POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Optimal control allocation problem analysis and implementation for in-orbit activities

Tesi di Laurea Magistrale in
Space Engineering - Ingegneria Spaziale

Author: **Emanuele Tagnin**

Student ID: 969511
Advisor: Prof. Pierluigi Di Lizia
Co-advisors: Gioacchino Scirè
Academic Year: 2022-23

Abstract

The thesis studies the control allocation problem, which defines how a control action shall be distributed among a set of redundant actuators, in the orbit servicing mission work frame, regarding attitude/pose control. In-orbit servicing missions, that have become more frequent in the last decade to cope with debris population increment and effective spacecraft life time enhancement, are characterized by tight requirements in terms of reliability, availability and flexibility, because these types of activities require a wide span of manoeuvres to be executed, that go from far and close proximity operations, to target docking and manipulation. To ensure the mentioned requirements, a spacecraft must be equipped with multiple and different actuators. Thus, an optimal distribution must be investigated, considering that the allocation has to be performed in a rapid and simple manner. The investigation starts from a literature review of the theoretical definition of the problem and the algorithms that are implemented. During this review, it emerged that the present methodologies apply specific algorithms for a tailored mission, in which the focus in not servicing work frame, and computational time is generally shaded by other drivers. Therefore a significant advancement that the investigation brought is that, not only a clear organization of the methodologies is reported, but a transposition of algorithms generally applied in the aeronautical field, which are inherently simple, fast and robust, was conducted. This operation required a complete reformulation of the characteristic quantities. In addition, the evaluating parameters to assess the methods performance are rigorously stated, which are the ability to optimize a cost function, the computational time, the accuracy, which is the difference between the required and provided action, and the algorithm capabilities when dealing with unattainable actions. Hence a comparison between the algorithms is performed for reaction wheels and reaction control thrusters. The first part, therefore, outlines a practical discerning tool that an AOCS engineer can consult to select the optimal and suited method for their own application.
The second part, instead, shows the design phase for a in-orbit servicing mission for modular spacecraft the Company is developing. The mission demonstrates how different and diverse manoeuvres can be covered by mild modification of the allocator scheme, which is the dual side of the innovations the thesis brings. In fact, operations that cover control

authority shift, hardware separation and pose control are performed without changing the control logic, which is completely disregarded in the analysis, but changing certain aspects in the allocation. In addition, a robustness analysis was conducted to test the reliability of the allocating scheme when dealing with variable scattering and measurement noise. In conclusion, the survey did not limit in the simple definition of different allocation methodologies, but transformed a high number DOFs optimum problem, that involves multiple constraints on the actuators, into an intuitive robust and computationally light element whose test on a real application gives timely and effective response.

Key Words: control allocation, in-orbit service mission, control robustness, optimal solution

Abstract in lingua italiana

La presente tesi studia il problema di allocazione del controllo, che definisce come una azione di controllo debba essere distribuita su un set ridondante di attutatori, considerando l'ambito il controllo di assetto/posa per missioni di tipo *in-orbit servicing*. Questa tipologia, sempre più frequente nell'ultima decade per sopperire alla crescita di detriti spaziali e per prolungare la vita utile dei satelliti, è caratterizzata da stringenti requisiti in termini di disponibilità, flessibilità, e affidabilità, in quanto queste attività richiedo che un ampio ventaglio di manovre venga eseguito, da *far* e *close proximity operations*, a *target docking/manipulation*. Per garantire tali requisiti, un satellite deve essere equipaggiato con diversi attuatori. Perciò, una soluzione ottimale deve essere investigata, considerando che l'allocazione deve essere eseguita in maniera intuitiva e rapida. Lo studio parte da una revisione della presente letteratura riguardante la definizione teorica del problema e i diversi algoritmi che vengono implementati. Durante tale revisione, è emerso che le metodologie applicano specifici algoritmi per particolari missioni non riguardanti l'ambito di *in-orbit servicing*, dove il tempo computazionale viene spesso oscurato da altri requisiti prioritari. Una significativa innovazione, dunque, è emersa: non solo viene riportata una chiara organizzazione dei vari metodi, ma una trasposizione di algoritmi semplici, rapidi e robusti, applicati per lo più nel settore aeronautico, è stata condotta. Questa operazione ha richiesto una completa riformulazione delle quantità caratteristiche in esame. Inoltre, i parametri impiegati per la valutazione delle performance degli algoritmi sono stati rigorosamente definiti: l'abilità di minimizzare un costo, il tempo computazionale, la accuratezza (differenza tra azione richiesta e prodotta) e la capacità di sopperire ad azioni al di fuori delle capacità degli attuatori. Dunque, un paragone tra i vari algoritmi è stato condotto per *reaction wheels* e *reaction control thrusters*. Questa prima parte delinea un pratico e utile strumento che un ingegnere AOCS può consultare per aiutarlo nella scelta della strategia ottimale e ideale per la sua applicazione.
La seconda parte, invece, mostra il design per una missione di *in-orbit servicing* per un satellite modulare che l'Azienda sta sviluppando. La missione dimostra come differenti manovre possano essere eseguite con semplici modifiche sullo schema di distribuzione, che è il secondo aspetto innovativo della tesi. Infatti, operazioni che vanno dallo sposta-

mento della autorità di controllo, alla separazione tra parti di satellite e controllo di posa sono condotte senza modificare la logica del controllore, che viene messa in secondo piano nella analisi, ma cambiando certi aspetti nella allocazione. Successivamente, uno studio di robustezza è stato definito per testare la affidabilità dello schema di allocazione nel momento in cui incertezze sulle variabili e rumore nelle misurazioni vengono introdotti. In conclusione, il lavoro non si è limitato nella semplice definizione dei vari metodi di allocazione, ma ha trasformato un problema di ottimo con alto numero di gradi di libertà, che considera diversi vincoli sugli attuatori, in un semplice, robusto, computazionalmente rapido elemento il cui test su una applicazione reale ha dato efficaci e solidi risultati.

Parole chiave: allocazione del controllo, missioni di servizio in orbita, robustezza del controllo, soluzione ottimale

Contents

Introduction

Control allocation studies how a control action shall be distributed among a set of available and redundant actuators, subject to certain hardware limitations on their produced output, called effort. In most of orbit servicing applications, spacecrafts are equipped with different and redundant actuators for two main reasons: reliability, and flexibility. Multiple actuators can cope with the potential failure of some of them, while still being able to execute the mission operations. Control allocation, then, becomes extremely useful, as it may be asked to reconfigure the remaining controls to allow the mission to progress. Flexibility, on the other hand, is a hard requirement in orbit services, since a wide span of different manoeuvres, actions and operations have to be carried out, minimizing certain mission key parameters, such as propellant mass, in most of the cases. This aspect can be granted only if redundancy is considered. Control allocation focuses on how the actuators at disposal have to produce a desired output. The main peculiarity of control allocation, with respect to any other control logic, is to separate the loop into two independent, but connected, blocks: the high level control and the low level control, as show in Figure 1. The first block contains the control law, which gives an output v, this output is then manipulated in the allocation and u is computed. This quantity is then fed into the system characteristic equations, and the state is integrated.

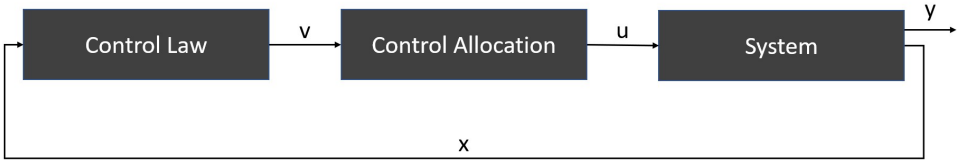

Figure 1: Block diagram describing the control loop

The control logic at a high level can be represented by various schemes, such as PID and LQR, and it is responsible for translating the system state error with respect to a given reference, taken as input, into a virtual effort v. Depending on the physical system in exam, v expresses different quantities, such as a force, torque or displacement. On the other hand, the low level control takes the action generated by the high level control and

distributes it among the set of actuators, translating it into an actuator-provided action u that is fed into the system, thus completing the control loop. Therefore, the high-level control is the part responsible of computing the virtual control action, while the low level shifts the quantity from a numerical world to a physical one, where the actuators are the ones responsible for generating the required action.
The question that naturally arises is why divide the control logic into two separate blocks. The primary reasons for doing so are threefold. First, against intuition, it actually simplifies the control definition, as it enables the user to separate how the control effort is defined from the actuators architecture. The control logic then can just compute the required action, and the allocation manages how this action is actually provided by the actuators. Secondly, a separate formulation gives more flexibility when it comes to designing an actual control system, as multiple iterations and corrections may change how the scheme is defined. If the control logic and actuator architecture are decoupled, a variation in one of the two does not necessarily imply that the whole control scheme needs to be modified. Finally most importantly, the control logic most of the times does not manage any actuator characterization such as hardware limitations, minimum and maximum outputs. This is the main driver when dealing with a control allocation problem: the necessity to express and highlight the physical features of the actuators in exam.
While control allocation for redundant sets of actuators in aircraft applications, both military and civil, has been extensively studied and analyzed in the literature, research on control allocation for space applications is comparatively less organized and thinner, especially in the context of in-orbit servicing or modular spacecraft. Typically, research in this field consists of individual papers or research essays that describe a specific strategy tailored to a theoretical mission [16, 22, 24–26]. Two main issues have arisen from literature research in this area. Firstly, few articles have attempted to compile the various methods and assess their unique features and capabilities, along with their optimal field of application. Secondly, it is unclear which parameters should be considered when evaluating these strategies, especially for applications that are characterized of long time spans. Although most algorithms are judged based on their ability to minimize a cost function, factors such as computational time, which is equally critical for on-board computing in real mission developments, are often ignored. This investigation aims to address these gaps by providing a comprehensive analysis of control allocation strategies for space applications, specifically in the context of in-orbit servicing and modular spacecrafts.
The objectives of the present dissertation are here listed:

- To collect the main strategies that have been used to solve the allocation problem, and to list their main features, in order to give the reader a practical tool to discern

which method is most suited for their application.

- To define the parameters that evaluate the control allocation procedure, to clarify which method is more suited for a certain application, with certain requirements.
- To demonstrate the application of the control allocation in a real developing mission scenario.

Even though the following analysis can be applied onto different system, the focus is brought onto the space activity work frame. The problem to be investigated in this survey is to distribute onto the actuators a virtual required torque in order to perform an attitude tracking problem. The actuators are defined by a characteristic effort, for example in the reaction wheels it is represented by the disk change of rotation rate. The effort is then translated into a torque through a function called effectiveness, which expresses the actuator architecture and on board disposition. The effort, moreover, is bounded in a feasible range, dictated by the hardware features. In the case of the reaction wheels, again, the lower and upper bounds may be the maximum change of rate.
The presented work is divided into three chapters. In the first part, the overall control loop model is described. It is made of fundamental building blocks that concur in the spacecraft attitude control problem definition. Here, all the elements that do not enter in the low level control field are briefly characterized. This part serves as an introduction and description on how the overall problem is modelled. The second chapter illustrates the allocation problem, how it is described and which parameters are fundamental drivers when assessing the performance. Different strategies are delineated and compared to determine their advantages and limitations when it comes to computational efficiency, implementation capability and such. In the third chapter, the control allocation is tested on a real developing mission scenario, in order to simulate the design of an AOCS subsystem for a in-orbit service satellite. The investigation pertains to a sensitive aspect of the company's project, which involves testing and verifying compliance with the AOCS requirements for pointing accuracy and fuel consumption. The study focuses on the early stages of the mission design (Phase A), where it is necessary to estimate requirements compliance. Throughout the description, hardware specifications and other critical parameters are omitted to ensure confidentiality. Nonetheless, the primary aim is to evaluate the control allocation subsystem and draw generalizable conclusions from this particular case, which is the focal point of the thesis. In this part the control allocation is evaluated also in terms of robustness with a Monte Carlo analysis, to test the behaviour of the allocation when the mission characteristics are off-nominal. Before introducing the investigation, it is necessary to define the convention followed regarding the vector expression. Throughout the survey, the variables are defined in the following manner:

- Vector are written in bold $\boldsymbol{a}$.
- Their norm in plain a or $||\boldsymbol{a}||$.
- The correspondent unitary vector $\hat{\boldsymbol{a}}$.
- The cross product as $\boldsymbol{a} \times \boldsymbol{b}$.

In addition, three main reference frames are considered: the inertial reference frame (IRF), the body reference frame (BRF) and the target reference frame (TRF). Vectors expressed in the first frame are reported with a capital letter, while with a lower case in the second and third ones.

Work innovations and radical progresses

As previously stated, there are multiple issues when dealing with control allocation that pertain to two different aspects. Regarding research field and literature, not only a clear organization and listing of the main strategies is required, but, most of all, the transposition of the strategies application from the aeronautical field to the space frame is necessary. In fact, efficient algorithms are mainly defined over deflectors for aircraft attitude control. On the other hand, regarding the actual implementation of the algorithms, it is necessary to define plainly which are the involving parameters that can be used to assess the performance, especially considering the computational efficiency, which has strict requirements for on-board installation. In addition to addressing these issues, other technical advancements are here presented that highlight the flexibility and reliability of the control allocation scheme:

- The application of fast and robust control allocation strategies, which are primarily studied and utilized in the aeronautical field [3, 5, 12, 20, 21], are also employed in the space industry on groups of actuators with different characteristics and formulations. While in aeronautics, the attitude control is generally handled through surface deflectors with angular displacement as the characteristic dimension, in space, reaction wheels and thrusters are commonly used. As a result, the manipulated quantities undergo a complete redefinition.
- The control allocation problem is rigorously stated, especially with the introduction of the cost function minimization (Chapter 2).
- The set of evaluating parameters for the control allocation is plainly defined, which are: output accuracy, defined as the error norm between required and produced actions (high level output v and low level output u), ability to minimize a cost

function, computational time and ability to respond to unattainable input action.

- Due to the shift of application, a stringent analysis is carried out when it comes to computational efficiency and unattainable required action response. In fact, on-board CPU must be extremely light for a reliable mission development and, since the analyzed mission in Chapter 3 has consistent inertia, thus high torque are required, it is necessary to exploit efficiently the redundancy at disposal.
- Definition of new solutions, as described in Section 2.5.
- The carried out analysis and comparison in Chapter 2, along with each method description, builds a reliable and intuitive tool that helps the AOCS designer to choose properly the correct strategy without influencing the control logic in any way.
- High number DOFs fast allocation is analyzed, and modular/cellular actuators architecture implementation is considered.
- Chapter 3 provides an example of the tool application in the design of a mission for in-orbit servicing that the Company is developing. The mission is a perfect demonstration of fast, flexible and reliable allocation for space work frame. As the spacecraft has to perform a wide span of manoeuvres, it is demonstrated how well suited is the allocation scheme when dealing with shifting control authority among actuators (section 3.2.2), response to sharp state variations (section 3.2.3), hardware variable scattering and measurements noise (section 3.3) and, most of all, the possibility to consider, without any change in the architecture, a pose control (section 3.2.4). It is vital to consider that all these operations are performed with intuitive, simple and straightforward modification in the allocation, without affecting in any way the high level control. In fact, in the analysis in Chapter 3, it is completely disregarded.
- Clear description, simulation and results analysis for a Monte Carlo simulation focused on the actuators architecture to show the robustness and response when the system is affected by scattering and noise.

To summarize, the listed operations can be seen as small but significant advancements with respect to the allocation state of the art. However, the key innovation is represented by the manipulation, and consequent transformation, of a high number DOFs optimum problem that is not treated as rigorously as in aeronautics, that involves actuators limitations, into an intuitive, robust and efficient element that completely disregards the controller and whose test on a real mission development gives a great and prompt response.

1 | Working frame analysis

Before introducing the control allocation analysis, it is necessary to describe all the elements that do not enter in this category, but are fundamental building blocks for the investigation. These blocks collect the parameters and quantities that enable the control allocation strategies to be evaluated and to simulate an orbiting spacecraft required to follow the target attitude. It is possible to define these main blocks as:

- Equation of Motions, dynamics
- Attitude parameters, kinematics
- Orbit propagation
- External disturbances, perturbations
- Control logic, high level control

Each block is interconnected to describe a closed control loop system, which simulates the effort of a set of redundant actuators to maintain a target attitude for a spacecraft, against external disturbances, during the orbital path. The control loop is shown in Figure 1.1. It not only emphasizes the interdependence and linkages between different blocks, but also delineates a logical progression that characterizes the simulation workflow: starting from the dynamics block, at each time instant, the angular velocity vector is integrated, which is then used to compute the body instantaneous attitude. The attitude is compared to the target one (which depend on the spacecraft orbit position), and the relative error is used as input for the controller. The required action to correct the body attitude, a net virtual torque, is split among the available actuators and the generated effect is plugged, along with perturbation, into the equations of motion and the loop repeats.

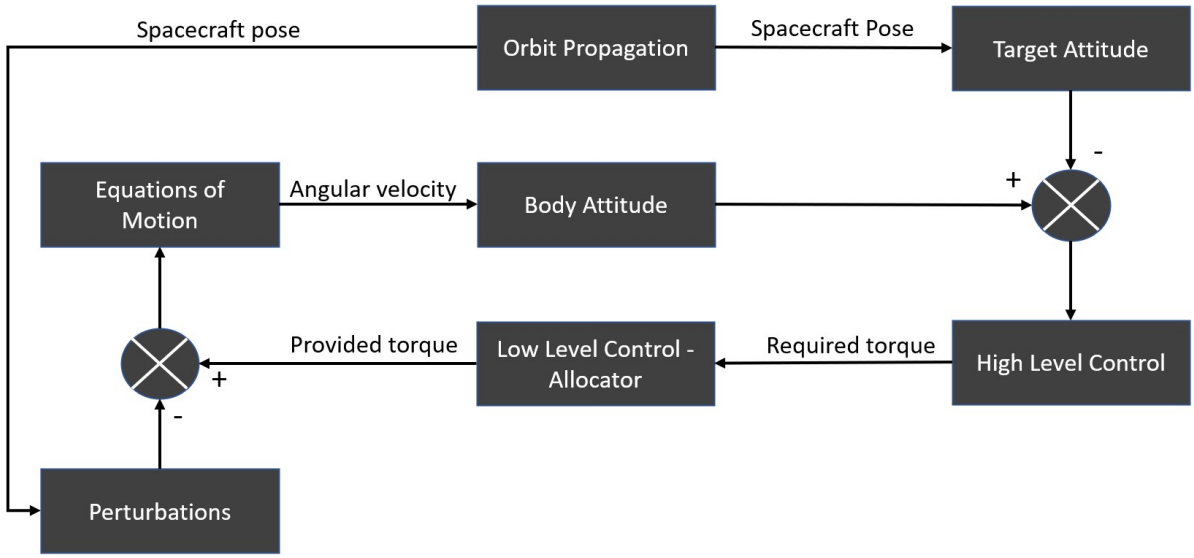

Figure 1.1: Refined control loop scheme for the attitude tracking problem

1.1. Equation of Motions (EoM)

The first block expresses the link between accelerations and external forces. Since only body attitude is considered, the accelerations are assumed to be angular throughout the analysis or, if not, clearly stated. Considering Euler equations of motion, the angular accelerations in 3-D space $\dot{\boldsymbol{\omega}}$ can be expressed in function of the spacecraft inertia matrix, angular velocity and external torques:

$$I\dot{\boldsymbol{\omega}} = (I\boldsymbol{\omega}) \times \boldsymbol{\omega} + \boldsymbol{T_{ext}} \tag{1.1}$$

Throughout the analysis, it is assumed that the body reference frame does not coincide with the principal axis reference frame, therefore the angular velocities are coupled through the inertia matrix, which is full, and the term $(I\boldsymbol{\omega}) \times \boldsymbol{\omega}$ is not zero. The integration, in addition, has to be solved numerically. $\boldsymbol{T_{ext}}$ contains the sum of all external effects, namely the actuator and environmental perturbation torques. The two elements, ideally, should give as output the necessary action to follow the prescribed attitude. For the simulation, the hypothesis of perfectly measured angular velocity is assumed, thus meaning that the solution of the Euler Equations and the measured angular velocities match. The assumption is considered to simplify the analysis, while taking into account noise does not actually affect the control allocation, since it will be a matter of performance degradation. It is possible, in any case, to apply a white noise onto the output of the EoM to model the error introduced in the act of measurement. In any case, if the

hypothesis is not considered, it will be plainly stated in the dissertation.

1.2. Kinematics

In order to mathematically depict the orientation of the body reference frame (BRF) with respect to an inertial reference frame (IRF), namely the spacecraft attitude, two different attitude parameters are employed: Direct Cosine Matrix (DCM) and Modified Rodrigues Parameters (RP). The former is utilized to define the relative orientation in relation to the IRF, to serve as an input for the control law, and to represent the desired attitude that the spacecraft should maintain throughout its orbital path. The latter is used as an indicator to evaluate the pointing accuracy of the satellite during simulation, which can be perceived as the discrepancy between the body and target attitude. The DCM is defined as a matrix transformation from one coordinate system into another, as shown in Equation (1.2), thus projecting a vector expressed in one reference frame, into another one. The DCM is orthogonal, which means that the opposite projection can be performed considering the transpose of the matrix.

$$\boldsymbol{x_{REF1}} = A_{body}\boldsymbol{x_{REF2}} \tag{1.2}$$

It is easy to understand that, while orbiting, if the body attitude changes, so does the DCM. This variation depends on the body angular velocity, as Equation (1.3) states.

$$\dot{A}_{body} = -\omega^{\wedge}A_{body},\ with\ \omega^{\wedge} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{1.3}$$

For what concerns the Modified Rodrigues Parameters, they are a modification of Euler Parameters that integrates the direction cosines of a rotation axis with tangent of half the rotation angle as three quantities. Recalling Euler Parameters definition, which consists in determining the attitude with four elements, that represent a vectorial part and a scalar one, the RP reduces these elements only considering a vectorial characterization, in 3-D. The relation between the two sets of parameters can be expressed as:

$$\boldsymbol{\sigma} = \frac{\boldsymbol{\beta}}{1+\beta_0} \tag{1.4}$$

As previously stated, RP are used for assessing the pointing performance of the spacecraft,

since they take into account far less variables, for an easier and simpler definition. Another key advantage that the modified RP possess is that the singularity is shifted to 360°, which is a rotational condition that is not encountered during the simulations. In addition, it is able to better represent the angular rotations when these quantities are small, which is the nominal condition in the analysis carried out. This fine property is highlighted in the relation between the RP and the euler axis and angle, as show in Equation (1.5).

$$\boldsymbol{\sigma} = \tan(\frac{\phi}{4})\boldsymbol{e} \tag{1.5}$$

For small angles, in fact, the RP can be linked directly with the euler angle divided by four. The modified RP can be also expressed in relation with the DCM, as the following equation illustrates:

$$\boldsymbol{\sigma} = \frac{1}{\xi(\xi+2)} \begin{bmatrix} A_{32} - A_{23} \\ A_{31} - A_{13} \\ A_{12} - A_{21} \end{bmatrix} \; with \; \xi = \sqrt{tr(A)+1} \tag{1.6}$$

In order to explain why RP are more suited for defining the accuracy with respect to the DCM, it is necessary to describe the target attitude and the error definition, which will be a direct input into the high level control.
As it orbits, the spacecraft is required to follow a prescribed attitude, represented by a direct cosine matrix A_{target}. The target DCM is defined by a set of three orthogonal unitary vectors. This orientation is the one the body attitude should match. For the simulation the following triad is used:

$$\begin{cases} \hat{\boldsymbol{x}} = \dfrac{\boldsymbol{h} \times \boldsymbol{s}}{||\boldsymbol{h} \times \boldsymbol{s}||} \\ \hat{\boldsymbol{z}} = \hat{\boldsymbol{n}} \times \hat{\boldsymbol{x}} \\ \hat{\boldsymbol{y}} = \hat{\boldsymbol{z}} \times \hat{\boldsymbol{x}} \end{cases} \tag{1.7}$$

The first vector reported in Equation (1.7) is the result obtained from the cross product between the orbit angular momentum vector $\boldsymbol{h}$ and the Sun position in body reference frame, scaled by the respective norm. The direction $\hat{\boldsymbol{z}}$ is, instead, computed as the cross product between the nadir direction and $\hat{\boldsymbol{x}}$, while $\hat{\boldsymbol{y}}$ completes the triad. Each unitary vector is computed at each time instant in order to update the target attitude. This particular orientation is characterized by two main features:

- The x axis is always normal to the Sun position.
- The z axis is always opposite to the orbit velocity.

The delineated attitude will be used to evaluate the performance of allocation methods in Chapter 2.
Now, the next step is to compute the discrepancy between the actual body attitude, and the target one. The disparity is expressed as the multiplication between the two DCMs A_{body} and A_{target}, as follows:

$$A_{err} = A_{body} A_{target}^{T} \tag{1.8}$$

If the two matrices are equal, then the error DCM will coincide with a diagonal matrix, due to the orthogonal property of this representation. Off-diagonal, non-zero terms will represent the mismatch of the two orientations, which has to be driven to zero. Geometrically, these terms can be seen as the three axis of the body and target reference frames difference. Zero error, in fact, would mean perfect overlapping between the two triads. Even though this is an intuitive definition of the pointing error, it is hard to evaluate numerically the accuracy, since nine variables have to be checked simultaneously, three to be at value 1, and the off-diagonal six to be zero. Moreover, as the error gets close to zero, it is hard to discern accuracy improvement with the DCM. RP, instead, makes use of just three parameters, which is easier and more manageable, while improving, with respect to the DCM, the accuracy trend visualization as the error grows thinner.

1.3. Orbit propagation

As the mission simulation evolves in time, the spacecraft follows an orbital path while keeping the prescribed attitude. It is important to determine the trajectory progression as the BRF position of key elements, such as celestial bodies, will change over time. Therefore the same target attitude previously discussed will vary in time. Moreover, environmental perturbations intensity and characterization strongly depend on the spacecraft pose in orbit, thus modifying the controller torque request. To propagate the orbit, an independent side block is defined which integrates, at each time step, the unperturbed keplerian elements. This block will take as input the assumed constants keplerian elements and the true anomaly to compute algebraically its variation and the spacecraft pose with respect to the IRF. Since the focus of this work is not the flight dynamics and the correct trajectory integration, this propagating method is used as it is extremely simple, computationally fast and efficient. The procedure is defined as follows:

Algorithm 1.1 Orbit propagation through keplerian elements

1: Compute $n = \sqrt{\frac{\mu}{a^3}}$

2: Compute $\dot{\theta} = \frac{n(1 + e\cos\theta)^2}{(1 - e^2)^{\frac{3}{2}}}$

3: Compute $R = a\frac{1 - e^2}{1 + e\cos\theta}$

4: Compute $p = a(1 - e^2)$

5: Compute $\boldsymbol{R} = R \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix}$

6: Compute $\boldsymbol{V} = \sqrt{\frac{\mu}{p}} \begin{bmatrix} -\sin\theta \\ e + \cos\theta \\ 0 \end{bmatrix}$

1.4. Environmental perturbations

To better reproduce a real mission application, a series of environmental perturbations are considered in the model, which have to be counter acted by the control scheme. These elements are described by a mathematical model that emulates the real effects on the spacecraft. The environment presence will degrade the pointing accuracy throughout the mission, because spurious unwanted torque will be applied onto the satellite during its motion, and thus major effort will be required by the actuators. For the simulation, the disturbances are modelled as pure spurious torque action, which will affect the target attitude tracking, while perturbations in terms of forces are neglected. Therefore it is key to underline an important assumption: if a perturbation element generates a force which is not applied on the spacecraft centre of gravity, the net translational contribution is not considered, while the rotational part is kept. This is due to the fact that the first part and the relative error in the pose that the force would generate does no does not affect directly the pointing performance and requirements, but the satellite navigation, a secondary element in the investigation. In addition, elements that energetically affect the orbit propagation are excluded, keeping the main focus on the attitude tracking problem. The intrusive environmental elements are:

- Drag due to the Earth's atmosphere
- Gravity Gradient
- Solar radiation pressure

Earth's magnetic field is not included, since it is generally hard to determine the dipole moment of the spacecraft in early stages of the development[1].

