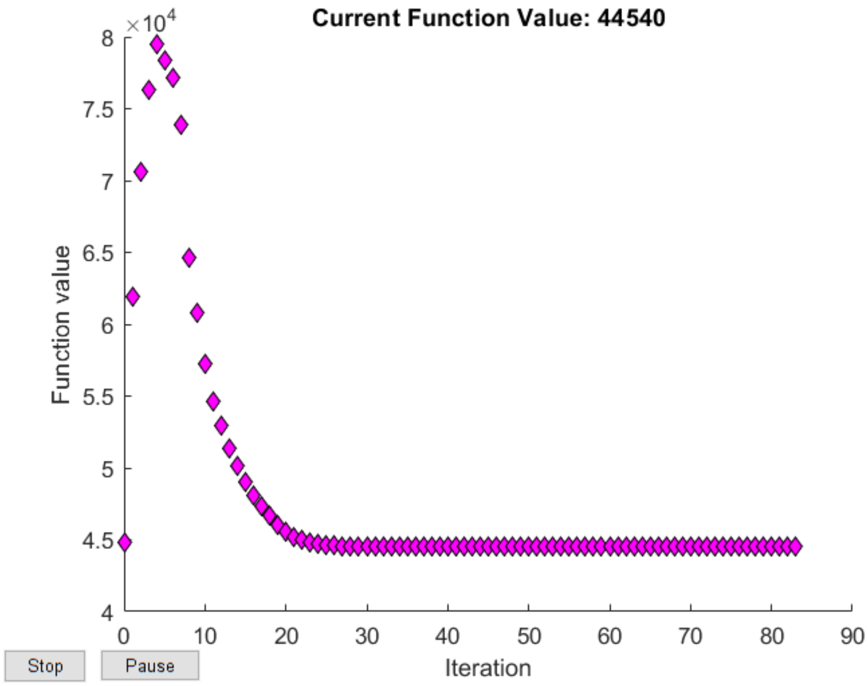

Figure 33: global solution run.

The thruster orientation for the global solution is shown in Figure 34. Figure 35 shows the comparison between the thrust trend of each thruster of the reference solution and of the global solution over time. The comparison between the overall thrust exerted by each thruster of the reference solution and the global solution is illustrated in Figure 36.

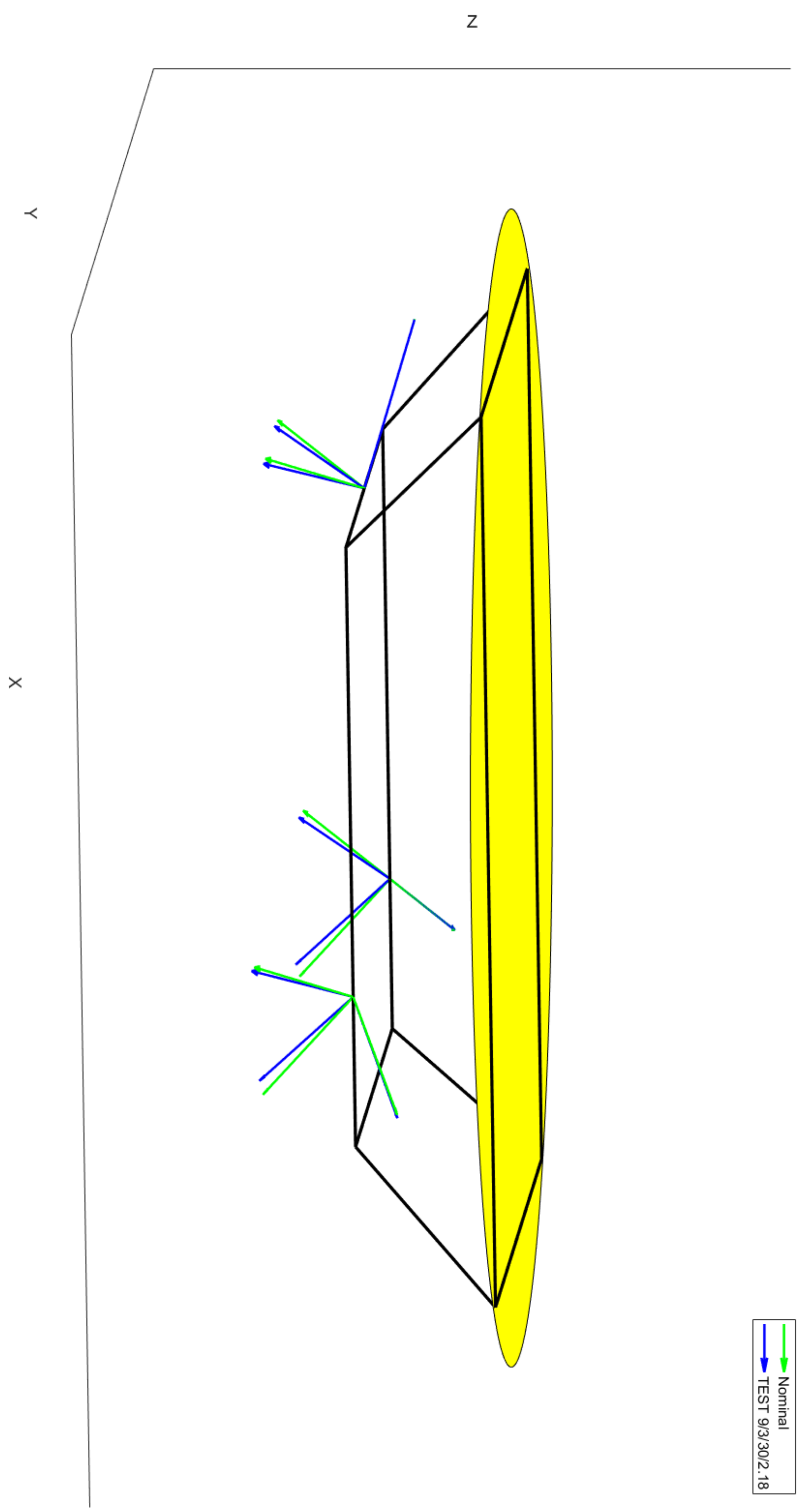

Figure 34: thruster orientation – global solution (Matlab).

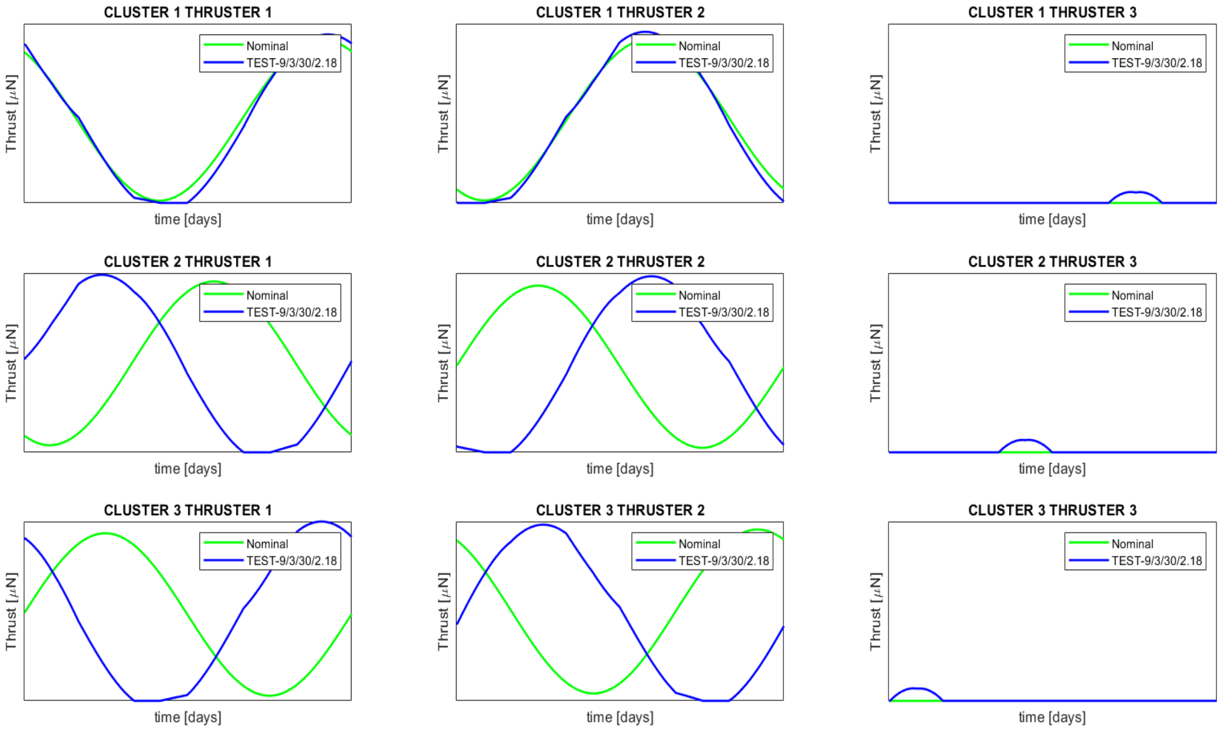

Figure 35: thrust over time – global solution (Matlab).

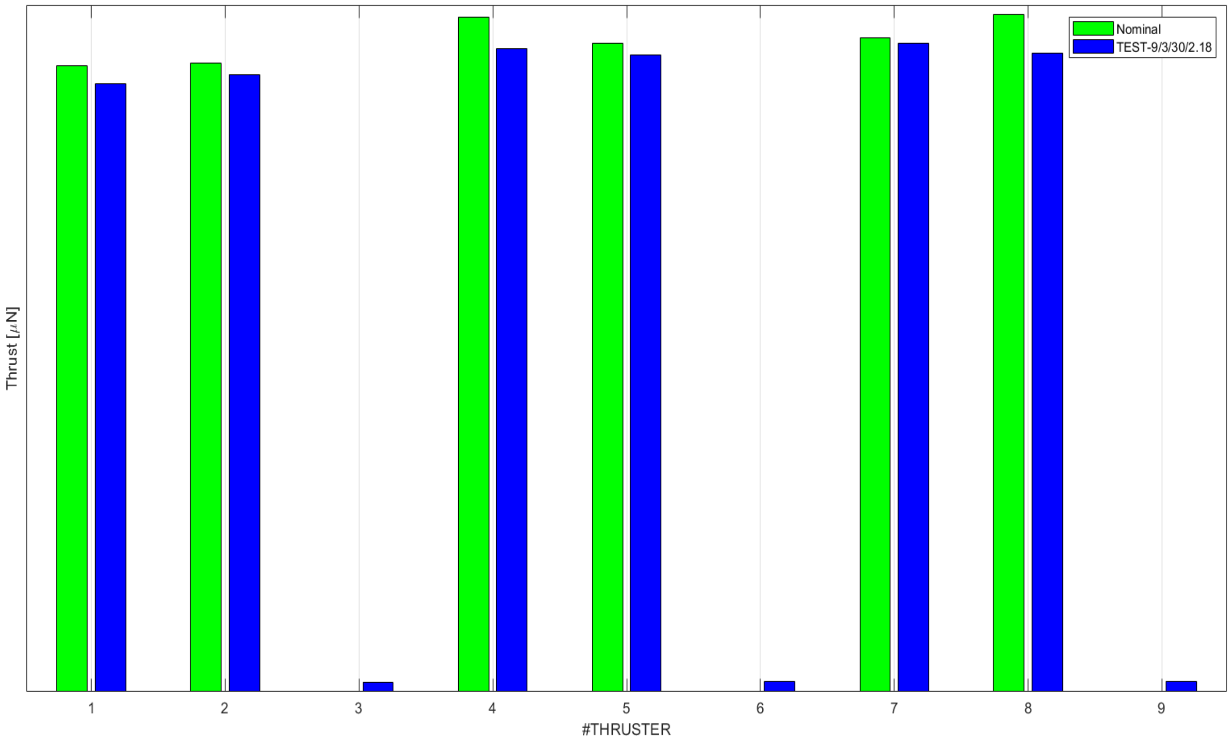

Figure 36: overall thrust for thruster – global solution (Matlab).

The Matlab analysis for the iteration 1 solution is reported in Figure 37.

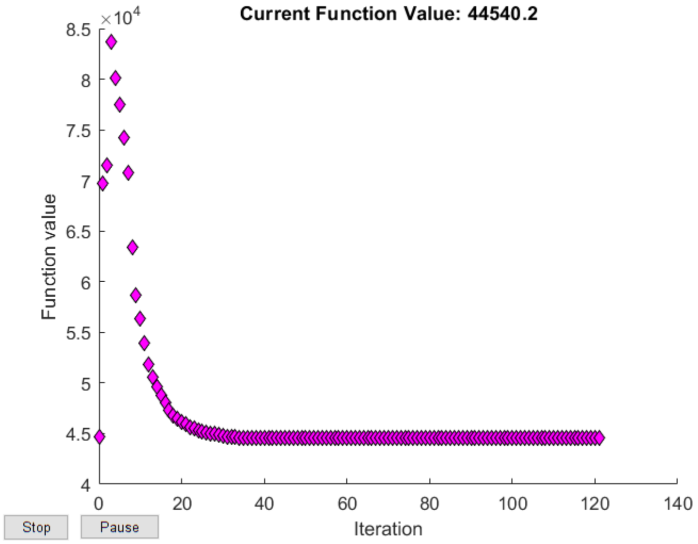

Figure 37: iteration 1 solution run.

The thruster orientation for the iteration 1 solution is shown in Figure 38. Figure 39 shows the comparison between the thrust trend of each thruster of the reference solution and of the iteration 1 solution over time. The comparison between the overall thrust exerted by each thruster of the reference solution and the iteration 1 solution is illustrated in Figure 40.

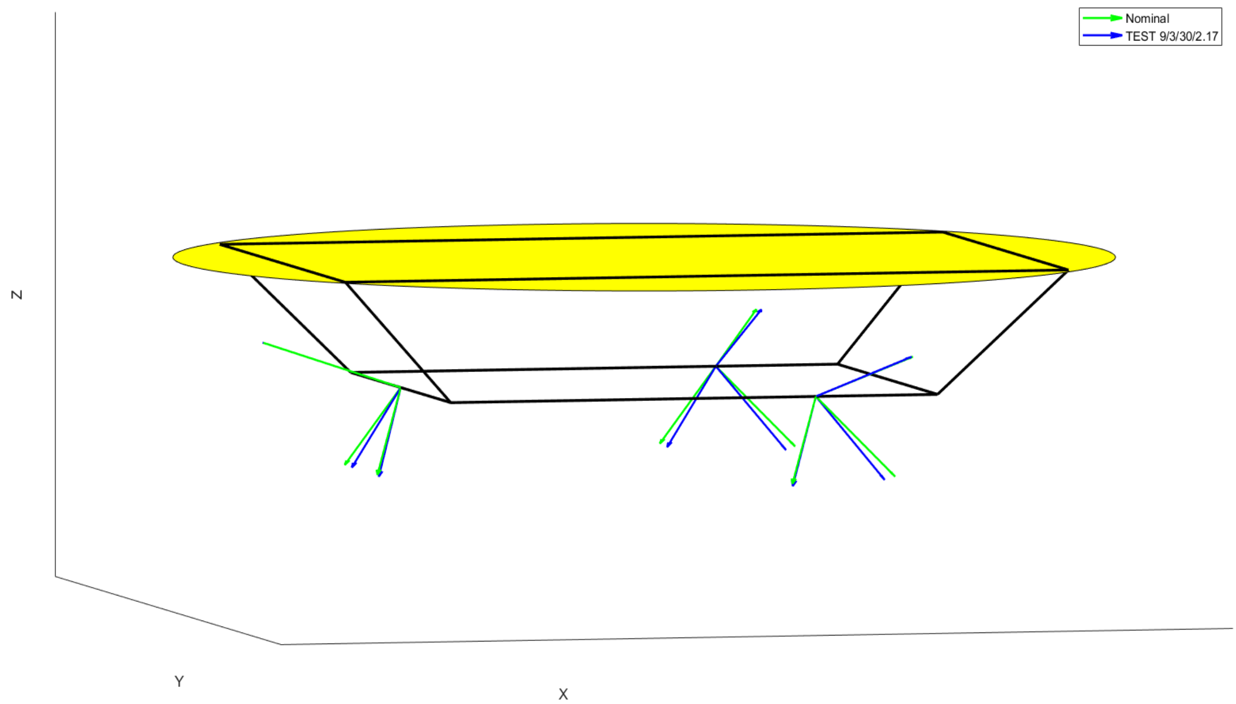

Figure 38: thruster orientation – iteration 1 solution (Matlab).

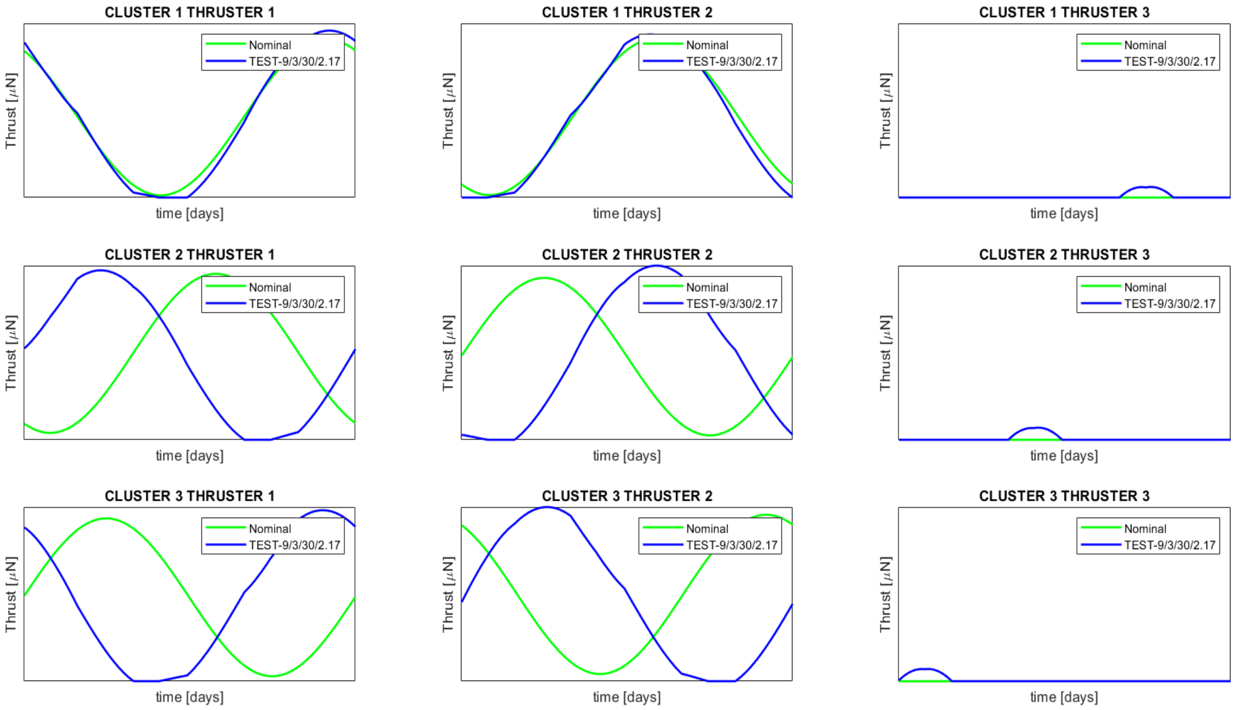

Figure 39: thrust over time – iteration 1 solution (Matlab).

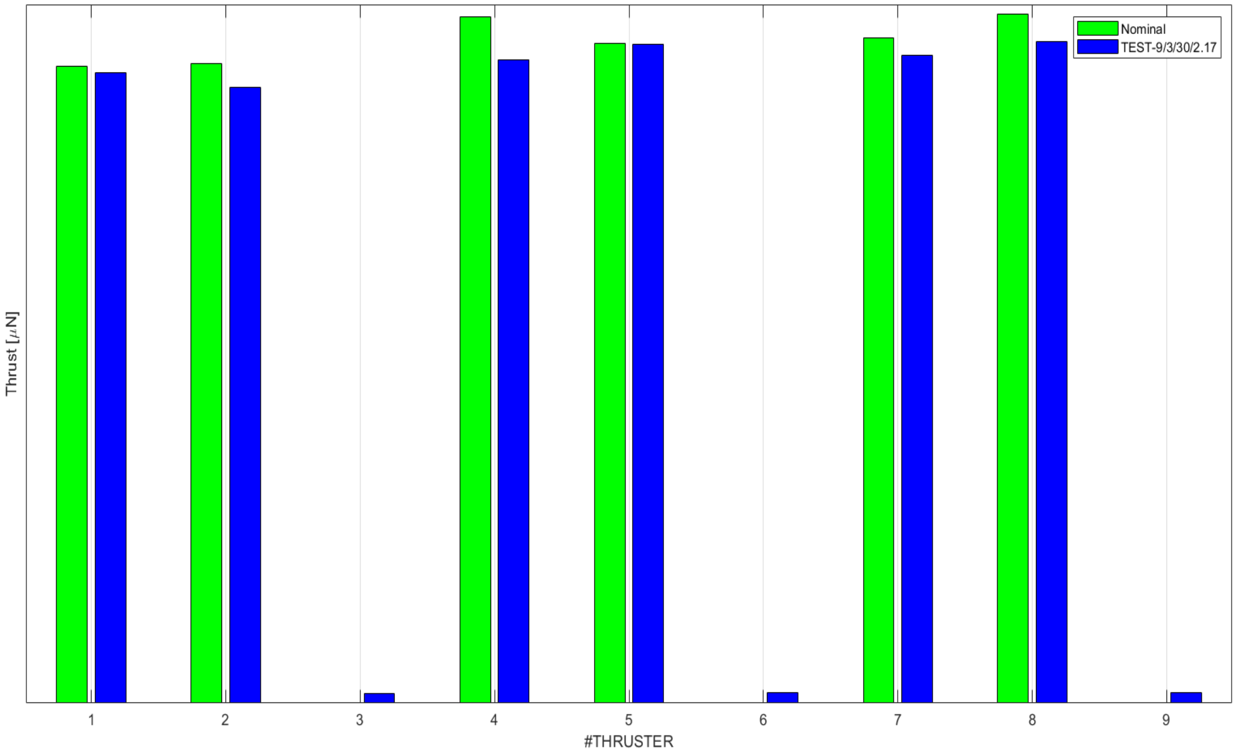

Figure 40: overall thrust for thruster – iteration 1 solution (Matlab).

The Matlab analysis for the iteration 2 solution is reported in Figure 41.

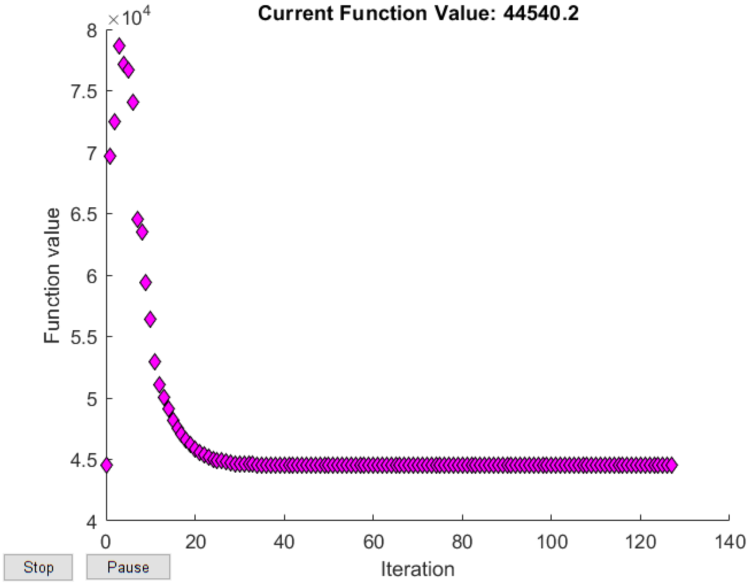

Figure 41: iteration 2 solution run.

The thruster orientation for the iteration 2 solution is shown in Figure 42. Figure 43 shows the comparison between the thrust trend of each thruster over time of the reference solution and of the iteration 2 solution. The comparison between the overall thrust exerted by each thruster of the reference solution and the iteration 2 solution is illustrated in Figure 44.

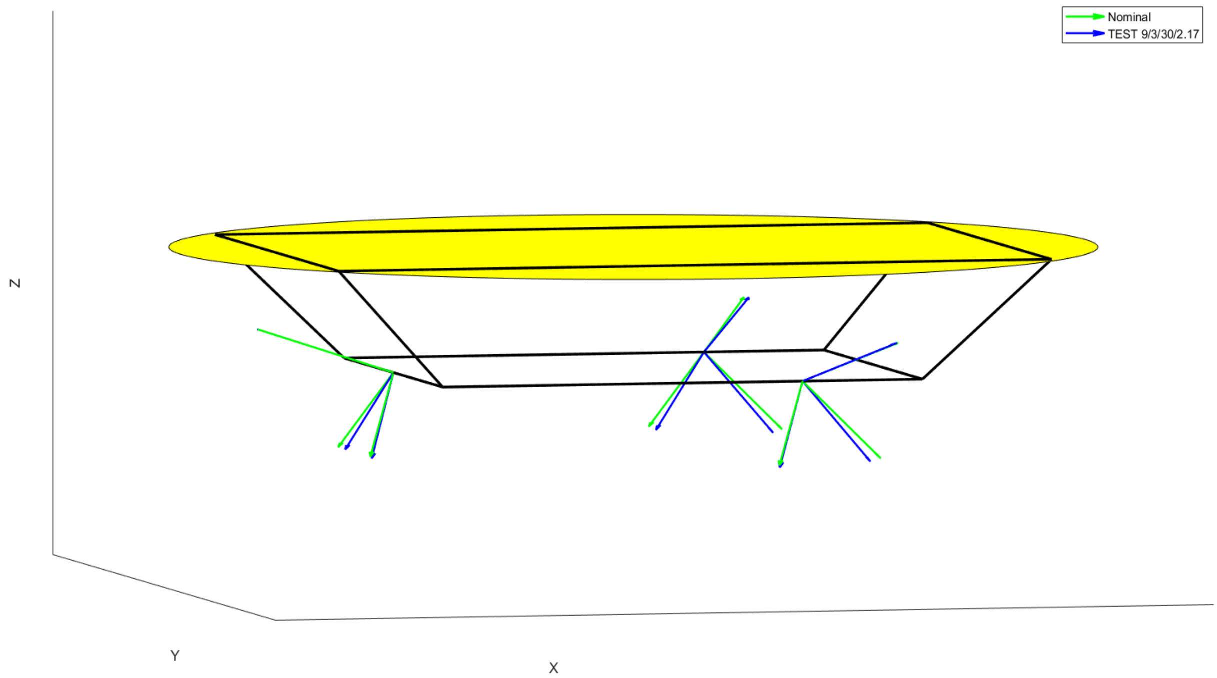

Figure 42: thruster orientation – iteration 2 solution (Matlab).

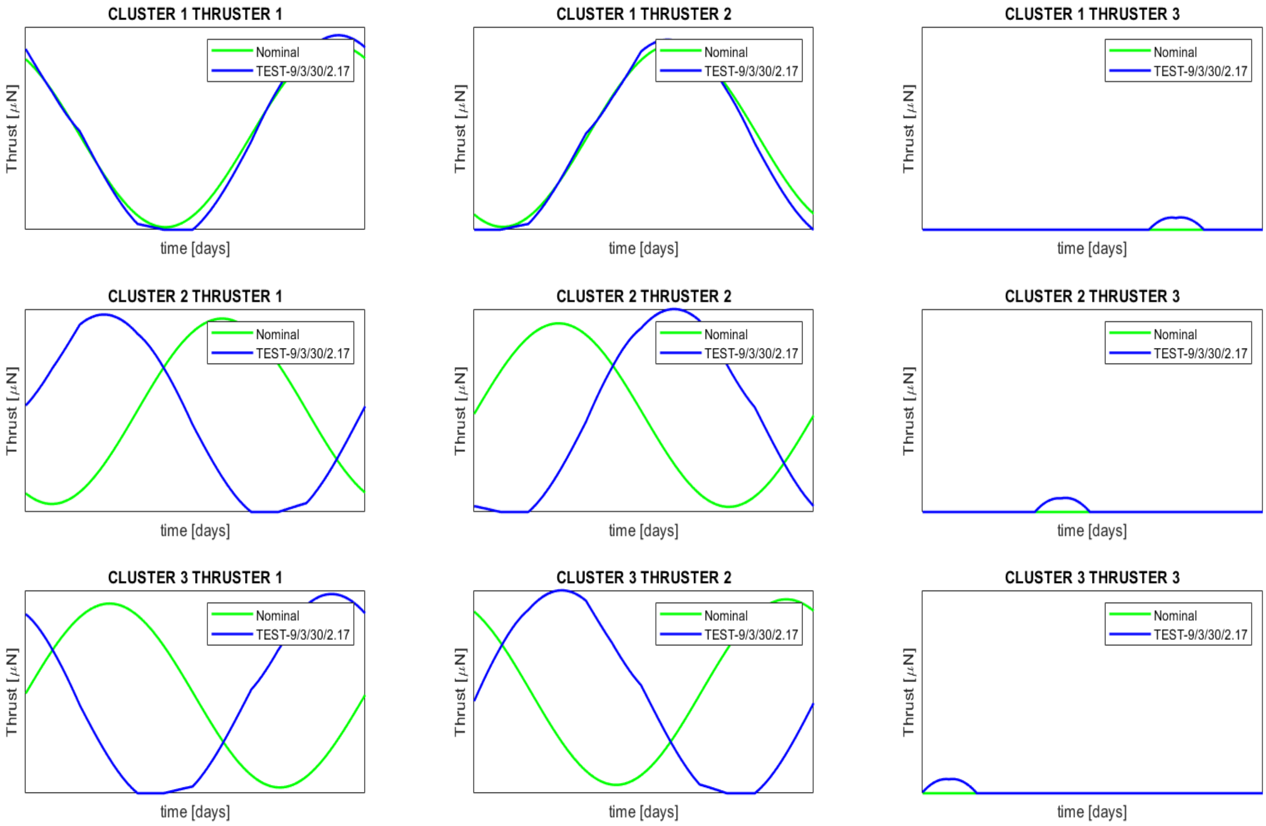

Figure 43: thrust over time – iteration 2 solution (Matlab).

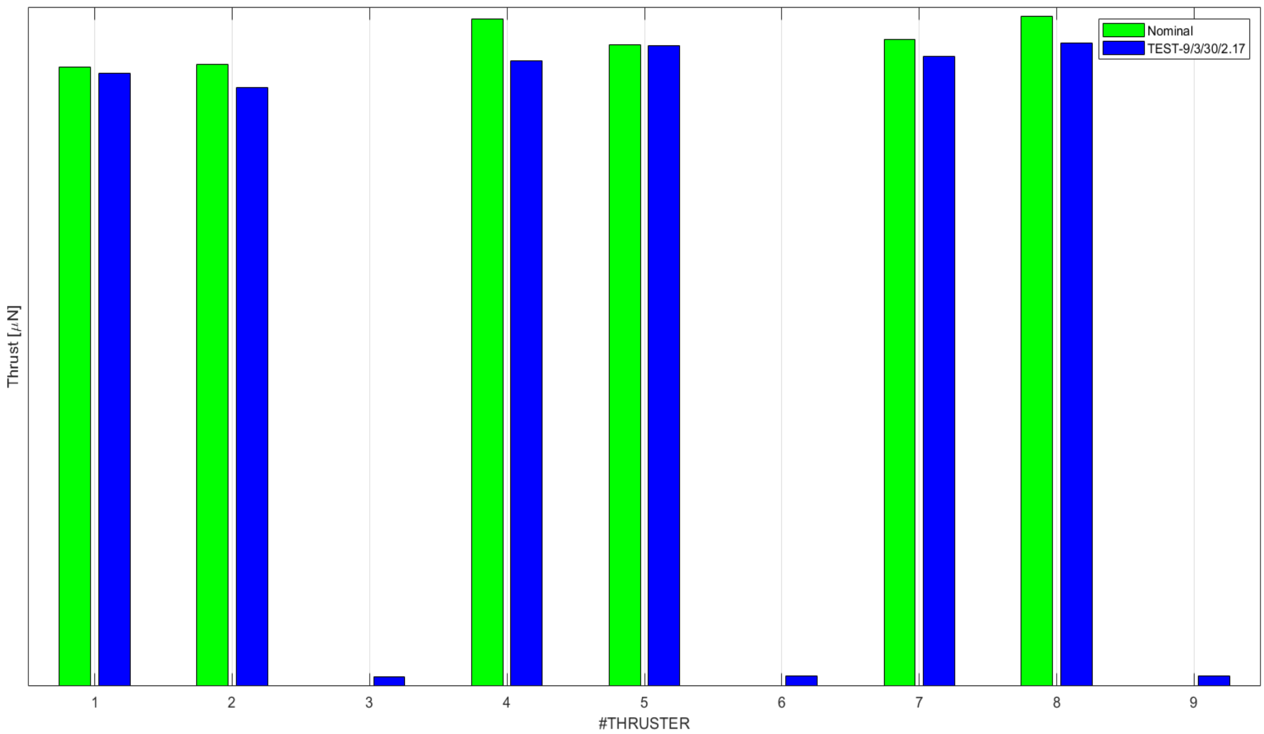

Figure 44: overall thrust for thruster – iteration 2 solution (Matlab).

The Matlab analysis for the iteration 3 solution is reported in Figure 45.

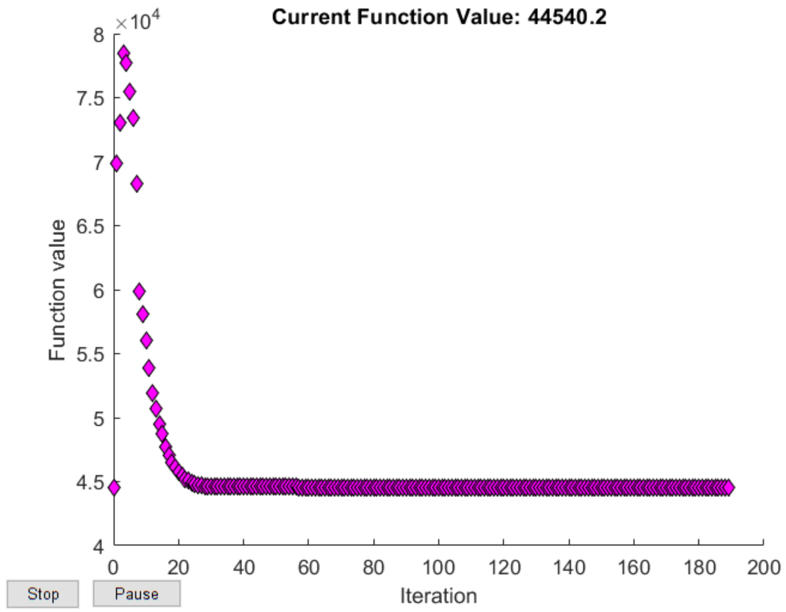

Figure 45: iteration 3 solution run.

The thruster orientation for the iteration 3 solution is shown in Figure 46. Figure 47 shows the comparison between the thrust trend of each thruster over time of the reference solution and of the iteration 3 solution. The comparison between the overall thrust exerted by each thruster of the reference solution and the iteration 3 solution is illustrated in Figure 48.

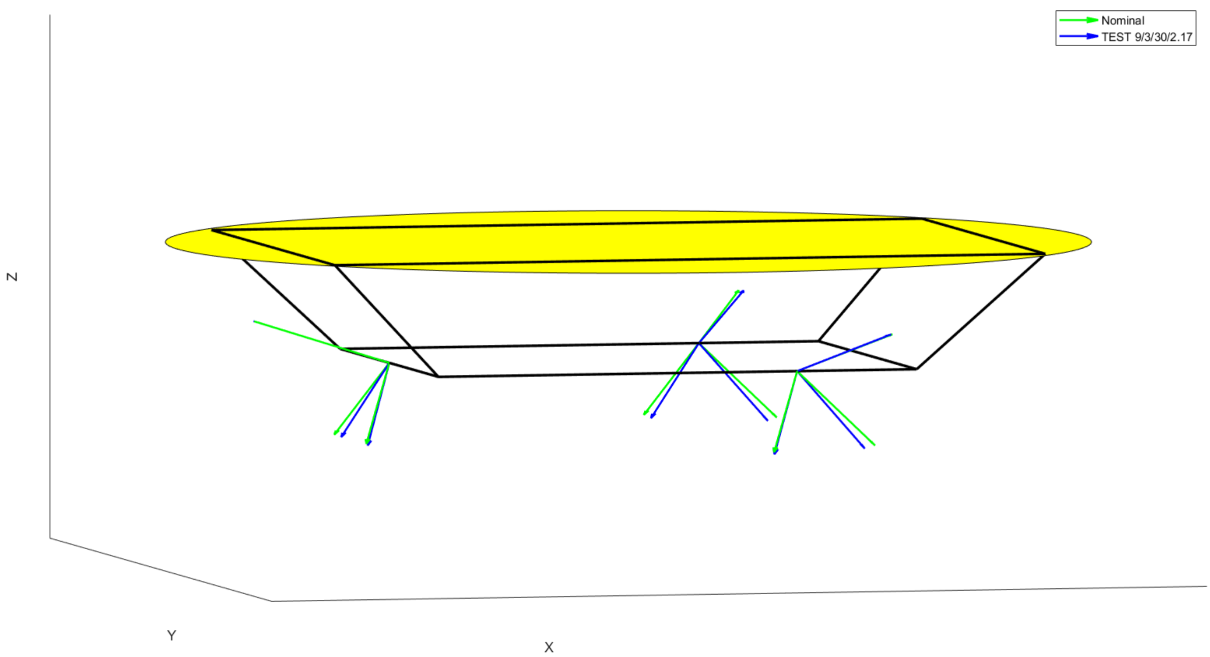

Figure 46: thruster orientation – iteration 3 solution (Matlab).

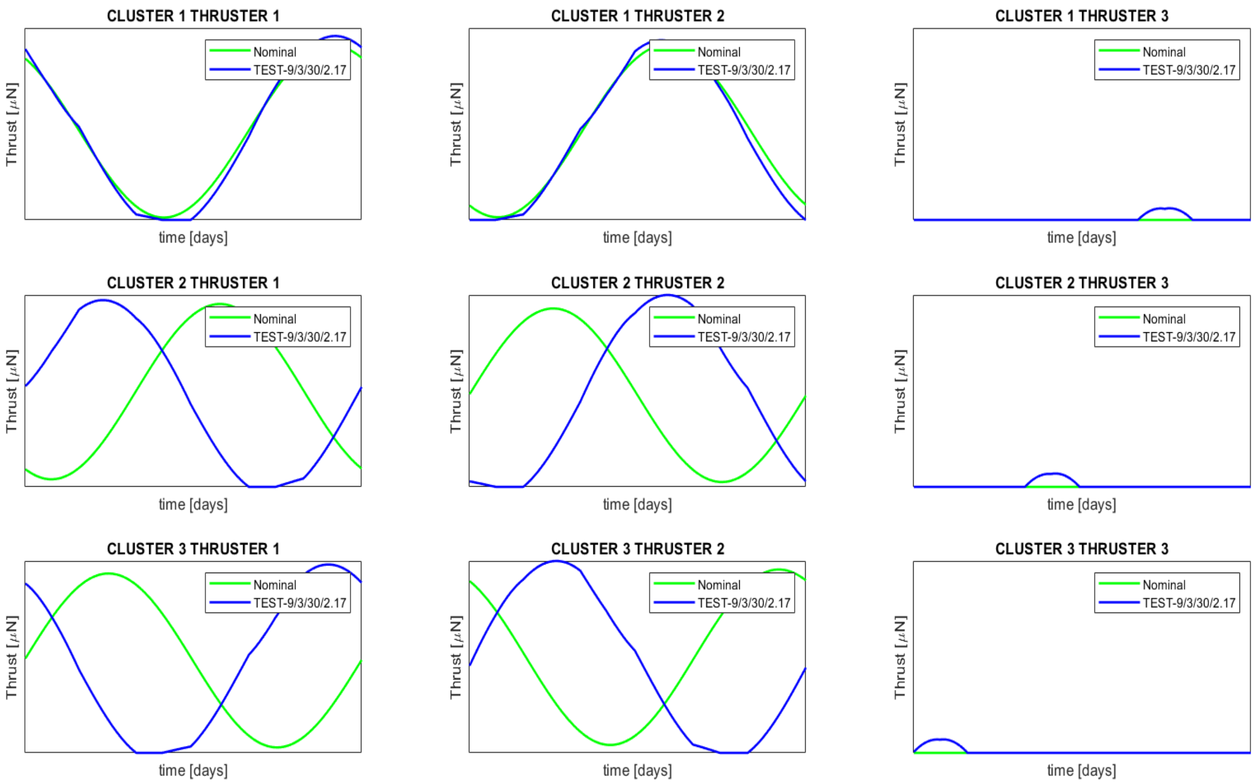

Figure 47: thrust over time – iteration 3 solution (Matlab).

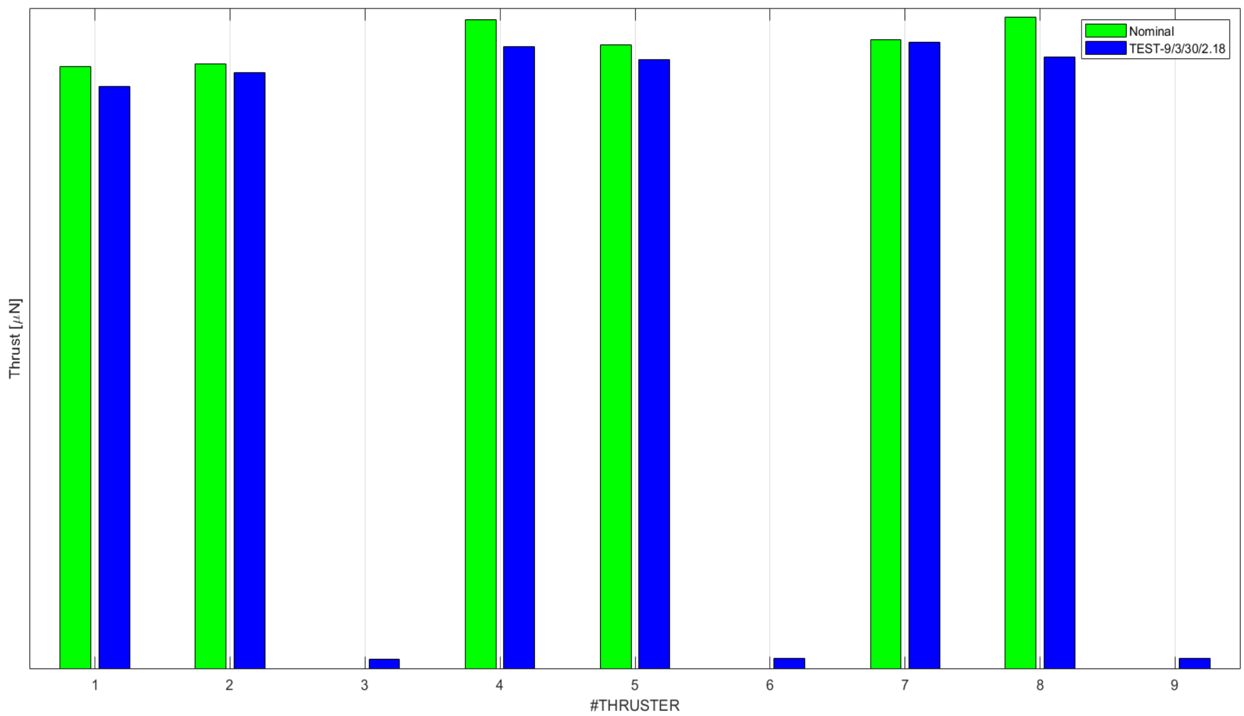

Figure 48: overall thrust for thruster – iteration 3 solution (Matlab).

5.2 SIX-THRUSTERS SCENARIO FOR DFACS

In the following table (Table 4) the results of the analyses made with CPLEX for the thrusters of DFACS for the scenario with six thrusters are summarized.

CPLEX ANALYSES			
Solution	Fuel consumption		Gain
	Subset of instants	Whole set of instants	Whole set of instants
Global	100%	100%	0%
Iteration 1	-	99.21%	0.79%
Iteration 2	-	98.96%	1.04%
Iteration 3	-	98.77%	1.23%
Iteration 4	-	98.72%	1.28%
Iteration 5	-	98.70%	1.30%

Table 4: fuel consumption of DFACS scenarios with six thrusters obtained with CPLEX.

Note that in this case all the runs of the iterations have been performed with the discretized model considering only the whole set of instants.

The thrusters' orientation for the global solution is shown in Figure 49. Figure 50 shows the comparison between the thrust trend of each thruster over time of the reference solution and of the global solution. The comparison between the overall thrust exerted by each thruster of the reference solution and the global solution is reported in Figure 51.

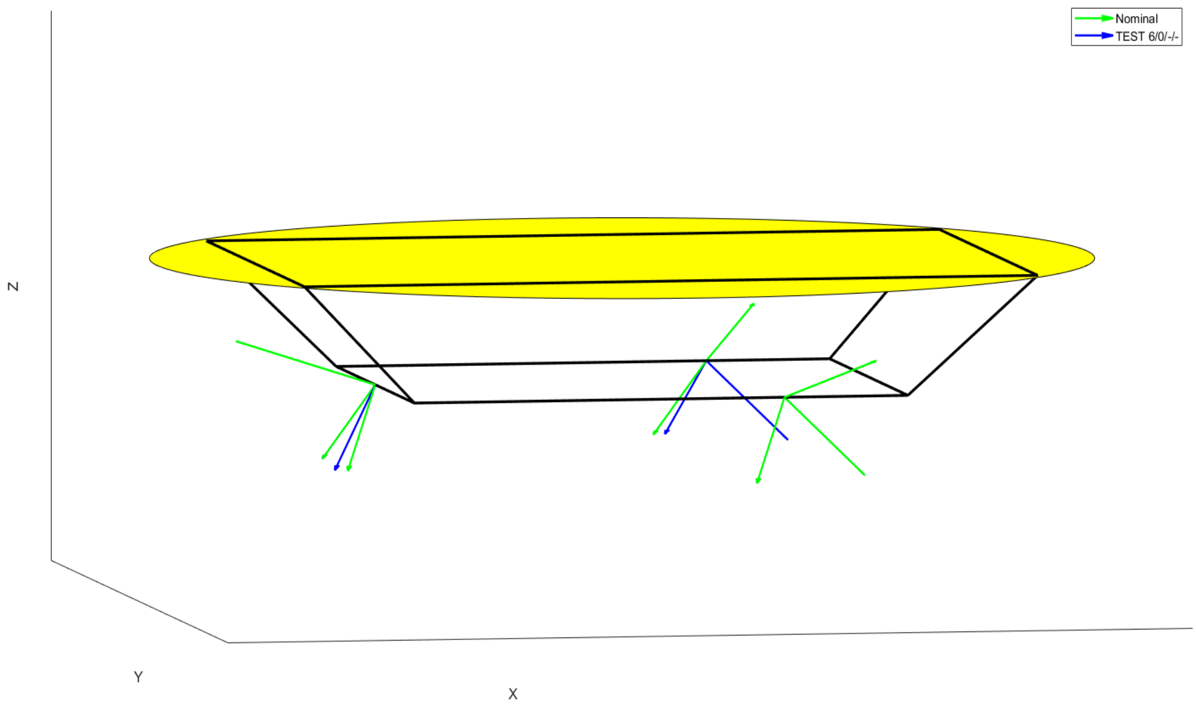

Figure 49: thruster orientation – global solution.

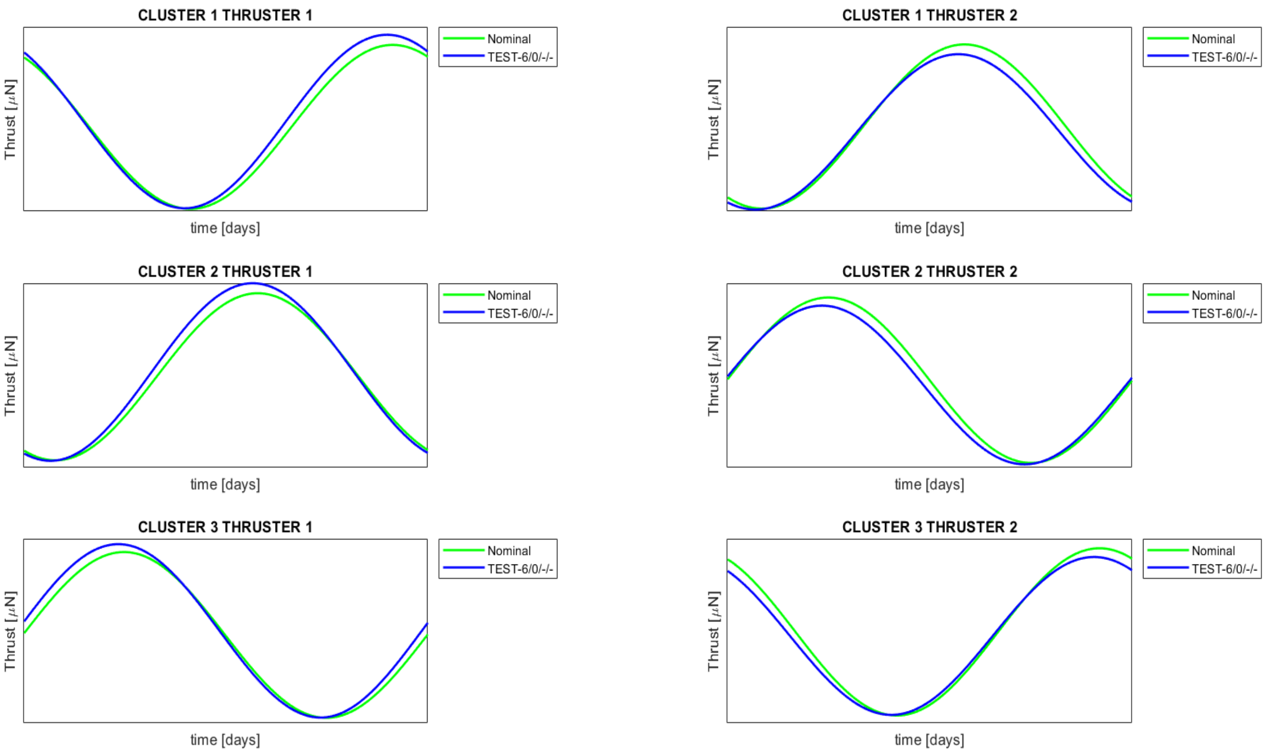

Figure 50: thrust over time – global solution.

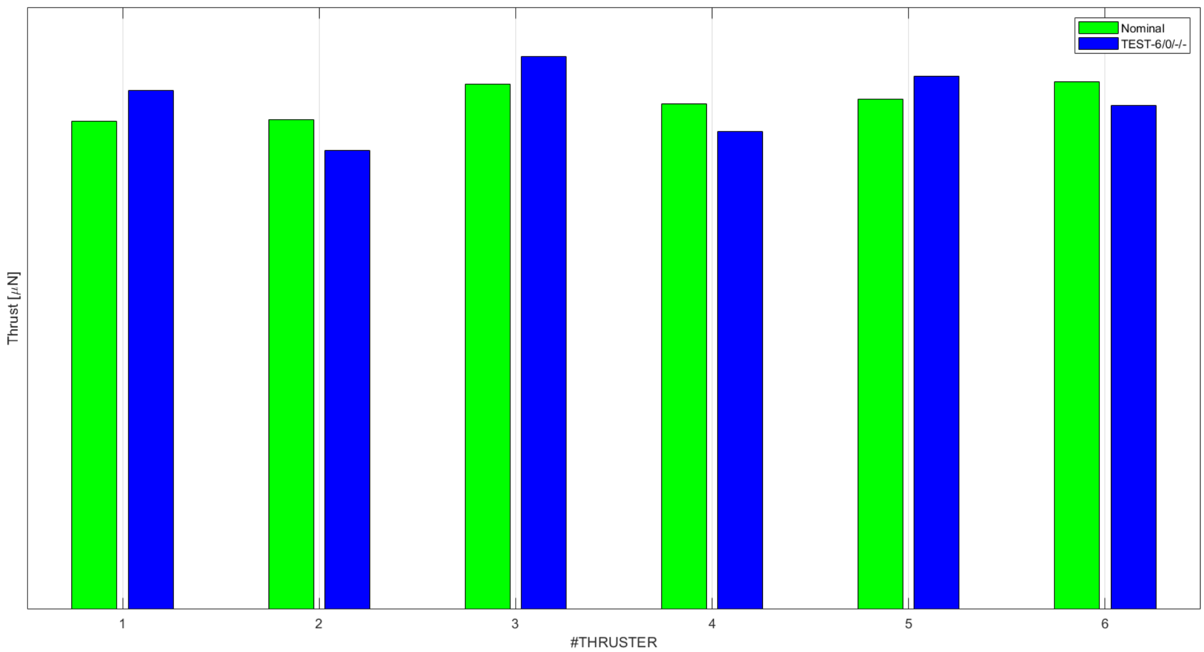

Figure 51: overall thrust for thruster – global solution.