1.4.1. Air drag

The presence of the atmosphere, even though rarefied, can affect the spacecraft attitude, especially if the satellite's net surface is wide and the orbit is low. Exchange of kinetic energy due to collision between the spacecraft and air molecules may generate a perturbing effect in the attitude. The impact can be modelled as a force generated onto the centre of pressure of a body, depending on the velocity and exposed body surface to the air flow. The force, which is not necessarily aligned with the centre of gravity, will generate a torque, the greater as the distance between the centre of mass and pressure increases. In the following equation the model is reported:

$$\boldsymbol{F_{drag}} = -\frac{1}{2} C_d \rho ||\boldsymbol{v}||^2 A_{net} \hat{\boldsymbol{v}} \tag{1.9}$$

Where:

- C_d is the drag coefficient of the spacecraft.
- ρ is the density of the air at orbiting altitude.
- $\boldsymbol{v}$ is the velocity vector of the satellite in BRF.
- A_{net} is obtained from the dot product between the normal to the area and $\hat{\boldsymbol{v}}$.

The force is assumed to be applied in the centre of pressure of the different bodies that composes the spacecraft, an example is shown in Chapter 3. The resulting torque is calculated, with respect to the centre of gravity, as:

$$\boldsymbol{T_{drag}} = \boldsymbol{r_{CoP}} \times \boldsymbol{F_{drag}} \tag{1.10}$$

In the computations, air is modelled as a continuum with density decreasing with respect to the altitude with an exponential model. As Equation (1.11) reports, the density is determined by the product between the standard value at sea level ρ_0 and an exponential factor. The latter contains the information of the altitude difference with respect H_0, which is the value corresponding to ρ_0. Ultimately, the difference is divided by a scaling constant H_{scale}.

[1] It is possible to assume a rough value and implement the Simulink block Earth's Magnetic Field nonetheless

$$\rho = \rho_0 e^{-\frac{H - H_0}{H_{scale}}} \tag{1.11}$$

The main issue regarding this model is defining the net area onto which the air molecules impinge, along with the estimation of the drag coefficient. Regarding the latter parameter, it is possible to use a satellite baseline and use the same coefficient as a preliminary analysis. Regarding the areas, it is possible to split the spacecraft main parts into simple geometrical shapes. Solar panels, for example, can be viewed as thin panels. Using symmetrical figures also helps to locate easily the centre of pressure. The main body, on the other hand, can be modelled as a cylinder or a prism.

1.4.2. Gravity Gradient Torque (GGT)

The target attitude that the spacecraft needs to track does not coincide with a stable configuration for the gravity gradient. GGT is then taken into account, as it is present throughout the orbit, and increases as appendages, such as solar panels, increase dimensions. The perturbation torque that arises from the non-uniform mass distribution, for a general inertia matrix is reported in Equation (1.12). The moment will depend on the earth constant μ, the inertia tensor and the orbital position of the spacecraft expressed in body reference frame $\boldsymbol{r}$.

$$\boldsymbol{T_{GGT}} = \frac{3\mu}{||\boldsymbol{r}||^5}(\boldsymbol{r} \times (I\boldsymbol{r})) \tag{1.12}$$

As for the air drag, the further from Earth, the lower the contribution. Regarding this type of perturbation, a purely rotational torque is generated.

1.4.3. SRP

Solar wind and electromagnetic flux coming from the Sun can generate a similar phenomenon to the air drag. In this frame, small particles can exchange momentum through the impact with the satellite exposed surface at light speed. Following the same procedure as for the air drag, it is possible to model this effect as a net torque, which will depend on the position of the centre of pressure and the net areas that are exposed to the Sun. As the produced torque will directly depend on the incoming radiation, it is going to be a combination of contribution of the reflected, absorbed and diffused radiation after the impact, and the power per unit area delivered by the Sun. The expression of the generated moment is written in the next equation:

$$\boldsymbol{T_{SRP}} = -P_w A_{net} \left[(1-\rho_s)\hat{\boldsymbol{s}} + (2\rho_s(\hat{\boldsymbol{s}} \cdot \hat{\boldsymbol{n}}_{\boldsymbol{surf}}) + \frac{2}{3}\rho_d)\hat{\boldsymbol{n}}_{\boldsymbol{surf}} \right] \tag{1.13}$$

Where:

- P_w is the ratio between the power per unit surface exerted from the Sun and the speed of light.
- ρ_s is the coefficient of the incoming radiation which is specular reflected.
- $\hat{\boldsymbol{s}}$ is the unitary vector representing the direction of the Sun in BRF.
- $\hat{\boldsymbol{n}}_{\boldsymbol{surf}}$ is the normal direction of the considered surface.
- ρ_d represent the percentage of the radiation diffused reflection.

The overall torque will be, then, the sum of the total incoming radiation, scaled by the reflective and absorbing properties of the materials which face the Sun. Of course it is easy to conclude that, if shaded by the Earth, the SRP torque will be zero. As a preliminary analysis, it is possible to retrieve the values of the coefficients from look-up tables or baselines.

1.5. High level control

The last block to be analyzed is the one that computes the required torque to track the target attitude. The task of this block can be fulfilled by different controller types. For the simulation, since it is required computational efficiency and speed, a rather simple but effective PID control is implemented. The choice is also driven by the fact that the PID control does not influence the control allocation as it simply computes the torque depending on the error fed. This condition is desirable in order to assess the performance of the allocators. As previously stated, the error can be highlighted from the DCM A_{err}, where its components will be defined as:

$$A_{err} = A_{body} A_{target}^T = \begin{bmatrix} 1 & \epsilon_z & -\epsilon_y \\ -\epsilon_z & 1 & \epsilon_x \\ \epsilon_y & -\epsilon_x & 1 \end{bmatrix} \tag{1.14}$$

As previously stated, the off-diagonal terms are related to the pointing error, in the three directions. If brought to zero, then A_{err} will be equal to an identity matrix, and the required attitude is being tracked successfully. If all the components are collected in the vector $\boldsymbol{\epsilon}$, the controller torque is computed as:

$$\boldsymbol{m} = \boldsymbol{K_p}\boldsymbol{\epsilon} + \boldsymbol{K_d}\dot{\boldsymbol{\epsilon}} + \boldsymbol{K_i}\int \boldsymbol{\epsilon} \tag{1.15}$$

As shown in Equation (1.15), the torque is described by a linear combination of three contributions, the integral, proportional and derivative part of the error, scaled by fitting gains, which can be computed through trial and error approach. Even though simple and trivial, it is possible to manipulate the gains to control the tracking response, which gives the user flexibility. Another advantage of this procedure is clearly the simplicity of the control logic definition, as the pointing error is directly available and plugged into the PID. The usage of A_{err}, moreover, gives the user a clear and straightforward view on the relation between attitude error and direction of the required torque, as the components of the vector $\boldsymbol{\epsilon}$ corresponds to the components of the vector $\boldsymbol{m}$.

2 | Control allocation methodologies

In this chapter, different strategies for control allocation are studied. The investigation is mainly driven by the accuracy of the solution, computational cost and ability to optimize mission parameters. An initial introduction to the control distribution problem and analysis of the main drivers, definitions and key parameters is carried out. Then, the different algorithms present in literature are shown, explained and evaluated in order to give a first idea of the main advantages and limitations of each strategy. Along the description, a pseudo-code algorithm is attached to better grasp the logical process and ease code creation. To further investigate the features of each algorithm, they are applied in a spacecraft control simulation, onto a set of redundant reaction wheels (RW) and reaction control thruster (RCT), for different orbital families. Throughout the whole survey, the attitude tracking problem is considered, with the high level control generating a required torque, which needs to be distributed onto the actuators.

2.1. Allocation theory

Initially, it is essential to elaborate on the concept of control distribution and elucidate the way in which the problem is formulated[1]. Control allocation studies how the vector of required action $\boldsymbol{m} \in \mathbb{R}^n$, output of the high level control, can be mapped over a set of actuator DOFs, which can be represented by variables collected in the vector $\boldsymbol{u} \in \mathbb{R}^m$, in an optimal and suited way. In the case taken into exam, $\boldsymbol{m}$ will represent the set of required control torque computed, as stated before, by the high level control, thus $n = 3$. On the other hand, $\boldsymbol{u}$ will represent the actuators degrees of freedom. Depending on the type of actuator in exam, the value in $\boldsymbol{u}$ can represent different physical quantities, for example, a magnitude, angular position and such. Each actuator generally is able to generate a single action, which is concatenated with the others in $\boldsymbol{u}$. Hence, the i^{th} element of $\boldsymbol{u}$ u^i is called effort of the i^{th} actuator. The set of these actuators can vary depending on the mission type and spacecraft configuration, thus the length of $\boldsymbol{u}$ is not fixed, but changes along the number of producible efforts.

[1] Throughout the whole chapter the same nomenclature and definitions in [5] are followed.

The two spaces where $\boldsymbol{m}$ and $\boldsymbol{u}$ lie, respectively the moment space $\mathbb{R}^n$ and control space $\mathbb{R}^m$, are linked together through a function called effectiveness $B : \mathbb{R}^m \rightarrow \mathbb{R}^n$. This function can be seen as the projection of the m-dimensional actuator effort, into a n-dimensional torque vector. In the case of RCT, for example, B contains the information regarding the position and orientation of the thruster with respect to the body reference frame, so that the effort of each thruster is converted into a moment. In the analyzed framework, B is assumed to be linear and time-invariant. The first hypothesis states that the effectiveness can be seen as a set of gains, that multiply $\boldsymbol{u}$. This is applicable in both RW and RCTs cases, where the effort u^i intrinsically generates a torque, which can be enhanced or reduced by the gain in B. In RCTs, for example, the effort, which can be linked to the thrust, is cross multiplied by the arm, which is a geometrical fixed constant, and depending on the arm length, the produced torque will vary as well. The geometrical information are collected in the effectiveness, which logically defines how "effective" is a certain actuator. The second aspect, instead, means that the actuators positioning and orientation does not change over time. This solution is commonly implemented in this kind of problem, since not only it simplifies the analysis, but also practical applications do not require B to change, especially if the actuators dynamic response is fast with respect to the system nominal working frequencies. However, it is possible to apply the following analysis including variations on the function definition if these are assumed instantaneous, or the update frequency is higher than the controller one. If these conditions are met, the following analysis is still applicable, since the system will have a compliant effectiveness B at each time instant. If the above hypothesis are considered, B can be represented as an $n \times m$ matrix, a linear mapping that connects control space ($\mathbb{R}^m$) and moment space ($\mathbb{R}^n$). Each i^{th} column of B will correspond to u^i, thus determining whether the considered effort can generate a torque in the n directions. So, for example, if the i^{th} column of B has a zero entry in the j^{th} row, the actuator u_i is not able to generate a torque in the j^{th} direction.

Regarding control allocation, the trivial problem is represented by having $m = n$, thus meaning that the distribution has only one solution, which can be determined by solving the linear system $\boldsymbol{m} = B\boldsymbol{u}$ with the effectiveness matrix being square. Though extremely simple and straightforward, this problem rarely represents a real case, where spacecrafts are equipped with multiple actuators for redundancy and failure occurrence. Moreover, redundant actuator architectures generate a more flexible and robust system, which are two vital aspects in space operations. Regarding specifically in-orbit servicing, redundancy is required to have a precise pointing accuracy and adaptable system to cope with different actions and activities. For example, modular and cellularized spacecrafts are inherently defined by groups of redundant actuators. Therefore, the studied case will consider $m > n$,

which implies that the linear system $\boldsymbol{m} = B\boldsymbol{u}$ has an infinite number of solutions. Of course the case $m < n$ is degenerate, since the actuator degrees of freedom are not able to generate independently an action in all n directions, which is not admissible in most of in-orbit applications. In addition to reliability, redundancy in the system adds the possibility of cost minimization, for example propellant mass or moment storage, which can be a driver in the mission. Therefore, control allocation is the investigation of an optimal set of control commands on the actuator sets among the different multiple solutions that can generate the required control torque while taking into account other factors important to the mission. The problem can be stated as reported in Equation (2.1).

$$find\ \boldsymbol{u}\ s.t.\ \begin{cases} \boldsymbol{m} - B\boldsymbol{u} = 0 \\ p\ is\ minimum \\ u^i_{min} \leq u^i \leq u^i_{max} \quad for\ i = 1:m \end{cases} \tag{2.1}$$

Where p is a scalar cost function. On the other hand, u^i_{min} and u^i_{max} define the feasible range boundaries of the control variable $\boldsymbol{u}^i$. This interval represents the actuator hardware limitations, namely maximum and minimum efforts. These minimum and maximum values, for each control variable, generate a feasible enclosed subset of the control space named Ω admissible or feasible control set, defined as:

$$\Omega = \{\boldsymbol{u} \in \mathbb{R}^m | u^i_{min} \leq \boldsymbol{u} \leq u^i_{max} \quad for\ i = 1:m\} \in \mathbb{R}^m \tag{2.2}$$

The subset is closed and bounded and represents a geometrical m-dimensional (m-D) polyhedron of feasible and admissible control, specifically it is the envelope of possible control efforts. The contour $\partial\Omega$ represents, instead, the region where at least one actuator has reached saturation, thus u_{min} or u_{max}. If a linear mapping is considered, it is possible to defined the n-D subset Φ which corresponds to Ω, called attainable moment set, which is the torque envelope the system can generate through its actuator configuration. Since B is linear time invariant, also Φ will be an enclosed and bounded subset. Moments which lie on the boundary of Φ are denoted by $\partial\Phi$, and represent the maximum torque the actuators can generate in a specific direction. Before introducing the analysis of the different methods, it is necessary to identify some key elements:

- The degree of redundancy is defined as $m - n$ and it expresses how much redundant is the actuator configuration. In the studied work frame, it is lower bounded to zero.
- A subspace is a lower dimensional portion of a higher dimensional space.

- Linear variety is a subspace which may have been translated from the origin. An n-dimensional linear variety is referred to as an n-flat.
- Residual $\boldsymbol{m}_{residual}$ is the numerical discrepancy from required to produced torque: $\boldsymbol{m}_{residual} = \boldsymbol{m} - B\boldsymbol{u}$.

In conclusion, each strategy is evaluated and compared each other under three main aspects: the computational efficiency and cost minimization, which are key for in-orbit operations, and accuracy in the problem solution reported in Equation (2.1).

2.2. Generalized Inverse (GI)

The first method to be analyzed is the rather trivial and most straightforward allocation through the Generalized Inverse. This strategy involves the generation of a matrix P which satisfies the equation $BP = I_n$, where I_n is an identity matrix nxn, while the matrix P will be mxn. P forms a basis for an n-dimensional subspace of $\mathbb{R}^m$, called P_s. Since this basis contains the origin, and as long as the origin is contained within Ω, these two elements are guaranteed to intersect. It is necessary to underline one of the requirements the control space has to cope with: Ω must contain the origin. By mapping the intersection set using B, all the moments for which the generalized inverse will allocate admissible controls can be seen. A simple and effective way to specify the Generalized Inverse is through the implementation of the pseudo-inverse. It solves through the minimization of the l_2 norm of the control variable vector. The problem statement is the following:

$$Minimize\ \boldsymbol{u}^{\boldsymbol{T}}\boldsymbol{u}\ \ s.t.\ B\boldsymbol{u} = \boldsymbol{m} \tag{2.3}$$

The problem is solved by defining a scalar function $H(\boldsymbol{u}, \lambda)$ such that:

$$H(\boldsymbol{u}, \lambda) = \frac{1}{2}\boldsymbol{u}^{\boldsymbol{T}}\boldsymbol{u} + \boldsymbol{\lambda}^{\boldsymbol{T}}(\boldsymbol{m} - B\boldsymbol{u}) \tag{2.4}$$

Where $\boldsymbol{\lambda}$ is a $nx1$ vector of Langrange multipliers which enforces the constraints on the problem. Hence, the partial derivatives with respect to $\boldsymbol{u}$ and $\boldsymbol{\lambda}$ of H are computed:

$$\begin{cases} \dfrac{\partial H(\boldsymbol{u}, \boldsymbol{\lambda})}{\partial \boldsymbol{u}} = \boldsymbol{u}^{\boldsymbol{T}} - \boldsymbol{\lambda}^{\boldsymbol{T}} B \\ \dfrac{\partial H(\boldsymbol{u}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \boldsymbol{m} - B\boldsymbol{u} \end{cases} \tag{2.5}$$

If the two equations are equal to zero, then a minimum is found with respect to H:

$$\begin{cases} \boldsymbol{u} = B^T \boldsymbol{\lambda} \\ B\boldsymbol{u} = \boldsymbol{m} \end{cases} \tag{2.6}$$

By plugging the first equation shown in Equation (2.6) into the second one, it is possible to solve for $\boldsymbol{\lambda}$, therefore the solution $\boldsymbol{u}$ can be calculated:

$$\begin{cases} \boldsymbol{\lambda} = (BB^T)^{-1}\boldsymbol{m} \\ \boldsymbol{u} = B^T(BB^T)^{-1}\boldsymbol{m} \end{cases} \tag{2.7}$$

Hence, a solution u is found by solving a minimum problem. Some important remarks can be stated:

- $(BB^T)^{-1}$ is an $n \times n$ matrix invertible if the effectiveness matrix has rank n.
- In this case the matrix $B^T(BB^T)^{-1}$ is the Generalized Inverse matrix P.
- $BP = I_n$, $\dfrac{\partial^2 H(\boldsymbol{u}, \boldsymbol{\lambda})}{\partial^2 \boldsymbol{u}} = 1$ and $\dfrac{\partial^2 H(\boldsymbol{u}, \boldsymbol{\lambda})}{\partial^2 \boldsymbol{\lambda}} = 0$, so the solution computed is a minimum.

The presented method is just one of the infinite possible solutions of the allocation problem. However, when specifying the generalized inverse, it is not possible to define freely all $m \times n$ elements, since it is required $BP = I_n$ to be granted. When a generalized inverse is identified, an n-D subspace in an m-flat is defined as well. Therefore the number of degrees of freedom is $(m-n)(n+1)$. Moreover, if the flat has a set of points to be fixed, the DOFs decreases as $(m-n)(n-p+1)$ where p is the number of fixed points. In this case, since the GI is a subspace, the origin is fixed. Finally, the overall number of DOFs is $(m-n)n$.
Another strategy to build the generalized inverse is through the implementation of a weighting matrix W, which gives $P = W(BW)^T[BW(BW)^T]^{-1}$. A diagonal matrix is typically used to represent W because it has a clear and intuitive interpretation. Each diagonal value measures how heavily weighted and preferred is one particular control degree of freedom with respect the others. If this procedure could give more flexibility regarding the limitations on the control and a way of expressing preference in the actuator activation, it is not efficient in terms of total tunable variables. In fact, for a diagonal matrix, there are m variable coefficients. For example, for four control variables and three torques, P has three DOFs, while W has four, thus meaning that there are generalized inverse which can not be generated through W. Even though this strategy can easily find a solution to the allocation problem, it is affected by multiple limitations. First, it does not

include the cost minimization, as it just focuses on the minimization of the control vector l_2 norm. Of course, the two aspects are generally correlated, but this method does not explicitly take the cost problem into consideration, shading part of the control allocation problem. Secondly, in the shown procedure, control value feasibility did not take part in the computations. Therefore the GI of a certain $\boldsymbol{m}$ may solve the problem with the generation of an unfeasible control vector, with elements that exceed the nominal working regime. It is required, then, a post process to enforce "manually" the actuator limitations onto the solution found. This will generate a residual error vector between the required torque and provided one. This major limitation affects intrinsically the method, and it is particularly detrimental since not only there may be a residual in terms of magnitude, but also direction, thus generating a coarse control response in the loop. On the other hand, as Algorithm 2.1 shows, it is clear to see the great advantage this strategy has, which is the simplicity and computational speed.

Algorithm 2.1 Generalized Inverse through the Pseudo-Inverse

1: Given B, $\boldsymbol{m}$ and the boundaries $\boldsymbol{u_{min}}$ and $\boldsymbol{u_{max}}$
2: Compute the pseudo-inverse $P = B^T(BB^T)^{-1}$
3: Compute the control vector $\boldsymbol{u} = P\boldsymbol{m}$
4: **for** $i = 1 : m$ **do**
5: **if** u^i *exceeds saturation* **then**
6: $u_i = u^i_{min}$ *or* u^i_{max}
7: **end if**
8: **end for**
9: Compute the residual $\boldsymbol{m}_{residual} = \boldsymbol{m} - B\boldsymbol{u}$

In conclusion, it is possible to see the GI through pseudo-inverse as as fast and rough allocation that disregards the geometrical definition of control and moment spaces. This method, then, is particularly useful when actuators are not affected by tight feasibility ranges, and fast solution must be computed.

2.3. Cascade Generalized Inverse (CGI)

In order to cope with the limitations regarding the Generalized Inverse [5], the CGI exploits the intrinsic redundancy of the system and redistribute the residual on the set of actuators that are not saturated. This is performed iteratively until all efforts are at maximum acceptable value or the residual is under required tolerance. The Cascade Generalized Inverse computes the solution through a specific GI, then checks each control

variable u_i of the vector $\boldsymbol{u}$. If it exceeds the valid range, namely it is greater than the saturation level, the value is brought to saturation. The control vector $\boldsymbol{u}$ will not produce the required torque, hence the residual is computed. This quantity is now fed again as input torque $\boldsymbol{m}$ into the control allocation scheme. This time, however, the effectiveness matrix is modified in order to omit the saturated actuator in the problem solving. Mathematically, this means removing the column B_i of the effectiveness matrix which corresponds to u_i that is at its limit value. Again, the distribution is performed through the Generalized Inverse, and exceeded variables are brought to saturation. The residual is newly computed and redistributed onto the set of available actuators. The effect of this redistribution effort is to enlarge the region in moment space for which the scheme will allocate admissible controls.
Even though the CGI is more capable, with respect to the GI, of coping with actuator limitations, and exploits in a more efficient way the redundant nature of the system, still it is an indirect way to solve the allocation problem, since the user is not able to actually impose the actuator range in a straight forward manner, but iteratively tries to lower the residual until the set of actuators are capable of coping with the redistribution. Moreover, it is key to understand that this procedure can be followed as long as the degree of redundancy $m-n \geq 0$, thus meaning that a reallocation is possible if the actuators left are able to actually produce the required residual torque. Therefore the residual at the end of the iterative cycle may be still nonzero. This procedure can be used with either pseudo-inverse, or generalized inverse weighted with a diagonal N matrix.

Algorithm 2.2 Cascade Generalized Inverse through the Pseudo-Inverse

1: Given B, $\boldsymbol{m}$ and the boundaries $\boldsymbol{u_{min}}$ and $\boldsymbol{u_{max}}$
2: Compute $\boldsymbol{m^0_{res}}$ and $\boldsymbol{u}$ through the GI algorithm
3: **while** $||\boldsymbol{m^k_{res}}|| > tolerance$ *or* $rank(B) \geq n$ **do**
4: Omit the column of B that corresponds to a saturated actuator
5: Perform GI with $\boldsymbol{m} = \boldsymbol{m^k_{res}}$ and reduced B
6: Compute $\boldsymbol{u^{k+1}} = \boldsymbol{u} + \boldsymbol{u^k}$ which is the accumulation of the control effort after each redistribution
7: Compute the new residual as $\boldsymbol{m^{k+1}_{res}} = \boldsymbol{m} - B\boldsymbol{u}^{k+1}$
8: **end while**

Even though this method is extremely intuitive, straightforward and easy to implement, it lacks of flexibility and does not consider again direct cost minimization. Moreover, as the GI, saturation constraints are manually applied to the solution, which may lead to a loss in the accuracy. In fact, the residual may be still both in magnitude and direction.

However, it is a step forward a more efficient solution with the basic GI, as the system is now more robust when it comes to deal with unattainable moments.

2.4. Null-Space solution (NS)

The Null-Space intersection method is particularly useful when tight constraints on the control variables are present, namely, configurations which have a particular definition of the feasible region Ω, or the control subspace does not comprise the surrounding of the origin. It is, then, through the effectiveness that the effort is translated into a torque. Considering RCTs with fixed orientation, for example, each thruster can generate, in its relative reference frame, only a contribution in one positive direction. Hence, Ω can approach the origin of $\mathbb{R}^m$ only from a positive region. With this particular configuration, the implementation of a Generalized Inverse could generate a negative output in the control vector $\boldsymbol{u}$, which does not meet the actuators constraints. The manual post process may require multiple actuators to saturation and the residual would be unbearable. Geometrically, if an allocation scheme asks a control to exceed its limit, then the vector position is outside the control space. The Null-Space solution is based on the traslation of the feasible control region Ω in order to intersect the required control vector while meeting up with the actuator constraints. In other words, it is possible to consider the Null-Space as a set of particular active actuator configurations different from zero that will generate a net zero torque or zero torque variation $B\boldsymbol{u}_{null} = \mathbf{0}$. By summing a null configuration and the particular one required for the control, a new solution is generated such that both control and actuator requirements are satisfied, as state in Equation (2.8).

$$\boldsymbol{m} = B\boldsymbol{u_p} + B\boldsymbol{u_{null}} \; with \; \boldsymbol{u_p} = P\boldsymbol{m} \tag{2.8}$$

The new control vector will be a combination of the effort required by the null space and the one required to solve the control allocation. Geometrically, this operation can be seen as a translation of Ω of a quantity defined by $\boldsymbol{u_{null}}$, an $m-n$ flat, thus shifting the feasible region and, possibly, intersecting the solution vector. This operation is granted by the fact that, for an $n \times m$ matrix B with rank equal to n, there exist a set of $m-n$ orthonormal vectors which forms a matrix $m \times (m-n)$, the Null Space[2] N of B, which is orhtogonal to P, the pseudo-inverse of B. Therefore it is possible to build the solution as a linear combination of the two contributions.
In order to find the vector $\boldsymbol{u_{null}}$, it is necessary to partition the solution obtained from

[2]This matrix can be easily computed through singular-value decomposition with the Matlab function null.

the pseudo-inverse $\boldsymbol{u_p}$ into two parts containing the saturated and feasible elements of the vector, respectively $\boldsymbol{u_s}$ and $\boldsymbol{u_f}$. The same procedure is applied for N, thus N_s and N_f are computed. The null control vector can be found as reported in Equation (2.9), with P_{N_s} being the pseudo-inverse of N_f, while $\boldsymbol{u_{lim}}$ the vector corresponding to $\boldsymbol{u_s}$ with entries equivalent to the saturated values[3].

$$\boldsymbol{u_{null}} = N(P_{N_s}(\boldsymbol{u_{lim}} - \boldsymbol{u_s})) \tag{2.9}$$

The presented method can be, although intuitive, quite expensive in terms of computational cost, since the different combinations of maximum and minimum values have to be tested in $\boldsymbol{u_{lim}}$ to find the feasible solution. Instead of applying the whole algorithm, it is possible to simplify the allocation problem, for a certain configuration of B. Considering the RCTs, the effectiveness matrix may be characterized in a way such that, if $\boldsymbol{u}$ has the same value for each component, the net torque is zero, because coupled nozzle will generate the same but opposite force. Hence a particular feasible solution $\boldsymbol{u_{null}}$ is immediately available, it will be characterized by a vector $m \times 1$ of same values, corresponding to the effort of $\boldsymbol{u}$ that is the furthest from the feasible region, which is going to be, generally, the most negative for the RCT case[4]. Therefore the final solution will have the minimum negative brought to zero, and the overall effort of the other actuators incremented. This process is extremely straightforward and simple but it is fundamental to understand that this shift needs to be applied to the whole control vector. If an actuator is particularly loaded, meaning that u_i is close to the saturation level, the added null contribution may let it exceed feasible range. It is still necessary a post process to ensure that $\boldsymbol{u}$ is still bounded within $\partial\Omega$. In Figure 2.1, an intuitive visualization of the the null vector sum operation is presented in a 2-D plot. As the orange axis shows, the application of the null vector can be seen as a general translation of the zero axis, so that the overall control effort will be brought to higher values, but compliant with the feasible ranges. As Algorithm 2.3 illustrates instead, the procedure can be seen as a particular case of post process in the Generalized Inverse method, where the actuator constraints are imposed through the null vector, while, in the GI, this is performed by checking each component of the control vector and enforce the saturation. In conclusion, it is clear to see that the method is extremely light regarding the computational cost, in expense of a higher general effort of the control vector, which could go against the mission cost minimization.