The thruster orientation for the iteration 1 solution is shown in Figure 52. Figure 53 shows the comparison between the thrust trend of each thruster over time of the reference solution and of the iteration 1 solution. The comparison between the overall thrust exerted by each thruster of the reference solution and the iteration 1 solution is illustrated in Figure 54.

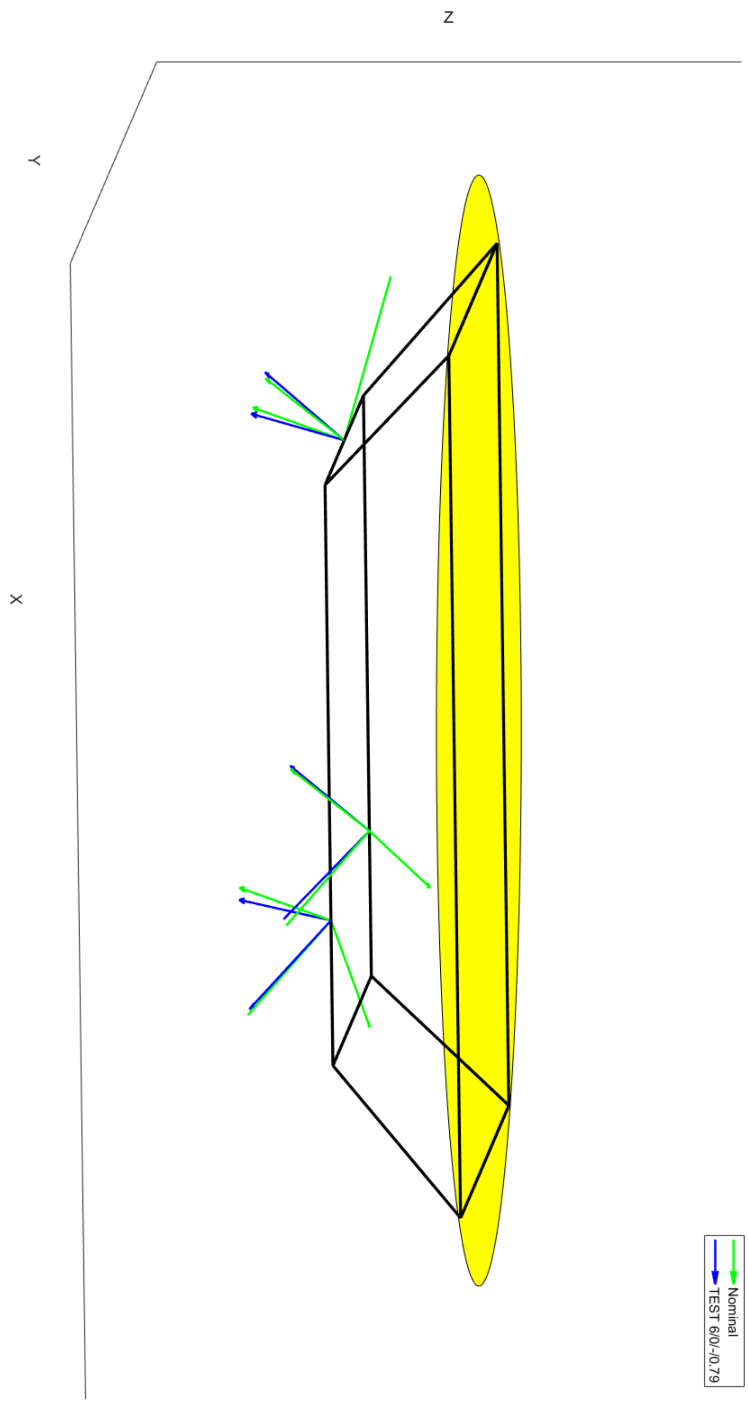

Figure 52: thruster orientation – iteration 1 solution.

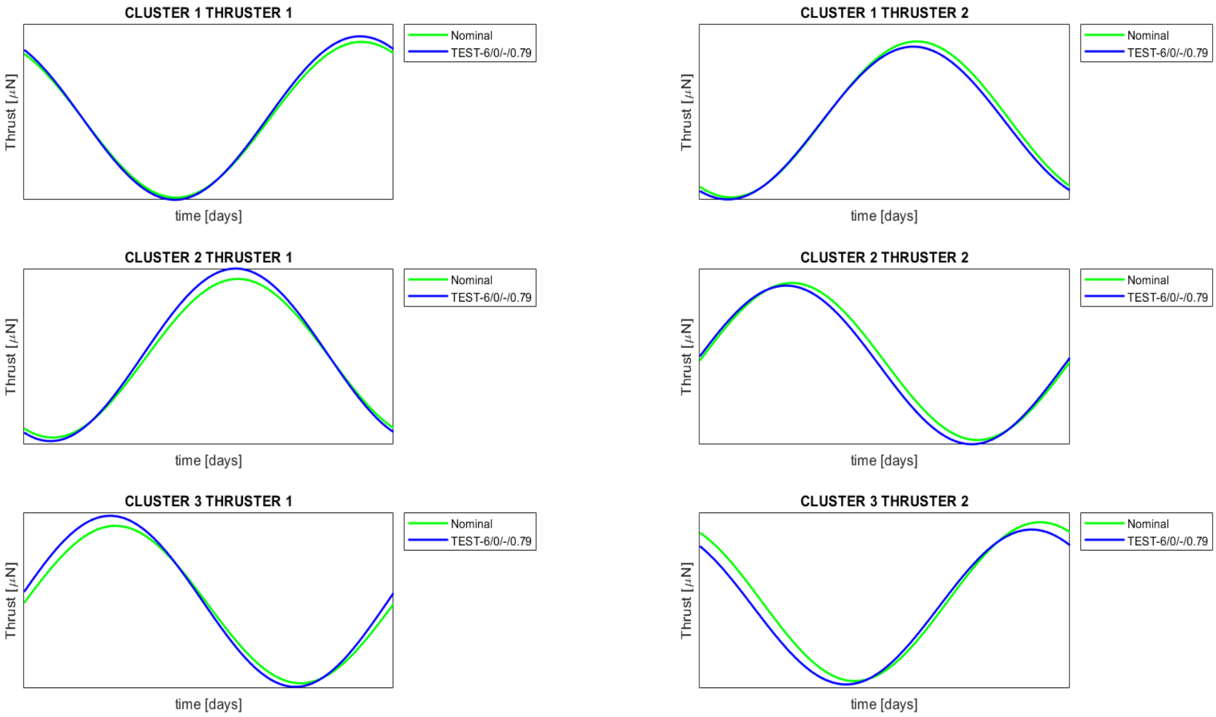

Figure 53: thrust over time – iteration 1 solution.

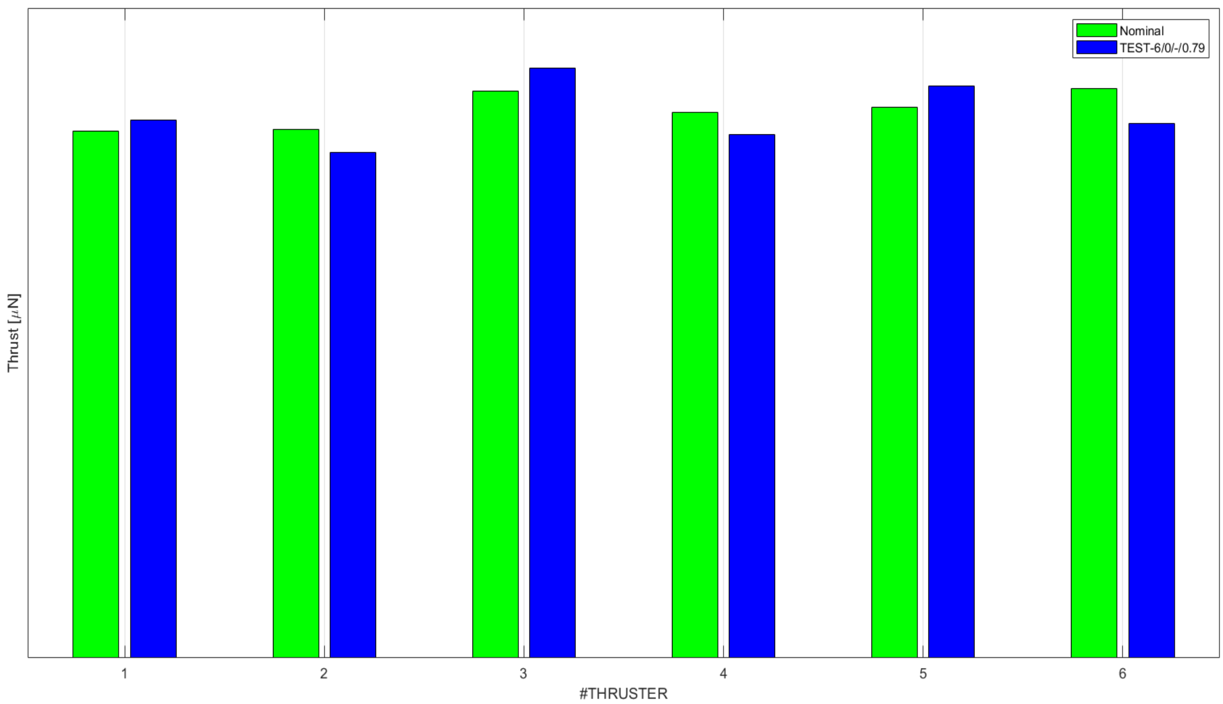

Figure 54: overall thrust for thruster – iteration 1 solution.

The thruster orientation for the iteration 2 solution is shown in Figure 55. Figure 56 shows the comparison between the thrust trend of each thruster over time of the reference solution and of the iteration 2 solution. The comparison between the overall thrust exerted by each thruster of the reference solution and the iteration 2 solution is illustrated in Figure 57.

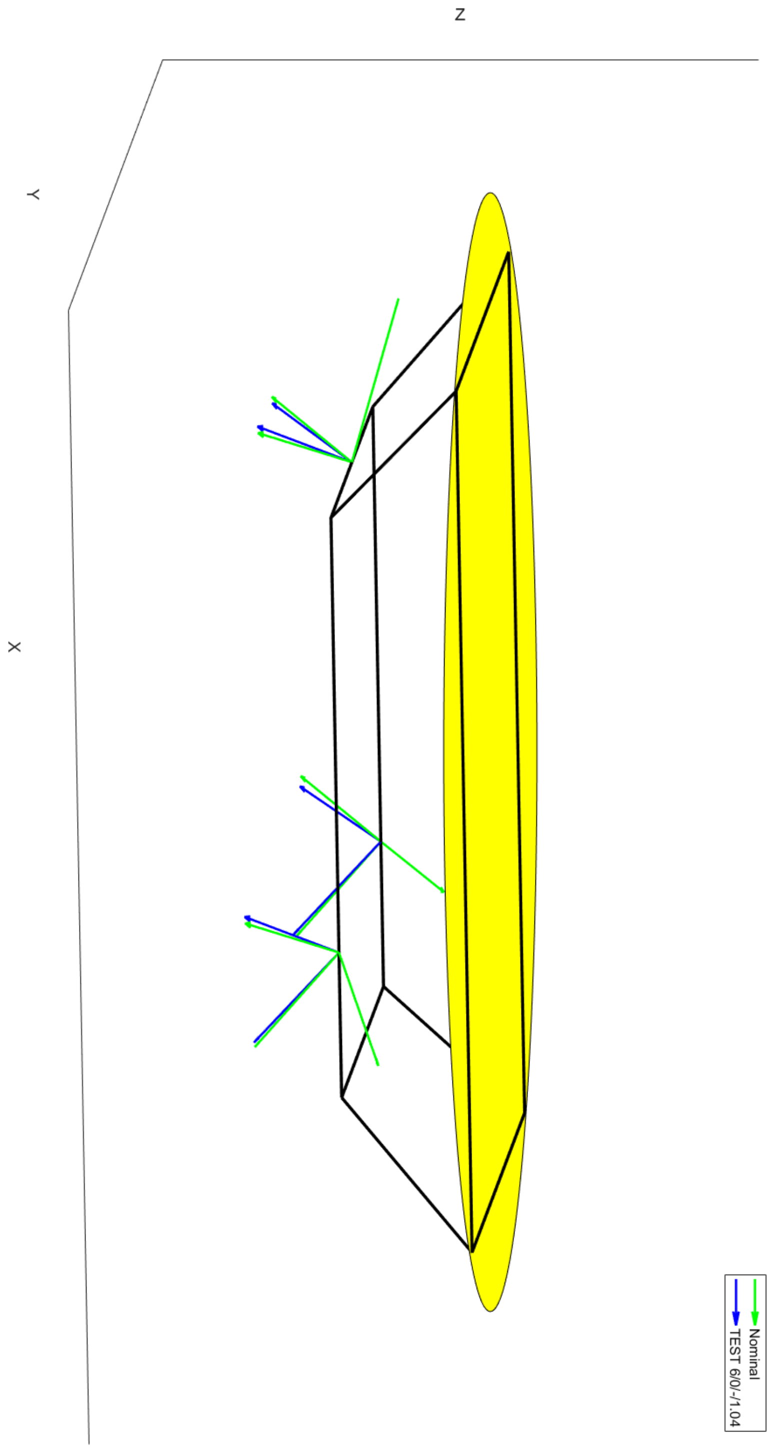

Figure 55: thruster orientation – iteration 2 solution.

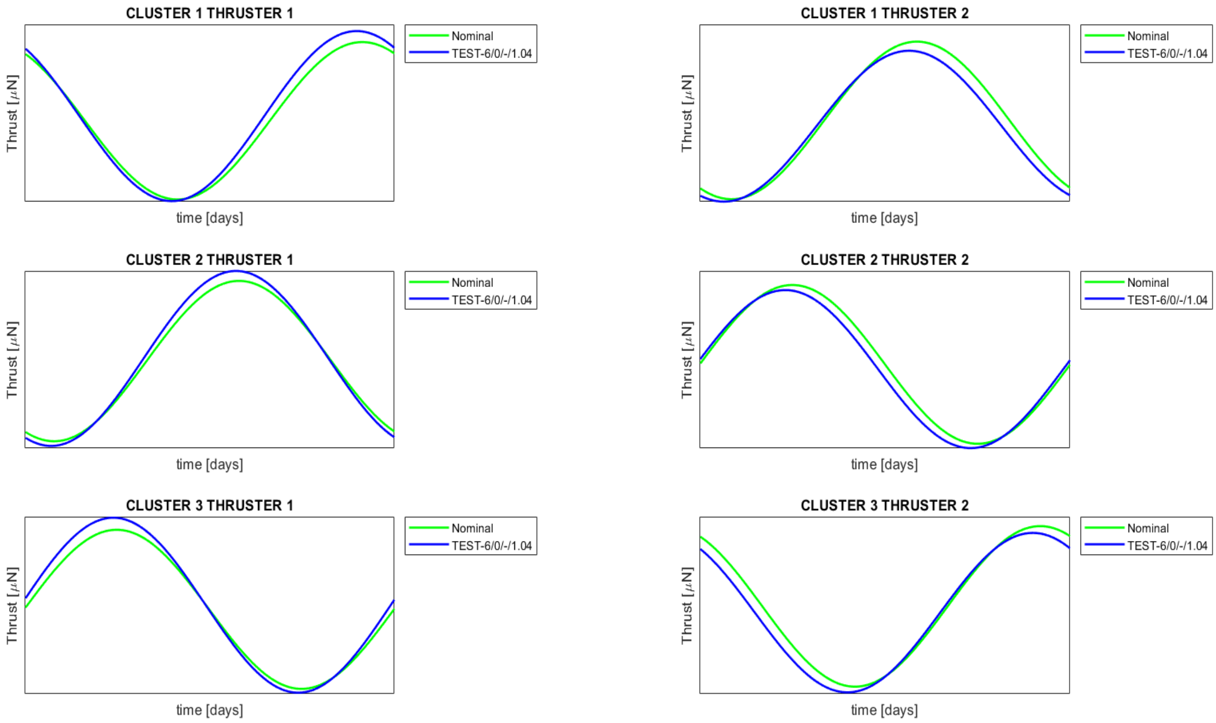

Figure 56: thrust over time – iteration 2 solution.

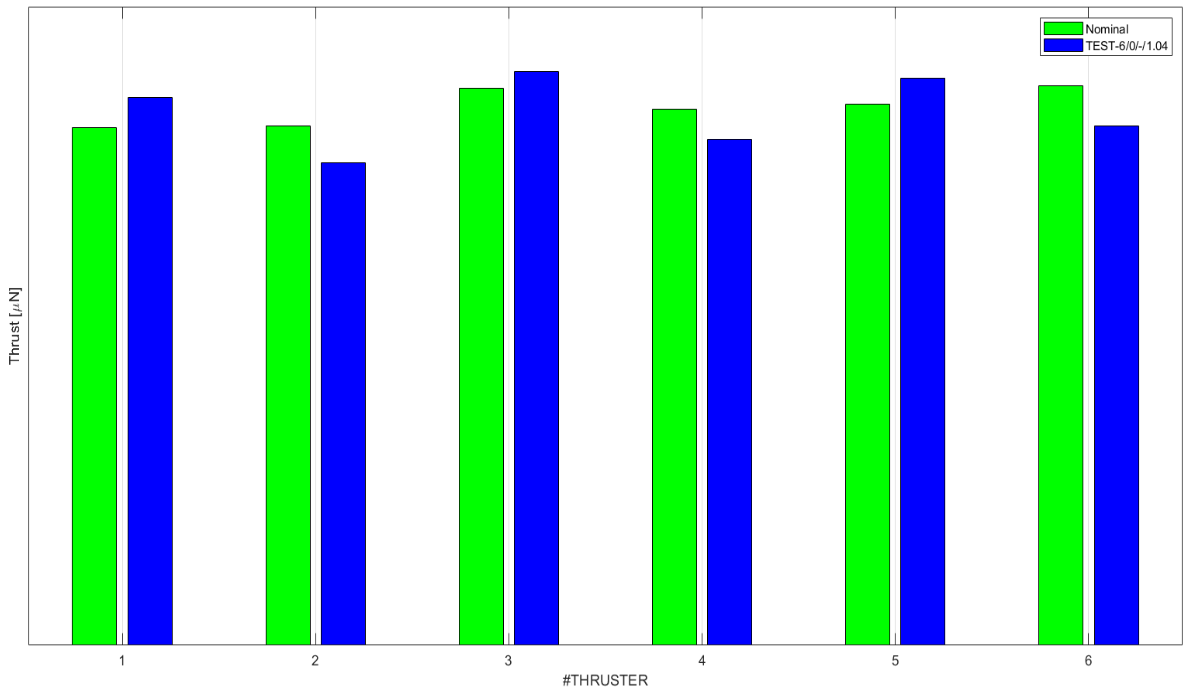

Figure 57: overall thrust for thruster – iteration 2 solution.

The thruster orientation for the iteration 3 solution is shown in Figure 58. Figure 59 shows the comparison between the thrust trend of each thruster over time of the reference solution and of the iteration 3 solution. The comparison between the overall thrust exerted by each thruster of the reference solution and the iteration 3 solution is illustrated in Figure 60.

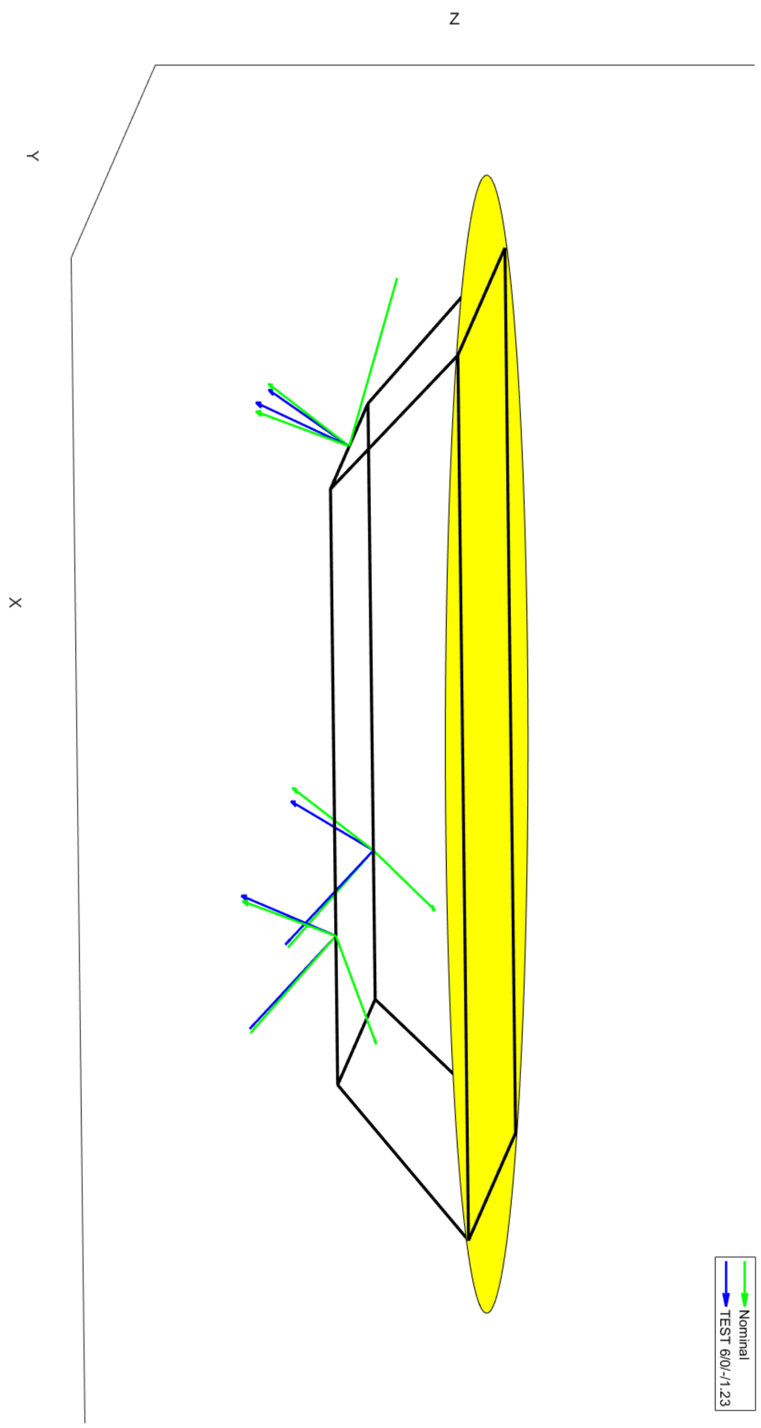

Figure 58: thruster orientation – iteration 3 solution.

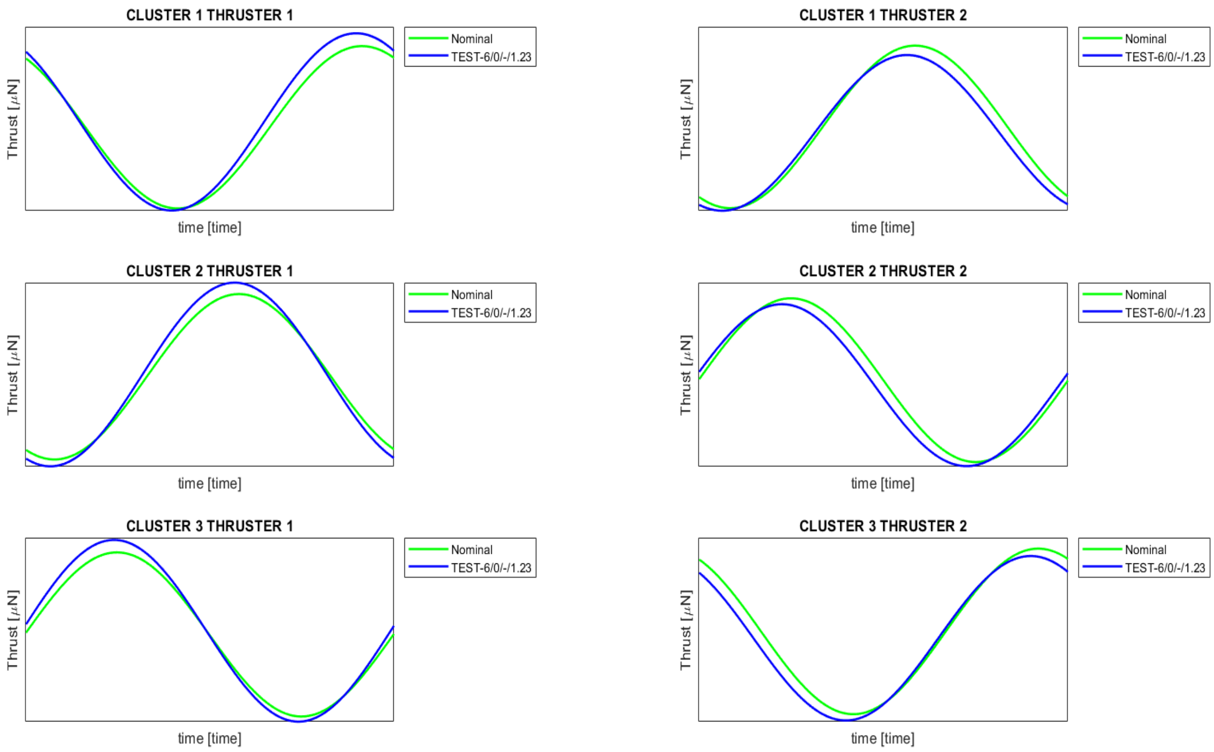

Figure 59: thrust over time – iteration 3 solution.