[3] Thorough explanation can be found in [5].

[4] This explanation is delineated to introduce the architecture in Section 3.1.

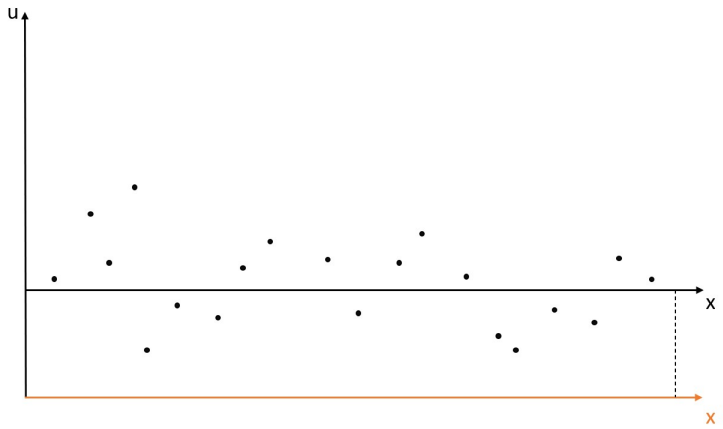

Figure 2.1: Visualization of the null vector sum effect

Algorithm 2.3 Null-Space allocation for lower bounded control vector - RCT example

1: Given B, $\boldsymbol{m}$ and the boundaries $\boldsymbol{u_{min}}$ and $\boldsymbol{u_{max}}$
2: Compute $\boldsymbol{u_p}$ through the GI algorithm
3: Compute the $\boldsymbol{u} = \boldsymbol{u_p} + |min(\boldsymbol{u_p})|$
4: Post process the result to check if the opposite saturation limit is reached

2.5. Direct Allocation (DA) and Pseudo-Direct Allocation (pDA)

The direct allocation method considers the direction of the required torque in the attainable moment set in Φ, it projects the unitary vector onto the control space in Ω and scales the magnitude to match $||\boldsymbol{m}||$ or to intersect the boundary $\partial\Omega$. Therefore the produced torque will be equal to the target one or, if it is not attainable through feasible range, the direction will coincide and the magnitude will be scaled to the maximum the actuators can provide. In order to implement this method, Φ has to possess two geometrical features:

- It must contain the origin and be convex.
- Points on the boundary of Φ correspond to unique points on the boundary of Ω.

The first condition is required in order to have a generic vector always intersect $\partial\Phi$ once or be completely enclosed in the volume of the attainable moment space. This requirement can be verified through Ω. If all controls in exam have positive and negative position limits, the origin of the control space is contained within Ω. Since a linear mapping B is being considered, if the origin in $\mathbb{R}^m$ is contained within Ω, then the origin of $\mathbb{R}^n$ will be contained in Φ as well. It is possible however, that a set of particular actuators is defined

such that the position limits spans only a positive or negative range. For example, fixed RCTs can go from 0 to maximum thrust. If this kind of situation occurs, it is necessary to shift the control limits such that the origin is contained in the control space. This shift can be obtained by adding a constant value $\Delta \boldsymbol{u}$ to the control variable vector such that:

$$\begin{aligned} u_i^{min} + \Delta u_i \leq u_i + \Delta u_i \leq u_i^{max} + \Delta u_i \\ u_i^{min} \leq u_i \leq u_i^{max} \end{aligned} \tag{2.10}$$

Therefore a new set of control variables, $\boldsymbol{u}'$, along with a new desired moment $\boldsymbol{m}'$ are defined:

$$\begin{aligned} \boldsymbol{m}' = \boldsymbol{m} + B\Delta \boldsymbol{u} \\ \boldsymbol{u}' = \boldsymbol{u} + \Delta \boldsymbol{u} \end{aligned} \tag{2.11}$$

On the other hand, to ensure the convexity of Φ, it is sufficient that Ω is convex[5], which is granted as long as the position limits of the actuators can be represented as planes in $\mathbb{R}^m$. In other words, it is sufficient that the variables are defined as $u_i^{min} \leq u_i \leq u_i^{max}$. Regarding the second requirement, which states that every point of $\partial\Phi$ corresponds to unique points in $\partial\Omega$, it is necessary that every $n \times n$ submatrix of B is full rank, thus meaning that each subset must be able to generate a torque in n-D directions. This is a stricter requirement over the actuators configuration, since it is necessary that each combinations of n actuators, and their respective columns of B must be able to solve the allocation $\boldsymbol{m} = B\boldsymbol{u}$. Configurations that are able to cope with this requirements are called purely redundant. In other words, the addition of actuators is considered only for a reliability purpose, or to increase the overall attainable moment set in terms of magnitude and not direction.

The technique of Direct Allocation can be applied to any problem where the degree of redundancy is greater than one. It requires the boundary of Φ to be determined to find the intersection of the torque $\boldsymbol{m}$ with $\partial\Phi$. A way to perform this operation would be to map all the vertices of the m-D polyhedron Ω into the moment space, and have a projection of the whole attainable moment set but that is extremely heavy in terms of computational cost. One could possibly compute this projection "offline" to save computational effort, but the algorithm would still require to search through the entire boundary surface.

Considering the geometry of Ω, instead, it is possible to define a way to perform the aforementioned operation, in an easier and more efficient manner. To better display the process, $n = 3$ case is studied. Therefore, $n-1$-D objects are 2-D faces, or planes, which

[5]The proof can be found in [5].

corresponds to two control free variables and $m-2$ fixed. All the possible combinations of couples among the m variables, while the other are set to their maximum and minimum values, represent all the faces of Ω. These faces are parallel when the same two controls are varying in $\mathbb{R}^m$. The face that belongs to $\partial\Omega$ is called facet. To align an axis of $\mathbb{R}^n$ perpendicular to a set of parallel faces, consider a nxn transformation matrix T. The rotated matrix will be TB. Since only one axis is to be specified, it is necessary for the algorithm one row $\boldsymbol{t}$ of TB. The i^{th} and j^{th} elements of $\boldsymbol{u}$ are considered, while the other $m-2$ are fixed. In order to find the rotation which will align $\hat{\boldsymbol{m}}$, the unitary vector of $\boldsymbol{m}$, perpendicular to a set of faces, $\boldsymbol{tB_i}$ and $\boldsymbol{tB_j}$ must be zero, where $\boldsymbol{B_{i,j}}$ are the columns of the effectiveness matrix corresponding to the given indexes. Hence the vector $\boldsymbol{t}$ can be computed as[6]:

$$\begin{bmatrix} B_{i1} & B_{i2} \\ B_{j1} & B_{j2} \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} + t_3 \begin{bmatrix} B_{i3} \\ B_{j3} \end{bmatrix} = \mathbf{0} \tag{2.12}$$

Where t_3 can be chosen arbitrarily (typically 1). If the built system is not full rank, the perpendicular vector can be found considering the other two 2×2 matrices, since the 3×3 submatrix of B is full ranked. Once compute the quantity $\boldsymbol{t}B$, the result will be a row vector with zero values in the respective i,j positions, which corresponds to the couple taken into account. The other vector elements, on the other hand, will have positive and negative values. With this information, it is possible to build a combination of three distinct control vectors $\boldsymbol{u_1}$, $\boldsymbol{u_2}$ and $\boldsymbol{u_3}$, which define the vertex of a facet in m-D space. All the $m-2$ elements are brought to their minimum or maximum value depending on the sign reported in the $\boldsymbol{t}B$ vector correspondent position, while the i,j elements will take a combination of minimum-maximum values to have the three distinct vectors. For example, if $i=1$ and $j=2$ the three vectors would take the following form:

$$u_1 = \begin{bmatrix} u_{min} \\ u_{min} \\ \boldsymbol{u}_{min/max} \end{bmatrix} \quad u_2 = \begin{bmatrix} u_{min} \\ u_{max} \\ \boldsymbol{u}_{min/max} \end{bmatrix} \quad u_3 = \begin{bmatrix} u_{max} \\ u_{min} \\ \boldsymbol{u}_{min/max} \end{bmatrix} \tag{2.13}$$

Where $\boldsymbol{u}_{min/max}$ represents the $m-2$ vector of minimum-maximum entries depending on the sign reported in $\boldsymbol{t}B$. Once the three vectors are computed, the correspondent quantities in moment space are calculated through the linear mapping:

$$\boldsymbol{m_1} = B\boldsymbol{u_1} \quad \boldsymbol{m_2} = B\boldsymbol{u_2} \quad \boldsymbol{m_3} = B\boldsymbol{u_3} \tag{2.14}$$

[6]Ibid [5]

As in control space, also in moment space the three vectors define the vertex of a facet. The actual facet can be computed considering the three following vectors:

$$\begin{bmatrix} \boldsymbol{m_1} \\ \boldsymbol{m_{1-2}} = \boldsymbol{m_1} - \boldsymbol{m_2} \\ \boldsymbol{m_{1-3}} = \boldsymbol{m_1} - \boldsymbol{m_3} \end{bmatrix} \tag{2.15}$$

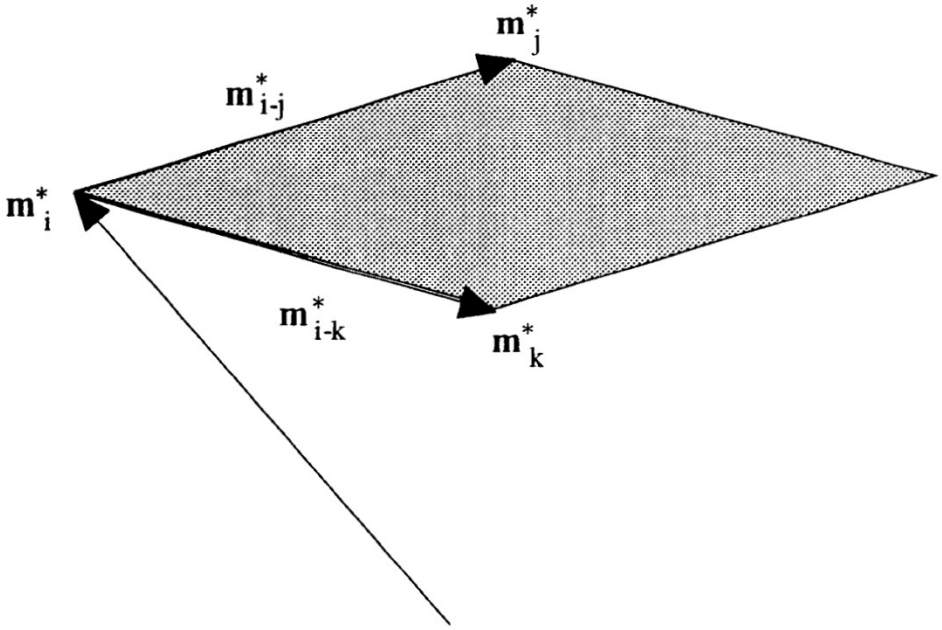

Figure 2.2: Facet of attainable moment set in $\mathbb{R}^n$ for a generic triad ijk

Now, it is necessary to check if the required torque lies onto the facet the three vectors identify (Figure 2.2). For a generic vector $\boldsymbol{m^*}$ to belong onto the face, it is necessary that the system reported in Equation (2.16) is satisfied, which states that the required torque, the product between the unitary vector and its magnitude, is a linear combination of three vectors.

$$\begin{cases} a\hat{\boldsymbol{m}}^* = \boldsymbol{m_1} + b\boldsymbol{m_{1-2}} + c\boldsymbol{m_{1-3}} \\ 0 \leq b \leq 1 \\ 0 \leq c \leq 1 \end{cases} \tag{2.16}$$

The constraints on b and c enforce the direction $\hat{\boldsymbol{m}}^*$ to intersect the currently considered facet. Therefore it is possible to solve the equation and check if the constants are compliant with the constraints:

$$\begin{bmatrix} \hat{\boldsymbol{m}}^* & -\boldsymbol{m_{1-2}} & -\boldsymbol{m_{1-3}} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \boldsymbol{m_1} \tag{2.17}$$

If $a > 0$ and the constraints in Equation (2.16) are satisfied, the required torque lies on the considered facet, otherwise, it is necessary to perform the same process with another couple of actuators. To compute the correct control $\boldsymbol{u}$, since B is linear, it is sufficient to operate the same projection:

$$\boldsymbol{u}^* = \boldsymbol{u_1} + b\boldsymbol{u_{1-2}} + c\boldsymbol{u_{1-3}} \tag{2.18}$$

$\boldsymbol{u}^*$ corresponds to the maximum attainable torque in the direction parallel to $\boldsymbol{m}$. if the required moment is feasible, then $a \geq ||\boldsymbol{m}||$ and the correct control variable output would be:

$$\boldsymbol{u} = \boldsymbol{u}^* ||\boldsymbol{m}|| / a \tag{2.19}$$

Hence, the scaling is applied to meet the performance requirements. If $a < ||\boldsymbol{m}||$, the required moment lies outside the feasible region and thus $\boldsymbol{u}^*$ is the final result. In this case, it is easy to understand that the computed torque will not meet with the controller requirements, but the direction is nonetheless preserved. If the conditions in Equation (2.16) are not met, a new couple of variables in $\boldsymbol{u}$ have to be considered, and the whole process is repeated.
With this procedure the control vector $\boldsymbol{u}$ can be efficiently computed without the knowledge of all the conformation of $\partial\Phi$. In addition, it can maintain the correct direction of the produced moment when this is not attainable, thus meaning that the residual will only be present in terms of magnitude. Even though the method can provide an optimal solution, it is affected by strict hypothesis in the definition of the effectiveness matrix, which limits its applications in real actuator configurations. Especially, if each $n \times n$ submatrix of the effectiveness is not full ranked, it is possible to apply this method loosening the restriction, in expense of having the probability of finding a sub-optimal solution. Consider, for example, a system which does not satisfy the second hypothesis. In this case the method is still applied with three important aspects:

- It is not always possible to find the perpendicular vector $\boldsymbol{t}$.
- The product $\boldsymbol{t}B$ may not have only two zero entries in the i, j positions.

- Constraints in Equation (2.16) may not be satisfied for each combinations of actuator couples due to other zero entries in $\boldsymbol{t}B$, thus leading to an approximation of the direction of $\boldsymbol{m}$.

Mathematically, the relaxation of the above hypothesis can lead to an error in the check of the facets, and it is possible that the direction $\hat{\boldsymbol{m}}$ will not belong to any of the planes computed, in other words that the constraints in Equation (2.16) are not satisfied. With these considerations, the method can still be applied taking into account all the outputs given by the different couples and selecting the one which suites the best the user requirements, for example, the output that minimizes the residual or a cost function.

Algorithm 2.4 Direct allocation

1: Given B, $\boldsymbol{m}$, $n_{couples}$, $\boldsymbol{t}B$ for each $n_{couples}$, saturation levels $\boldsymbol{u_{max}}$ and $\boldsymbol{u_{max}}$
2: **for** $i = 1 : n_{couples}$ **do**
3: $\boldsymbol{index_0} = \text{find}(\boldsymbol{t}B = 0)$
4: $\boldsymbol{index_{max}} = \text{find}(\boldsymbol{t}B > 0)$
5: $\boldsymbol{index_{min}} = \text{find}(\boldsymbol{t}B < 0)$
6: Create $\boldsymbol{u_1}$, $\boldsymbol{u_2}$ and $\boldsymbol{u_3}$ considering $\boldsymbol{u_{max}}$ and $\boldsymbol{u_{min}}$ in positions $\boldsymbol{index_{max}}$ and $\boldsymbol{index_{min}}$
7: Compute $\boldsymbol{m_1}$, $\boldsymbol{m_2}$ and $\boldsymbol{m_3}$
8: Solve the linear system $[\hat{\boldsymbol{m}} - \boldsymbol{m_{1-2}} - \boldsymbol{m_{1-3}}][a\ b\ c]^T = \boldsymbol{m_1}$
9: **if** $0 \leq b \leq 1$ *and* $0 \leq c \leq 1$ **then**
10: $\boldsymbol{u}^* = \boldsymbol{u_1} + b\boldsymbol{u_{1-2}} + c\boldsymbol{u_{1-3}}$
11: **if** $a \geq ||\boldsymbol{m}||$ **then**
12: $\boldsymbol{u} = \boldsymbol{u}^*||\boldsymbol{m}||/a$
13: Break cycle
14: **else**
15: $\boldsymbol{u} = \boldsymbol{u}^*$
16: Break cycle
17: **end if**
18: **else**
19: Proceed with the next couple of actuators
20: **end if**
21: **end for**

The Algorithm 2.4 reports the procedure for the Direct Allocation method. If the pseudo Direct Allocation is used, instead, line 9 is not implemented. Instead, for example, a check on the feasibility of the solution $\boldsymbol{u}$ calculated can be used. Then, the cycle does

not break but all combinations outputs are stored and the suited option is taken as final result. As selector to choose among the solutions, it is possible to use an external index which directly depends on the actuator efforts (in RCTs, for example, the propellant mass flow rate to be minimum). In conclusion, it is important to underline that pDA is less effective when dealing with unattainable moments. If the input unfeasible solution is far greater than what the actuators can produce, and the substitute check is used, it may occur that no output is computed, since the solution, during the cycle, is stored only if it satisfies the actuator ranges. One could ease these constraints with a certain tolerance and, through a post process, enforce the necessary boundaries, with a possible loss in the direction match between required and furnished torque (which is the strength of this method). If $\boldsymbol{m}$ is instead in the neighborhood of $\partial\Phi$, the output should be found as stated in line 15 in Algorithm 2.4. Even though the strict definition, this method actually takes into account the geometrical view of the problem, as the feasible actuator range have a key role in the computations.

2.6. Allocation through function minimization

The previous methods try to resolve the allocation problem by looking directly into the definition of Φ and Ω, feasibility and the generation of a control effort to cope with the required moment. Hence the driver of the problem was the compliance $\boldsymbol{m} = B\boldsymbol{u}$, namely the accuracy, while the mission cost is completely disregarded or it is affected indirectly. The dual side of the problem is selecting as a driver the minimization of a scalar function that represents a vital aspect of a mission, namely p. The resulting control vector will be the one that minimizes p while taking into consideration the saturation constraints. This scalar value may represent propellant mass consumption, or momentum storage increment, depending on the application, but the procedure is general. This family of allocators collects the different algorithms that solve the following problem:

$$minimize_{\boldsymbol{u}}\ p\ s.t \begin{cases} \boldsymbol{u}_{min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{max} \\ \boldsymbol{m} = B\boldsymbol{u} \end{cases} \tag{2.20}$$

Which is a manipulation of the problem stated in Equation (2.1). The main difference is that now the focus is brought onto the minimization of p, and the consequent definition of $\boldsymbol{u}$ in compliant ranges. The research is then carried out through feasible regions determined by the active constraints, for example the valid actuator ranges. If the minimum investigation will move towards an unfeasible region, a function, called barrier, will increase the objective cost, thus leading the process back to a feasible region. Some of

these algorithms, since they are based on an iterative method, require an initial guess solution of the input variable which will generate an estimate of the cost function. The dependency on an initial point highly affects the solution found, since it will dictate the course of the research, which could be close to a local minimum where the algorithm "may fall", thus not finding the absolute optimal solution. An example of such algorithms is the Linear Programming (LP) [4], which solves the linear problem stated in Equation (2.21).

$$minimize_{\boldsymbol{u}} \ \boldsymbol{f}^T\boldsymbol{u} \ s.t \begin{cases} \boldsymbol{u_{min}} \leq \boldsymbol{u} \leq \boldsymbol{u}_{max} \\ B\boldsymbol{u} = \boldsymbol{m} \end{cases} \tag{2.21}$$

In this case, the cost function is directly defined by the sum of the control variables each multiplied by a weighting factor. The resolution is performed through the `Matlab` built-in function `linprog`. In the problem formulation, it is also possible to add a set of inequality constraints, which are not required for the allocation problem solution. If the cost function can not take the form expressed in Equation (2.21) or the problem is affected by non-linear constraints, another algorithm shall be implemented, such as the built-in function `fmincon`, which can solve directly Equation (2.20). The key vantage of these kinds of methods is that the optimal solution, starting from the "correct" initial guess, is calculated. On the other hand, they are affected by two major detrimental factors. Not only they are generally computationally heavy algorithms, but also are not able to cope with unattainable moments. If Direct Allocation is considered, for example, it is capable of preserving the direction of the required torque, while the magnitude will be the maximum actuators can provide. If solution lies in unfeasible regions, the algorithm will not be able to converge, and coarse output, or even no result may be produced. Regarding the advantages that this type of algorithms present, they are characterized by an intuitive and straight implementation of feasibility regions, and may take into account multiple, different constraints at the same time.

2.7. Weighted Least Square solution (WLS)

The whole set of methods seen so far have a common feature, which is the focus on one respective aspect of the control allocation problem. In fact, the solution can be accuracy or cost-driven, meaning that the problem is solved giving attention to, for example, how the allocation is accurately describing the required torque, minimizing the residual, while affecting indirectly the second part, which is cost minimization. The same is applied vice versa. This matter can be considered both an advantage or a limitation depending on the applications and mission requirements. It is possible to define a method such that both

aspects are considered at the same time, giving the user the possibility to select which aspect is to be prioritized. Particularly suitable for this kind of problem is the optimal solution through Weighted Least Square minimization[7], which merges the two drivers into a single cost function, as written in Equation (2.22).

$$\boldsymbol{u} = arg\ min_{\boldsymbol{u_{min}} \leq \boldsymbol{u} \leq \boldsymbol{u_{max}}}\ \gamma ||W_a(B\boldsymbol{u} - \boldsymbol{m})||^2 + ||W_p(\boldsymbol{u} - \boldsymbol{u_p})||^2 \tag{2.22}$$

The above problem statement declares that the solution will be a trade-off between the two 2-norm cost functions. The first part represents the accuracy of the solution, which is the residual. The second part, instead, represents the cost minimization, stated as the difference between the control command set $\boldsymbol{u}$ and the desired one $\boldsymbol{u_p}$. Concerning space applications, the actuators effort can be generally linked to the parameter, or cost, that is necessary to minimize, such as power or propellant consumption. By reducing the overall intensity of the variables in $\boldsymbol{u}$, it will eventually reduce the cost. If RCTs are considered, for example, each element in the control vector can be seen as the fraction of thrust the engine can exert. If all the thrusters are the same, then $\mathbf{0} \leq \boldsymbol{u} \leq \mathbf{1}$, where the lower bound represents the switched off condition, and the upper one maximum thrust. The same thrust is linked to the propellant mass flow rate of each nozzle through the specific impulse, which can be assumed, for the analysis, to be constant:

$$\dot{m}_{prop} = \frac{T}{I_{sp} g_0} \tag{2.23}$$

Therefore, the linear dependency, as Equation (2.23) shows, demonstrates that the lower each actuator effort, the lower the consumption. Moreover it is possible to state, mathematically, that:

- $For\ each\ x,\ if\ f(x) \geq min(f(x))\ and\ g(x) \geq min(g(x))$
- $f(x) + g(x) \geq min(f(x)) + min(g(x))$
- $min(f(x) + g(x)) \geq min(f(x)) + min(g(x))$

Which corresponds to the RCTs case, where u_i is lower bounded with 0, being the absolute minimum. The sum of each effort, if minimum, will guarantee the minimum for the overall propellant mass consumed. Thus, the cost minimization (propellant for RCTs) is directly present in the problem definition, if $\boldsymbol{u_p}$ is selected to be a zero entries vector, representing the minimum possible effort of the actuators. The same discussion can be done for

[7]Nomenclature and procedure are followed as in [18].

the reaction wheels. Of course, it is clear to see that, if only the cost minimization is considered, represented in Equation (2.22) by the second part, the optimal solution will be a null control vector, which goes against the attitude control problem (first part in Equation (2.22)). The combination of the two contributions, which represents a trade-off between cost savings and accuracy, shows the great utility of the method: the freedom given to the user to select, depending on the mission requirements, the characterization of the allocator. The solution obtained could guarantee lower accuracy but save propellant or vice versa.
Each quantity in Equation (2.22) is then multiplied by a weighting factor, respectively $W_a \in \mathbb{R}^{n\times n}$ which defines how relevant each element of the residual vector has to be with respect to the overall cost, and $W_p \in \mathbb{R}^{m\times m}$, which states the same but applied on the control variables in $\boldsymbol{u}$. Generally for simplicity, these matrices are diagonal and have the same values if no prioritization is required among control DOFs. Moreover, γ is an additional constant to further highlight the accuracy total cost with respect the other one. Generally γ should be greater than 10^6, otherwise the solution could be unfeasible or the residual would be too high. This parameter is introduced due to the fact that, generally, the accuracy has to be more relevant with respect to cost minimization to avoid the condition in which the solution gets too close to the null, but feasible, solution $\boldsymbol{u_p}$.
In order to solve the WLS, an **Active Set Algorithm** is implemented[8] which rewrites the cost function in Equation (2.22) as follows:

$$\gamma||W_a(B\boldsymbol{u}-\boldsymbol{m})||^2 + ||W_p(\boldsymbol{u}-\boldsymbol{u_p})||^2 = ||A\boldsymbol{u}-\boldsymbol{b}||^2 \tag{2.24}$$

$$A = \begin{bmatrix} \gamma W_a B \\ W_p \end{bmatrix},\ \boldsymbol{b} = \begin{bmatrix} \gamma W_a \boldsymbol{m} \\ W_p \boldsymbol{u_p} \end{bmatrix} \tag{2.25}$$

And solves:

$$\boldsymbol{u} = argmin||A\boldsymbol{u}-\boldsymbol{b}||\ s.t.\ \boldsymbol{u_{min}} \leq \boldsymbol{u} \leq \boldsymbol{u_{max}} \tag{2.26}$$

The algorithm solves the minimization problem through a sequential resolution of equality constrained problems. At each iteration, some inequality constraints are treated as equality, defining the working set WS, while the other are not considered. The optimum investigation is carried out by perturbing of a quantity $\boldsymbol{p}$ the result of the previous iteration thus computing $\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \boldsymbol{p}$. The residual is updated as well with the new

[8] Ibid in [18]

solution. If $\boldsymbol{u}^{k+1}$ is not compliant with the constraints, a constant α is defined such that $\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \alpha \boldsymbol{p}$ is feasible, where α represents the closest distance among the actuator variables, between their values and respective boundaries. This operation is necessary in order to have the variable brought to saturation. The correspondent element is, then, omitted from the working set, as it is fixed. The process is repeated until a certain number of actuator variables, that generate an unfeasible solution after the perturbation, are eliminated from the working set. Once a feasible solution is found, it is necessary to check the compliance with the problem constraint $A\boldsymbol{u} - \boldsymbol{b}$, which is carried out by calculating the lagrangian multipliers $\boldsymbol{\lambda}$ associated with the problem. Since the problem is describing a quadratic programming (Equation (2.24)), with the minimization of the squared norm of the cost function, a necessary condition that ensures an optimal solution, as stated in [17], is that all components of $\boldsymbol{\lambda}$ are strictly non-negative. If a certain number of multipliers are negative, it is then necessary to deactive the constraint associated with the most negative λ. For the algorithm initialization, a feasible initial guess should be implemented for fast convergence, since the algorithm depend on the residual $\boldsymbol{d} = A\boldsymbol{u} - \boldsymbol{b}$. In Algorithm 2.5 the procedure for computing the WLS solution is shown.