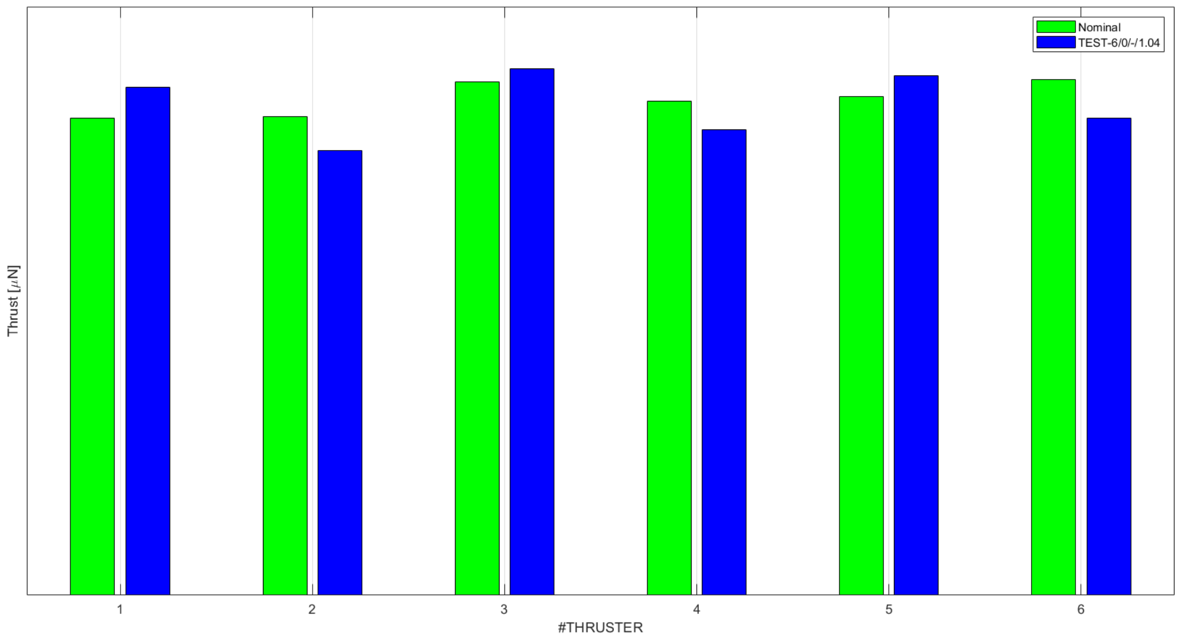

Figure 60: overall thrust for thruster – iteration 3 solution.

The thruster orientation for the iteration 4 solution is shown in Figure 61. Figure 62 shows the comparison between the thrust trend of each thruster over time of the reference solution and of the iteration 4 solution. The comparison between the overall thrust exerted by each thruster of the reference solution and the iteration 4 solution is reported in Figure 63.

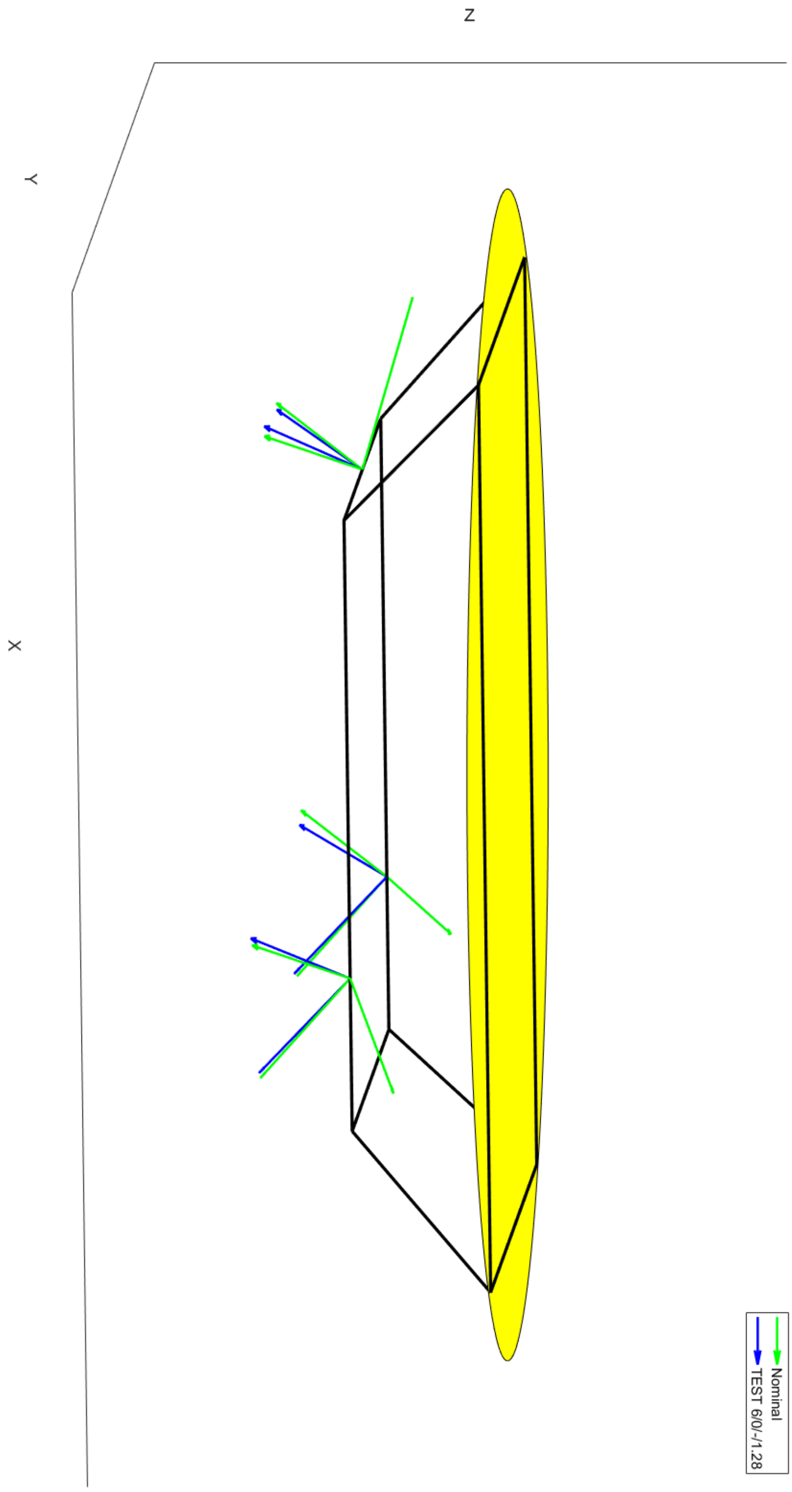

Figure 61: thruster orientation – iteration 4 solution.

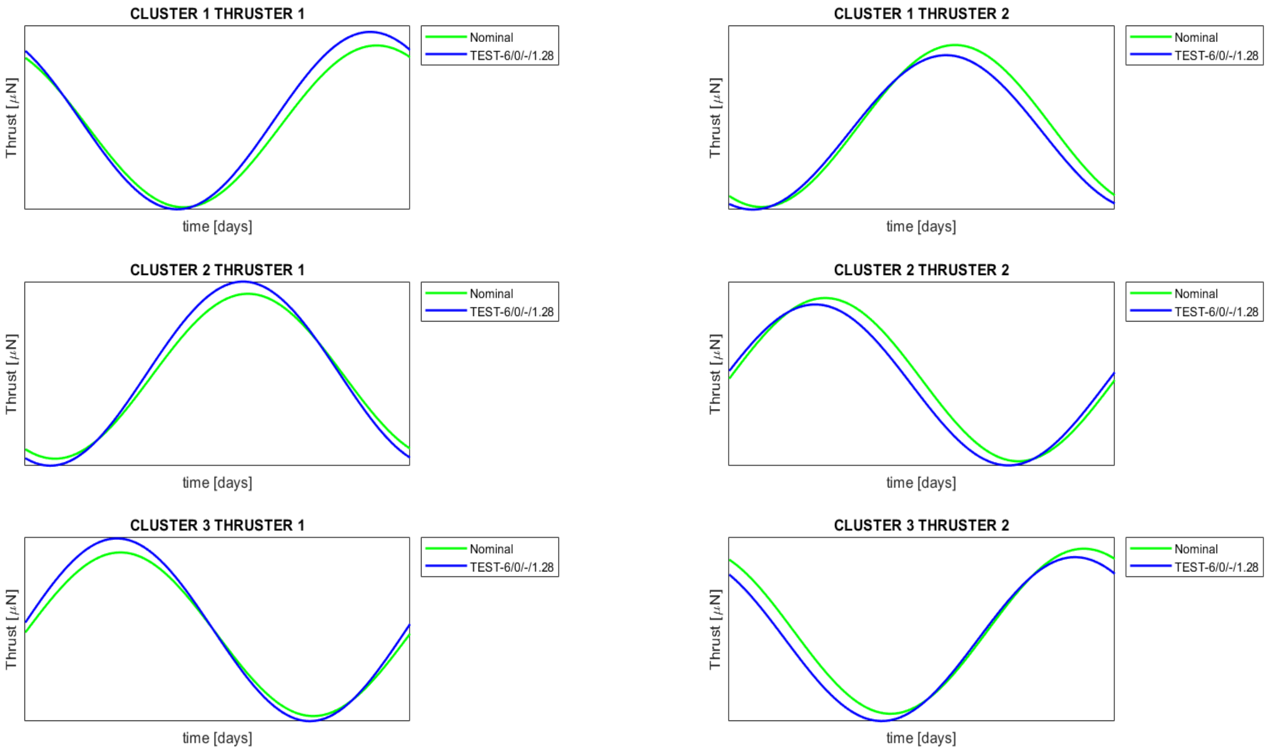

Figure 62: thrust over time – iteration 4 solution.

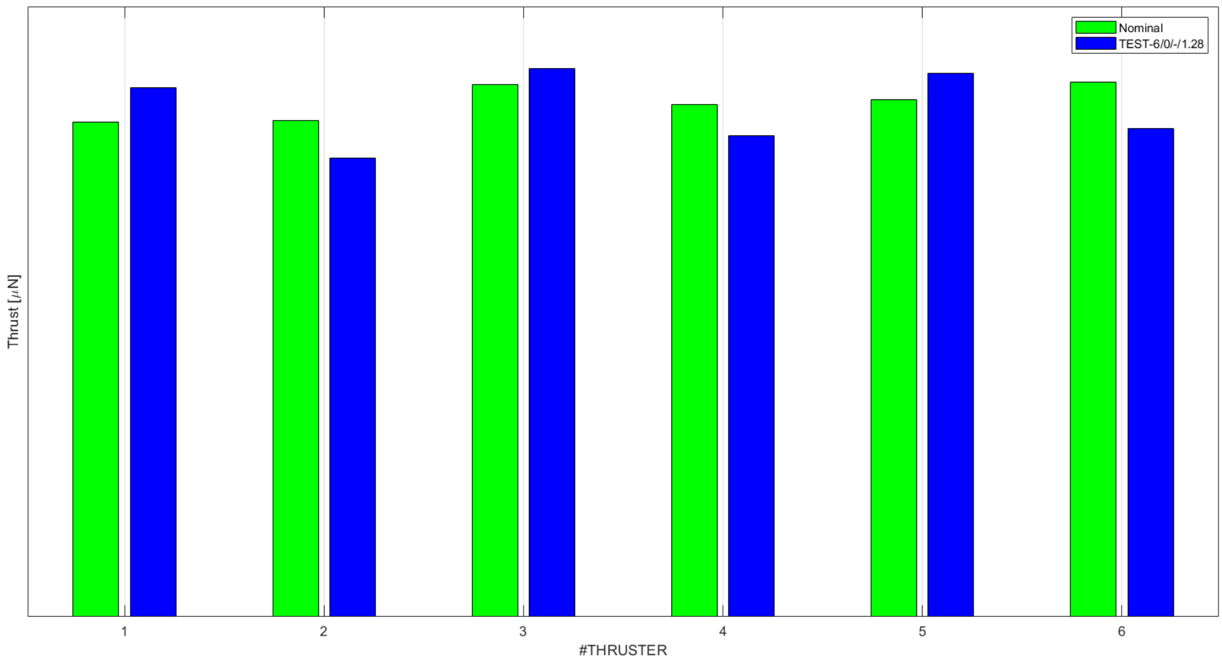

Figure 63: overall thrust for thruster – iteration 4 solution.

The thruster orientation for the iteration 5 solution is shown in Figure 64. Figure 65 shows the comparison between the thrust trend of each thruster over time of the reference solution and of the iteration 5 solution. The comparison between the overall thrust exerted by each thruster of the reference solution and the iteration 5 solution is illustrated in Figure 66.

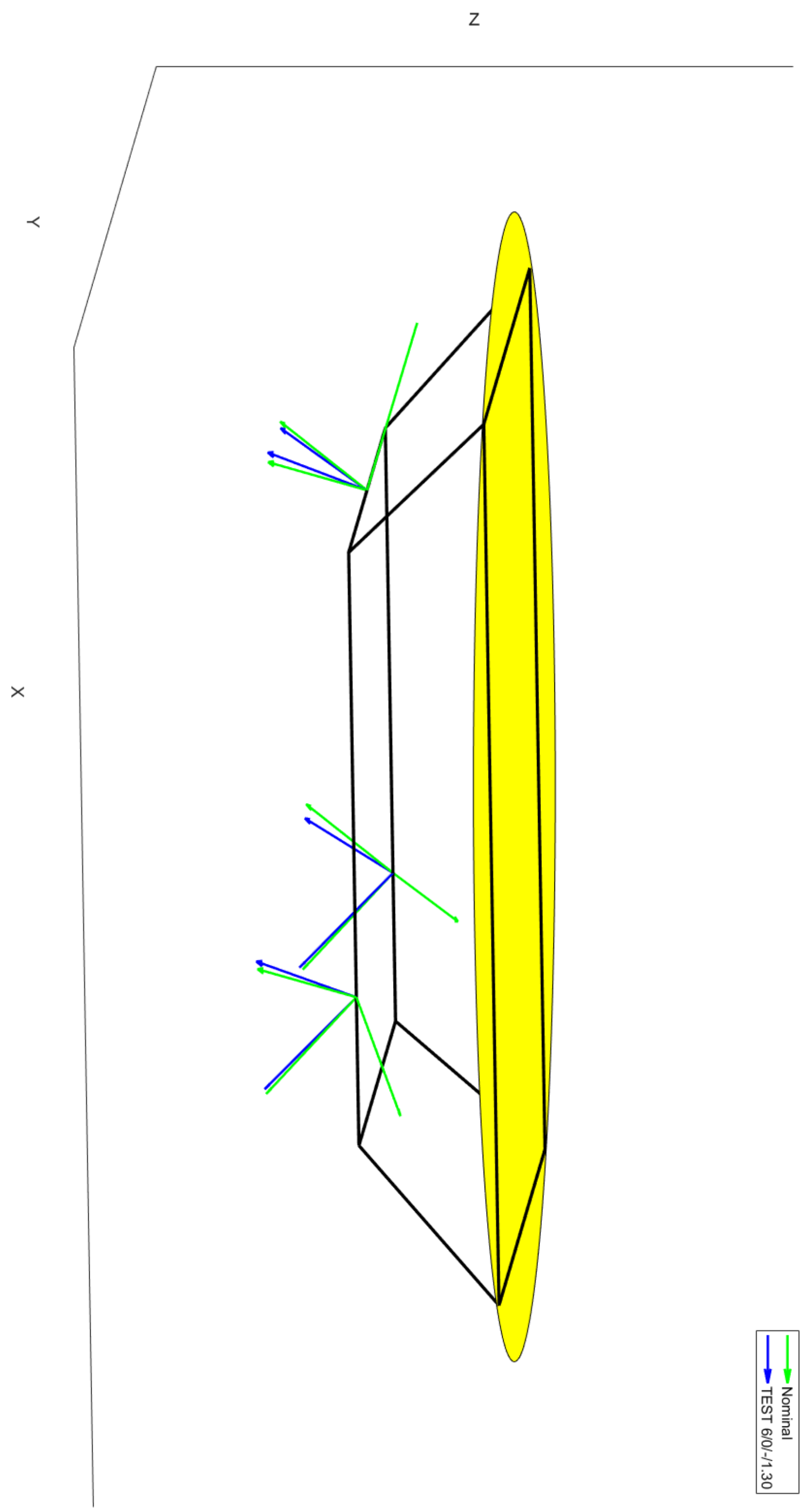

Figure 64: thruster orientation – iteration 5 solution.

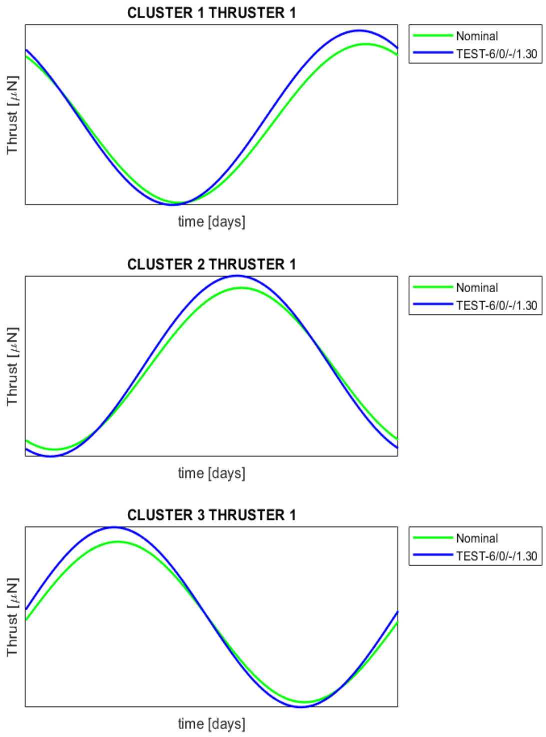

Figure 65: thrust over time – iteration 5 solution.

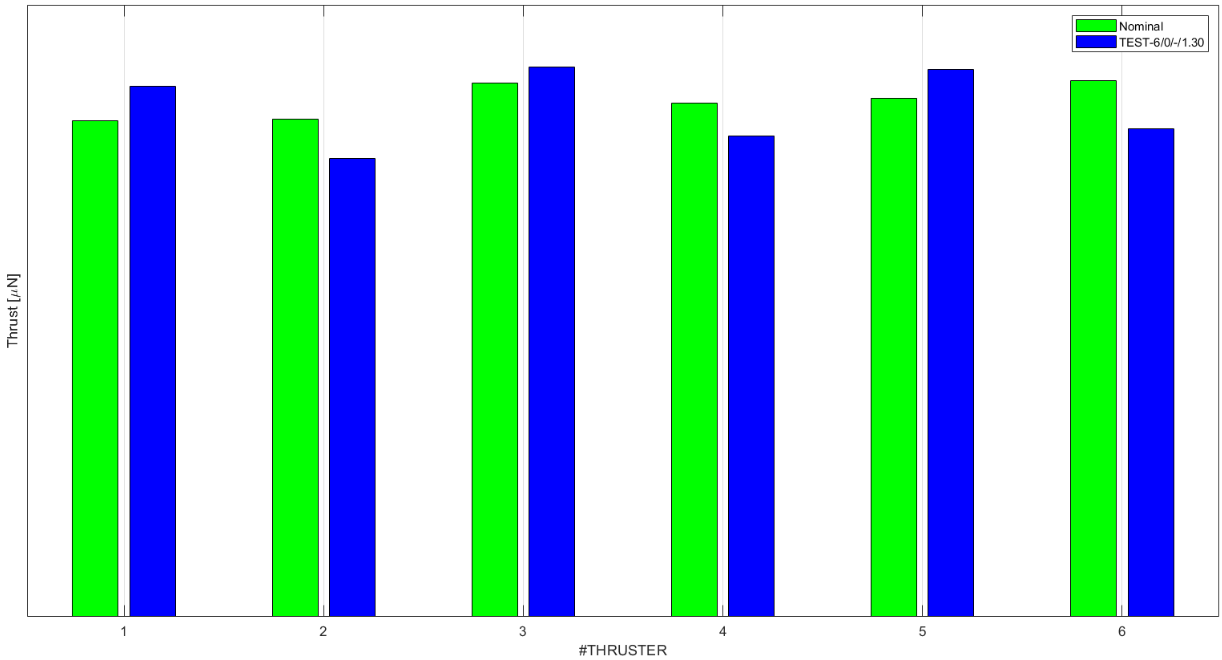

Figure 66: overall thrust for thruster – iteration 5 solution.

In the following table the results of the analyses made with MATLAB for the thrusters of DFACS for the scenario with six thrusters are summarized.

MATLAB ANALYSES		
Solution	Fuel consumption	Gain
	Whole set of instants	Whole set of instants
Global	102.14%	-
Iteration 1	98.63%	1.37%
Iteration 2	104.66%	-
Iteration 3	102.09%	-
Iteration 4	-	-
Iteration 5	-	-

Table 5: fuel consumption of DFACS scenarios with six thrusters obtained with Matlab.

The only solution which allows a gain in fuel consumption is iteration 1. For this reason more details on thrusters' orientation and thrust are given only for this solution. The global solution, the iteration 2 solution and the iteration 3 solution lead to an increase in fuel consumption, while the optimization with Matlab does not even start for the iteration 4 and iteration 5 solutions.

The Matlab analysis for the global solution is reported in Figure 67.

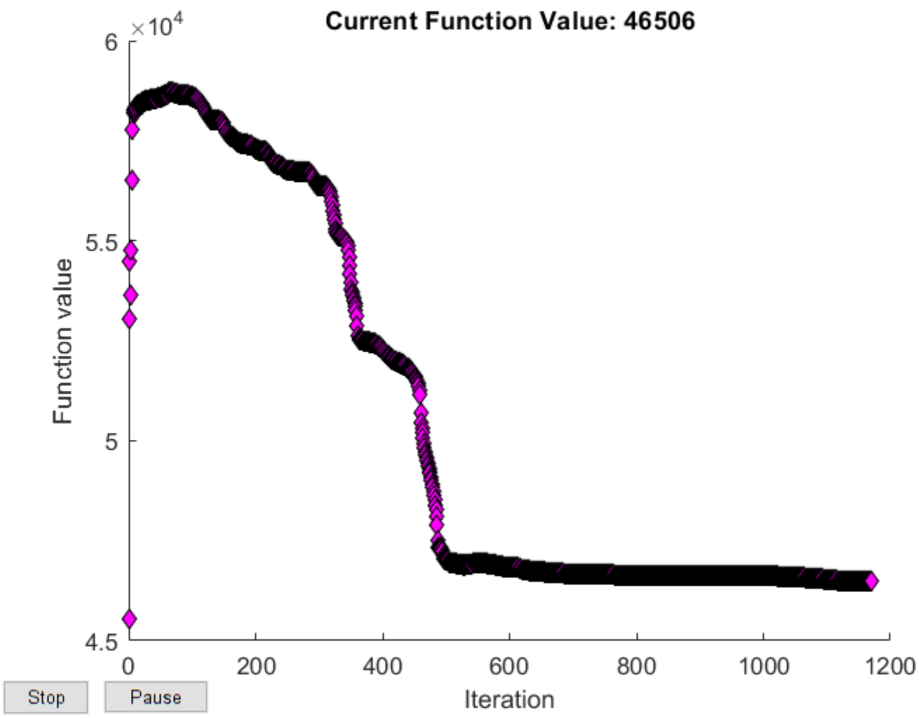

Figure 67: global solution run.

The Matlab analysis for the iteration 1 solution is reported in Figure 68.

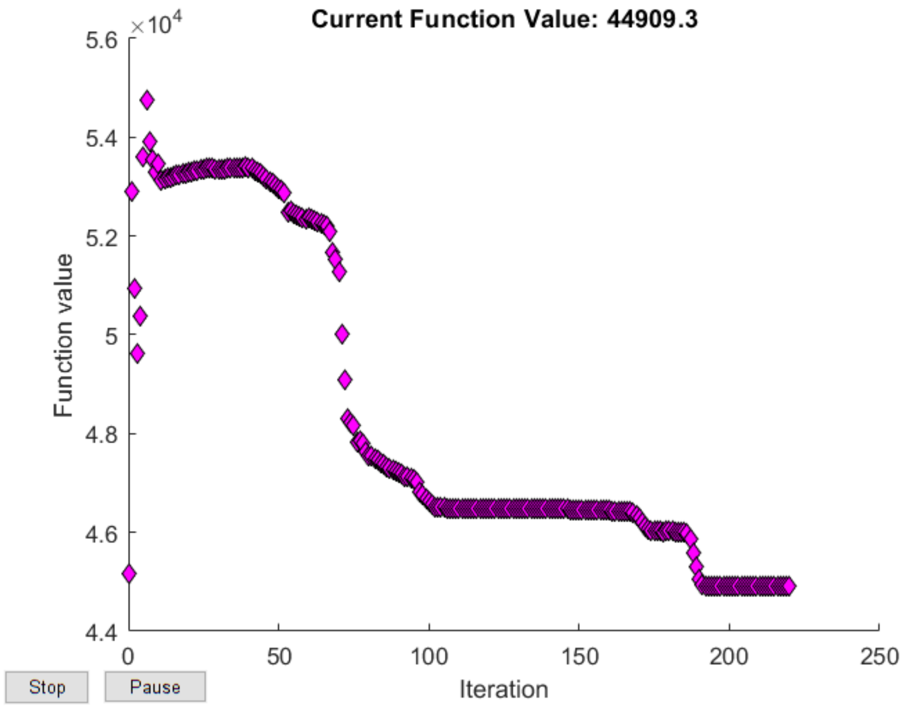

Figure 68: iteration 1 solution run.

The thruster orientation for the iteration 1 solution is shown in Figure 69. Figure 70 shows the comparison between the thrust trend of each thruster over time of the reference solution and of the iteration 1 solution. The comparison between the overall thrust exerted by each thruster of the reference solution and the iteration 1 solution is illustrated in Figure 71.

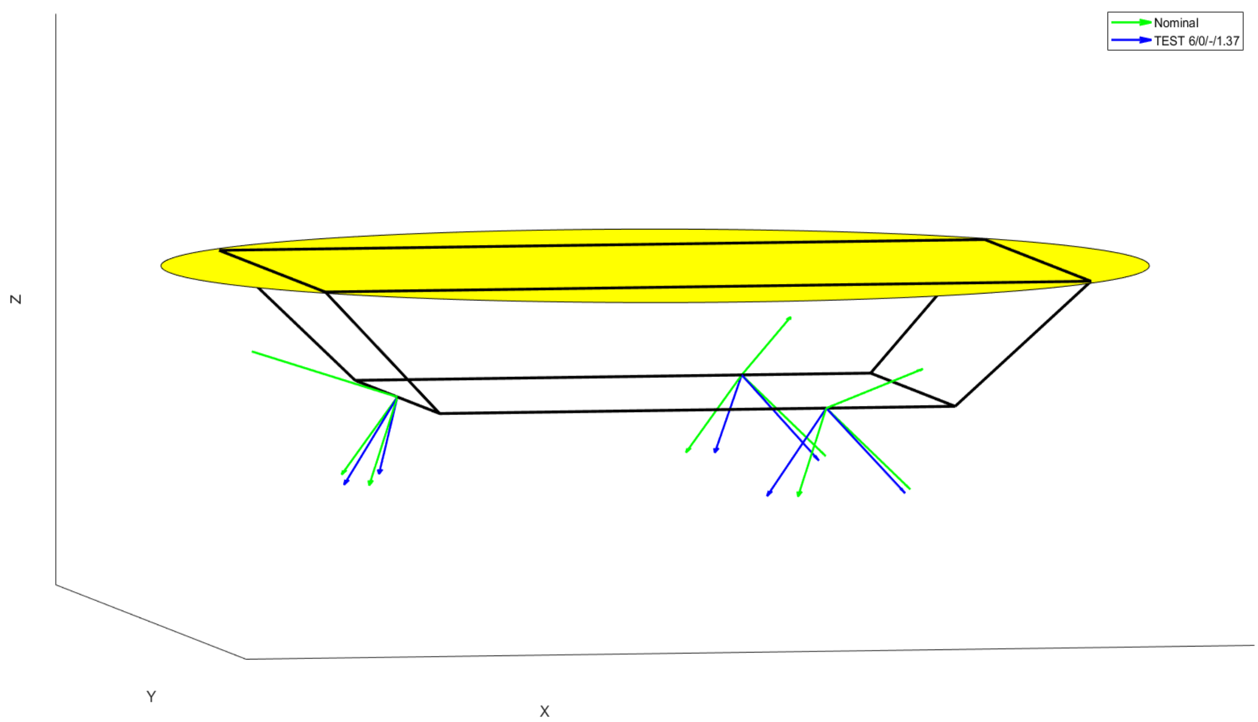

Figure 69: thruster orientation – iteration 1 solution (Matlab).

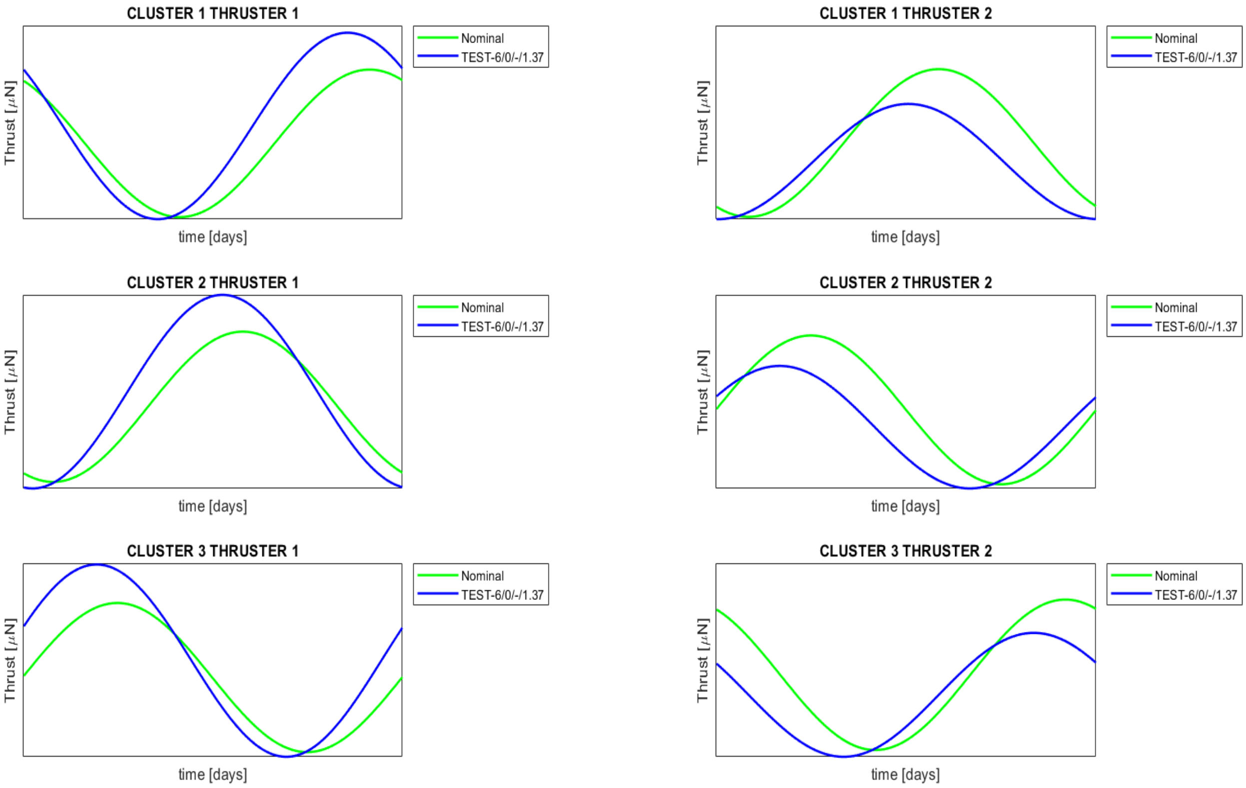

Figure 70: thrust over time – iteration 1 solution (Matlab).

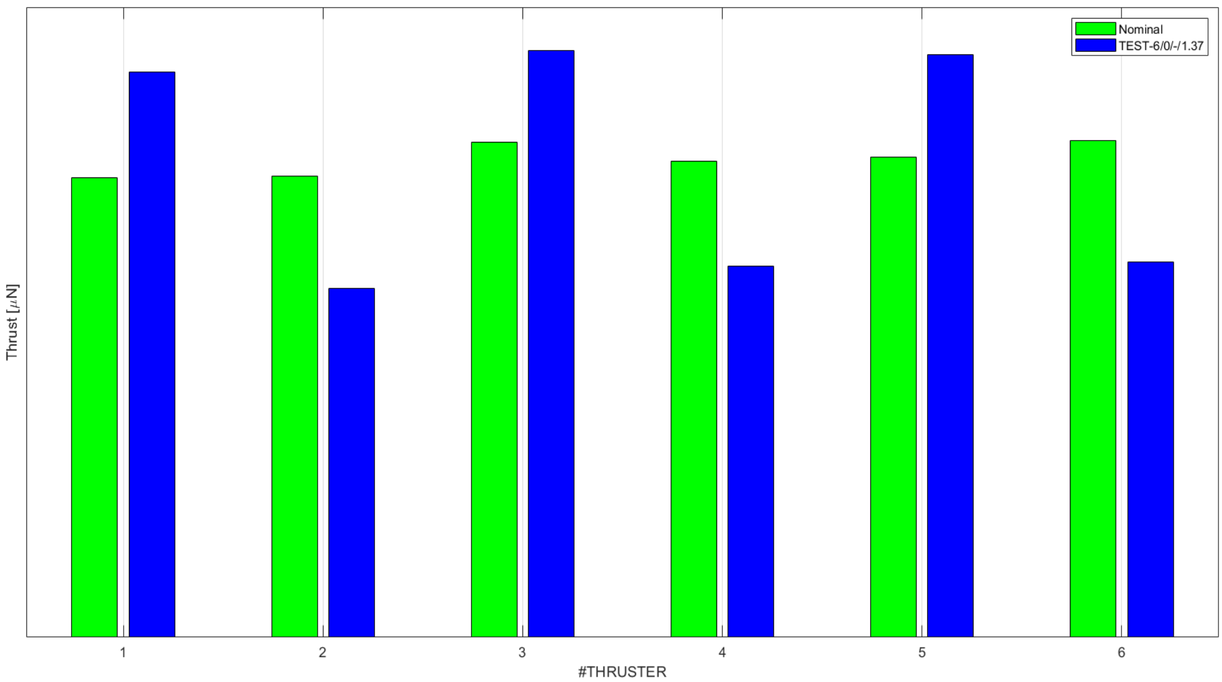

Figure 71: overall thrust for thruster – iteration 1 solution (Matlab).

The Matlab analysis for the iteration 2 solution is reported in Figure 72.

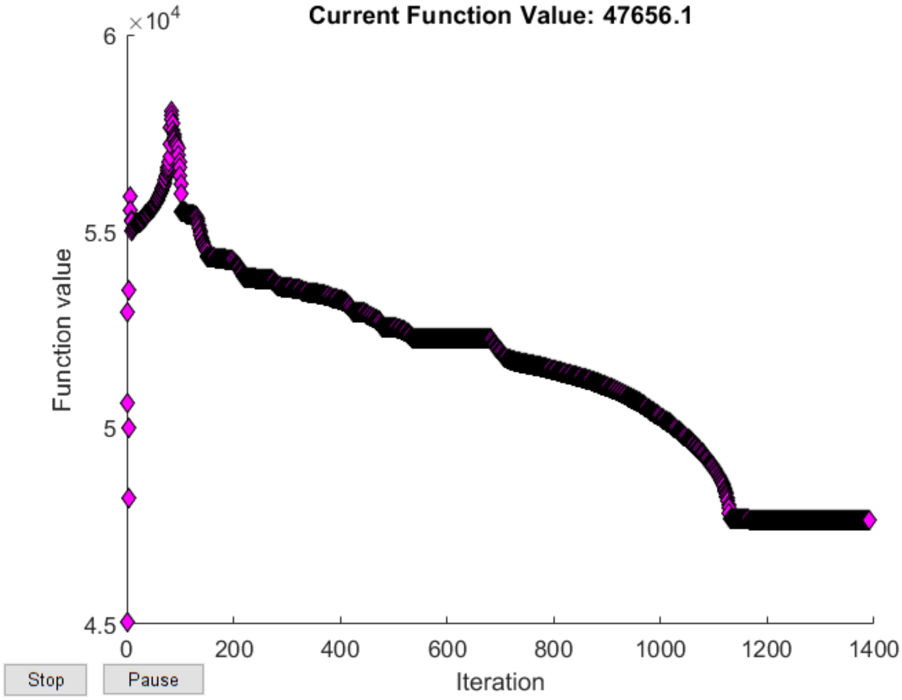

Figure 72: iteration 2 solution run.

The Matlab analysis for the iteration 3 solution is reported in Figure 73.

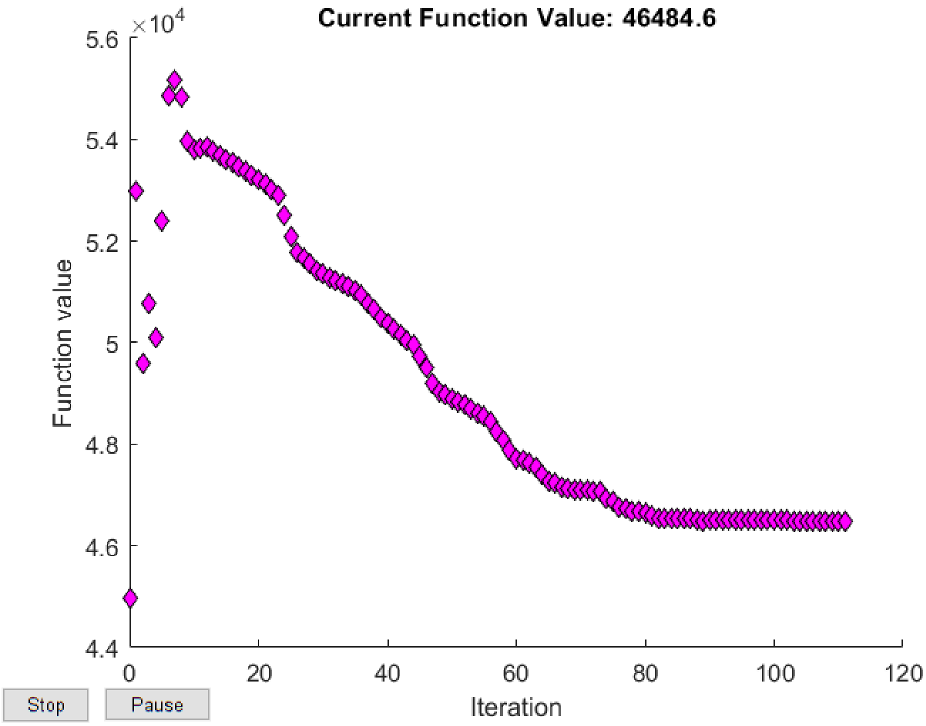

Figure 73: iteration 3 solution run.

5.3 BOUND ANALYSIS

The results of the analyses made to qualify the solutions obtained for the DFACS thrusters are summarized in Table 6.

Time Set 1				
Scenario	Solution	Fuel Consumption	LP-Bound	Gap
9 thrusters	Best CPLEX	97.97%	90.31%	7.82%
9 thrusters	Best Matlab	97.81%	90.31%	7.67%
6 thrusters	Best CPLEX	98.69%	90.31%	8.49%
6 thrusters	Best Matlab	98.63%	90.31%	8.43%
Time Set 2				
Scenario	Solution	Fuel Consumption	LP-Bound	Gap
9 thrusters	Best CPLEX	97.97%	90.31%	7.82%
9 thrusters	Best Matlab	97.81%	90.31%	7.67%
6 thrusters	Best CPLEX	98.69%	90.31%	8.49%
6 thrusters	Best Matlab	98.63%	90.31%	8.43%

Table 6: fuel consumption and relative LP-bound for the best solution obtained with CPLEX and Matlab for both scenarios with nine and six thrusters.

It should be noted that only the best solution obtained with CPLEX and the best solution obtained with Matlab for each scenario have been considered. The best solution for the nine-thrusters scenario obtained with CPLEX is the iteration 3 solution and the one obtained with Matlab is the global solution. The best solution for the six-thrusters scenario obtained with CPLEX is the iteration 5 solution and the one obtained with Matlab is the iteration 1 solution. The fuel consumption and the bound values are given in percentage of the fuel consumption of the reference scenario, obtained with the continuous model considering the two sets of 122 instants.

5.4 SPACECRAFT HANDLING

The results of the analyses made to evaluate the handling of the spacecraft in other phases as compared to scientific one are summarized in Table 7. The thruster layout considered in the continuous model for the different phases is the one obtained with the best solution for each scenario.

9 THRUSTERS	
Phase	Fuel Consumption
Science	97.82%
Disturbances of the antenna during the scientific measurements	100.61%
Variation of the antenna orientation while in science orbit	587.39%
Variation of the antenna orientation while in transfer orbit	infeasible
Main engine de-pointing (1)	110.66%
Main engine de-pointing (2)	105.09%
6 THRUSTERS	
Phase	Fuel Consumption
Science	98.63%
Disturbances of the antenna during the scientific measurements	infeasible
Variation of the antenna orientation while in science orbit	infeasible
Variation of the antenna orientation while in transfer orbit	infeasible
Main engine de-pointing (1)	infeasible
Main engine de-pointing (2)	infeasible

Table 7: fuel consumption for the best scenarios with nine and six thrusters for perturbances conditions different from those in science phase.

5.5 SIX-THRUSTERS SCENARIOS FOR AOCS

At first the maximum thrust considered is fixed to a value of 40 mN. Taking into account the thruster orientation of the reference scenario and running the iteration, a solution has been obtained with an error of 24.1% on the total torque request. The thruster orientation is shown in Figure 74 and the thrust over time is shown in Figure 75.

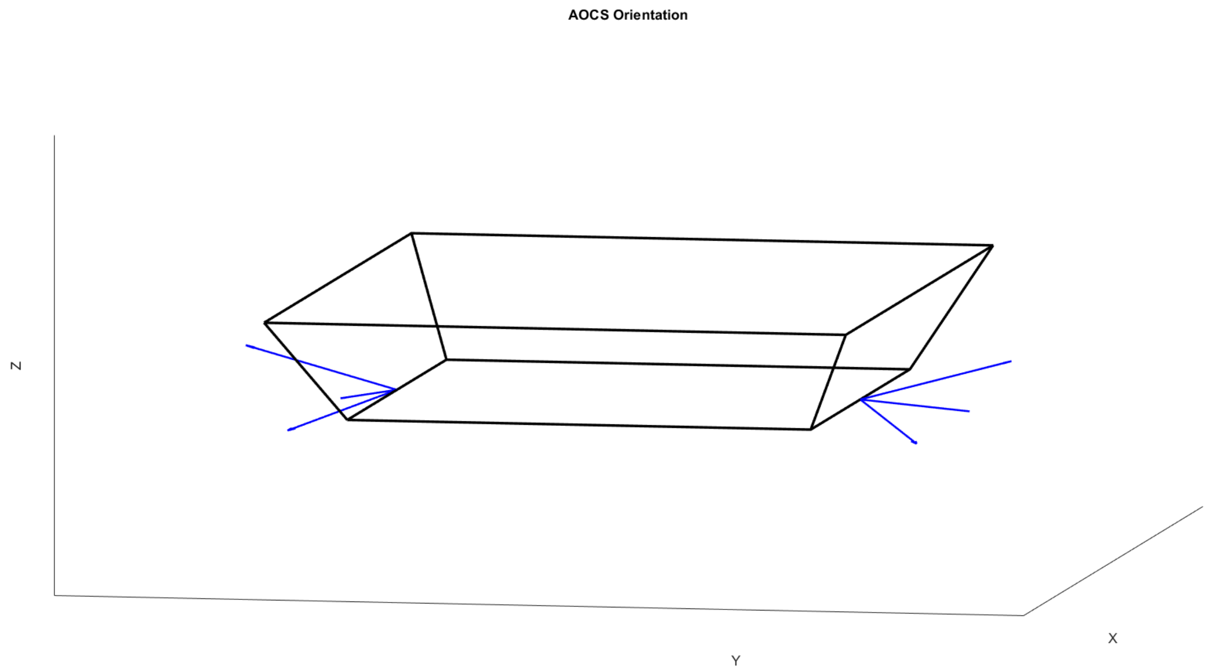

Figure 74: thruster orientation for the reference scenario.

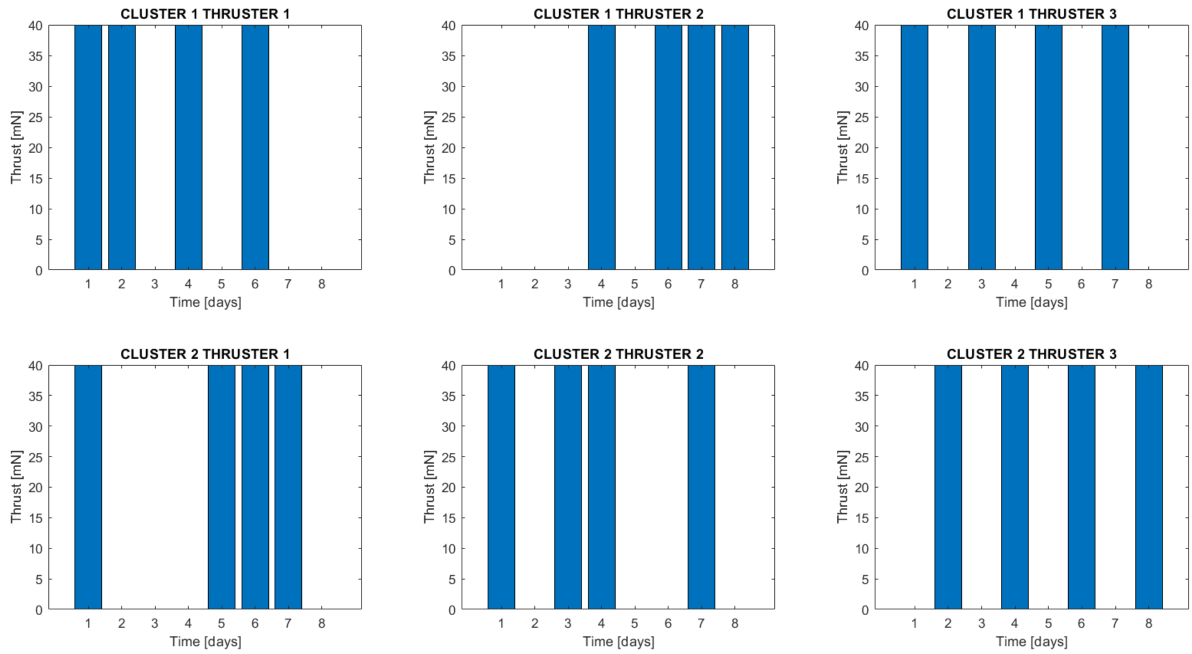

Figure 75: thrust over time for the reference scenario.

The comparison between torque requested and torque provided by the solution at every instant around the x,y and z axis is shown respectively in Figure 76, 77 and 78.

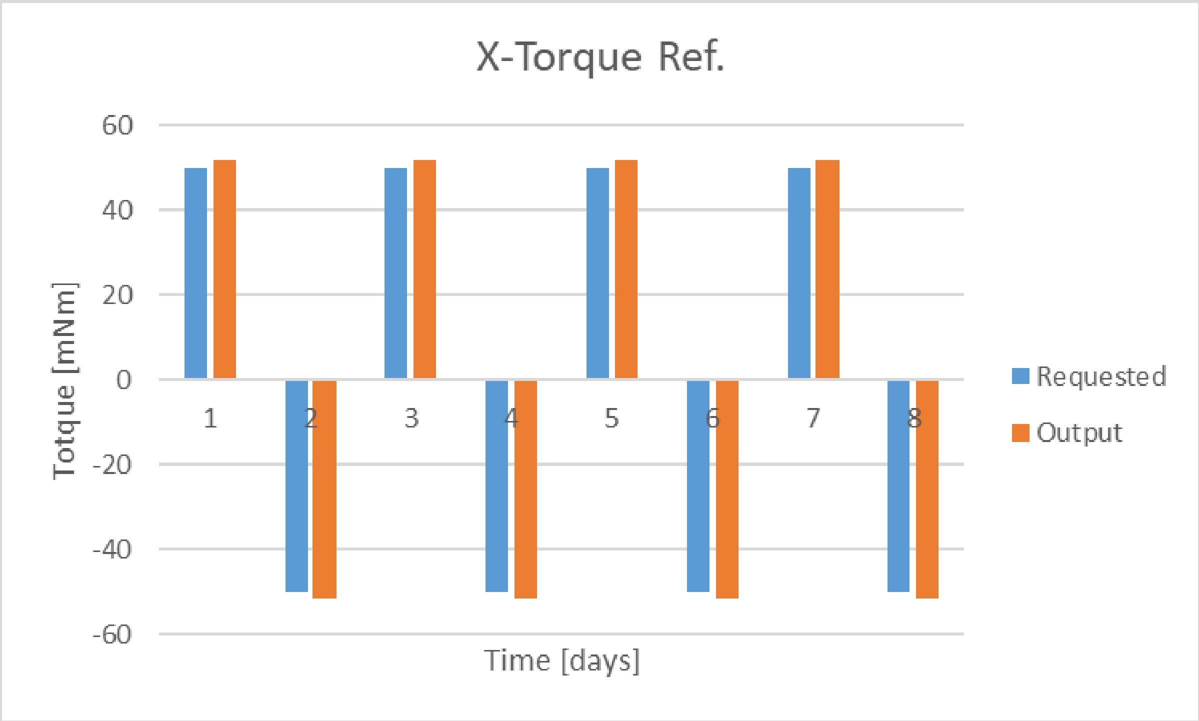

Figure 76: comparison between requested and provided torque around x-axis for the reference scenario.