Algorithm 2.5 Weighted Least Square with Actibe Set Algorithm

Given A, $\boldsymbol{b}$, $\boldsymbol{u}^0$, saturation levels $\boldsymbol{u_{min}}$ and $\boldsymbol{u_{max}}$
Define WS which contains the active inequality constraints at $\boldsymbol{u}^0$
Compute the initial error $\boldsymbol{d^0} = \boldsymbol{b} - A\boldsymbol{u}$
for $k = 1 : n_{iter}$ **do**
    Define the matrix A which corresponds to WS
    Solve the equation $A\boldsymbol{p} = \boldsymbol{d}$
    Apply the perturbation onto the variable $\boldsymbol{u}^k$
    if $\boldsymbol{u}^k + \boldsymbol{p}$ is feasible **then**
        $\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \boldsymbol{p}$
        Update $\boldsymbol{d} = \boldsymbol{d} - A\boldsymbol{p}$
        Compute Lagrangian multipliers $\boldsymbol{\lambda} = WS(A^T\boldsymbol{d})$
        if $\boldsymbol{\lambda} \geq 0$ **then**
            $\boldsymbol{u}^{k+1}$ is the optimal solution, break the cycle
        else
            Remove the constraint associated with the most negative λ from the working set
        end if
    else
        Compute α such that $\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \alpha\boldsymbol{p}$ is feasible.
    end if
end for

The initial working set can contain all the actuator DOFs, then a part of them will be excluded during the cycle. The algorithm is able to compute efficiently and rapidly an optimal solution, with the unique feature of a direct control on both aspect of the control allocation problem, which gives high flexibility and freedom to the user.

2.8. Strategies comparison

All the described methods have their own points of strength and weakness. It is necessary to highlight them through a real implementation on a set of redundant actuators. To perform this task, a series of simulations are executed, with same conditions for actuator architecture, orbital elements, pointing requirements and control logic, namely merging all the information from Chapter 1 and the control allocation algorithms. In the simulation, a spacecraft is required to follow a prescribed target attitude against perturbation using the set of redundant actuator at disposal. The control allocation will be assigned to

the different methodologies. To evaluate the strategies and have a numerical comparison between each other, a set of key parameters are selected, that generally represent the accuracy, efficiency and optimality of the algorithms:

- The cost, which can be represented by a scalar value, such as propellant mass, vital to the mission life span.
- Pointing accuracy, as the norm of the error between target and body attitude in terms of Rodrigues parameters.
- The residual, namely the error between required and produced torque, which evaluates how precisely the solution is calculated.
- The computational efficiency, which will take into account how much time the simulation takes to complete.

In addition, the behaviour when dealing with unattainable moments is examined to assess the reliability of the allocation in critical conditions. The methodologies are divided into two groups, applied respectively onto RW and RCT, whose effort is assumed to be continuous. On the first set of actuator the GI, CGI, DA and WLS methods are implemented. On the other hand, for RCT, the NS, LP, pDA and WLS are used. For each algorithm, three operating orbits are chosen to span different working conditions:

	SSO Dawn-Dusk	GTO	GEO
Perigee Altitude [km]	500	180	35786
Eccentricity [-]	0	0.6	0
Inclination [°]	90	0	0
Perigee anomaly [°]	0	0	0
RAAN [°]	0	0	0

Table 2.1: Orbital elements for the simulations

2.8.1. RW allocation comparison

Before introducing the evaluation, it is necessary to show the configuration of the wheels, which is reported in Equation (2.27).

$$B = \begin{bmatrix} 0.579 & -0.579 & -0.579 & 0.579 \\ -0.579 & -0.579 & 0.579 & 0.579 \\ 0.579 & -0.579 & -0.579 & 0.579 \end{bmatrix} \quad and \quad -0.2 \leq \boldsymbol{u} \leq 0.2 \tag{2.27}$$

As reported in the above equation, each column of the effectiveness matrix represents the orientation of each of the four wheels, in the three directions xyz. Each row, instead, represents the contribution of the wheels to generate a torque in the specific direction. B has no zero entry, meaning that all the wheels provide a moment in each direction, and that the architecture is purely redundant[9]. Additionally, the degree of redundancy is equal to 1. In conclusion saturation levels limit the effort of the actuators in a feasible, bounded region. The simulation time considered is 50000s, starting from a matching condition between target and body attitude.
Regarding the SSO scenario, the low altitude will generate intense perturbation due to pressure drag and gravity gradient. So, generally, the effort of the actuators will be greater with respect to, for example, the GEO case.

SSO Dawn Dusk	GI	CGI	DA	WLS
Storage at t_f [s]	20.79	20.79	20.57	19.17
Mean pointing error [arcsec]	116.86	116.86	116.90	116.86
Mean residual [Nm]	∼0.00	∼0.00	∼0.00	∼0.00
Computational time ratio [-]	0.90	0.90	0.99	1.00

Table 2.2: Performance evaluation for SSO case

As Table 2.2 shows[10], the GI and CGI methods solves the allocation problem faster than the other two, in expense of a more rough optimization, since the storage at the end of the mission is higher. On the other hand, DA and WLS are able to reduce the cost with a higher computational time. Generally, regarding this scenario, the four methodologies are able to solve the allocation problem, as the residual and the pointing accuracy, intrinsically linked, are reasonably low, thus meaning that the required torque always lie within Φ. Even though the difference in the results do not seem relevant, the life span of years of a in-orbit servicing mission could benefit greatly with the efficiency increment of WLS or DA. For example, the first storage saturation occurrence is delayed by three

[9]It is therefore possible to apply the strict Direct Allocation method.
[10]The results are reported in terms of ratio between the algorithm computational time and the maximum one to decouple the evaluation from the machine efficiency that tested the algorithms.

hours using WLS with respect to GI.

GTO	GI	CGI	DA	WLS
Storage at t_f [s]	30.80	30.80	29.90	28.83
Mean pointing error [arcsec]	31.31	31.31	29.08	29.08
Mean residual [Nm]	∼0.00	∼0.00	∼0.00	∼0.00
Computational time ratio [-]	0.89	0.89	0.96	1.00

Table 2.3: Performance evaluation for GTO case

In GTO case, a similar trend is shown in Table 2.3. First, as in the SSO case, both GI and CGI perform equally, since no saturated wheels is detected during the allocation. Secondly, the performance are comparable to the previous case, where DA and WLS are more accurate but slow compared to GI and CGI and vice versa. The difference in the strategies is more marked due to the increment in the perturbation elements, especially atmosphere drag, being the perigee of the orbit at 180 km. Even though each algorithm is able to cope with the required torque, the storage is rather different, making the WLS the best choice, but the less computationally efficient. In conclusion, it is possible to state that, as the perturbation elements increase in intensity, the strategies differ in the computed solution, and the efficiency as well. On one hand, the GI computing a fast and rough solution, and the WLS and DA, taking more time to refine the results.
In conclusion, the same considerations can be applied for the GEO case, which is the least perturbed case, thus the error is almost null. Of course it is necessary to underline that these numerical results depend on the different blocks discussed in Chapter 1, which, if varied, can affect the performance, especially the pointing accuracy, since it depends on the definition of the control logic (in this case the gains selected), but the behaviour of the strategies is conserved.

Equatorial GEO	GI	CGI	DA	WLS
Storage at t_f [s]	14.28	14.28	13.74	13.74
Mean pointing error [arcsec]	8.34	8.34	8.34	8.34
Mean residual [Nm]	∼0.00	∼0.00	∼0.00	∼0.00
Computational time ratio [-]	0.88	0.88	0.96	1.00

Table 2.4: Performance evaluation for GEO case

For what concerns the response to unattainable moments, a simulation was performed testing the different algorithms with increasing required torque, to analyze at which input value the allocator would fail. Therefore, the capacity of the attainable moment set is studied.

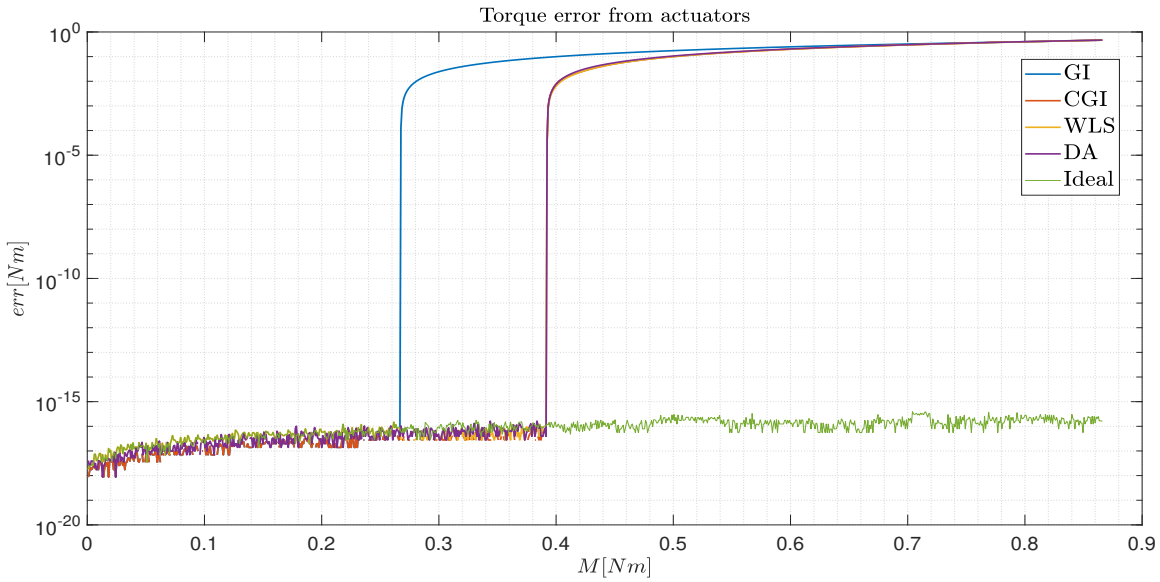

Figure 2.3: Residual as the required torque grows

In Figure 2.3 it is possible to see how the different algorithms try to cope with a required torque increasing in intensity. The graph plots in semi-logarithmic scale the norm of the residual in function of the norm of the input required torque $\boldsymbol{m}$, simulating a condition where the spacecraft has to execute a manoeuvre that exceeds the actuators capabilities. The thin green line, indicated as "ideal", represents the residual if a boundless allocation is implemented, in order to have a comparison of what should be the actual solution. The generalized inverse is the first to be incapable of producing the required torque, since no further operations is done other than the pseudo-inverse. On the other hand, the other three strategies are more suited to treat unattainable moments, as they delay the failure condition. Geometrically, it is possible to view the enhancement as an overall augmentation of the admissible control set Ω with respect to the simple GI: regarding the CGI, the second distribution can avoid early saturation, while for DA and WLS, a more suited solution is directly computed. Of course, also these more accurate algorithms will eventually meet their limits. In the upper right part of the graph, the strategies reach a plateau. This is due to the fact that, as the actuators get saturated and are not able to cope with the input moment, the residual, defined by the difference norm between required and produced torque $||\boldsymbol{m} - B\boldsymbol{u}||$, will continue to grow, since $\boldsymbol{u}$ is practically

fixed at its maximum attainable value. Therefore the values reported in the x and y axis will eventually match. Finally, it is important to underline that this trend may vary in terms of minimum unattainable moment value, due to the fact that a certain direction of torque increment was selected as an example, but the consideration still apply for any other direction.

2.8.2. RCT allocation comparison

As done for the reaction wheels, first, it is necessary to display the hardware actuators configuration that is implemented for the evaluation. One of the possible effectiveness definition is the following: assuming the RCTs to be fixed, the effectiveness matrix will collect the cross product of the nozzle thrust and its position with respect to the centre of gravity. Therefore, each column will contain the torque each nozzle can provide in BRF. The control vector $\boldsymbol{u}$, instead, will represent the fraction of thrust to be furnished. For example, if the nozzle needs to generate half the maximum thrust, its correspondent value in $\boldsymbol{u}$ will be equal to 0.5, if switched off it will be 0. In this way it is possible to decouple the thruster specific characterization from the control problem. This operation can be performed as long as the thrust profile is assumed continuous or the actuator response is faster than the controller update frequency.

$$B = \left[\begin{array}{cccccc} \boldsymbol{r}^1_{\boldsymbol{nozzle}} \times \boldsymbol{T}^1 & \cdots & \boldsymbol{r}^i_{\boldsymbol{nozzle}} \times \boldsymbol{T}^i & \cdots & \boldsymbol{r}^m_{\boldsymbol{nozzle}} \times \boldsymbol{T}^m \end{array} \right] \; and \; 0 \leq \boldsymbol{u} \leq 1 \tag{2.28}$$

This configuration is more problematic with respect to the wheels case, where all the variables are able to produce a torque in the three directions, and the system is purely redundant. First, since a nozzle, in its relative frame is able to produce a one direction effort, the control vector is bounded to a positive region. Moreover, in this scenario, it may be possible to select three nozzles which are not able to provide a 3-D moment, thus it is not guaranteed purely redundant configuration. As previously carried out for the RW, the RCT allocation methods are evaluated for the three orbits, for a time span of 3600 s.

SSO Dawn-Dusk	NS	pDA	LP	WLS
Consumed propellant [kg]	0.19	0.18	0.07	0.09
Mean pointing error [arcsec]	264.04	264.79	264.82	263.78
Mean residual [Nm]	∼0.00	∼0.00	∼0.00	∼0.00
Computational time [s]	4.66	14.64	1067.92	16.12
Computational time ratio [-]	4.40e-03	1.37e-02	1	1.51e-02

Table 2.5: Performance evaluation for SSO case

Looking at Table 2.5, the first thing to emerge is the extremely high computational time for computing the optimal solution through the linear programming. This is due to two important aspects: first, at each integration step, an optimal solution search has to be evaluated. In addition, `Simulink` is not able to solve optimization problems that require undefined variables. A possible solution to this issue is to force `Simulink` to transfer the function related to the optimization problem onto `Matlab` and solve it on this platform, but this procedure is extremely inefficient[11]. As already highlighted by the wheels analysis, the basic null space solution is able to rapidly compute a rough solution. On the other hand WLS prioritize the optimization of the cost function. The pseudo-Direct Allocation demonstrates that the algorithm, loosening the hypothesis defined previously, not only solves the problem, but finds a better solution with respect to NS in a reasonable time lapse. Of course, since all the combinations of facets have to be checked, the computational efficiency is likely to be lower than strict direct allocation. In conclusion, each algorithm is able to solve the allocation as the residual is zero.
For what concerns the transfer orbit, the results are presented in the next table:

GTO	NS	pDA	LP	WLS
Consumed propellant [kg]	5.71e-03	8.86e-03	3.78e-03	4.22e-03
Mean pointing error [arcsec]	22.68	22.68	22.68	22.68
Mean residual [Nm]	∼0.00	∼0.00	∼0.00	∼0.00
Computational time [s]	4.02	7.76	1094.05	15.71
Computational time ratio [-]	3.70e-03	7.10e-03	1	1.44e-02

Table 2.6: Performance evaluation for GTO case

[11] In appendix A a comparison between the strategies is also performed using only Matlab to assess if the same conclusions can be drawn.

As the SSO work frame, again the linear programming is affected by a heavy computational burden, while the other strategies have the same order of magnitude, with Null Space being the fastest. It is possible to see that the pDA efficiency decreases, as its computed cost overcomes the NS one. This phenomenon is clearly highlighted in Table 2.7, where, since the perturbations are even lower, the controller requires even less effort, and thus the optimality again is subject to a decrement. This is probably due to the fact that, as the required torque reduces in intensity, a small imprecision in the approximation of the direction of the facet (line 8-9 in Algorithm 2.4 for pDA) becomes more relevant, and the imprecision is more visible. However, all the procedures are still able to solve the allocation problem.

Equatorial GEO	NS	pDA	LP	WLS
Consumed propellant [kg]	6.95e-04	11.85e-04	5.17e-04	5.76e-04
Mean pointing error [arcsec]	12.92	12.96	12.92	12.95
Mean residual [Nm]	∼0.00	∼0.00	∼0.00	∼0.00
Computational time ratio [s]	4.02	6.81	1012.12	16.72
Computational time ratio [-]	4.00e-03	6.70e-03	1	1.65e-02

Table 2.7: Performance evaluation for GEO case

Once the nominal performance are evaluated, the next step is to investigate the response to unattainable moments. Again, the allocators are required to deal with a continuously increasing torque, in order to underline which algorithm is able to extend the feasible region the furthest.

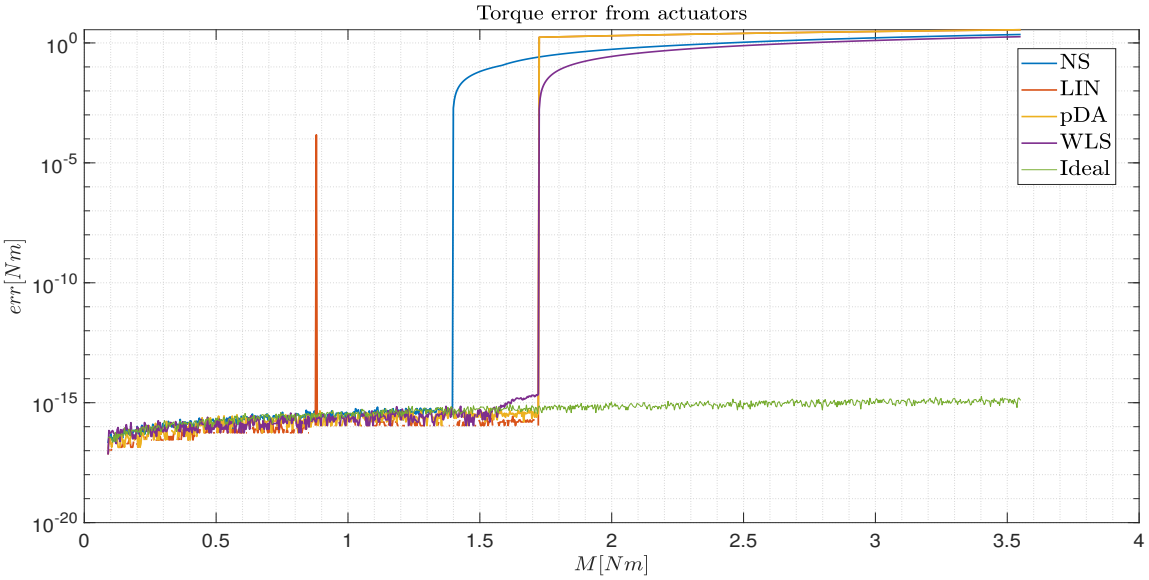

Figure 2.4: Residual as the required torque grows

In Figure 2.4, the four algorithms, compared to an ideal unbounded allocator, are tested. It is clear to see that the Null Space strategy, which resemble a pseudo-inverse distributor, performs the worse, while the pseudo-Direct and Weighted Least Square are able to prolong the working, feasible condition. Shaded by the pDA in the right part of the graph, the Linear Programming, instead, is affected by discontinuities as the required torque grows, which are visible in the central part of the plot. This happens because the algorithm always tries to find the solution in the feasible range, and if a high unfeasibility is detected, the algorithm will provide imprecise results. The spurious spike is probably due to numerical tolerance and ill-conditioning of the problem. In conclusion, the same plateau is present, as the discussed conclusion holds in this case as well. Clearly, as for the RW, also in this case the transition from feasible to unattainable is instantaneous, as it marks the crossing of $\partial\Phi$ and, consequently, $\partial\Omega$.
The studied methods, with their unique features and different definitions, assemble an extremely useful tool that enables the user to navigate through the allocation problem and find the best tailored solution for different applications, taking into account multiple and varying drivers. In fact, the "best" solution is not unique, as it varies depending on the application. The analysis then becomes handy when it is necessary to decide which method implement based on the design requirements. The next step is to implement and study the control allocation on an actual mission for in-orbit servicing, presented in the following chapter.

3 | Mission Simulation and Monte Carlo Analysis

Up to this point, the control loop defined in Chapter 1 has been fully described, and various allocation methodologies have been investigated. The abilities of these strategies were evaluated using a "test bench" application that focused solely on numerical applications, in order to assess the accuracy, performance, and efficiency of the allocating algorithms. The next step involves verifying whether the control allocation is capable of handling the complexities and demands of an actual mission in development. This is essential in order to demonstrate another critical feature that the control allocation scheme should possess: reliability in a real scenario. In this chapter, a real in-orbit servicing mission is defined, including its own characterization, work frame, and conceptual operations. This will enable the assessment of whether the allocation architecture can fulfill the requirements necessary for the spacecraft to execute its tasks. The chapter is divided into three main parts, which outline a logical workflow for the design and testing of an attitude control subsystem:

- Mission definition, architecture and work frame, where the spacecraft, its operations, and functioning environment are delineated.
- Mission phases description: divided into sections for each phase, the different actions and manoeuvres the satellite has to execute are delineated furthermore. The performance are evaluated in terms of accuracy, optimization and computational efficiency.
- Robustness analysis, executed through a Monte Carlo simulation to assess reliability and ability to cope with unexpected errors in the architecture.

3.1. Mission architecture and work frame

The mission consists of a series of in-orbit servicing operations. The spacecraft, defined *Module A (MA)*, shall carry the client spacecraft *Module B (MB)* to target position. The

two, then, separate so that the client satellite can perform its required operations. After the detachment, a movable thrust vector, present on *MB*, is activated and its effects have to be counter acted by the attitude control system. Therefore, it possible to divide the overall mission into four main phases:

- **Phase I - *MAMB* phasing**: the two satellites, connected, execute a phasing manoeuvre to reach target position.
- **Phase II - *MB* desaturation**: the client spacecraft damps out the momentum storage collected throughout the phasing.
- **Phase III - *MAMB* separation**: the client separates from the carrier, as the latter has to counter the perturbation due to the detachment.
- **Phase IV - *MB* thrust vectoring compensation**: the client spacecraft *MB* has to operate a thrust vectoring subsystem that generates a series of perturbing reaction forces and torques. These quantities must be opposed by a pose control system.

The overall spacecraft, divided into two main parts, can be modelled as two cylinders, one on top of the other, the carrier *MA* and the client *MB*. The thrust vector part is considered as an external body that just transfers, by means of reaction force and moment, its operating effects. The main difference in the two bodies is the presence in *MA* of wide solar wings, which affect especially the solar radiation pressure contribution, and impose a significant difference in the mass distribution between the two spacecrafts. All the fundamental parts are assumed to be perfectly rigid and uniform bodies. This rather simple model, show in Figure 3.1, is used to estimate the perturbation terms and the actuators positioning within the spacecraft. In Figure 3.1, in addition, the body reference frame is shown, with the x axis that runs parallel to the panels longer edge, the z axis along the cylinder axis of the connected parts and the y axis to complete the triad.

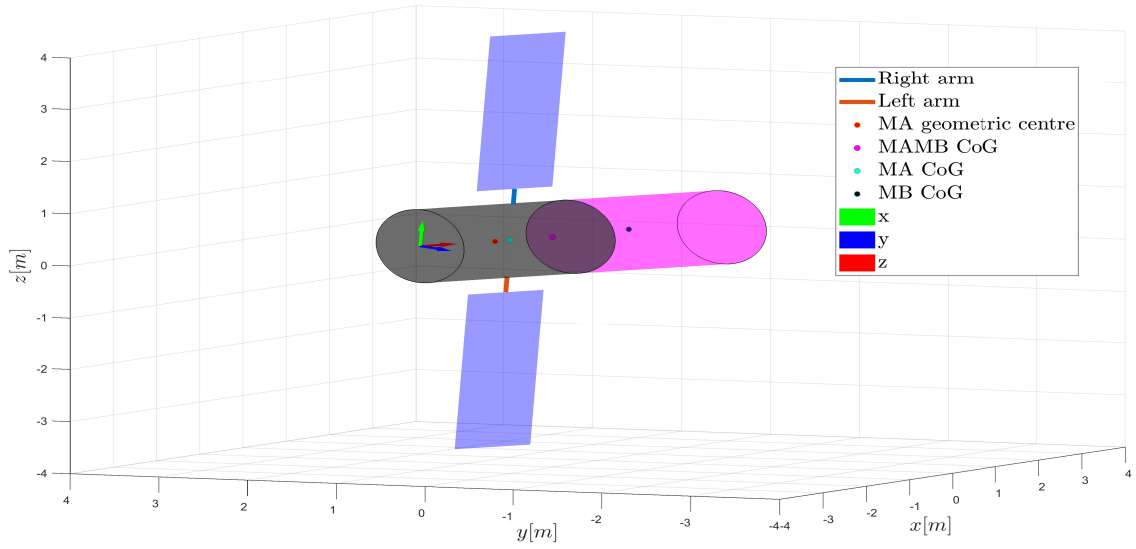

Figure 3.1: *MAMB* attached spacecraft model

In Table 3.1, on the other hand, the geometrical properties of the spacecraft are reported.

Data	Value
***MA* height**	2 m
***MB* height**	2 m
***MA* radius**	0.7 m
***MB* radius**	0.7 m
Solar panel length	5 m
Solar panel width	1 m

Table 3.1: Spacecraft data

Regarding mass and inertia parameters, more refined data are used, in order to have relatable results. For MA and MB, the parameters are here shown, in body reference frame:

$$I_A = 1e-06 * \begin{bmatrix} 7.02e+09 & -4.05e+08 & -1.88e+07 \\ -4.05e+08 & 1.505e+10 & 9.34e+07 \\ -1.88e+07 & 9.34e+07 & 1.02e+10 \end{bmatrix} [kgm^2] \tag{3.1}$$

$$I_B = 1e-06 * \begin{bmatrix} 1.03e+09 & -6.93e+06 & -4.03e+07 \\ -6.93e+06 & 1.18e+09 & 3.87e+07 \\ -4.03e+07 & 3.87e+07 & 8.12e+08 \end{bmatrix} [kgm^2] \tag{3.2}$$

$$I_{AB} = 1e-06 * \begin{bmatrix} 2.08e+10 & -3.68e+08 & -2.03e+08 \\ -3.69e+08 & 2.87e+10 & 3.54e+07 \\ -2.03e+08 & 3.54e+07 & 1.11e+10 \end{bmatrix} [kgm^2] \tag{3.3}$$

$$\boldsymbol{r}_{\boldsymbol{CoG}}^{\boldsymbol{A}} = 1e-03 * \begin{bmatrix} 4.87e+00 \\ -1.62e+01 \\ 1.21e+03 \end{bmatrix} [m] \tag{3.4}$$

$$\boldsymbol{r}_{\boldsymbol{CoG}}^{\boldsymbol{B}} = 1e-03 * \begin{bmatrix} 6.08e+01 \\ -5.33e+01 \\ 8.12e+02 \end{bmatrix} [m] \tag{3.5}$$

$$\boldsymbol{r}_{\boldsymbol{CoG}}^{\boldsymbol{AB}} = 1e-03 * \begin{bmatrix} 1.56e+01 \\ -8.02e+00 \\ 1.77e+03 \end{bmatrix} [m] \tag{3.6}$$

Both satellites are defined by a full inertia matrix, as the selected frame does not coincide with the principal inertia axis, thus meaning that couplings effects are present in the equation of motions. Both *MA* and *MB* are equipped with the same sets of reaction wheels and reaction control thrusters. For what concerns the first group of actuators, the same features reported in Section 2.8.1 are implemented, namely effectiveness matrix and hardware ranges. The same goes for the propulsion system, where the maximum thrust is now equal to 0.5 N, and the minimum producible is 1 mN. The configuration is shown in the next illustration.

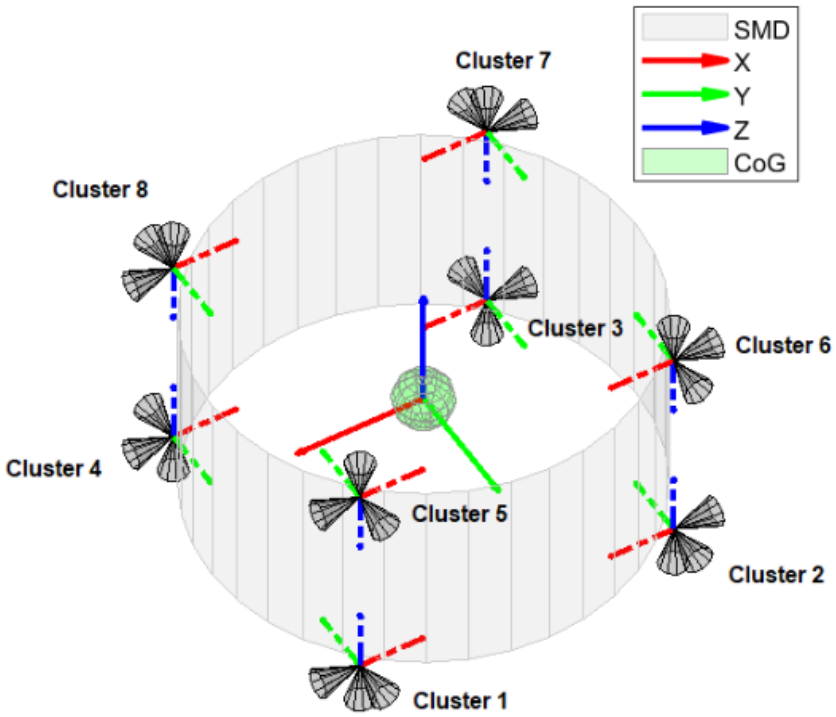

Figure 3.2: *MA* and *MB* RCT configuration

As Figure 3.2 shows, 24 cold gas thrusters are displayed in an angular configuration. Here it is easy to notice that, if all switched-on, the overall contribution is zero. The effectiveness matrix still collects the contribution of the cross product between arm and thrurst direction. This value is then scaled by the control vector $\boldsymbol{u}$ which represents how widely open is the valve. Throughout the mission, it is assumed that the dynamic response of the valve that controls the propellant flow, and the wheels change of rate are faster than the control logic, selected to 8Hz. In addition, the actuators quantization is neglected so, by merging the two assumption, the effort is always treated as continuous. The selected control logic is a variable gain PID controller, with a numerical constraint on the output, where, if the module of the required action is lower than 1e-04, then the controller treats it as a 0, to simulate a real hardware lower limitation. Therefore $||\boldsymbol{m}|| \geq 1e-04$. The spacecraft servicing orbit is assumed to be an equatorial GEO. Again, all the building blocks discussed in Chapter 1 are plugged in the model, especially the perturbing elements. Throughout the simulation, it is assumed that all the operations are performed on the same orbit. The assumption is given for two main factors: first it simplifies the analysis, secondly the displacement from the selected orbit are reasonably low (in the order of metres), as well as the characteristic simulation time spans, and precise flight dynamics and spacecraft positioning is not the aim of this study, rather the

spacecraft control. Finally, the analysis is carried out considering, if not stated differently, a purely attitude control problem, with the state vector represented by the angular velocity $\boldsymbol{\omega}$.

3.2. Conceptual operations

Once the overall system is described, it is possible to delineate what the spacecraft shall do to complete the mission, the attitude to track and the perturbations the actuators have to cope with. As previously stated, the mission can be divided into a phasing, desaturation, separation and operation manoeuvres. Each phase is characterized by different control authority and allocation, because different pointing requirements are applied. In each phase, the capacity of the allocation to guarantee the mission success is analyzed, in the same terms seen in Chapter 2, such as accuracy, computational efficiency, which becomes a key issue in this mission, and ability to optimize the action distribution.

3.2.1. Phase I - *MAMB* Phasing

The first task the satellite has to execute is to bring the client spacecraft to a target orbit. This operation, in terms of attitude control, translates into a tracking problem, where *MAMB* connected spacecraft has to follow a prescribe orientation in space[1]. The tracking shall be performed with the implementation of reaction wheels to save propellant, since the perturbations are small, the prescribed attitude does not change sharply and there are no tight constraints on the pointing error. Since the two parts are connected, the characteristic inertia is rather consistent, so the control effort is distributed among *MA* and *MB* wheels, in order to have high reliability and attainable moment space increment. However *MB* shall receive a small percentage of effort, in order to keep its storage as low as possible, to avoid pre-loading before the operations in Phase II. The two factors have conflicting objectives. On one hand, the required torque must be distributed to the *MB* actuators. On the other hand, the aim is to minimize their usage. To find a solution, a trade-off is necessary, which involves creating a weighting vector. This vector ensures that a greater percentage of the total required torque is assigned to *MA*, while the remaining small residual is assigned to *MB*. If the wheels on *MA* will reach torque or storage saturation, also *MB* will contribute fully to the control effort definition. This approach achieves a balance between maximizing saturation delay and minimizing *MB* storage. Specifically, the weight vector is selected based on a trade-off between the delay

[1]The prescribed trajectory comes from an optimization analysis performed by the company, under confidentiality.

in saturating the storage of *MA* and the need to keep *MB* storage to a minimum. The chosen priority vector is $\boldsymbol{w} = [0.1, 1]^T$, and it determines the initial distribution of control torque between *MA* and *MB*, which can be expressed as:

$$\mu_i = \frac{1}{w_i}\frac{1}{\sum \frac{1}{\boldsymbol{w}}} \; and \; \left[\begin{array}{cc} \boldsymbol{m}^{\boldsymbol{A}}_{\boldsymbol{Input}} & \boldsymbol{m}^{\boldsymbol{B}}_{\boldsymbol{Input}} \end{array} \right] = \boldsymbol{\mu m} \tag{3.7}$$

Then, the allocation receives the torques allocated to each actuator. As discussed in Section 2.7, the Weighted Least Square algorithm is a highly suitable choice for distributing the torques, as it offers both computational efficiency and accuracy. This selection is motivated by the need to minimize storage and the fact that, considering a trade-off analysis, the WLS method outperforms other options in this regard. In the next table, relevant results regarding this first phase are reported:

Data	Value
Initial epoch	02-Jan-2035 00:01:00
Simulation time	27.88e04 s
Computational time	3.94e02 s
Mean attitude error norm	8.09e01 arcsec
Mean residual norm	2.63e-02 Nm
***MA* final storage**	48.64 Nms
***MB* final storage**	35.69 Nms

Table 3.2: Performance during Phase I

The simulation requires a relatively high computational cost due to a wide time span and the necessary interpolation of the prescribed attitude[2]. The average error and residual in this phase are higher compared to the results obtained in the previous chapter. Two primary factors contribute to this outcome. Firstly, the target attitude has a more rapid variation. Secondly, the gains in the high-level control are reduced to slow storage increment. Despite this, the reported error is still deemed acceptable, as this phase is not critical. The next set of figures, instead, show the trend of important parameters for the whole simulation.

[2]The prescribed attitude required to be numerically "smoothed".

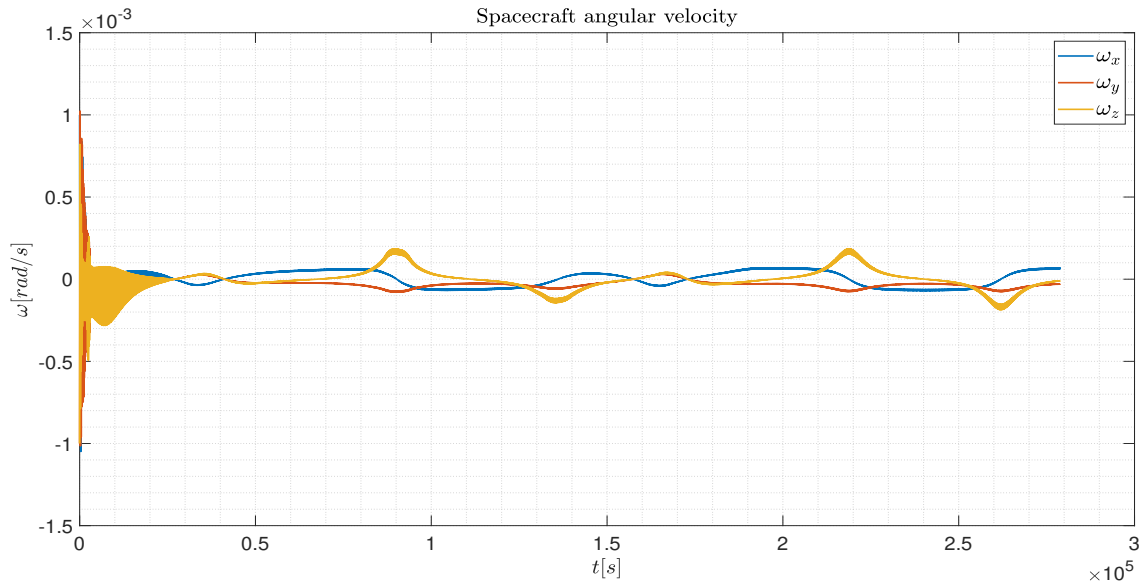

Figure 3.3: *MAMB* angular velocity

In Figure 3.3, the angular velocity vector has in the far left part of the graph, a rather perturbed and coarse behaviour. This is due to the initial angular body orientation that greatly differs from the required one. In the first part, in fact, the ability of the loop to converge to the target is tested. The same trend in visible in Figure 3.4.

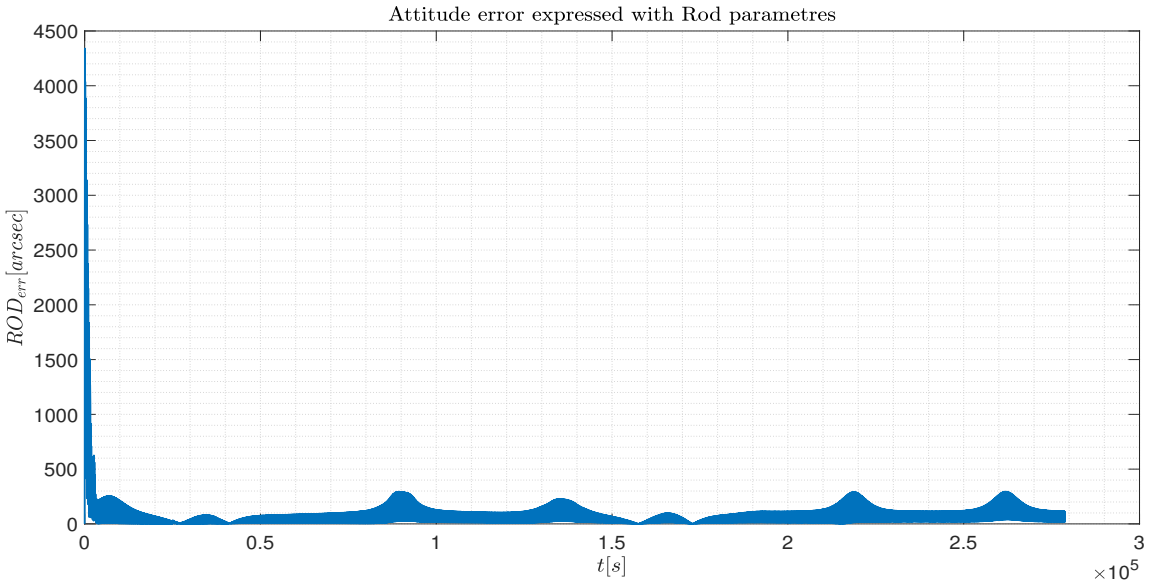

Figure 3.4: *MAMB* attitude error

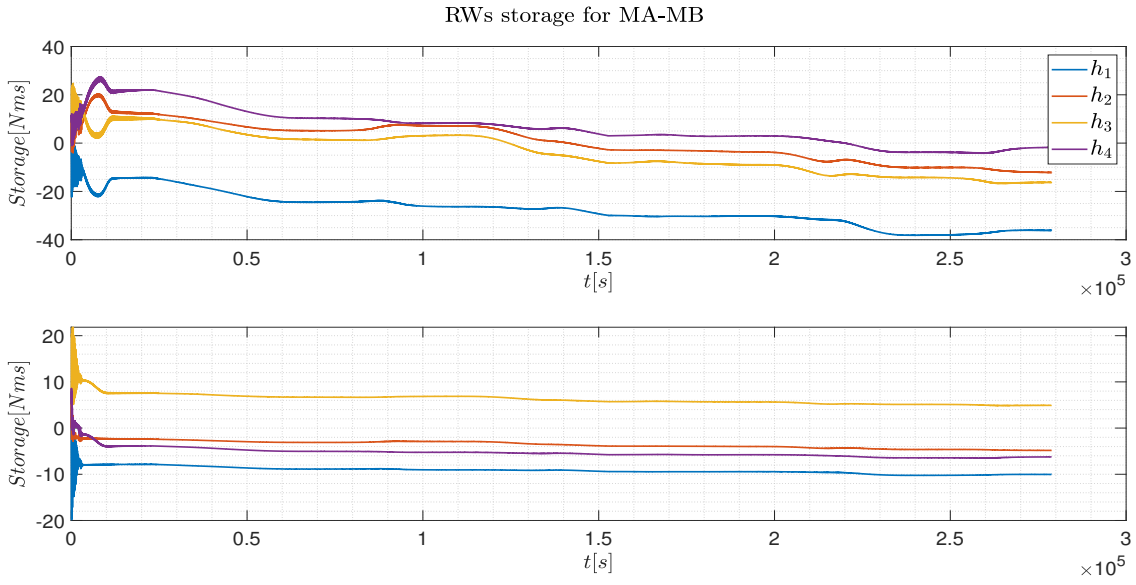

Figure 3.5: *MA* and *MB* reaction wheel storage

Again, the high effort of the reaction wheels in the first part of the phase is clearly visible, as the storage greatly increases and then varies more gently. In Figure 3.6, instead, it is possible to see the residual spikes, meaning that the required torque is unattainable. Then, after damping out the angular velocities, the tracking is performed in a feasible allocation region, as the residual gets close to zero.

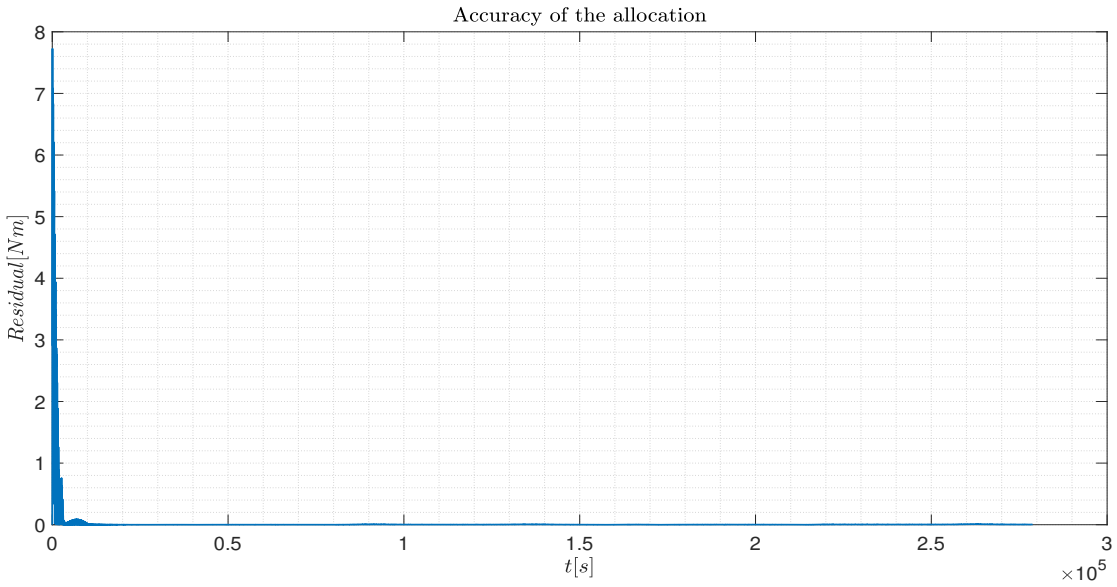

Figure 3.6: Residual between required and provided torque

It key to understand that the behaviour seems to be sharp due to the plot features, as the time span is rather wide. To sum up, it is clear that implementing a concurrent action between thrusters and reaction wheels would be adequate to reduce the pointing error if the requirements were more stringent.

3.2.2. Phase II - *MB* desaturation

Before the separation between the two spacecrafts can take place, it is necessary to empty the momentum storage accumulated during the previous phase in *MB*. To perform this task, the control effort that was earlier distributed onto the two different sets of RWs, is transferred completely to *MA*, while *MB* takes, as an input, a virtual torque that is necessary to allow the wheels to deplete their stored energy. This torque can be expressed as the product of the effectiveness matrix and the opposite saturation value, multiplied by the sign of the i^{th} wheel storage to have the fastest possible depleting time. For instance, if the storage has a positive value, the wheel must exert the maximum negative effort until the storage decreases to zero. The task is deemed finished when the storage falls below a specified tolerance in the actual implementation. Of course it is clear to understand that, if the reaction wheels are damping out the storage, they are transmitting an undesired torque that has to be counter acted by the other available actuators. This manoeuvre also highlights the strength in the redundancy, and the flexibility of the controller to perform different tasks simultaneously. The virtual produced torque, therefore, is managed, with respect to the control logic, as a perturbation that the controller has to cope with, plugged directly into the equations of motion.
As the manoeuvre occurs, the control authority is shifted from the *MA* reaction wheels to *MB* RCTs, linearly over the course of 100 s, until the whole required torque is obtained by means of the thrusters. This transfer is executed for two main reasons: first, the thrusters are able to provide a higher net torque with respect to the reaction wheels. Moreover, it is the first procedure to decouple the two satellites in terms of control authority and distribution, thus preparing the control loop to the separation in Phase III. As for the reaction wheels, the WLS allocation method is implemented for the RCT subsystem.
With regards to the configuration depicted in Figure 3.2, the thrusters positioned along the z axis do not play a role in attitude control during this stage. This is because their contribution, which results in a torque around the x and y axis, is comparatively smaller due to the shorter net arm given by their geometric arrangement as compared to the other nozzles aligned with the x and y axis. Additionally, the decision to exclude them aims to simplify the problem characteristic dimension, from $m = 24$ to $m = 16$ and to conserve propellant. As mentioned earlier, the effectiveness matrix employed in this

phase is the one presented in Equation (2.28). At this stage, the spacecraft is required to point inertially a fixed orientation, with the target attitude matching the last targeted orientation in the previous phase.

Data	Value
Initial epoch	05-Jan-2035 05:27:34
Simulation time	3.00e02 s
Computational time	5.13 s
Propellant consumed	2.37e-01 kg
Mean attitude error norm	3.47e02 arcsec
Mean residual norm	5.77e-03 Nm
***MA* final storage**	42.42 Nms

Table 3.3: Performance during Phase II

In Figure 3.7 and Figure 3.8, the spacecraft angular velocity and pointing accuracy are shown. It is clear to see that the added perturbation element, the virtual torque, stresses more the actuators, especially the reaction wheels. In fact, greater error is present in the first part of the simulation. This behaviour is also highlighted in Figure 3.11, where the residual has a spike during the wheels operations, thus proving the authority transfer to be necessary: as the required effort by the reaction wheels is moved to the RCT, the error quickly decreases. In Figure 3.10, on the other hand, the storage trend of both *MA* and *MB* are presented. Clearly, it is easy to understand that, the greater the storage in the client spacecraft, the longer it will take to damp it out. Therefore the dependancy on the trade-off distribution from the previous phase is plainly underlined.

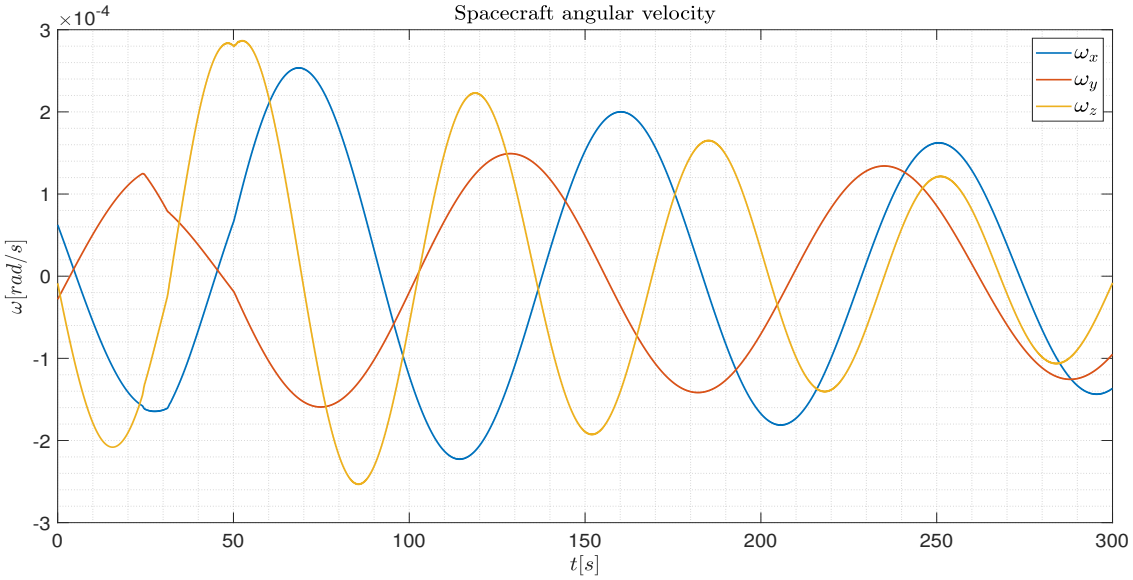

Figure 3.7: *MAMB* angular velocity

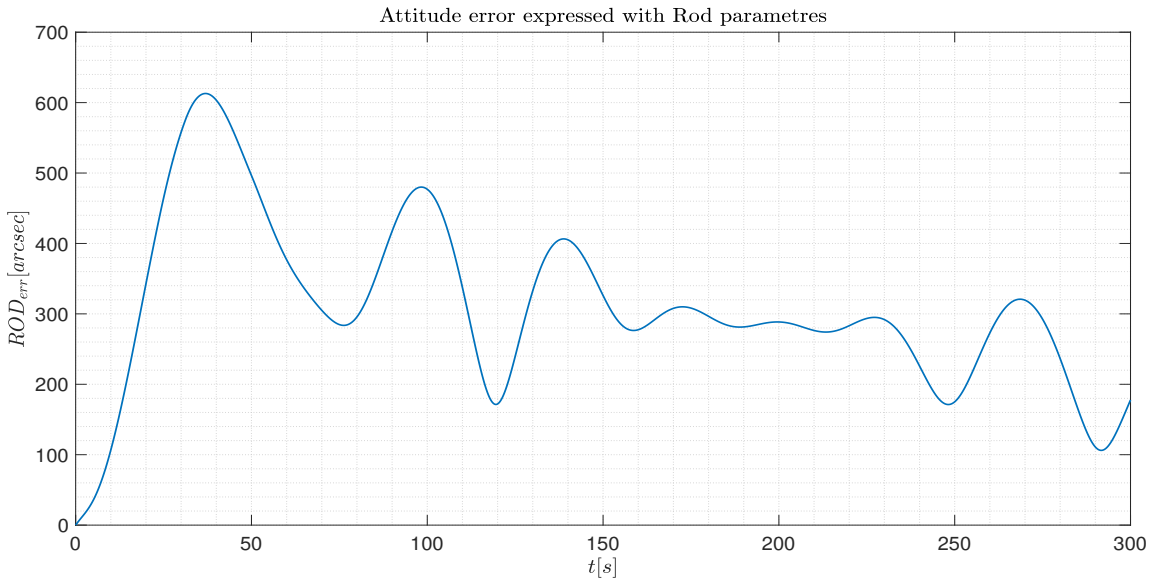

Figure 3.8: *MAMB* attitude error

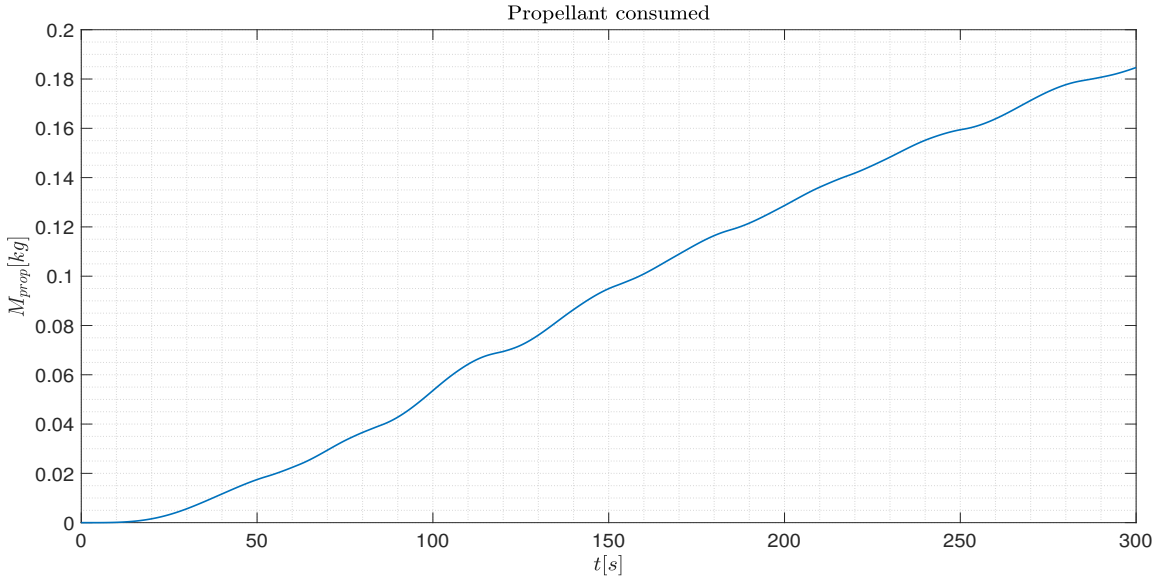

Figure 3.9: *MB* consumed propellant mass

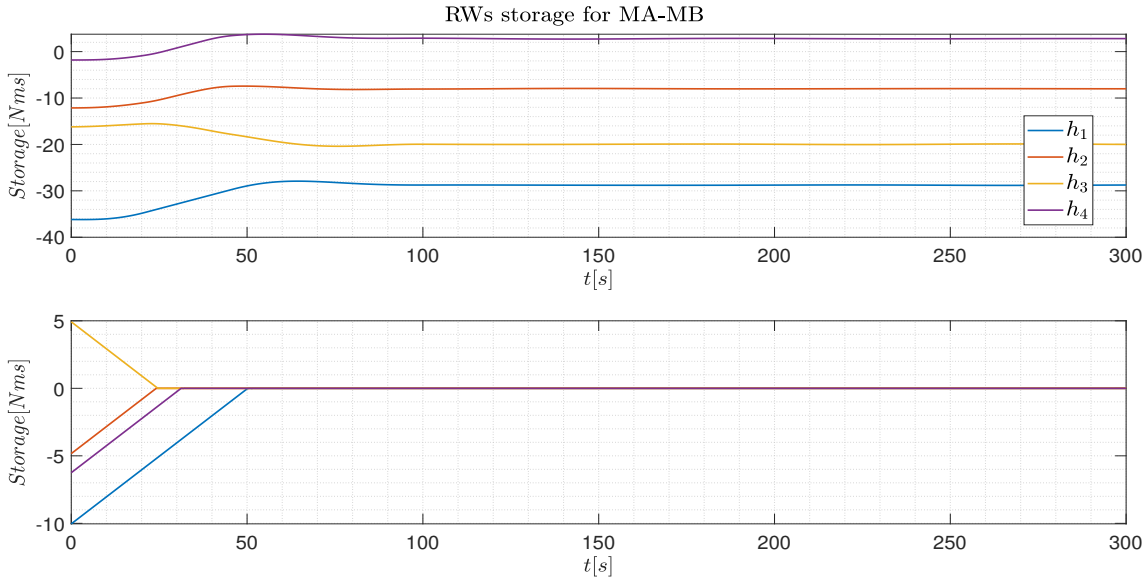

Figure 3.10: *MA* and *MB* reaction wheel storage

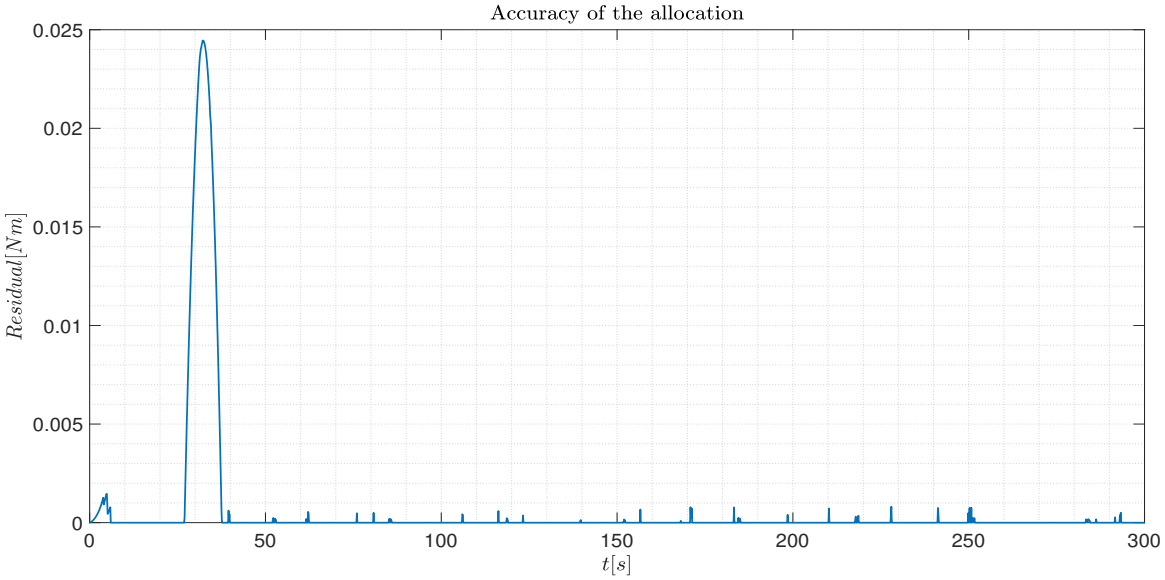

Figure 3.11: Residual between required and provided torque

To summarize, it is important to note that the performances achieved in this phase are directly influenced by the distribution chosen in the previous phase. For example, if Phase I involves a different distribution with more *MA* loading, the MB storage at the beginning of Phase II will be lower than in the presented case. This will result in lower required propellant mass, as the overall damping torque will be reduced, but at the expense of increased *MA* storage. One way to understand this is by visualizing the energy involved, which must be conserved and can be redistributed based on the mission requirements. It may be more advantageous to have a higher *MA* storage or to consume more propellant during Phase II, depending on the scenario. It is worth noting that this energetic consideration holds true for all the other phases as well.