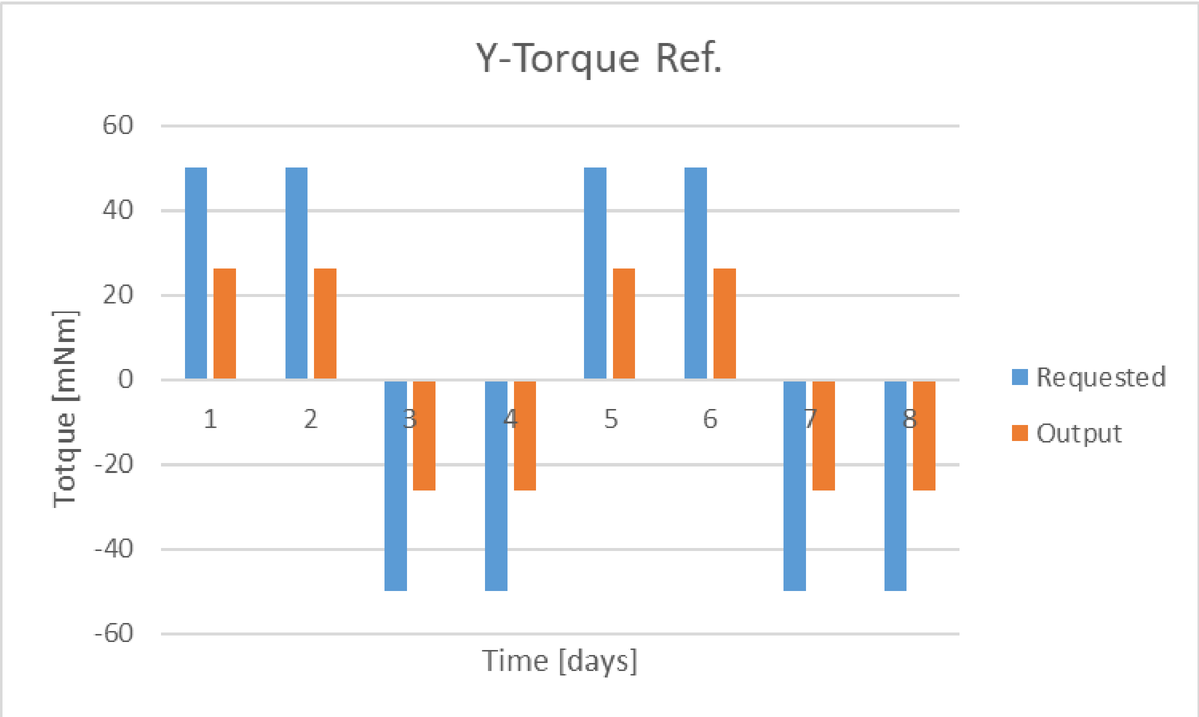

Figure 77: comparison between requested torque and provided torque around y-axis for the reference scenario.

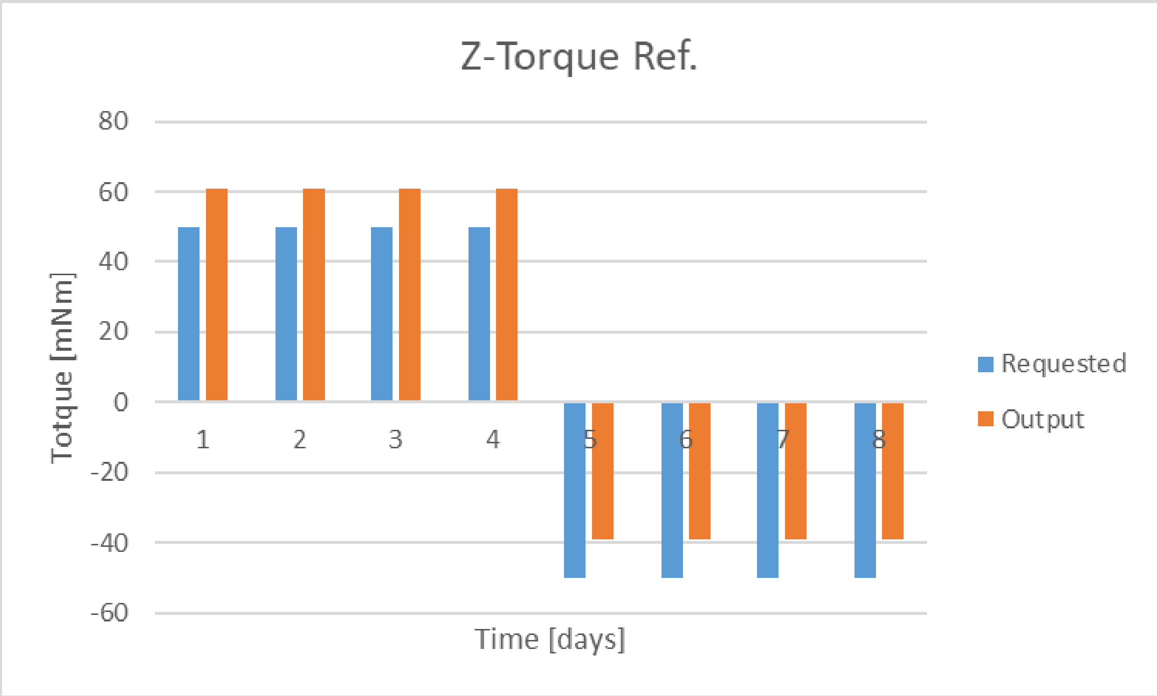

Figure 78: comparison between requested torque and provided torque around z-axis for the reference scenario.

The first scenario has no feasible solution even considering the thrust continuous.

The second scenario has a feasible solution considering the thrust as continuous. The thrust over time is shown in Figure 79.

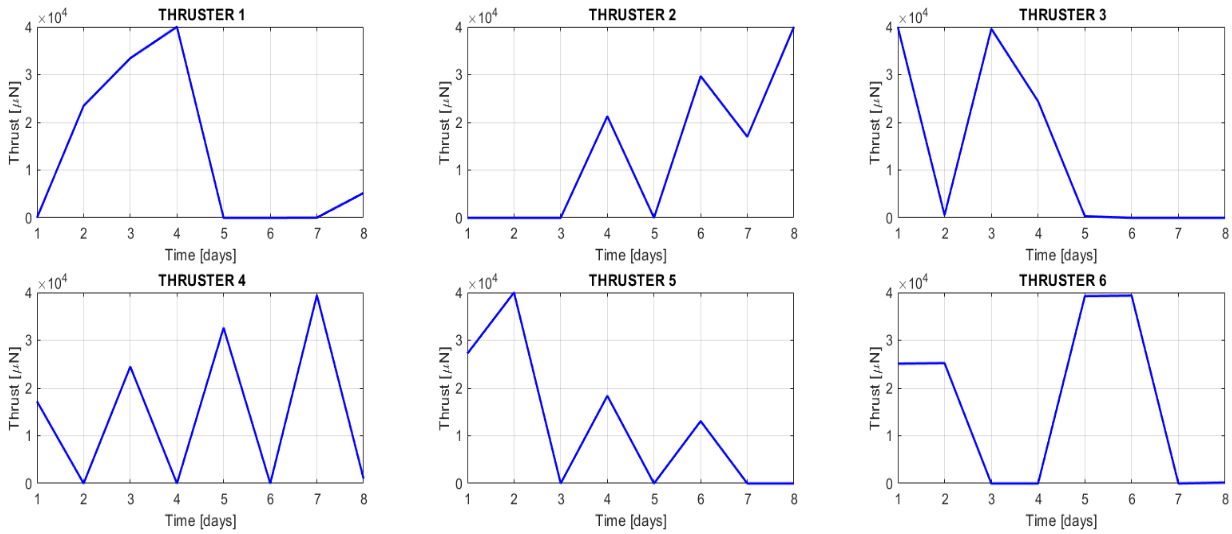

Figure 79: continuous thrust over time for each thruster for the second scenario.

The thruster orientation obtained after the iteration is shown in Figure 80. The best solution is obtained with a minimum total error of 20.3% on the total torque request. The thrust over time is shown in Figure 81. The comparison between torque requested and torque provided by the solution at every instant around the x,y and z axis is shown respectively in Figure 82, 83 and 84.

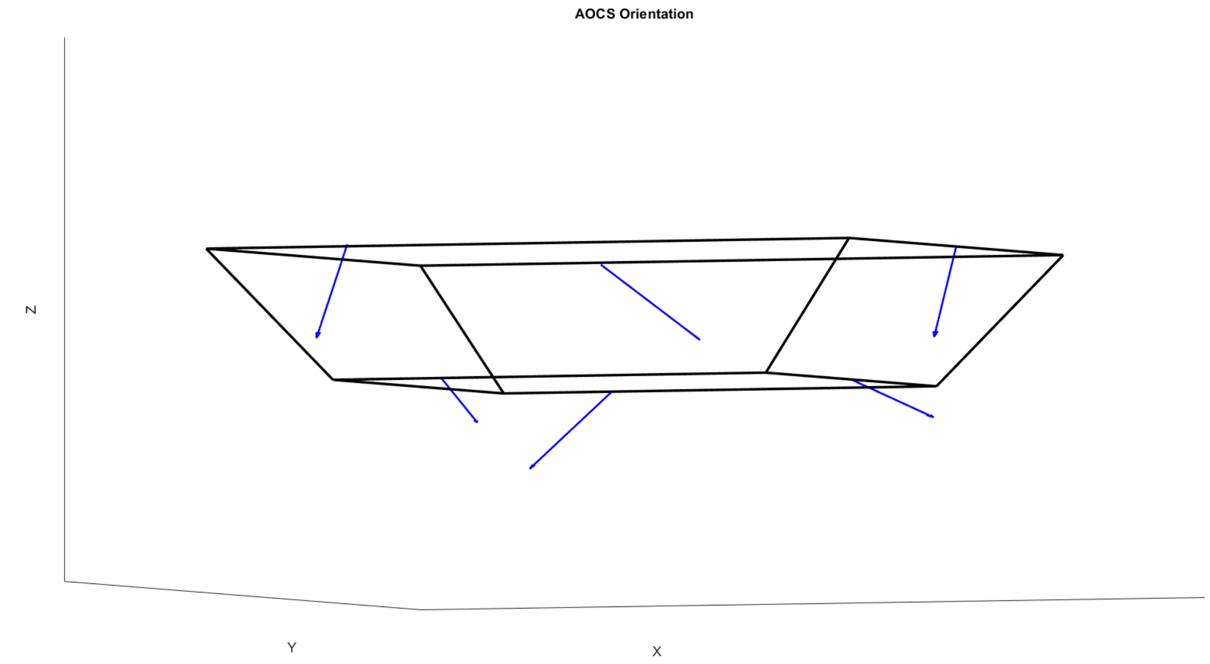

Figure 80: thruster orientation for the second scenario.

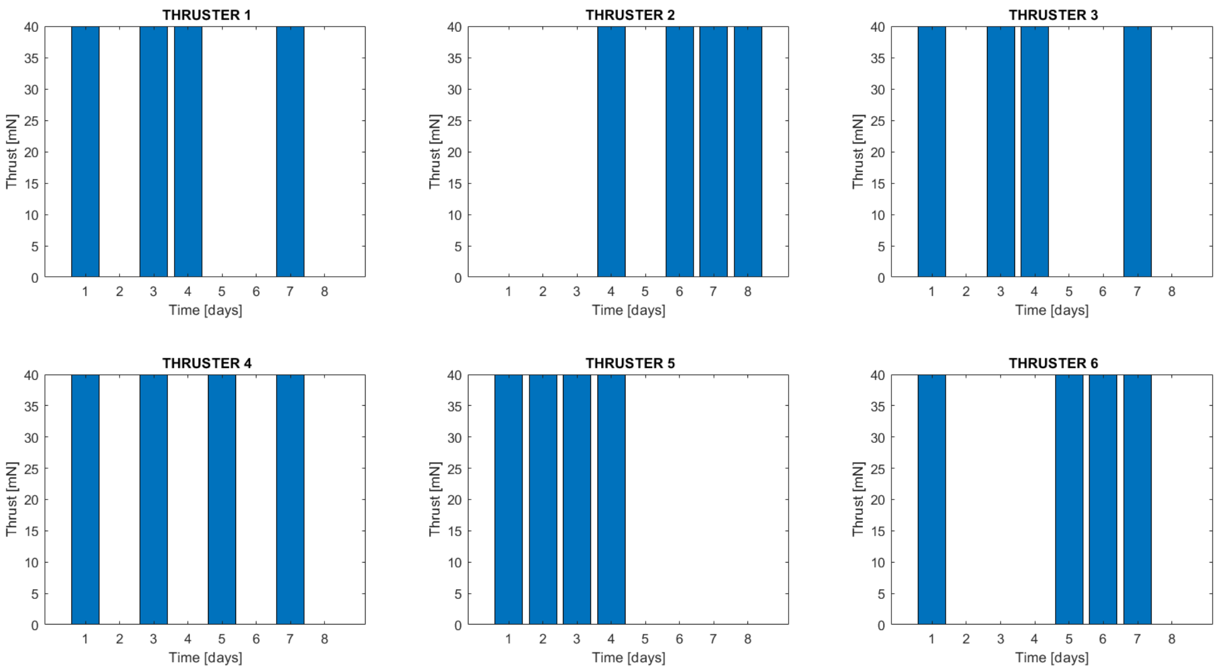

Figure 81: thrust over time for each thruster for the second scenario.

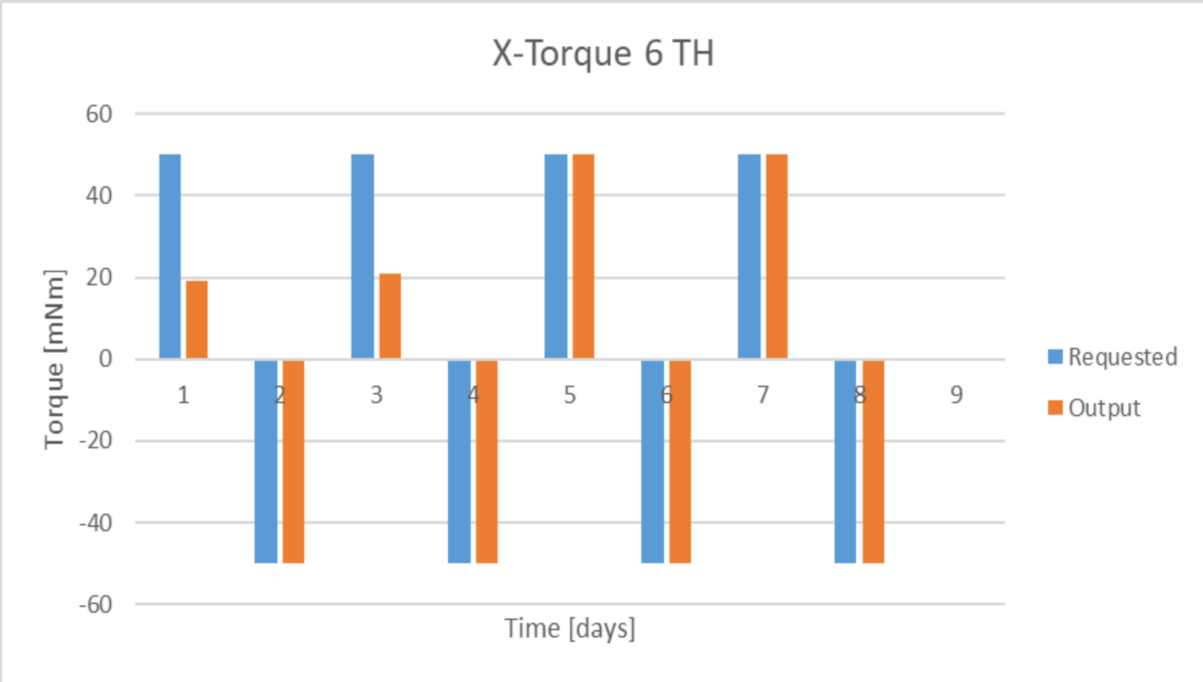

Figure 82: comparison between requested and provided torque around x-axis for the second scenario.

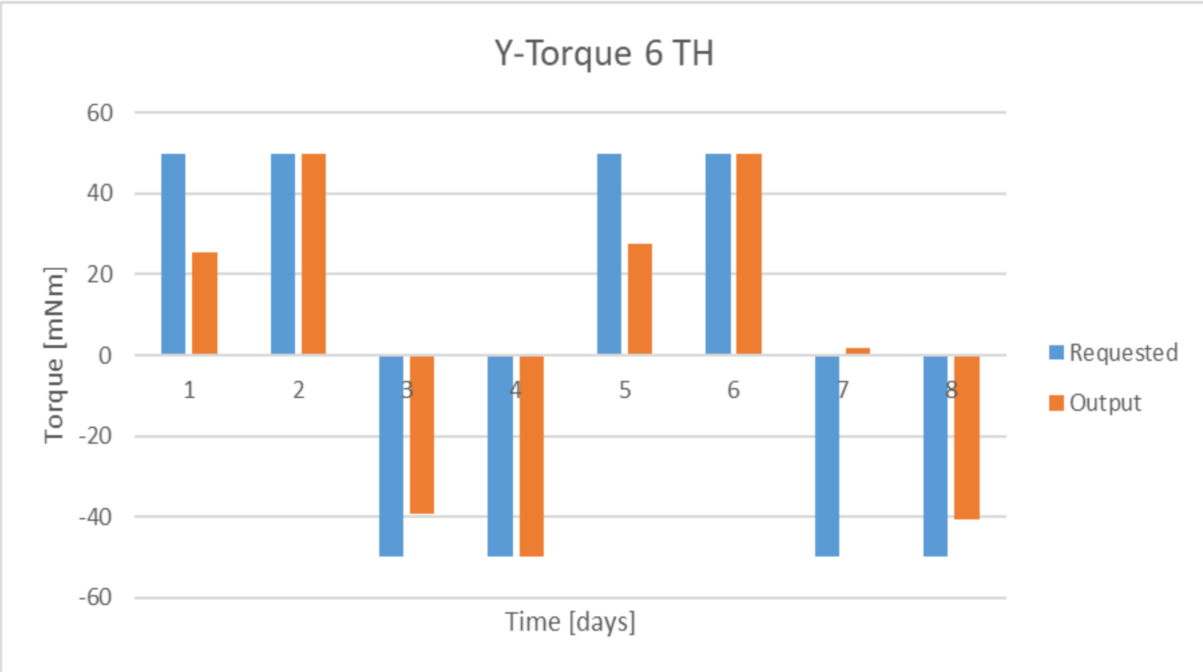

Figure 83: comparison between requested and provided torque around y-axis for the second scenario.

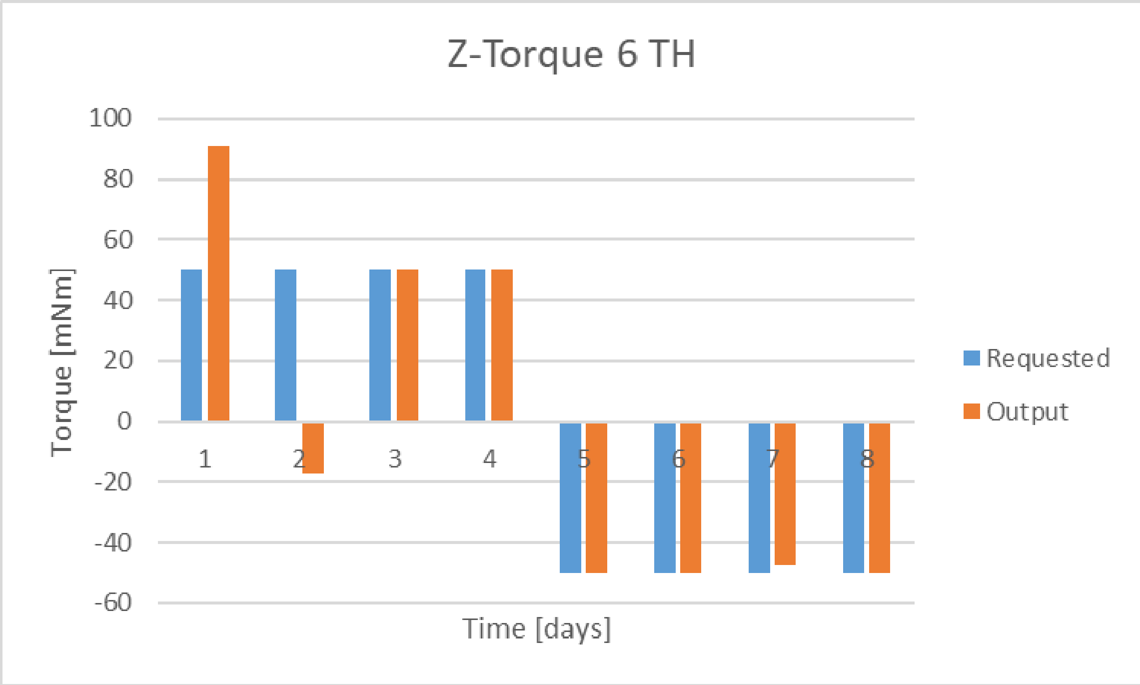

Figure 84: comparison between requested and provided torque around z-axis for the second scenario.

Subsequently a maximum thrust value greater than 40 mN has been considered. The scenario taken into account is the first one. The minimum value of maximum thrust which permits to obtain a feasible solution considering the thrust continuous is 65 mN. The thrust over time is shown in Figure 85.

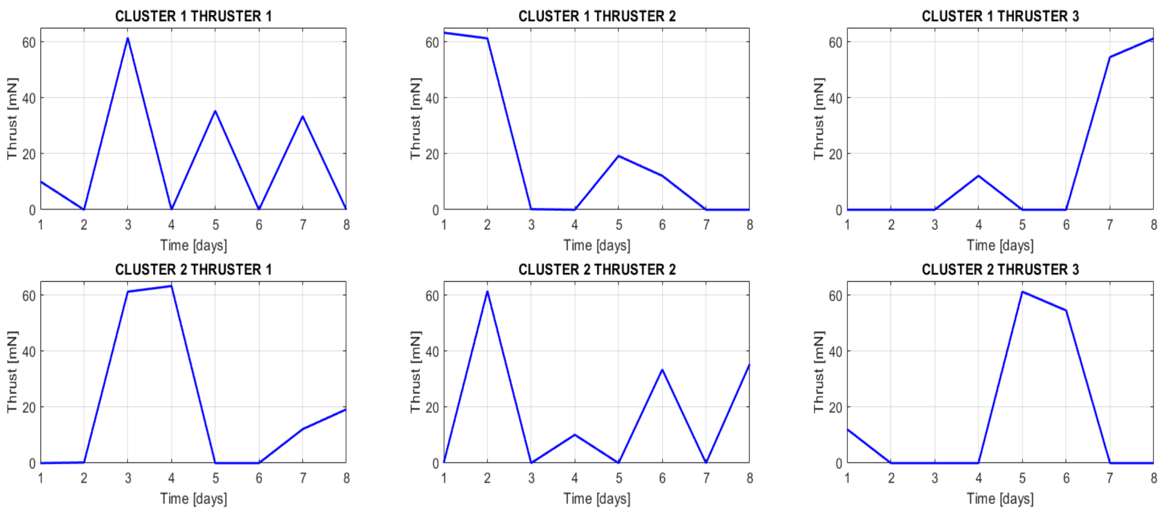

Figure 85: continuous thrust over time for each thruster for the first scenario with a maximum thrust of 65 mN.

The thruster orientation obtained by the iteration is shown in Figure 86. The best solution is obtained with minimum total error of 7.75% on the total torque request. The thrust over time is shown in Figure 87. The comparison between torque requested and torque provided by the solution at every instant around the x,y and z axis is shown respectively in Figure 88, 89 and 90.

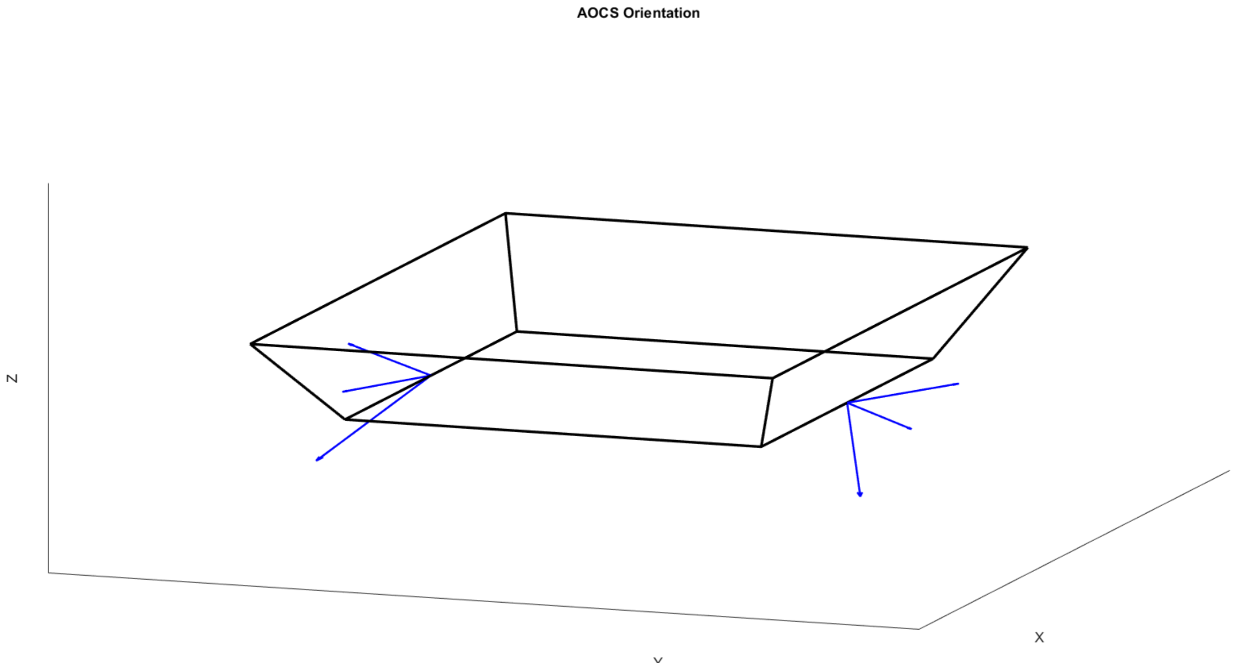

Figure 86: thruster orientation for the first scenario with a maximum thrust of 65 mN.