3.2.3. Phase III - *MAMB* separation

Phase III consists in the separation between the two spacecrafts. Once the storage in *MB* is emptied, *MB* detaches from *MA*, following a straight trajectory parallel to the z axis[3]. This manoeuvre is modelled as an instantaneous division between *MA* and *MB*, in which an impulsive variation of the mass and inertia properties is introduced. This phase is particularly critical in terms of the pointing accuracy, since any kind of collision must be avoided, or the mission may completely fail.

[3]Of course the characteristic time and distances enable a relative dynamics work frame, and the trajectory curvature of the orbital path can be neglected.

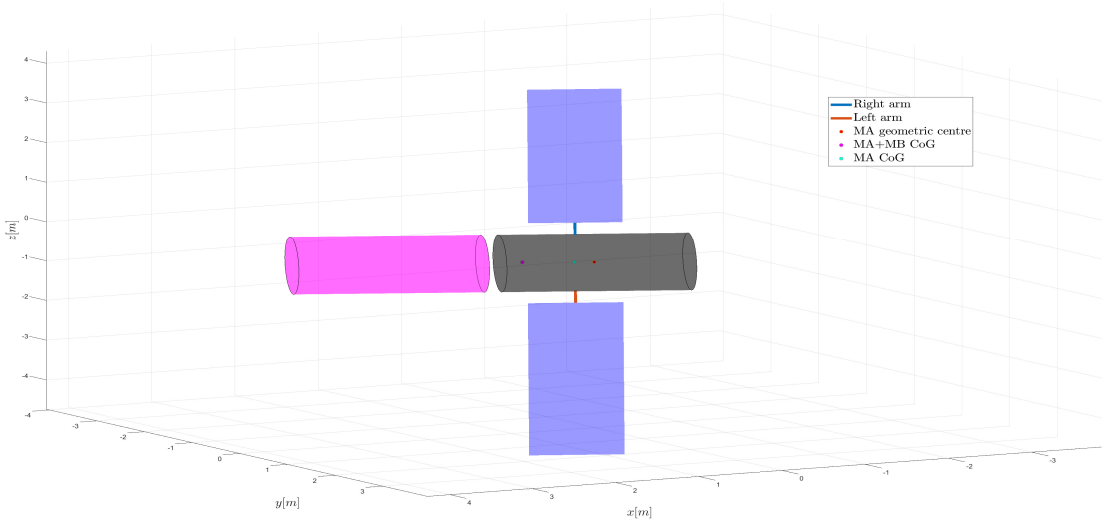

Figure 3.12: Spacecrafts detachment visualization

In this phase, the attention is brought onto the first spacecraft, *MA*, because it is equipped with solar panels that can hit *MB*, so it is the spacecraft with tighter pointing constraints. To assess the performance, an impulsive perturbing torque is applied to *MA* to simulate the separation load. This quantity is applied instantaneously at 50s during the simulation, drawn randomly from an interval of $\pm 50N$ in the three directions. Regarding control, the high-level control is disabled for a brief period before and after the separation to prevent a collision when the two bodies are in close proximity. This is because a sudden control reaction could result in an impact. Instead, low-level control authority is restored to the MA reaction wheels, and the WLS algorithm is employed.

Data	Value
Initial epoch	05-Jan-2035 05:32:34
Simulation time	5.00e02 s
Computational time	2.14 s
Mean attitude error norm	2.51e02 arcsec
Mean residual norm	2.22e-01 Nm
***MA* final storage**	43.57 Nms

Table 3.4: Performance during Phase III

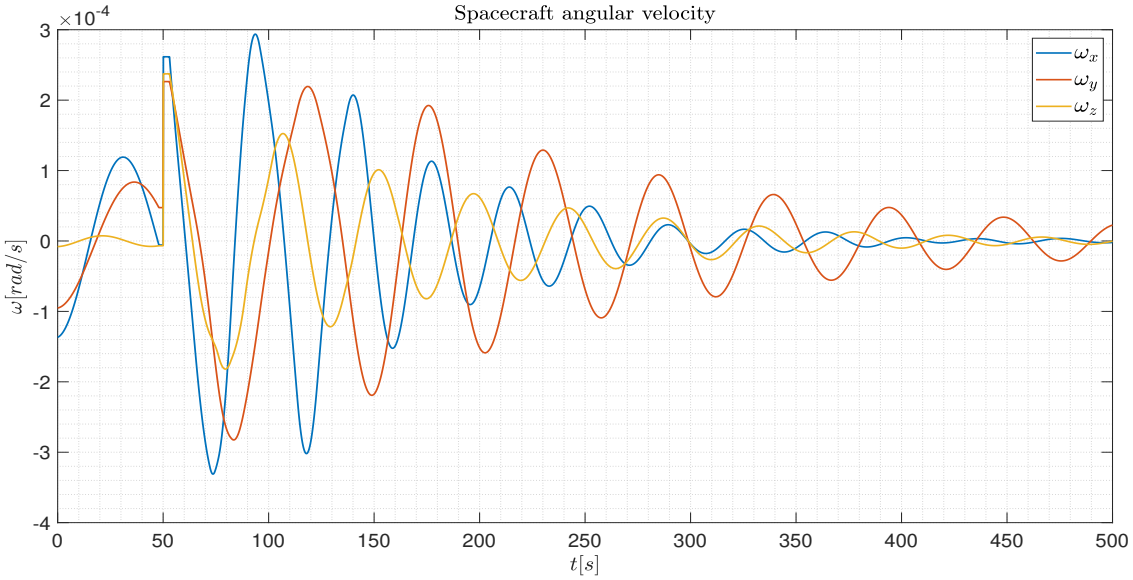

Figure 3.13: *MA* angular velocity

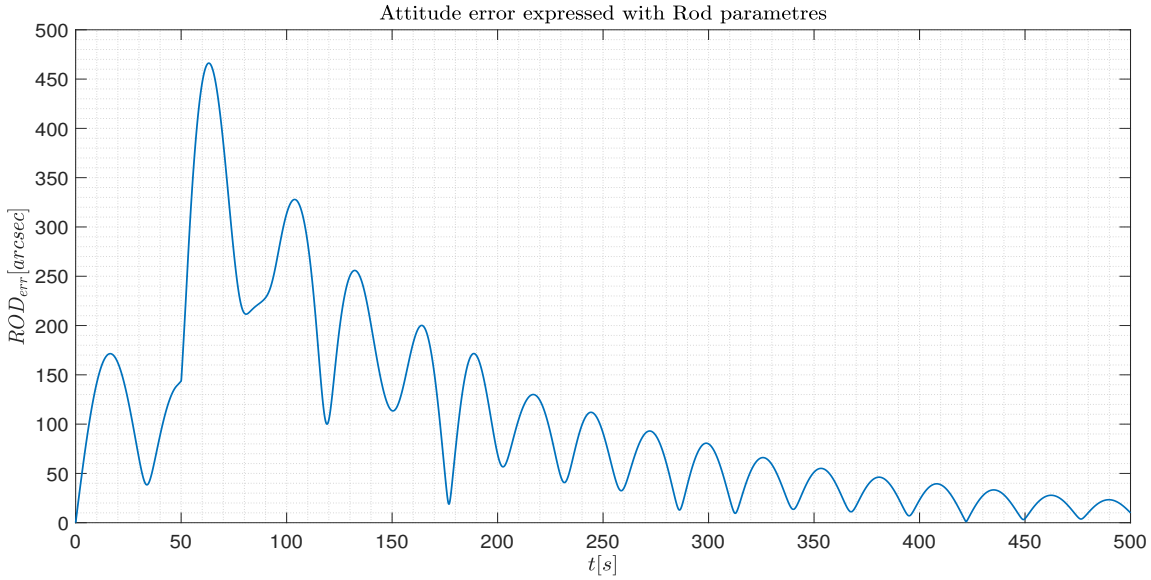

Figure 3.14: *MA* attitude error

As Figure 3.13 and Figure 3.14 report, the impulse load introduction is clearly visible at $t = 50s$, where the analyzed quantities are subject to an almost discontinuous variation, a spike trend. Then, the controller is switched back on and the pointing error decreases throughout the rest of the simulation, stabilizing the spacecraft. The same behaviour is also highlighted in the next graphs, where the storage is characterized by a smooth trend before the separation, then sharp variation to cope with the disturbance and, finally, the

progressively damped oscillations as the body attitude approaches the target one. In Figure 3.16, on the other hand, it is clear to see the presence of unattainable moments early after the separation, with a decreasing trend as the system goes back to regime.

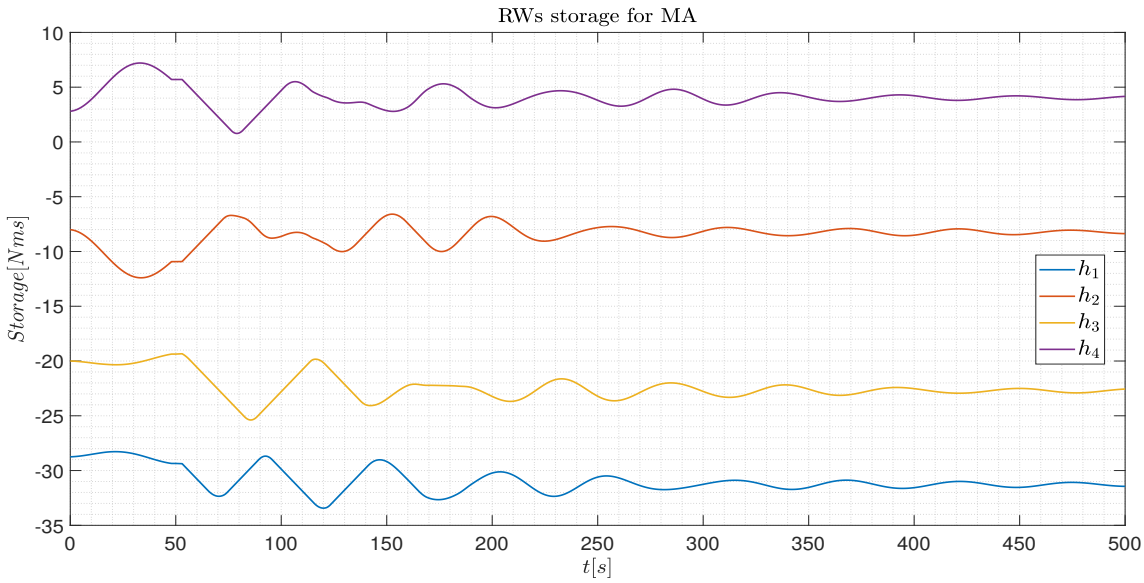

Figure 3.15: *MA* reaction wheel storage

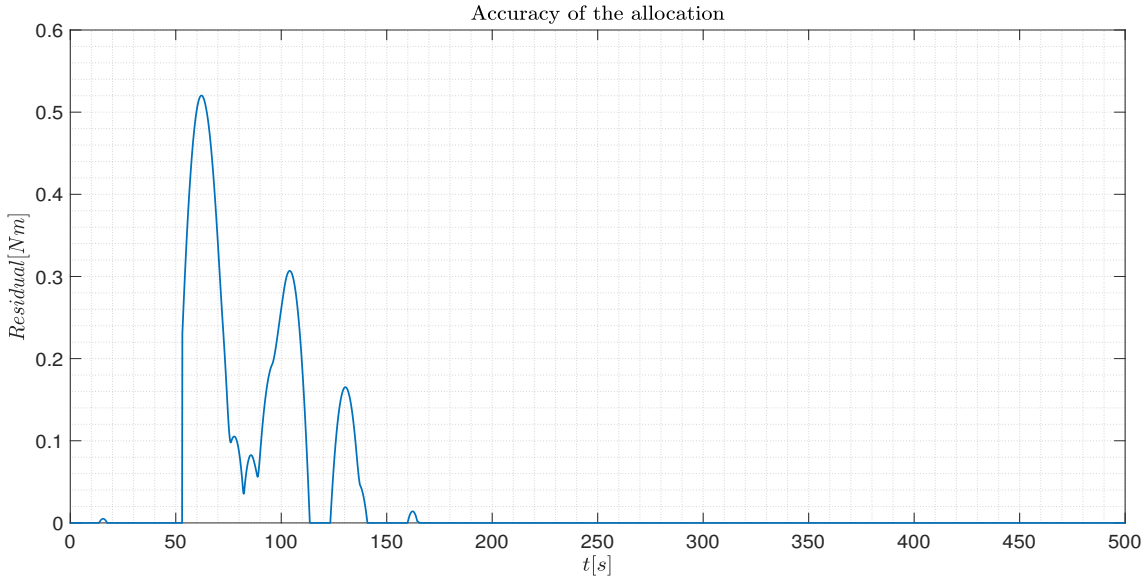

Figure 3.16: Residual between required and provided torque

In conclusion, this phase not only can be used to assess the possibility of reallocation when strong unattainable moments are present, but also to see how the controller behaves when dealing with discontinuities or sharp variations in the system state.

3.2.4. Phase IV - *MB* thrust vectoring compensation

The client spacecraft *MB*, after separation, has to execute a set of operations that require a thrust vectoring action. The resultant effects on the spacecraft are a reaction torque and force applied to *MB* centre of gravity. Hence, it is vital that the control loop balances these perturbing elements. The procedure followed for the analysis is the definition of a decoupled model: the structural parts that compose the thrust vectoring, and its contributions, are treated as external elements with respect to the rest of the spacecraft. In this way two subsystems, respectively spacecraft and thrust vector, are defined. Focusing on *MB*, the satellite will see the other part actions as an external disturbing effect in terms of torque and force, applied at the interface between the two bodies. If the reaction force and moment are perfectly countered, the spacecraft would appear as a fixed base to the thrust vectoring system. Thus, it is evident that the spacecraft must be capable of resisting the reaction contributions for operations in this phase to proceed. However, given the presence of perturbing net forces and the need for precise manoeuvres, it is necessary to consider the overall pose of the spacecraft in the problem analysis. For what concerns the equation of motions, the translational acceleration are added to the state vector:

$$\begin{bmatrix} M\ddot{\boldsymbol{x}} \\ I\dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F_{ext}} \\ (I\boldsymbol{\omega}) \times \boldsymbol{\omega} + \boldsymbol{T_{ext}} \end{bmatrix} \tag{3.8}$$

The pointing criteria for this phase remains unchanged from the previous one, meaning that the spacecraft must maintain a fixed orientation. Additionally, it must resist any external vectoring forces that may cause it to shift from its current position. Regarding the high level control, the same logic implemented for the attitude control is used for the displacement control, where the error to be plugged in the PID control is the drift from the origin of the target BRF attached to the *MB* centre of mass, since it represents the stable, fixed, unperturbed, initial condition. Considering the low level control, on the other hand, the implemented strategy is the following: first the required torque is allocated onto the reaction wheels, by means of the WLS algorithm. Then a daisy chaining approach is used, which consists in a sort of allocation method that distribute the effort considering a chain of saturating actuators. The available actuators are divided into subsequent groups, each able to solve the allocation. The required torque is allocated onto the first group, then, if saturation is detected, the residual is allocated onto the second group and so on. In the considered case, when the reaction wheels are not able to produce the required moment, the RCTs enter in actions, providing the left residual torque. The procedure can be seen in the next series of equation.

$$\boldsymbol{m_{RW}} = B_{RW}\boldsymbol{u_{RW}} \tag{3.9}$$

$$\boldsymbol{m_{residual}} = \boldsymbol{m} - \boldsymbol{m_{RW}} = B_{RCT}\boldsymbol{u_{RCT}} \tag{3.10}$$

This method is particularly useful when unattainable moments are expected, but it is still vital to preserve the propellant mass. From an energetic point of view, it is preferred to store the momentum rather than expend propellant mass. The chain is necessary to avoid the presence of residual, which could interfere with the target pose definition, a delicate and even stricter requirement in this phase than the others. Hence, the allocation will distribute first the action onto the wheels, then, if the torque is unattainable, the RCT will compensate the residual. Concerning this set of actuator, the effectiveness matrix is now modified to take into account also the allocation for net force production. Since the problem now considers also the CoG position, the B_{RCT} matrix will be augmented to manipulate six degrees of freedom, thus $n = 6$. The upper part of the matrix (line from 1 to 3 per m columns) will now contain the maximum thrust each nozzle can produce in the three directions xyz. The second part (line 4 to 6 per m columns), instead, will be equal to the one defined in Equation (2.28). Considering the fact that fine pointing is required, and relevant loads are applied, all the thrusters concur in the allocation problem, thus $m = 24$. The final effectiveness will be $B \in \mathbf{R}^{6\times 24}$.
As stated before, this approach is useful if a large presence of unattainable moments is present, but it is still preferable to save fuel. In this scenario, the profile model of thrust and torque transmitted to the spacecraft base are shown in Figure 3.17, and it is clearly visible that the actuator effort, which ideally should be the opposite of the quantities in the graphs will stress more the control loop.

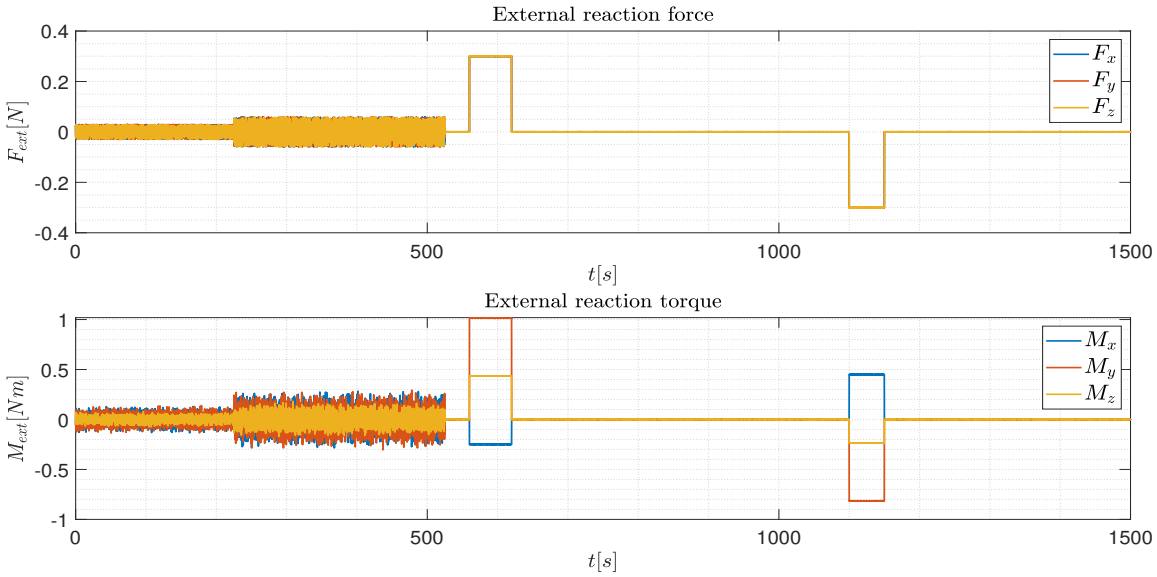

Figure 3.17: Reaction forces and moments applied at the interface

It is possible to divide the trend into three main phases:

- First part with random white noise, until approximately 600s.
- Step both in force and torque, which lasts roughly 300s.
- Second step, after 1000s of simulation.

For what concerns the environmental perturbation, they are still related only to the attitude, as their contribution to the translation part can be neglected when compared to the values reported in the above figure.
In the next table, the results of the Phase IV simulation are reported. It is immediately visible that the concurrent usage of both actuator sets keep the residual sufficiently low. The pointing accuracy, also, stays below reasonable tolerance. In conclusion, the norm of the mean drift from nominal resting position of the gravity centre is outlined, proving that the pose control loop works successfully.

Data	Value
Initial epoch	05-Jan-2035 05:40:54
Simulation time	15.00e02 s
Computational time	2.09e01 s
Propellant consumed	7.64e-01 kg
Mean attitude error norm	5.13e01 arcsec
Mean residual norm	8.14e-06 Nm
***MB* final storage**	6.52 Nms
***MB* drift mean norm**	3.36e-04 m

Table 3.5: Performance during Phase IV

As done for the other phases, the trend of different quantities key to the simulation analysis are here displayed. First, in Figure 3.18 and Figure 3.19, the trend of angular velocity and tracking error are reported. It is possible to notice the corresponding trend shown in Figure 3.17: an initial perturbed trend, then a set of spikes corresponding to the first step, and a second at the latter.

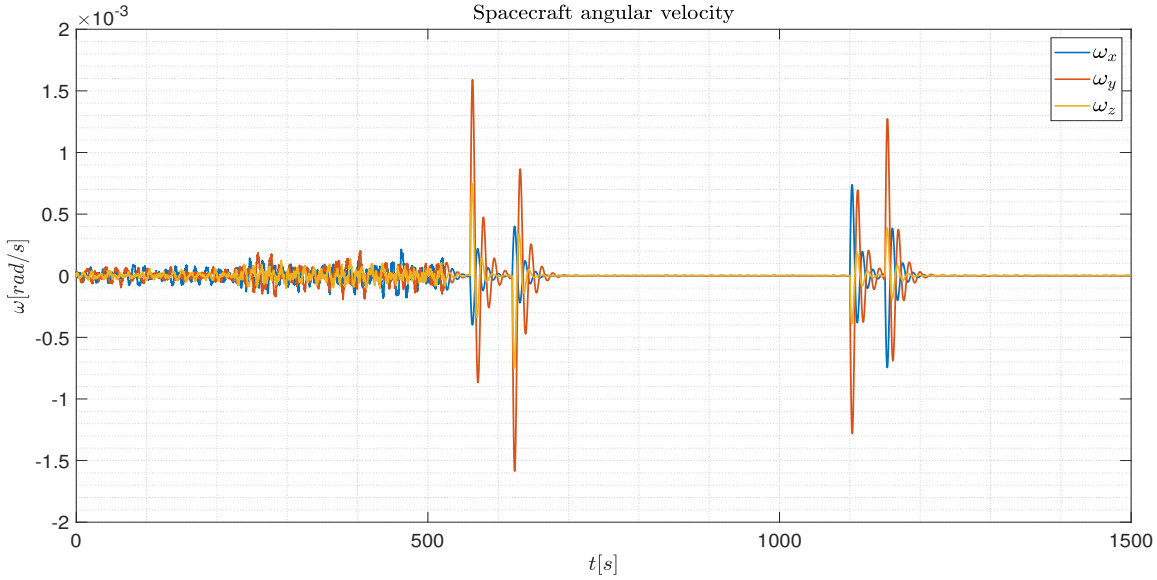

Figure 3.18: *MB* angular velocity

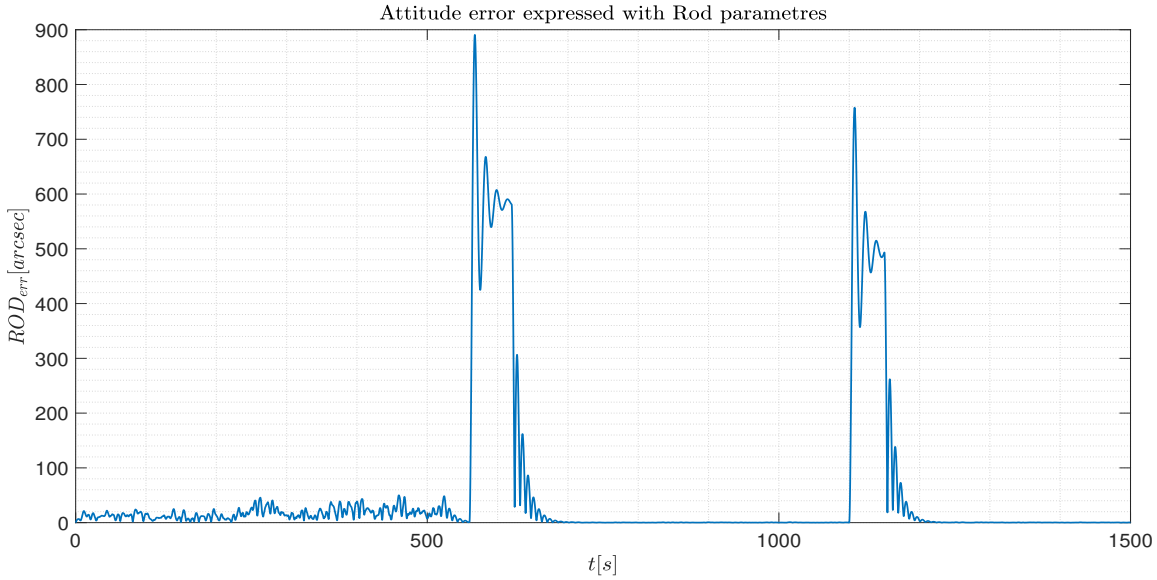

Figure 3.19: Pointing accuracy

In terms of cost, Figure 3.20 displays the storage level, while Figure 3.21 shows the propellant consumption. As before, the graphs exhibit three distinct sections: a small increase in both measures, followed by plateaus dividing the first and second impulses. Regarding storage, the torque accumulation is dampened during the second impulse due to its opposite sign compared to the first impulse.

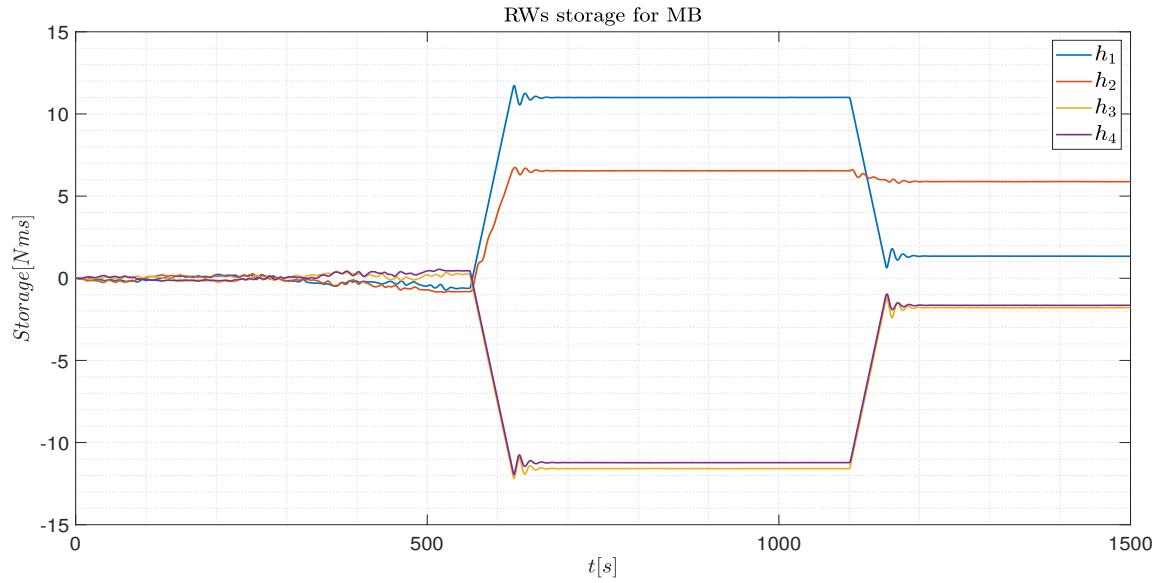

Figure 3.20: *MB* storage

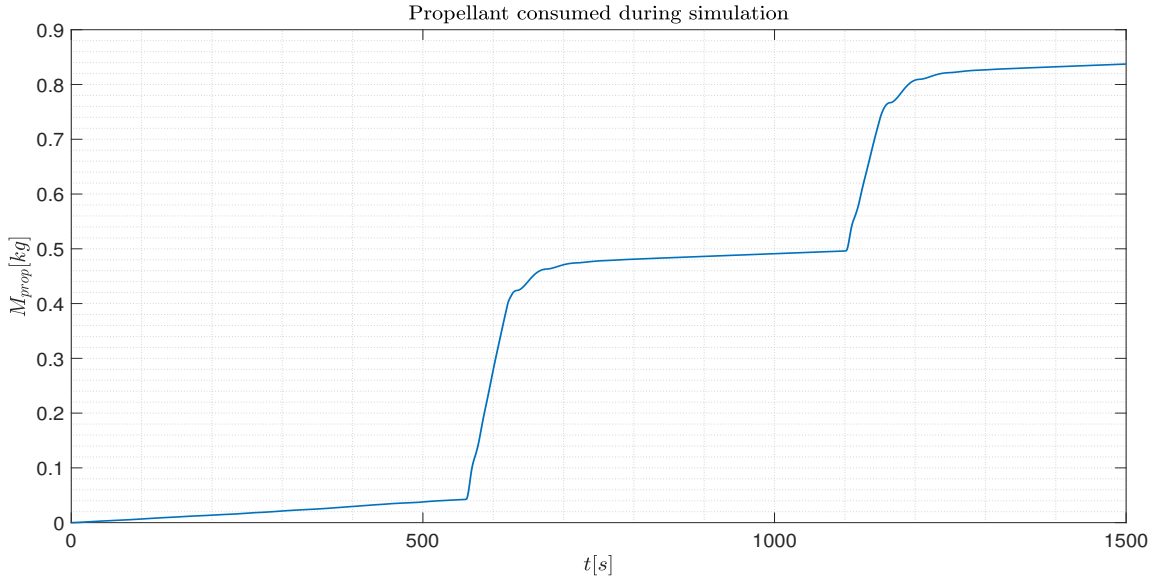

Figure 3.21: *MB* consumed propellant

Regarding accuracy in the allocation, the residual is presented in Figure 3.22. It is characterized by a practically zero value apart from two spikes. The second higher spike is probably due to the first impulse magnitude, which is rather heavy and especially faster than the control logic update frequency. The first one, instead, may also be due to the incapacity, in that instant, of the reaction wheels to provide the requested torque, whose random value may be one of the highest through the whole first simulation part. The residual should then be fed to the RCT, but the output from the allocation does not overcome the hardware threshold of 1e-3 N to active the RCT valve. Therefore a slight error is present, which is still reasonably low. This aspect can be also detected from Figure 3.23, as the RCTs are active only when the steps occur. On the other hand, Figure 3.24 delineates the centre of mass divergence from the origin. The main translation, corresponding to the spikes, are quickly counter acted by the RCTs, as the maximum displacement is less than 4 cm. In conclusion, the performance obtained in this phase not only ensure a precise pointing, but also demonstrate the flexibility in the control authority that shifts between RW and RCT and, most of all, that the allocation problem can be easily solved even when considering an overall pose definition, with a compliant effectiveness matrix.

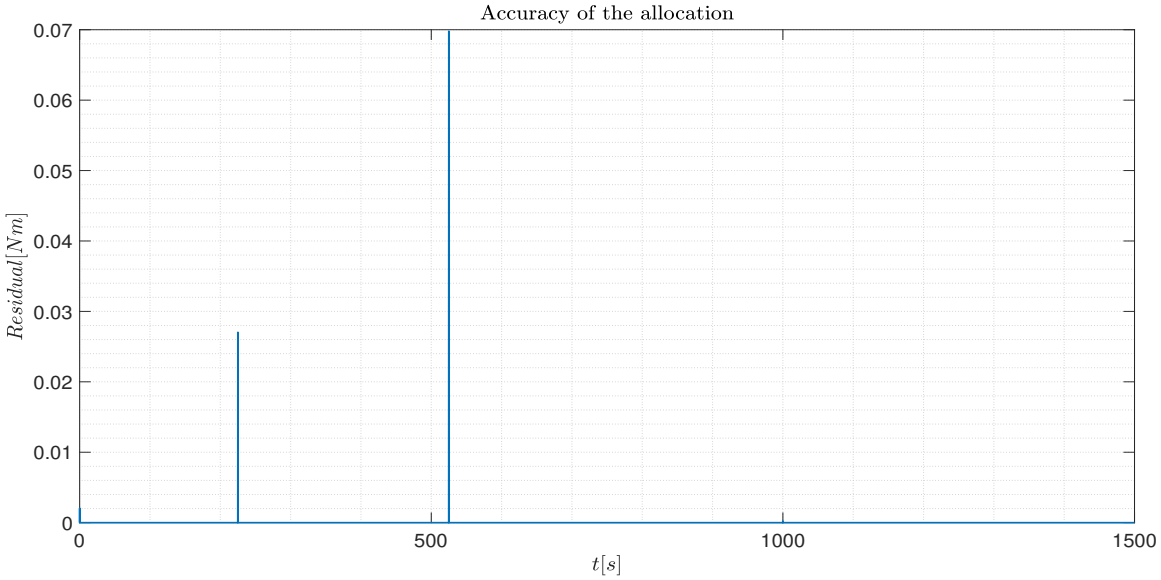

Figure 3.22: Residual between required and provided torque

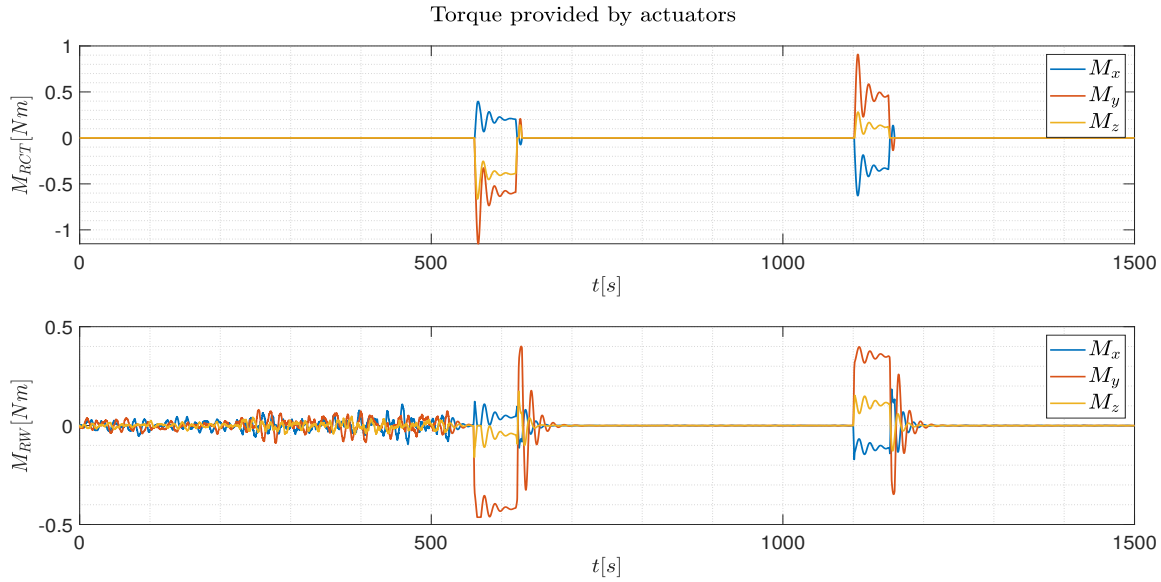

Figure 3.23: RW and RCT provided torque

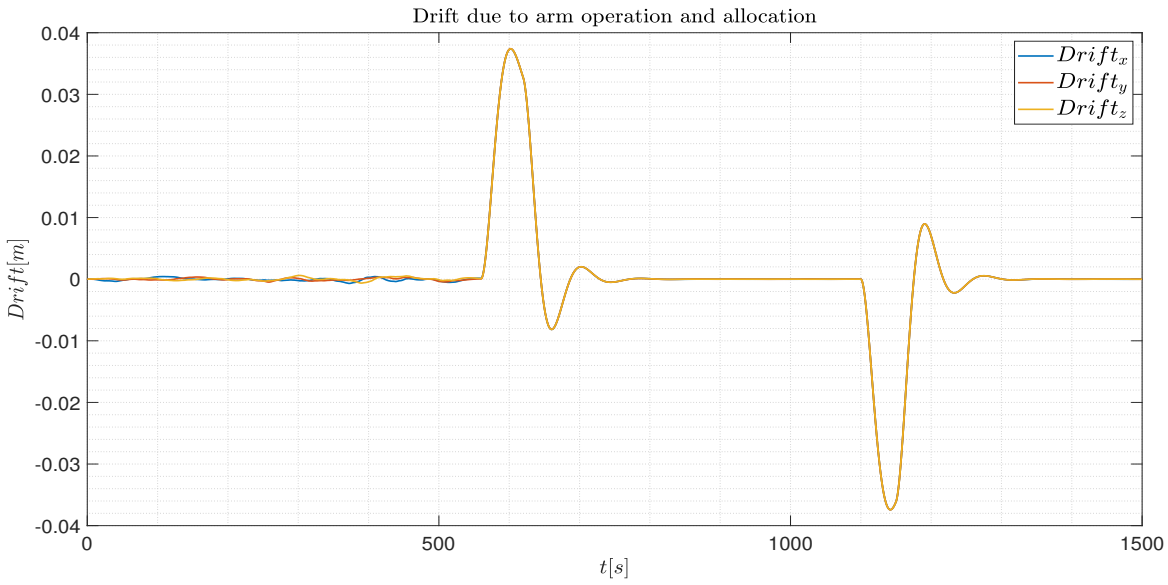

Figure 3.24: CoG drift from nominal position

3.3. Robustness analysis

So far, the various operations, actions, and requirements have been examined using a nominal mission profile, which represents an ideal analysis of how all the conceptual operations should be carried out. It is now key to investigate how the system behaves when conditions are not matching the nominal ones. The aim of this investigation are mainly two:

- To assess the robustness of the mission profile, especially in the delicate Phase III and IV.
- To examine the behaviour of the allocation method when uncertainties are introduced into the control scheme.

Therefore, a Monte Carlo simulation is defined, with the introduction of uncertainties and errors in the loop. First, it is necessary to identify which elements can be affected by noise and uncertainties. Regarding the first element, white noise is added to the angular velocity vector and body attitude DCM, in order to simulate the intrinsic errors that are bore in the measurements, respectively, from gyros and sensors. So, the control scheme will not deal with the true quantities, but with their estimation. Naturally, the pointing error is still computed with respect to the real quantities, while the high level control will receive the affected ones, thus leading to generally worse performance with respect the nominal case. For what concerns the second aspect, the system hardware is going to be

affected by uncertainties that derive from the manufacturing and handling processes. For example the centre of gravity position may differ from the one in the ideal mathematical model. It is required that all the input values that represent a hardware component are scattered with a specific uncertainty. This will generate a cloud of probability distribution around the mean, which is the mathematical model value. The parameters that are going to be scattered are the centre of gravity positions, the inertia matrices, the orientation of the reaction wheels, the orientation of the nozzles and the thrust profile each can release. The mean value and diagonal covariance matrix are defining characteristics of every distribution. The next step is to create a random initial spacecraft configuration, simulate the mission and collect data to assess the performance. To create a random set of parameters for the different initial configurations, a pool can be created, which encompasses all possible distributions of the scattered variables. Next, various random extractions are made, each corresponding to different initial mission conditions. For instance, a single extraction could have its center of gravity located at a specific position, but a subsequent extraction may cause this value to shift to another position within a certain range based on the covariance matrix, centered around the mean. These initial configurations are propagated throughout the mission's phases II, III and IV, as their profile is more delicate with respect to the coasting initial phase, that is not included in the investigation. The simulation resulting parameters, especially cost and accuracy, derived from each sample are then recorded. By repeating this process for each sample, it is possible to generate the sample mean and sample covariance for each crucial quantity involved in the mission's accomplishment, thereby evaluating the system's robustness. The mission definition remains the same exposed through this chapter.
In Figure 3.25 and Figure 3.26, respectively, the cumulative density functions for the propellant mass consumed in Phase II and IV are reported. The graphs show the distribution of cumulative probability for a sample to occur in the simulation regarding the propellant mass. In other words, the two plots illustrate how likely is the system to consume a certain amount of fuel. The mean values of these probability distribution are, correspondingly, 1.16 kg for Phase II and 1.62 kg for Phase IV. The consumption increment is clearly due to the added noise in the measurement and generally coarser actuators capacity.

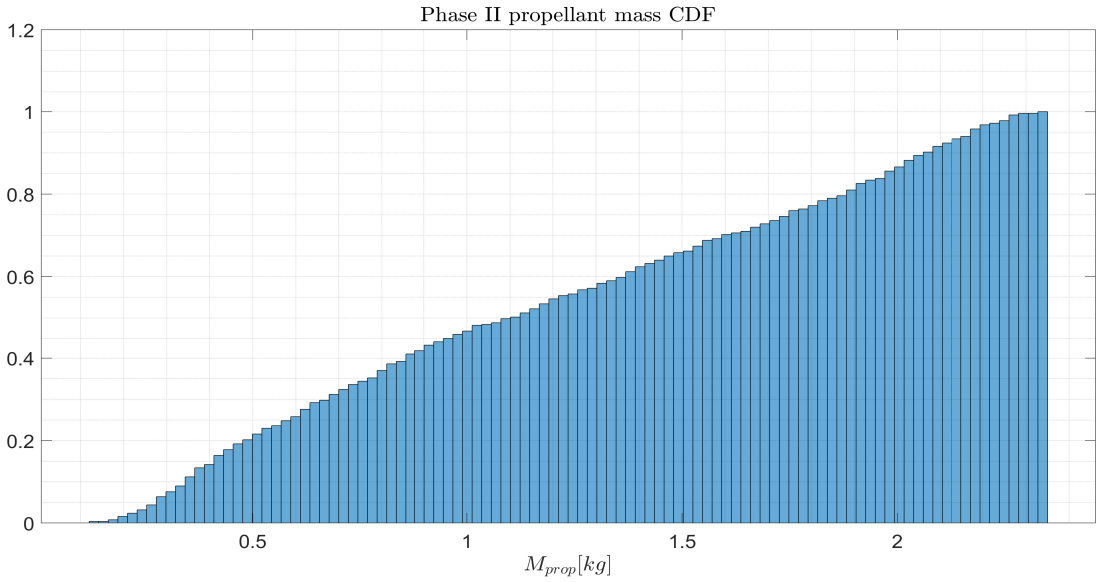

Figure 3.25: Cumulative density function for the consumed propellant mass in Phase II

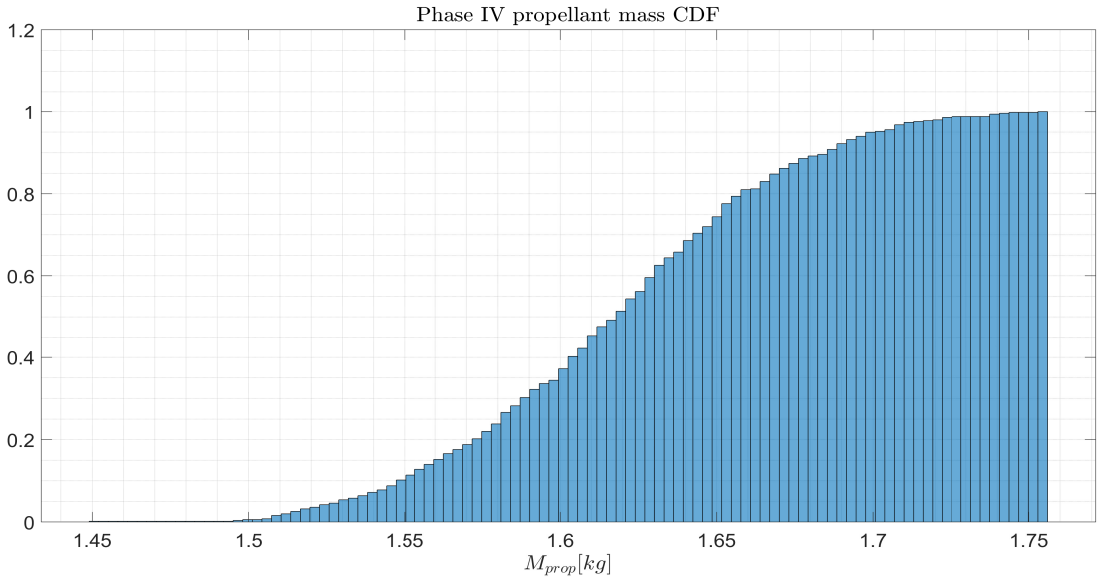

Figure 3.26: Cumulative density function for the consumed propellant mass in Phase IV

Regarding the storage profile during the phases, in Figure 3.27, Figure 3.28 and Figure 3.29 the CDF of the storage norm at the end of each stage are presented. The corresponding mean values are 47.27 Nms, 45.72 and 7.61 Nms. The first two plots are related to the *MA* storage in Phase II and III, while the third to Phase IV *MB* storage.

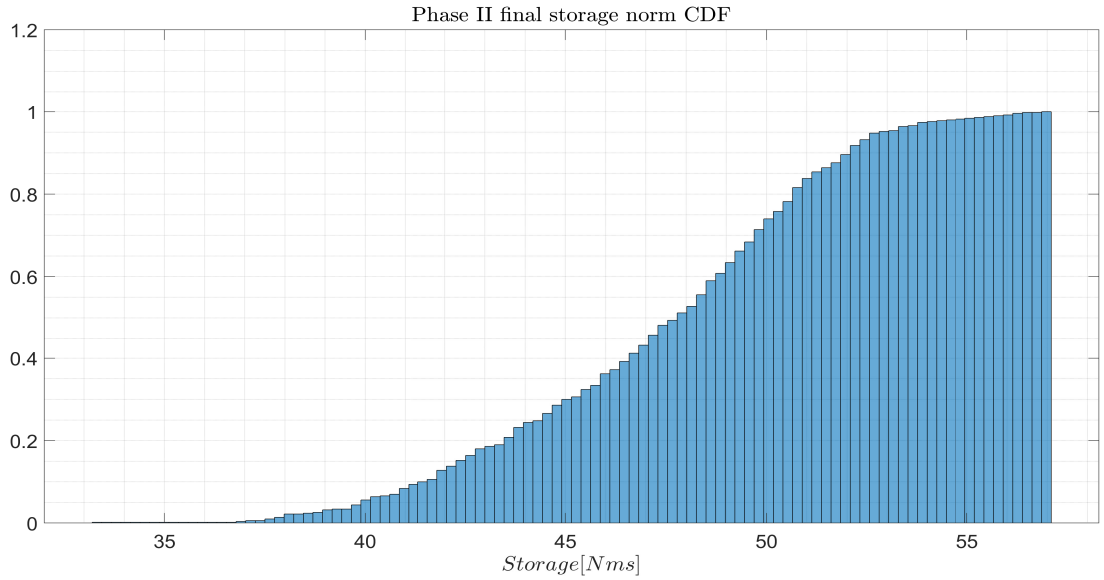

Figure 3.27: Cumulative density function for final storage norm in Phase II

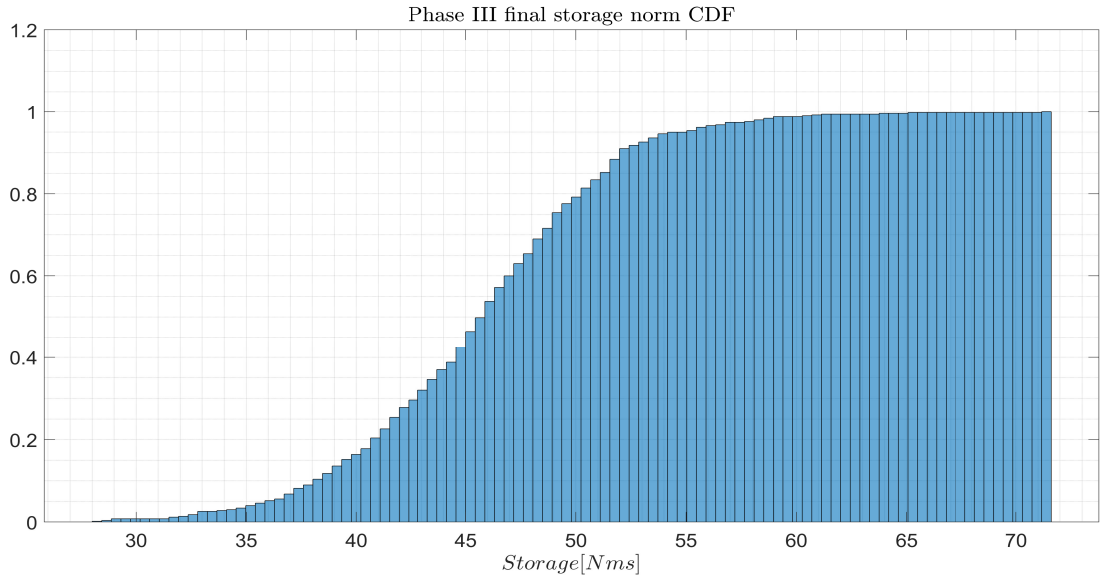

Figure 3.28: Cumulative density function for final storage norm in Phase III

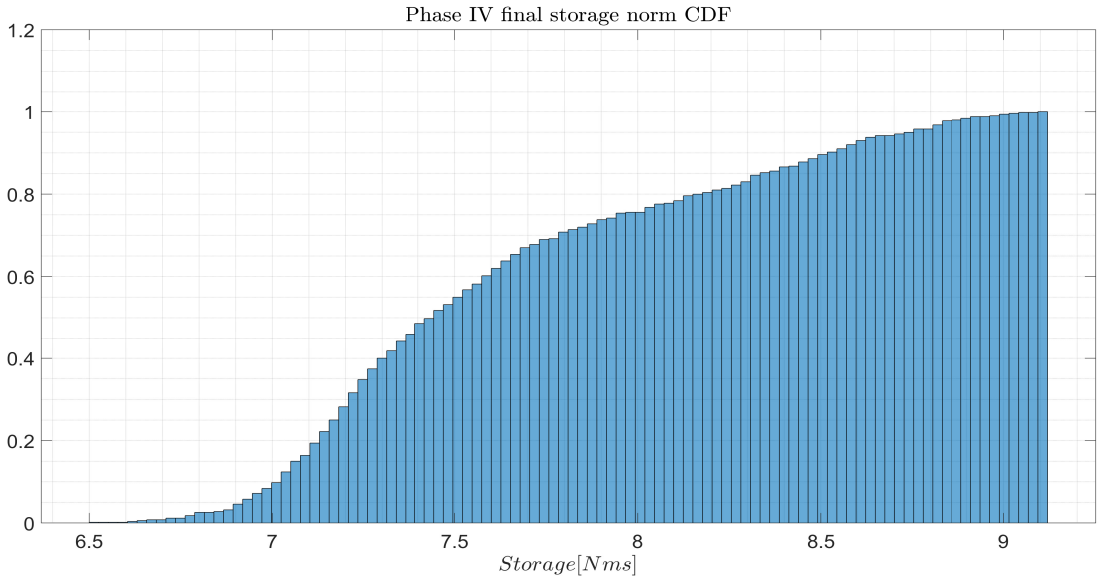

Figure 3.29: Cumulative density function for final storage norm in Phase IV

In the following images, instead, the pointing accuracy is displayed. In each graph, three curves are delineated: for the whole vector that describe the trend of the error, at each instant, the mean value is computed through the different samples, in order to obtain a mean trend for the whole simulation. The same is done for the minimum and maximum values. Therefore the error envelope is completely characterized, by the most probable trend, the best and worst case scenarios. Again the RP error norm is displayed.