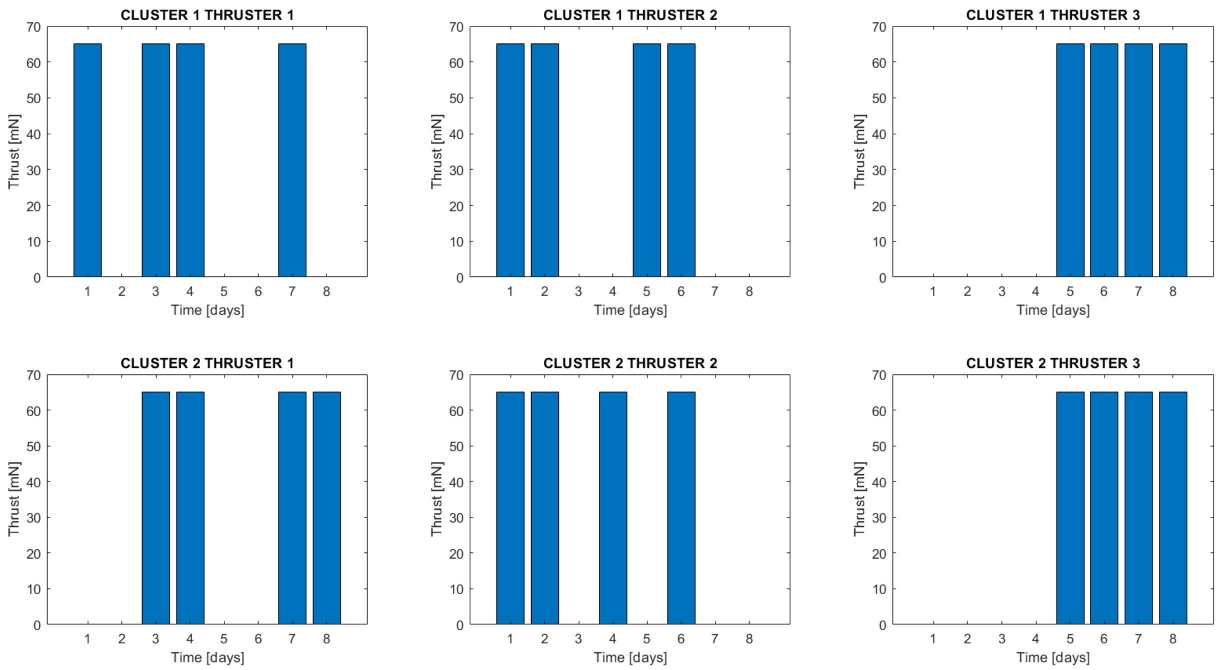

Figure 87: thrust over time for each thruster for the first scenario with a maximum thrust of 65 mN.

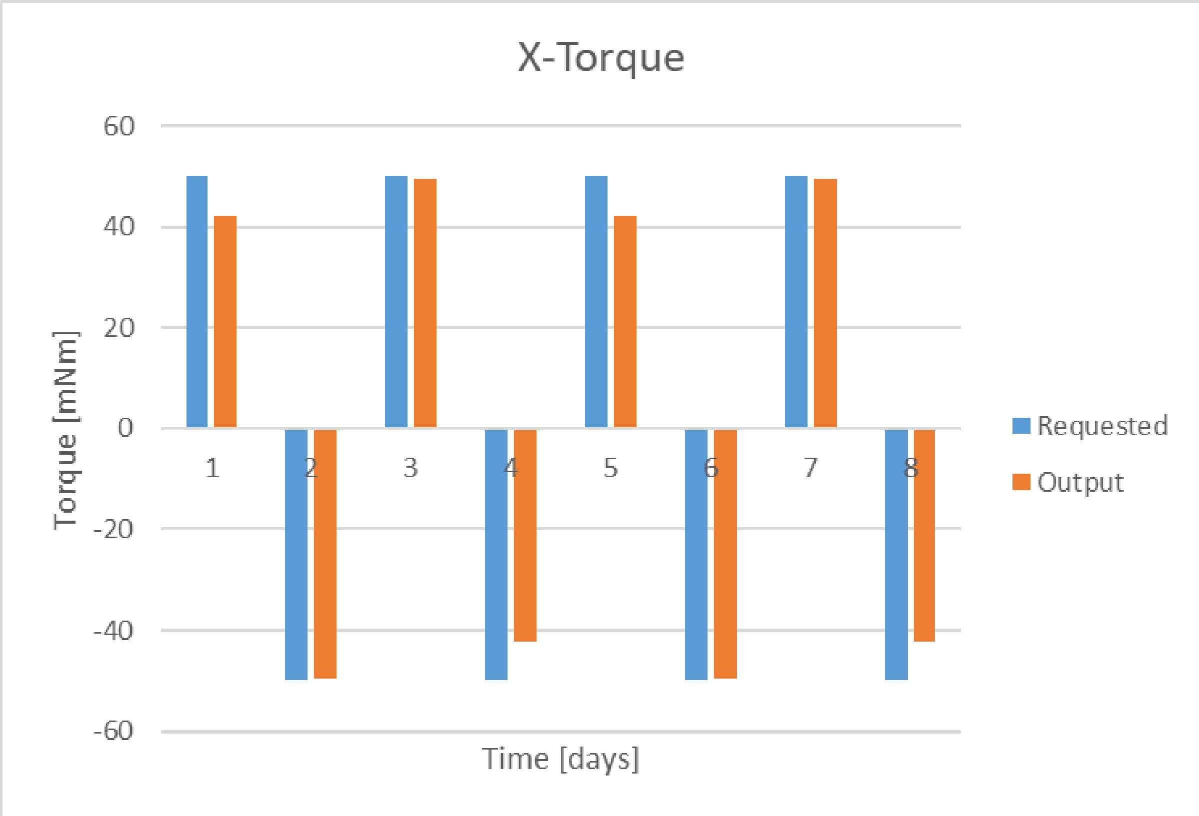

Figure 88: comparison between requested and provided torque around x-axis for the first scenario with a maximum thrust of 65 mN.

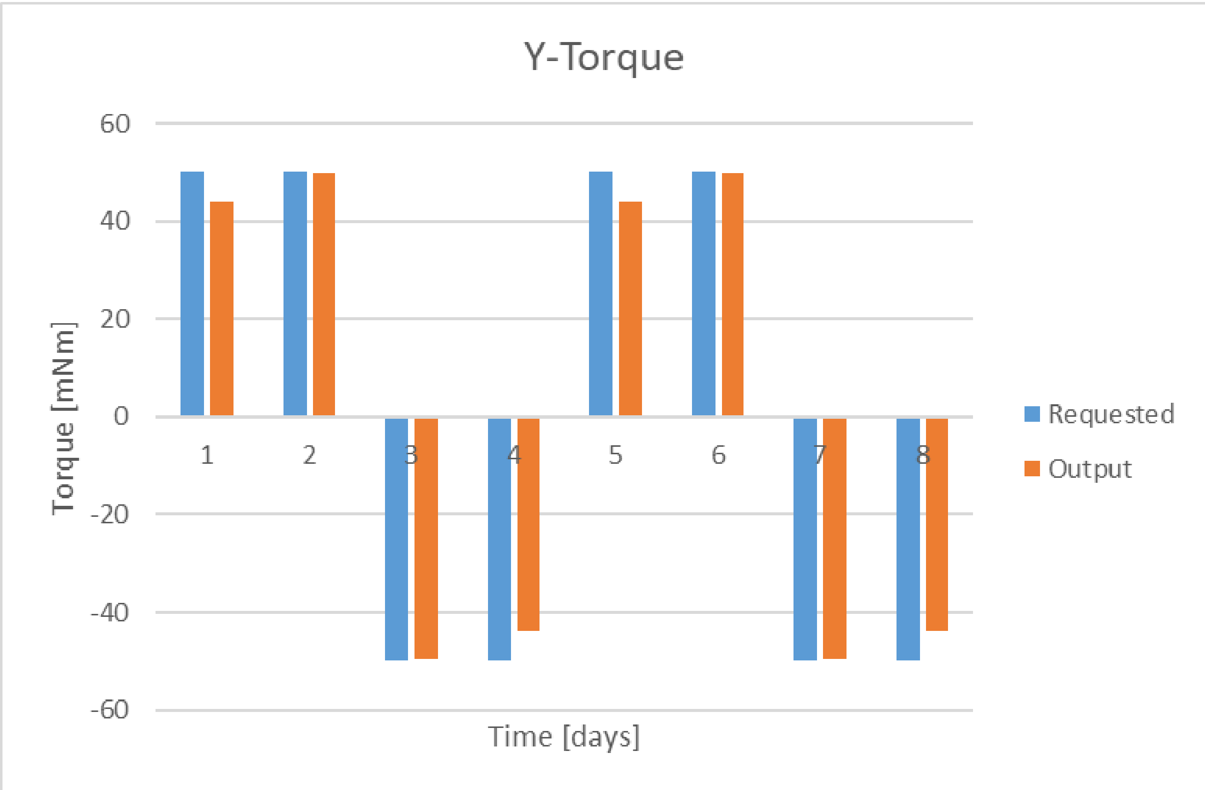

Figure 89: comparison between requested and provided torque around y-axis for the first scenario with a maximum thrust of 65 mN.

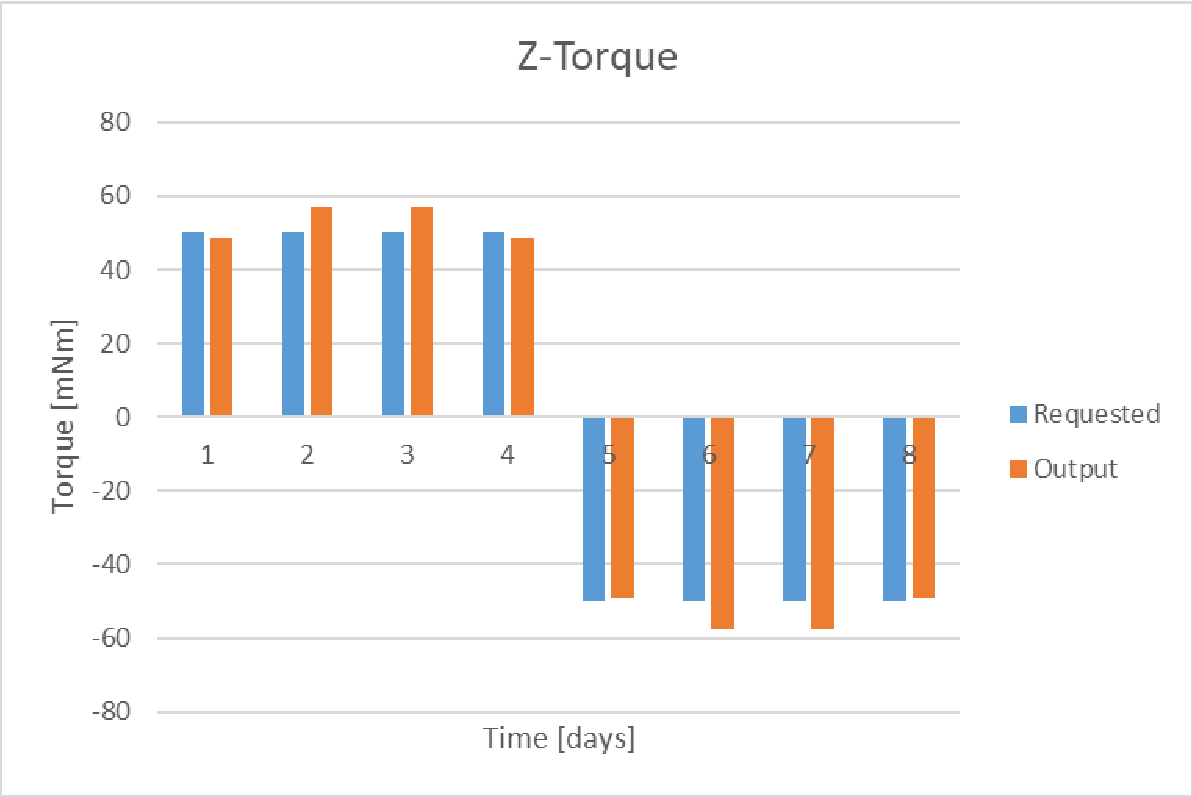

Figure 90: comparison between requested and provided torque around z-axis for the first scenario with a maximum thrust of 65 mN.

Subsequently a maximum thrust value of 70 mN is considered. The thrust over time is shown in Figure 91.

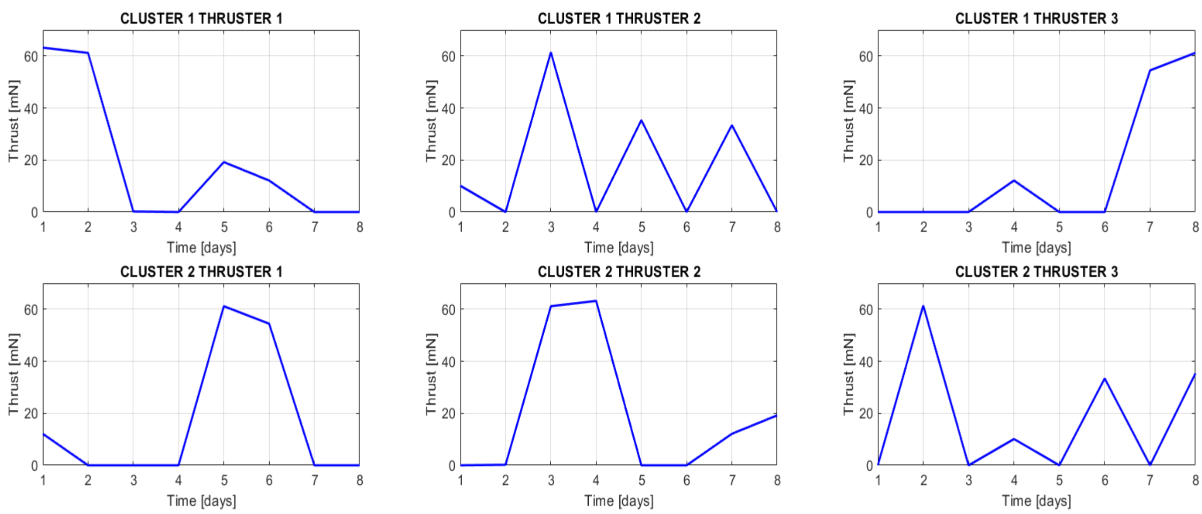

Figure 91: continuous thrust over time for each thruster for the first scenario with a maximum thrust of 70 mN.

The thruster orientation obtained after the iteration is shown in Figure 92. The best solution is obtained with minimum total error of 7.33% on the total torque request. The thrust over time is shown in Figure 93. The comparison between torque requested and torque provided by the solution at every instant around the x,y and z axis is shown respectively in Figure 94,95 and 96.

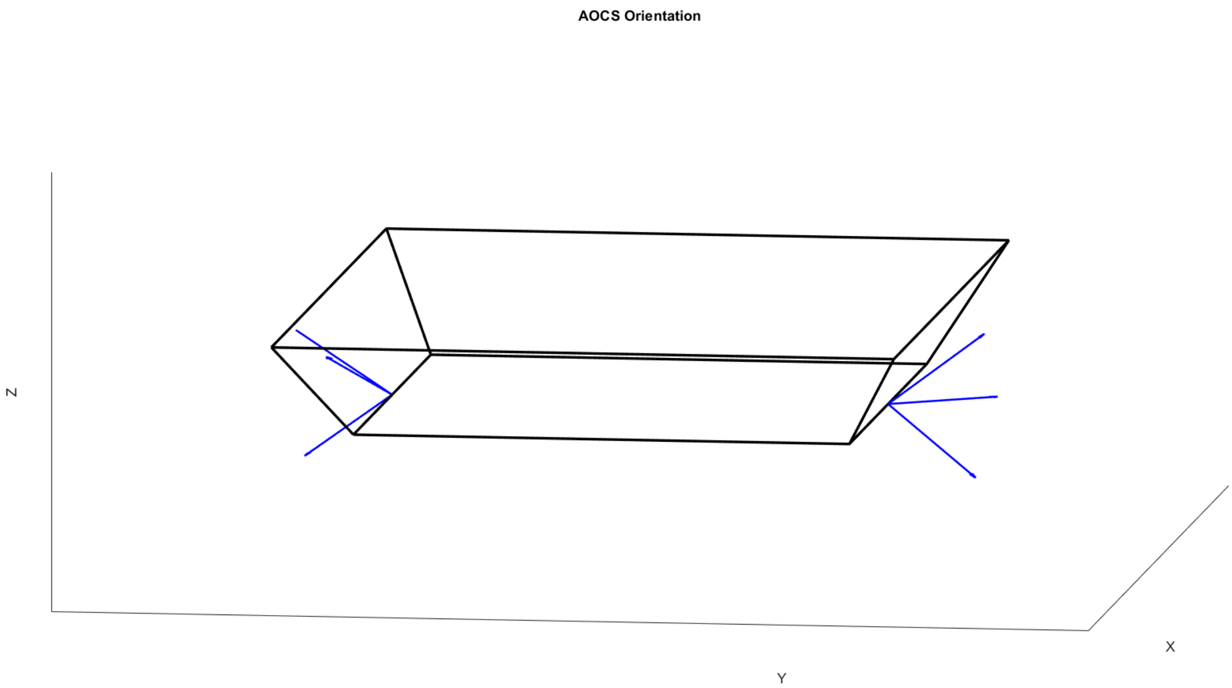

Figure 92: thruster orientation for the first scenario with a maximum thrust of 70 mN.

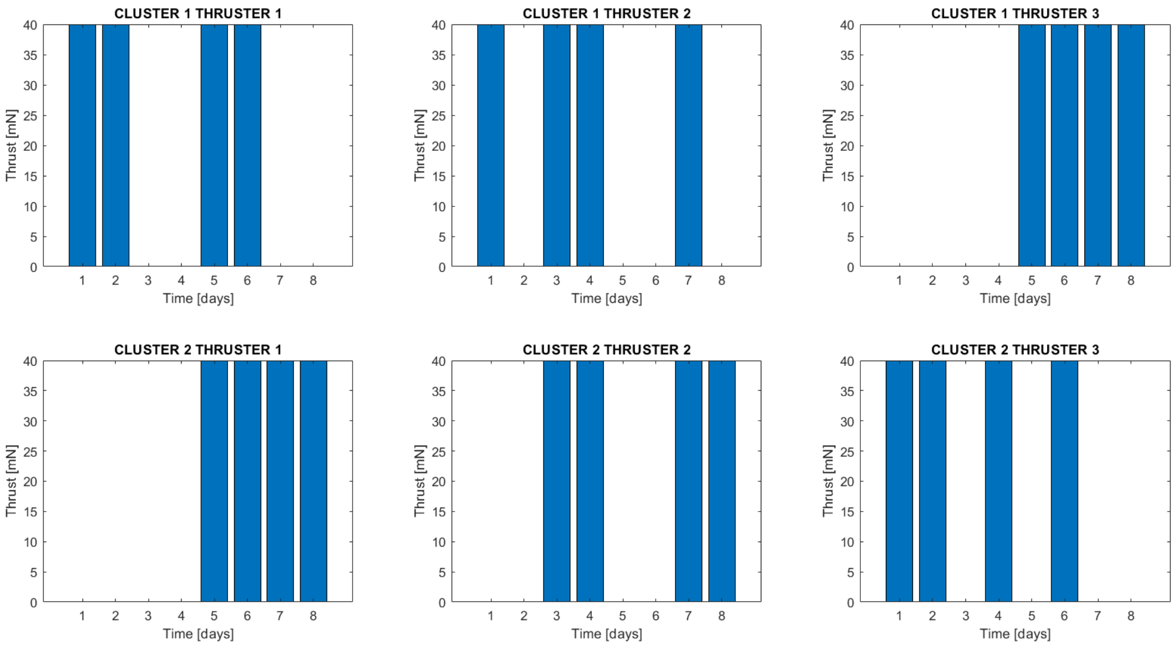

Figure 93: thrust over time for the first scenario with a maximum thrust of 70 mN.

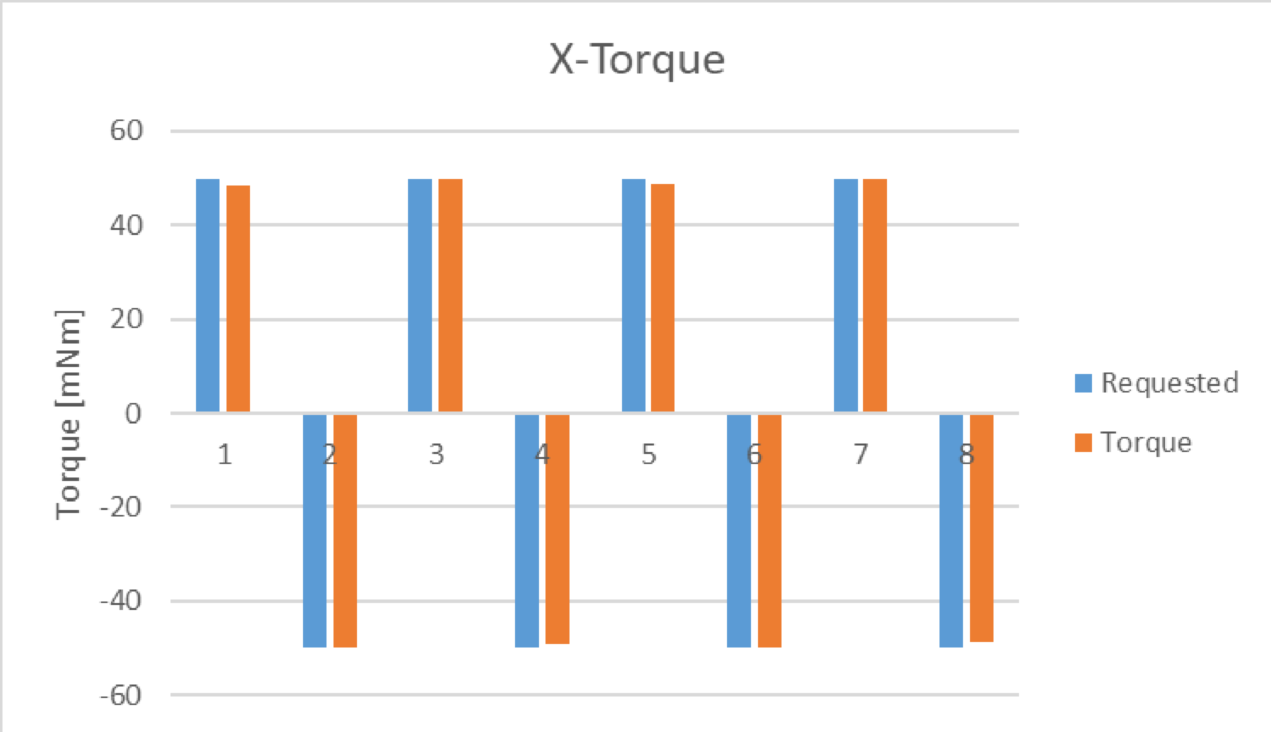

Figure 94: comparison between requested and provided torque around x-axis for the first scenario with a maximum thrust of 70 mN.

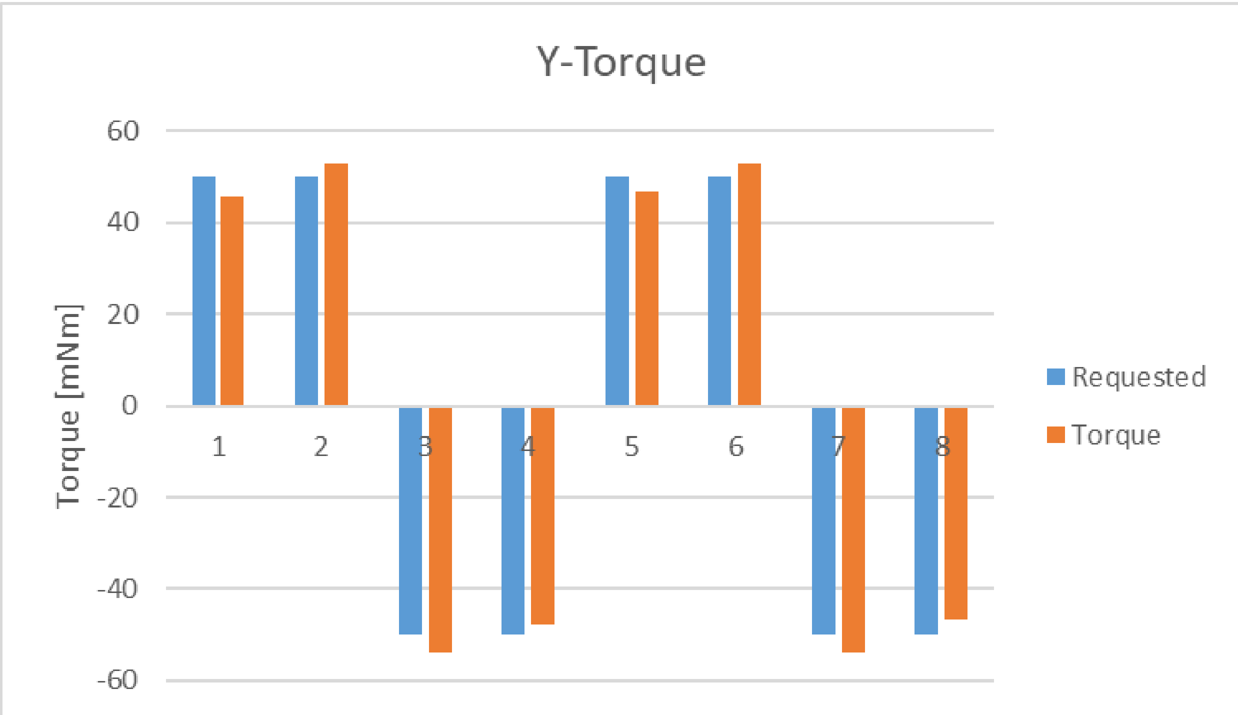

Figure 95: comparison between requested and provided torque around y-axis for the first scenario with a maximum thrust of 70 mN.

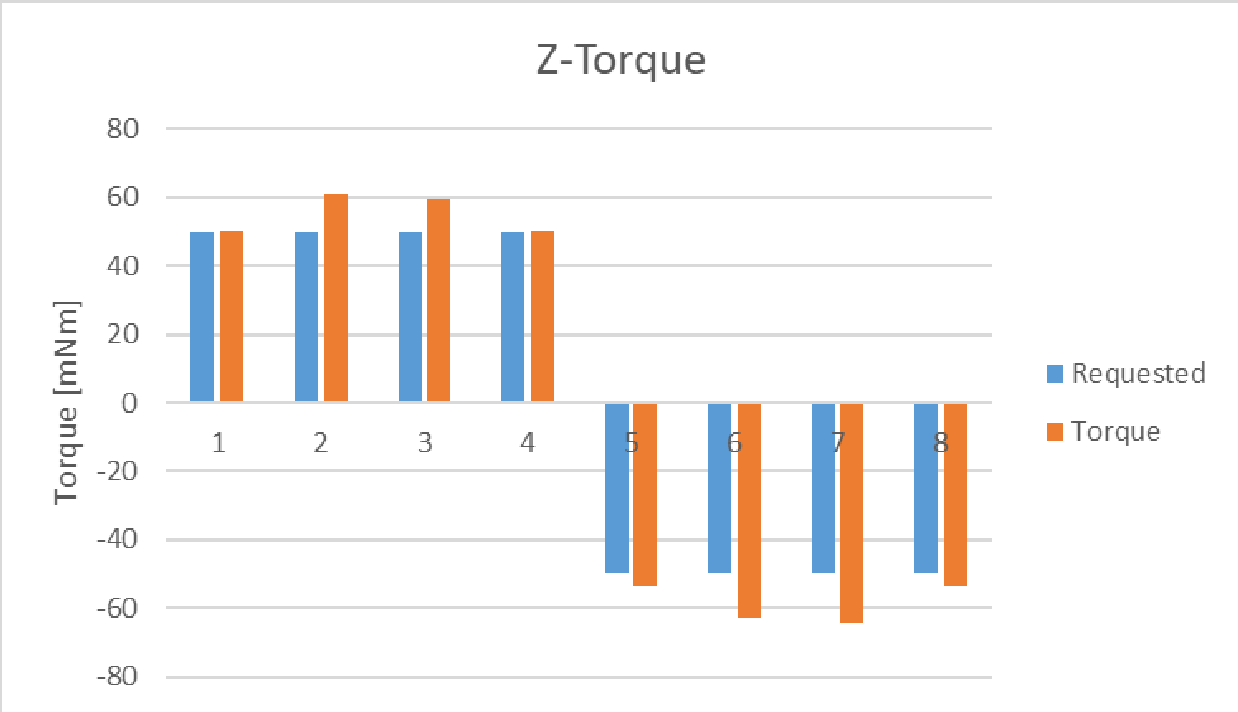

Figure 96: comparison between requested and provided torque around z-axis for the first scenario with a maximum thrust of 70 mN.

5.6 EIGHT-THRUSTERS SCENARIO FOR AOCS

The maximum thrust considered has a value of 40 mN.

The scenario with eight thrusters has a feasible solution considering the thrust continuous. The thrust over time is shown in Figure 97.

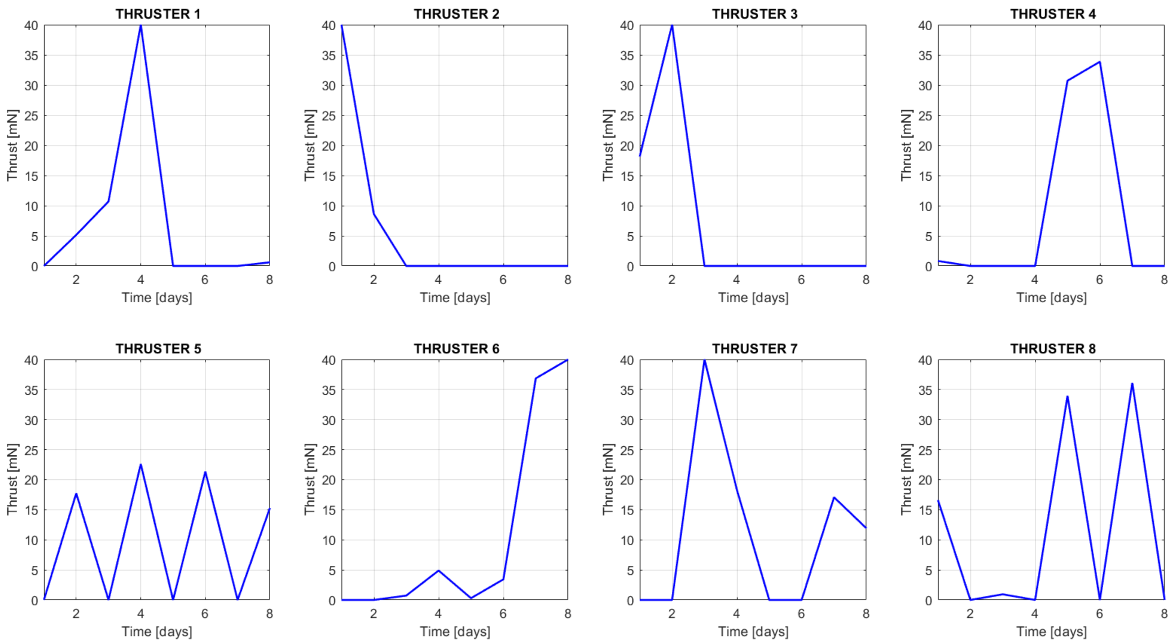

Figure 97: continuous thrust over time for the third scenario.

The thruster orientation obtained after the iteration is shown in Figure 98. The best solution is obtained with a minimum total error of 10.0% on the total torque request. The thrust over time is shown in Figure 99. The comparison between the torque requested and the torque provided by the solution at every instant around the x,y and z axis is shown respectively in Figure 100, 101 and 102.

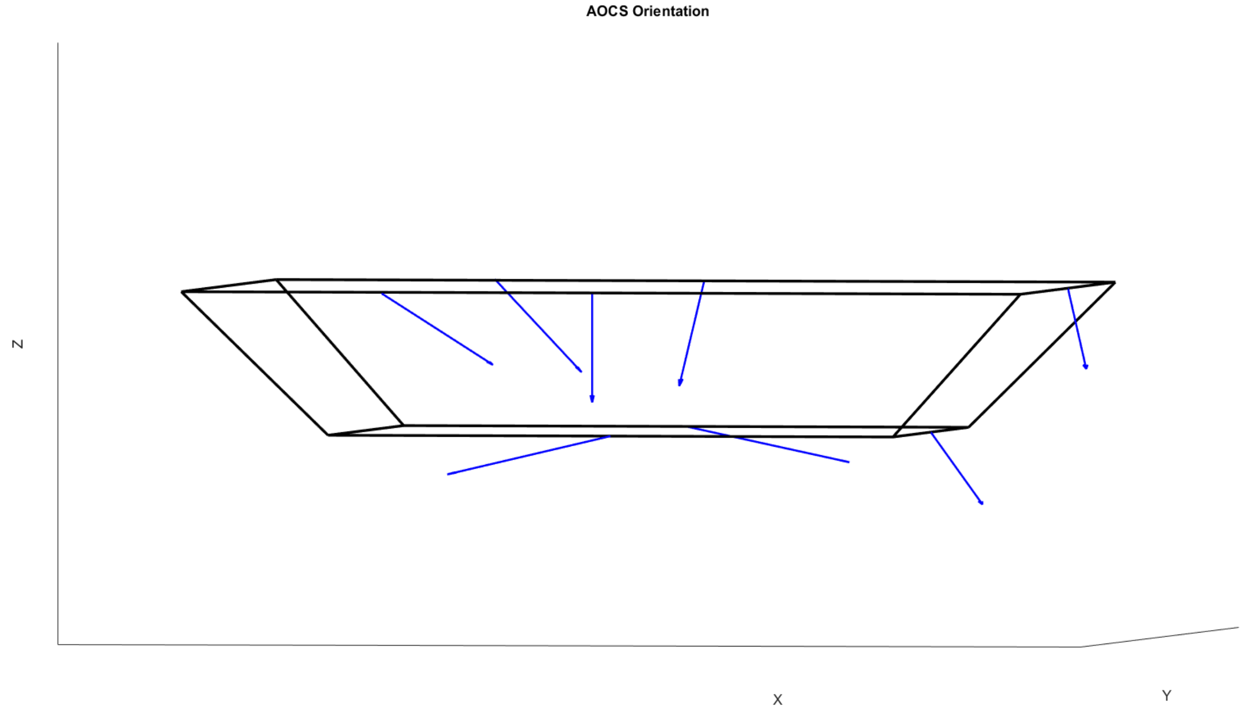

Figure 98: thruster orientation for the third scenario.

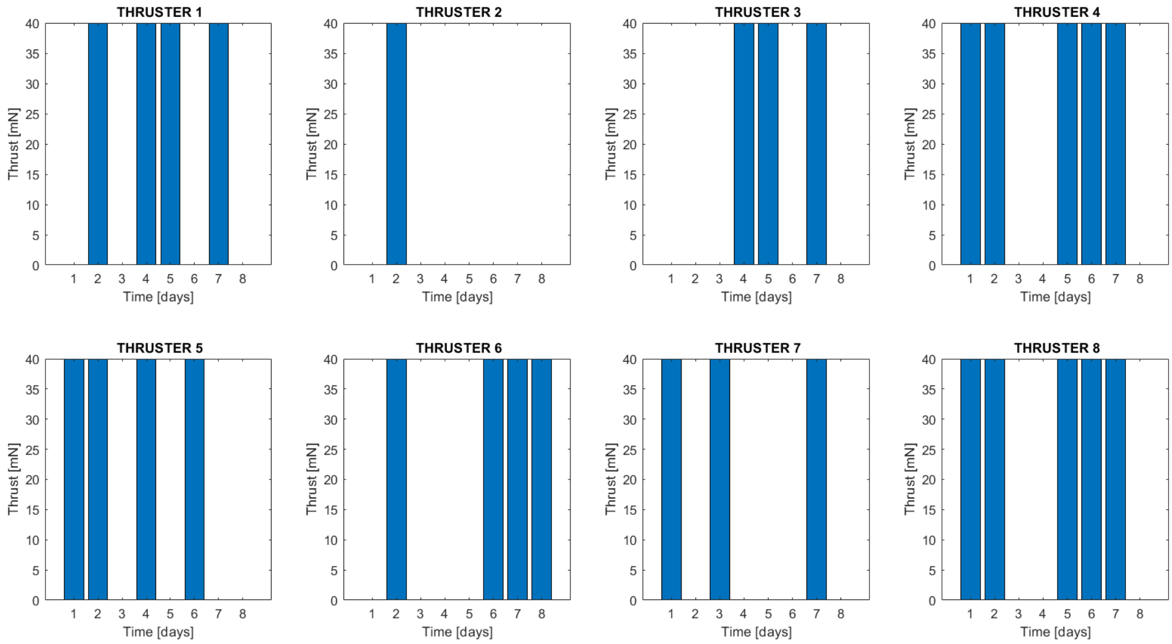

Figure 99: thrust over time for each thruster for the third scenario.

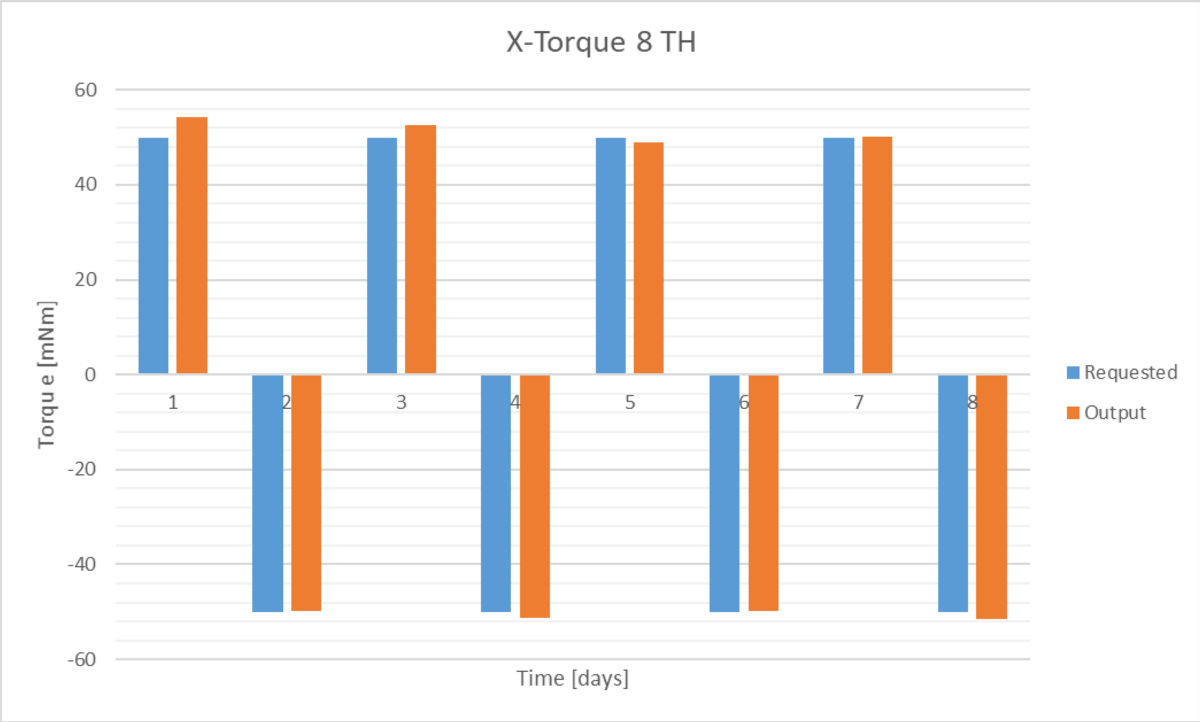

Figure 100: comparison between requested and provided torque around x-axis for the third scenario.

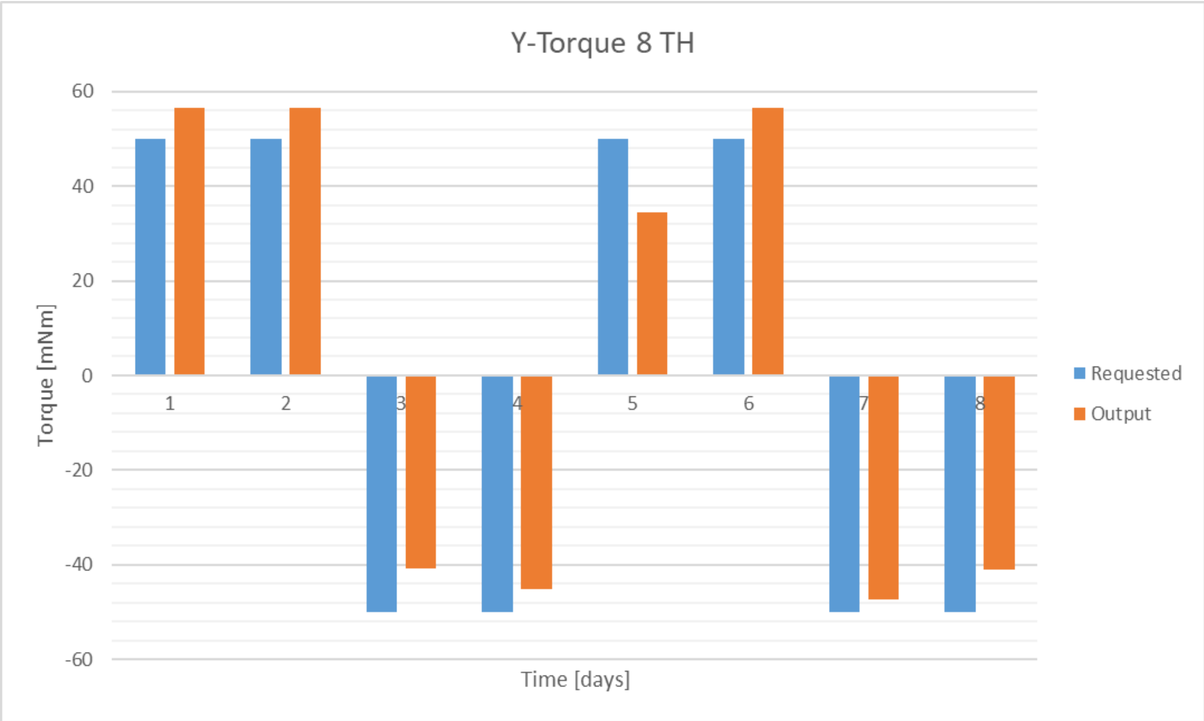

Figure 101: comparison between requested and provided torque around y-axis for the third scenario.

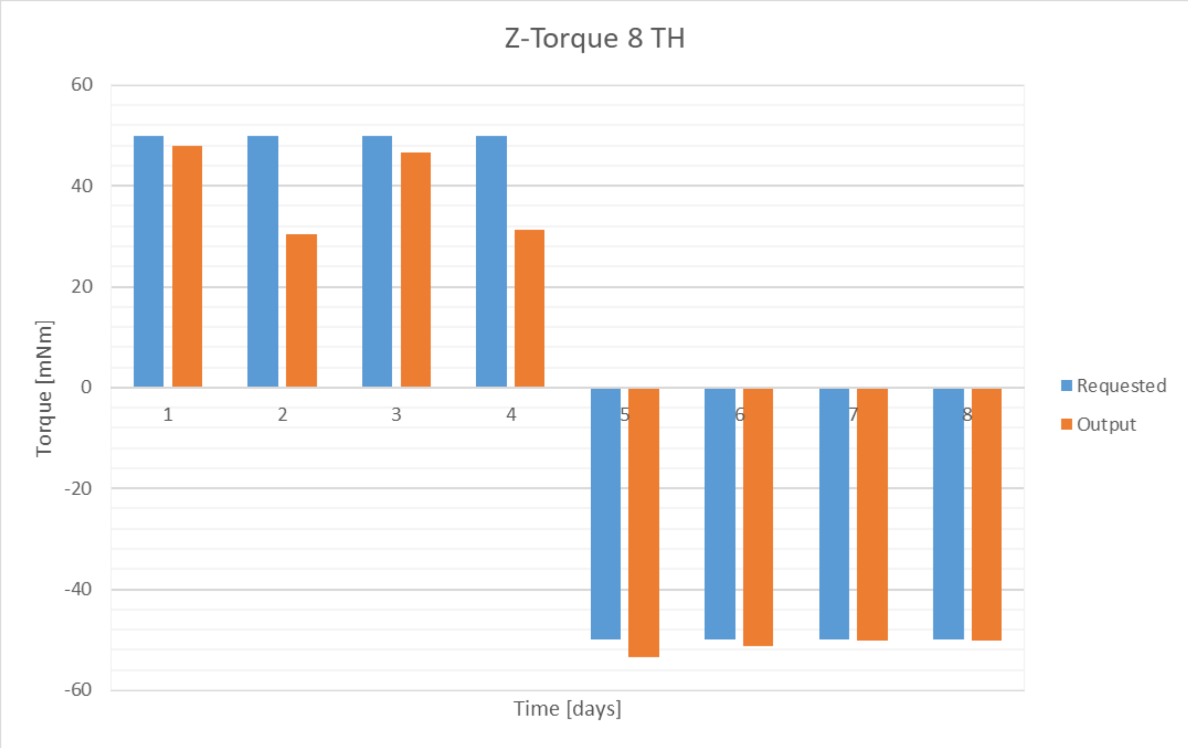

Figure 102: comparison between requested and provided torque around z-axis for the third scenario.

5.6 EXPERIMENTAL ANALYSIS SUMMARY

The results of the first part of the analyses performed in this thesis is related to the DFACS (Drag Free Attitude Control System) of the LISA (Laser Interferometer Space Antenna) spacecraft. The optimization methods used have led to different thruster layouts with a fuel consumption lower than the reference one provided by ESA. In all cases the thruster orientation is very similar to the one of the reference solution. The best solution yields an improvement of 2.18% in terms of fuel consumption. The corresponding layout provides for the presence of three clusters of three thrusters each, with one thruster oriented upwards for each cluster. This solution has been obtained with Matlab, but the previous analyses performed with CPLEX have been fundamental to find the initial solution for Matlab. The layout with six thrusters also implies a fuel consumption lower than the reference one, but the improvement is smaller than the one obtained with the layout with nine thrusters.

The lower-bound analysis shows that the gap between the solution value and the lower-bound found is very small, therefore the solution cannot be improved significantly.

The scenario with nine thrusters is able to operate in conditions different from the science phase. It can especially compensate the disturbances of the antenna during the scientific measurements, the variation of the antenna orientation while in science orbit and further the main engine de-pointing while in transfer orbit. The only condition not compensated by this scenario is the variation of the antenna orientation while in transfer orbit. The scenario with six thrusters is not able to operate in conditions different from the science phase mentioned above.

The last part of the analyses in this thesis is related to the AOCS (Attitude and Orbit Control System) of the LISA spacecraft. The results show that the solution error decreases with the increase of the number of thrusters and with the increase the maximum thrust which can be supplied by each thruster.

CONCLUSIONS AND FUTURE DEVELOPMENTS

The research work discussed in this thesis has been carried out in support of the LISA (Laser Interferometer Space Antenna) mission. The analyses performed concern the thruster layout optimization problem. Two different scenarios have been considered, i.e. DFACS (Drag Free Attitude Control System) and AOCS (Attitude and Orbit Control System). The results derived from the whole experimental analysis performed have provided a significant contribution both to the present and the upcoming phases of the LISA study.

The best layout obtained for DFACS provides for the presence of three clusters of three thrusters each, with one thruster per cluster oriented upwards. This solution yields an improvement of fuel consumption by 2.18% as compared to the reference solution provided by ESA.

Another important piece of information comes from the lower-bound analysis: fuel consumption cannot improve significantly beyond 2.18% since the gap between the best solution and the lower-bound found is very small.

The scenario with nine thrusters that brings about the best fuel consumption is very promising since it can compensate forces and torques other than those of the science phase: it can operate in all conditions considered other than the science phase, excepting one.

Solutions for the scenario with six thrusters have also been obtained, but the improvement in fuel consumption is lower than 2.18% and, furthermore, this scenario cannot operate in conditions other than the science phase.

With regard to the AOCS layout analysis, no solutions without a certain error (w.r.t. the torque request) have been obtained. Nonetheless, it has been noted that the error decreases with the increase in the number of thrusters and with the increase in the maximum thrust which can be supplied by each thruster.

Future studies and developments on this work could include:

- a search for a more accurate lower-bound, considering a finer discretization and a set of time steps greater than the ones used in this thesis.
- a search for a thruster layout capable to compensate all the disturbances reported in this thesis, starting from the optimization in the most critical conditions.
- a reformulation of the model for the AOCS, which allows considering the thrust as a discontinuous variable from the beginning of the analysis. Otherwise, the approach used in this thesis can be followed, further increasing the number of thrusters or the maximum thrust until the error is cancelled.

References

[1] Danzmann K.: LISA Laser Interferometer Space Antenna. A proposal in response to the ESA call for L3 mission concepts. (2017)

[2] M. Armano et al.: Sub-Femto-g Free Fall for Space-Based Gravitational Wave Observatories: LISA Pathfinder Results. Phys. Rev. Lett., vol. 116, p. 231101 (2016)

[3] Bartusiak M.: Einstein's Unfinished Symphony: The Story of a Gamble, Two Black Holes, and a New Age of Astronomy. (2000)

[4] Anselmi A., Cesare S., Dionisio S., Fasano G., Massotti L.: Control Propellant minimization for the Next Generation Gravity Mission. In: Fasano G., Pintér J.D., Eds., Modeling and Optimization in Space Engineering – State of the Art and New Challenges. Springer International Publishing Switzerland. (2019)

[5] Fasano G.: Dynamic System Control Dispatch: A Global Optimization Approach. In: Fasano G., Pintér J.D., Eds., Modeling and Optimization in Space Engineering – 2019. Springer International Publishing Switzerland. (2019)

[6] Goldreich O.: Computational Complexity: A Conceptual Perspective. Cambridge University Press. (2008)

[7] Williams H. P.: Model Building in Mathematical Programming, 5th ed. (2013)

[8] Hillier F. S., Lieberman G. J.: Introduction to operation research. (2001)

[9] Nocedal J., Wright S.J.: Numerical Optimization. (2006)

[10] Lanzillotti A.: Optimization Methods for Thruster Layout. (2019)

[11] Horst R., Pardalos P. M., Thoai N. V.: Introduction to Global Optimization, Second Edition. Kluwer Academic Publishers (2000)

[12] https://www.ibm.com/it-it/analytics/cplex-optimizer

[13] https://it.mathworks.com/products/matlab.html

[14] https://lisa.nasa.gov

[15] https://www.lisamission.org

[16] Byrd R. H., Gilbert J. C., Nocedal J.: A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming, Mathematical Programming, Vol 89, No. 1, pp. 149–185. (2000)

[17] https://it.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html

Ringraziamenti

Un sentito ringraziamento al Dott. Giorgio Fasano, per i suoi insegnamenti ed il suo prezioso aiuto nello svolgimento di questa tesi. Ringrazio il Dott. Stefano Cesare, per l'opportunità di partecipare ad un progetto così importante e per la costante disponibilità. Ringrazio inoltre la Prof.ssa Manuela Battipede, per il suo entusiasmo e per la sua disponibilità.
Un ringraziamento speciale va ai miei genitori, per avermi permesso di affrontare questo percorso con tranquillità, per avermi sempre dato il loro appoggio ed il loro supporto.
Ci tengo a ringraziare mia sorella Giorgia perché con la sua dolcezza e la sua simpatia riesce sempre a farmi sorridere.
Grazie a tutto il resto della mia famiglia: nonni, zii, cugini, le piccole Alice e Tea, per l'affetto e i momenti di gioia passati insieme.
Un grazie particolare ad Antonio, per avermi sopportata anche in sessione e nel periodo pre-laurea quando ero praticamente insopportabile, ma soprattutto perché mi fa sentire amata ogni giorno.
Ringrazio Lidia per tutte le avventure passate insieme, per aver condiviso momenti belli e meno belli, per aver sempre il consiglio giusto da dare e per la sua positività instancabile e contagiosa.
Ringrazio Rosario perché, nonostante le nostre discussioni inutili in casa, so che è un amico su cui poter sempre contare (alla fine voglio il voto al discorso, mi sto impegnando a farlo meglio di quello che faccio di solito per il brindisi del compleanno).
Grazie a Francesca, per aver reso piacevoli anche giornate passate a studiare, a fare progetti, esercitazioni, etc, e per le nostre chiacchierate durante le "gite" alla Romana.
Grazie a tutte le amiche lontane: Silvia, Marta, Elisabetta, Laura, che sento sempre vicine.
Infine, grazie a tutti i compagni di pause caffè, serate techno, cene (e pranzi) random, partite a Citadels, etc. che hanno reso questi anni a Torino indimenticabili.

Federica