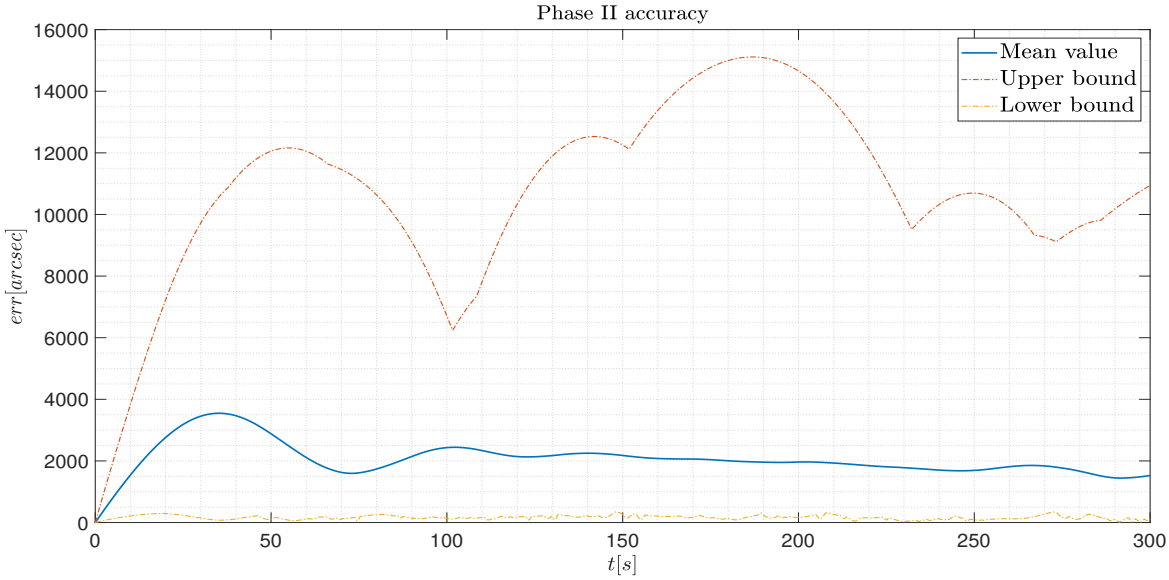

Figure 3.30: Accuracy in Phase II

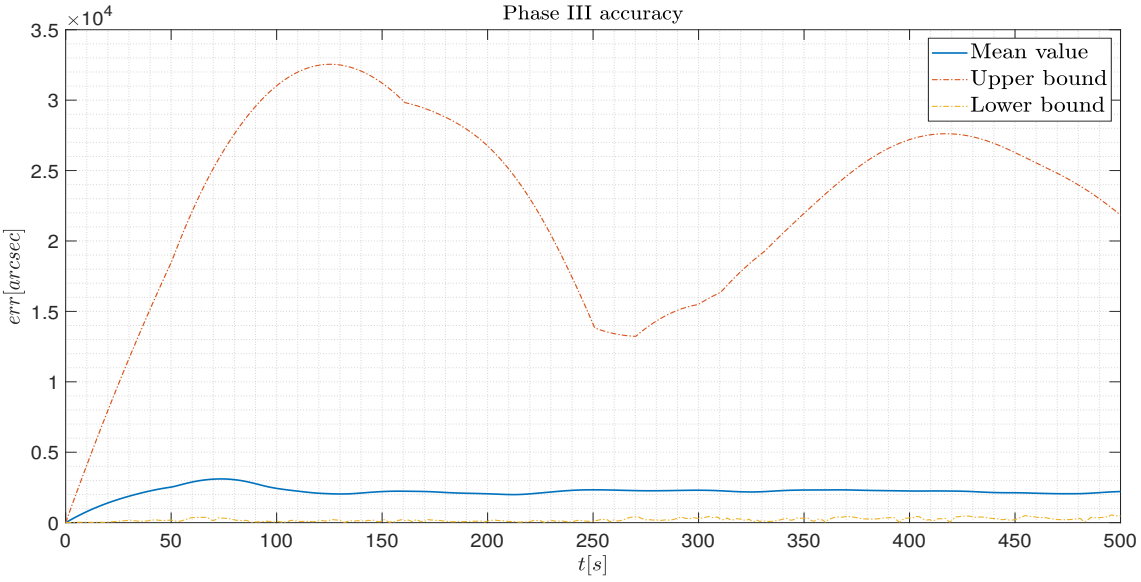

Figure 3.31: Accuracy in Phase III

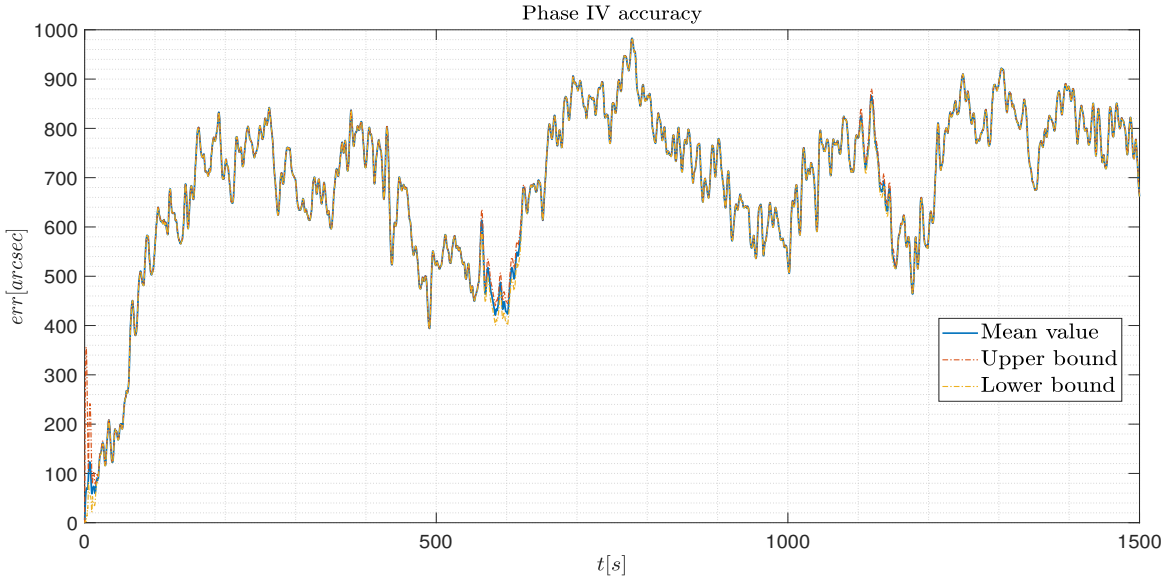

Figure 3.32: Accuracy in Phase IV

In Figure 3.30 and Figure 3.31, the mean performance are generally worse with respect to the nominal mission case, while the worse case demonstrate that the pointing could be compromised. Considering the maximum error reported in the red curve in Phase II, assuming small rotations, the maximum displacement at the tip of the spacecraft is estimated to be 0.31 m, while in Phase III lower than 0.30 m. If, on one hand, the mean value trends are still feasible, it may be necessary to assess if the error in the worse case

is still acceptable or if the design should be revisited, for example, considering tighter requirements for the manufacturing process. The pointing accuracy in Figure 3.32, even though rather worse with respect to the nominal case, is still manageable. Moreover, the spiked trend seen in Figure 3.19 is covered by the noise measurements. Finally, the same approach is followed by the drift in Phase IV. In the next plots, the displacement envelope is defined. Clearly, since the displacement is mainly due to the thrust vectoring reaction force, the trend throughout the samples is practically the same.

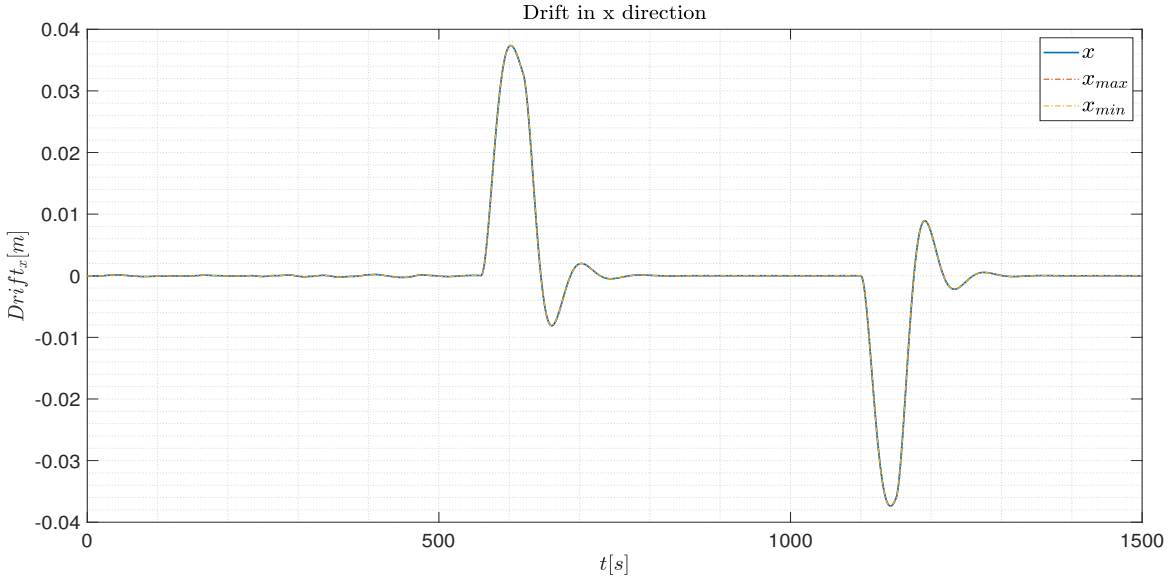

Figure 3.33: Drift in the x direction

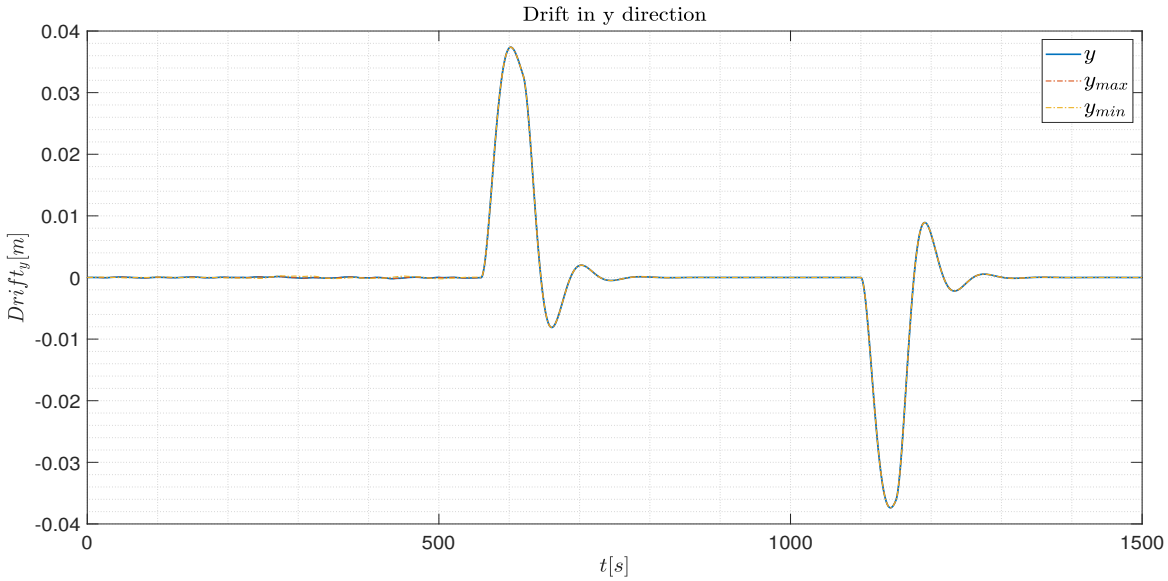

Figure 3.34: Drift in the y direction

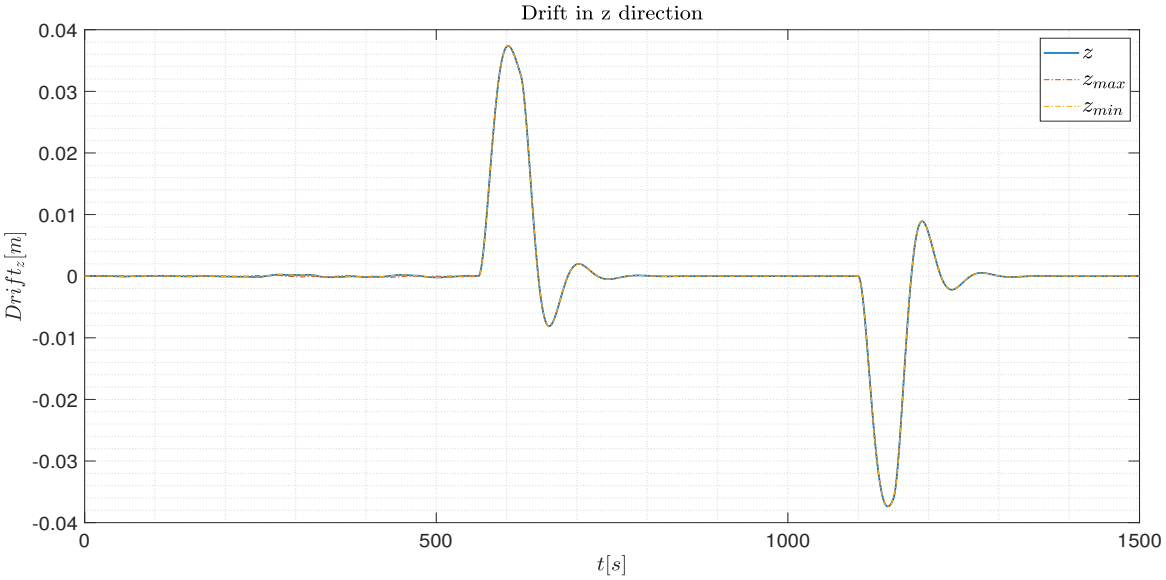

Figure 3.35: Drift in the z direction

The previous plots demonstrate the mission design and implemented allocation strategy's ability to handle uncertainties and measurement noise. For the sake of brevity, only the performance with the WLS algorithm implemented were reported, but it is possible to state that the conclusions drawn in Chapter 2 can be applied also in this case, regarding the performance and overall efficiency. This final survey on robustness concludes the overall analysis of control allocation. Initially, various strategies were examined and their features were highlighted in a nominal attitude control simulation. Later, a developing mission was introduced, and the control allocation was further studied. This workflow not only showcases the benefits outlined in the introduction but also serves as the first step in designing the control loop for an in-orbit servicing mission.

4 | Conclusions and future developments

The main objective of the presented work is to provide a practical and valuable tool that assists the reader in selecting the most appropriate and customized solution for achieving optimal, precise, and prompt control allocation. Through the investigation, it has become apparent that there is no one "perfect" strategy among those proposed, as each has its own advantages and disadvantages that prioritize a specific aspect of the control allocation problem. The optimal solution depends heavily on the mission scenario, architecture, environment and requirements. Throughout the survey, the different allocation strategies were studied and divided into two main groups. First, allocation methods that focus on the accuracy, minimizing the residual and indirectly influencing the cost minimization aspect. On the other hand, the second group collects the methods that prioritize the cost minimization when a certain set of constraints are applied to the problem. To summarize, it emerged from the analysis that:

- Straightforward and intuitive solutions such has the Generalized Inverse are able to compute extremely fast a rough solution.
- The Cascade Generalized Inverse is able to increase the performance of the Generalized Inverse when it comes to deal with unattainable moments, exploiting efficiently the inherent redundancy of the system.
- Null Space solution is a simple allocation strategy especially suited when applied to the RCT system. Still, the solution computed can be enhanced.
- To increase the efficiency in terms of cost minimization, linear programming is able to find a better solution, but is characterized by high computational times.
- Direct Allocation settles in between optimal solutions and Generalized Inverse, as it is faster with respect the first family of strategies and computes a more refined solution with respect to the second one. The major points of weakness are the tight constraints that characterize the problem, but, as seen in Section 2.5, they can be

loosen at the expense of optimality degradation in the computed solution.

- Weighted Least Square, on the other hand, is particularly useful when it is necessary to take into consideration, at the same time, both the accuracy and optimality of the computed solution, as the user is free to weight the interested aspect.

As stated in the introduction, the main, key advantage of control allocation is the decoupling between the high level and low level control. In fact, the high level control was completely disregarded in the whole analysis, becoming just a black box that took as input the attitude error and gave, as result, the required torque. This procedure not only extremely simplifies the overall control subsystem formulation, but also enables the user to develop the two blocks separately, which gives high flexibility when dealing with preliminary design corrections and iterations. In addition, not all control designs methods are able to handle redundancy, or actuator failure. Therefore the decoupling is required for a more relatable, realistic and practical application when developing an AOCS subsystem.
As stated at the beginning of the thesis, the presented work did not limit in the simple description and evaluation of the different allocation methods, but, once the assessing terms were strictly stated, explained and transposed into the space field, it built a reliable instrument that helps choosing the tailored method for an application without influencing the control logic in any way. Moreover, it showed how vital and suited is the application of the control allocation in a real orbit servicing mission scenario. In fact, in Phase I computational efficiency for long time span simulation is demonstrated, in Phase II the inherent flexibility of the allocation, while in Phase III the ability to manage high intensity impulses, sharp system state variations and unattainable moment response. In Phase IV, instead, the allocation was considered for a pose control. With the Monte Carlo analysis, finally, also the robustness was tested. Again, the control logic was not modified, but it was just a matter of manipulating properly the effectiveness matrix. Therefore a seemingly complex problem is reduced into a simple and intuitive operation.
In present literature, few examples of control allocation study can be found when dealing with in-orbit servicing missions, which is an extremely vast topic. These types of missions, that are becoming increasingly required to avoid debris creation and to make space a more affordable environment, need intrinsically high flexibility in the AOCS subsystem, which can be granted by redundant actuators. Naturally, the next steps involve improving control allocation performance, discovering new customized solutions, and incorporating time-varying effectiveness matrix into the analysis, particularly when dealing with movable appendages like, for example, robotic arms or thrust vectoring. In fact, these components could assist with attitude control instead of being treated as an external, decoupled element. The joint effects, namely reaction force and torque, could be

used, instead of being treated like perturbations, as actuating elements with a certain producible effort, depending on the ratio between base and appendage mass. Considering a robotic arm, for example, due to the conservation of angular momentum, the movement of this appendage would generate a reaction onto the basis, thus providing an action that affects the body attitude. This application could be particularly useful when the arm and basis have comparable masses. The approach avoids the need for propellant mass consumption to counteract their movements and instead incorporates them into the control loop. In addition, the motion of the appendage has the potential to alter the center of gravity and impact the effectiveness matrix. This change may result in certain subsets of actuators being favored over others, particularly in the case of RCT where the effort is highly dependent on the geometric definition of the torque arm. The appendage effort, clearly, could enter in the same effectiveness matrix definition as any other actuator, with, if required, a weighting factor for prioritization. Although the resulting model would be highly complex, requiring a multi-body model with time variable inertia parameters and a time-varying effectiveness, it could represent a significant advancement in fine pointing manoeuvres and close proximity operations that are crucial regarding in-orbit servicing.
To sum up, the proposed inquiry establishes the initial phase for the aforementioned examination and constructs a useful instrument to effectively handle and enhance the control allocation in the closed loop control system.

Bibliography

[1] Kinematics: Describing the Motions of Spacecraft. URL https://www.coursera.org/learn/spacecraft-dynamics-kinematics#syllabus.

[2] F. Bernelli. Slides on attitude definition. *Spacecraft Attitude Dynamics Master Course*, 2020.

[3] A. Björck. Numerical methods for least squares problems. *Choice Reviews Online*, 34(03):34–1602, 11 1996. doi: 10.5860/choice.34-1602.

[4] M. Bodson. Evaluation of Optimization Methods for Control Allocation. *Journal of Guidance Control and Dynamics*, 25(4):703–711, 7 2002. doi: 10.2514/2.4937.

[5] K. A. Bordignon. Constrained control allocation for systems with redundant control effectors. *Dissertation submitted to the Faculty of the Virginia Polytechnic Institute and State University*, 12 1996.

[6] K. A. Bordingnon and W. Durham. Closed-form solutions to constrained control allocation problem. *Journal of Guidance Control and Dynamics*, 18(5):1000–1007, 9 1995. doi: 10.2514/3.21497.

[7] P. Di Lizia. Notes on Covariance Matrix. *Spacecraft Guidance and Navigation Master Course*, 2021.

[8] C. Duan, S. Zhang, Y. Zhao, and X. Kong. Robust Control Allocation among Over-actuated Spacecraft Thrusters under Ellipsoidal Uncertainty. *Abstract and Applied Analysis*, 6 2014. doi: 10.1155/2014/950127. URL http://downloads.hindawi.com/journals/aaa/2014/950127.pdf.

[9] W. Durham. Constrained Control Allocation. *Journal of Guidance, Control, and Dynamics,*, Vol. 16, No. 4:717–725, 8 1993. doi: 10.2514/3.21072.

[10] W. Durham. Constrained control allocation - Three-moment problem. *Journal of Guidance Control and Dynamics*, 17(2):330–336, 3 1994. doi: 10.2514/3.21201.

[11] W. Durham. Attainable moments for the constrained control allocation problem.

Journal of Guidance Control and Dynamics, 17(6):1371–1373, 11 1994. doi: 10.2514/3.21360.

[12] W. Durham. Computationally Efficient Control Allocation. *Journal of Guidance Control and Dynamics*, 24(3):519–524, 5 2001. doi: 10.2514/2.4741.

[13] P. Efimova and D. Shymanchuk. Dynamic model of space robot manipulator. *Applied mathematical sciences*, 1 2015. doi: 10.12988/ams.2015.56429.

[14] D. Enns. Control allocation approaches. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 98–108, 1998. doi: 10.2514/6.1998-4109.

[15] M. Ghobadi, M. Shafaee, and M. J. Nadoushan. Reliability Approach to Optimal Thruster Configuration Design for Spacecraft Attitude Control Subsystem. *Journal of Aerospace Technology and Management*, 6 2020. doi: 10.5028/jatm.v12.1112. URL `https://doi.org/10.5028/jatm.v12.1112`.

[16] C. Haitao, P. Huang, Y. Zhang, Z. Meng, and Z. Liu. Distributed Control Allocation for Spacecraft Attitude Takeover Control via Cellular Space Robot. *Journal of Guidance Control and Dynamics*, 41(11):2499–2506, 9 2018. doi: 10.2514/1.g003626.

[17] P. Henri and J. Scruggs. Constrained Optimization Using Lagrange Multipliers. *Department of Civil and Environmental Engineering, Duke University*, 2020. URL `https://people.duke.edu/~hpgavin/cee201/LagrangeMultipliers.pdf`.

[18] O. Härkegård. Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation. *Conference on Decision and Control*, 12 2002. doi: 10.1109/cdc.2002.1184694.

[19] O. Härkegård. Backstepping and control allocation with applications to flight control. *Digitala Vetenskapliga Arkivet*, 1 2003.

[20] O. Härkegård. Dynamic Control Allocation Using Constrained Quadratic Programming. *Journal of Guidance Control and Dynamics*, 27(6):1028–1034, 11 2004. doi: 10.2514/1.11607.

[21] O. Härkegård and S. T. Glad. Resolving actuator redundancy—optimal control vs. control allocation. *Automatica*, 41(1):137–144, 1 2005. doi: 10.1016/j.automatica.2004.09.007.

[22] J.-H. Jin. Modified Pseudoinverse Redistribution Methods for Redundant Controls Allocation. *Journal of Guidance Control and Dynamics*, 28(5):1076–1079, 9 2005. doi: 10.2514/1.14992.

[23] J.-H. Jin, B.-K. Park, Y.-W. Park, and M.-J. Tahk. Attitude control of a satellite with redundant thrusters. *Aerospace Science and Technology*, 10(7):644–651, 10 2006. doi: 10.1016/j.ast.2006.04.005.

[24] X. Lang and A. H. J. De Ruiter. A control allocation scheme for spacecraft attitude stabilization based on distributed average consensus. *Aerospace Science and Technology*, 106:106173, 11 2020. doi: 10.1016/j.ast.2020.106173.

[25] X. Lang and A. H. J. De Ruiter. Distributed optimal control allocation for 6-dof spacecraft with redundant thrusters. *Aerospace Science and Technology*, 118:106971, 11 2021. doi: 10.1016/j.ast.2021.106971.

[26] W. Liu, Y. Geng, B. Wu, and J. D. Biggs. Distributed Constrained Control Allocation for Cellularized Spacecraft Attitude Control System. *Journal of Guidance Control and Dynamics*, 45(2):385–393, 2 2022. doi: 10.2514/1.g006266.

[27] D. G. Luenberger and Y. Ye. Linear and nonlinear programming. *Mathematics and Computers in Simulation*, 28(1):78, 2 1986. doi: 10.1016/0378-4754(86)90092-3.

[28] G. Ma and X. Tan. Dynamic Near-Optimal Control Allocation for Spacecraft Attitude Control Using a Hybrid Configuration of Actuators. *IEEE Transactions on Aerospace and Electronic Systems*, 56(2):1430–1443, 4 2020. doi: 10.1109/taes.2019.2934697.

[29] G. Ma, B. Li, and Y. Zhang. Nonlinear Proportional-Derivative Control Incorporating Closed-Loop Control Allocation for Spacecraft. *Journal of Guidance Control and Dynamics*, 37(3):799–812, 4 2014. doi: 10.2514/1.61815.

[30] J. A. M. Petersen and M. Bodson. Constrained quadratic programming techniques for control allocation. *IEEE Transactions on Control Systems and Technology*, 14 (1):91–98, 1 2006. doi: 10.1109/tcst.2005.860516.

[31] J. W. Petersen and M. Bodson. Interior-Point Algorithms for Control Allocation. *Journal of Guidance Control and Dynamics*, 28(3):471–480, 5 2005. doi: 10.2514/1.5937.

[32] E. Sadien, M. Carton, C. Grimault, L. E. Romana, C. Roos, A. Birouche, and M. Basset. A detailed comparison of control allocation techniques on a realistic on-ground aircraft benchmark. *HAL (Le Centre pour la Communication Scientifique Directe)*, 7 2019. doi: 10.23919/acc.2019.8814718. URL `https://hal.archives-ouvertes.fr/hal-02449260/document`.

[33] H. Schaub. 15: Modified Rodrigues Parameters Definitions, . URL

https://www.coursera.org/lecture/spacecraft-dynamics-kinematics/15-modified-rodrigues-parameters-definitions-gSNEv.

[34] H. Schaub. 18: MRP to DCM Relation, . URL https://www.coursera.org/lecture/spacecraft-dynamics-kinematics/18-mrp-to-dcm-relation-Lo0Q6.

[35] P. Servidia. Control allocation for gimballed/fixed thrusters. *Acta Astronautica*, 66 (3-4):587–594, 2 2010. doi: 10.1016/j.actaastro.2009.07.023.

[36] P. Servidia and R. Pena. Spacecraft thruster control allocation problems. *IEEE Transactions on Automatic Control*, 50(2):245–249, 2 2005. doi: 10.1109/tac.2004.841923.

[37] Q. Shen, S. Zhu, and E. K. Poh. Robust Control Allocation for Spacecraft Attitude Tracking Under Actuator Faults. *IEEE Transactions on Control Systems and Technology*, 25(3):1068–1075, 5 2017. doi: 10.1109/tcst.2016.2574763.

[38] O. Sordalen. Optimal thrust allocation for marine vessels. *Control Engineering Practice*, 5(9):1223–1231, 9 1997. doi: 10.1016/s0967-0661(97)84361-4.

[39] S. Tang, S. Zhang, and Y. Zhang. A Modified Direct Allocation Algorithm with Application to Redundant Actuators. *Chinese Journal of Aeronautics*, 24(3):299–308, 6 2011. doi: 10.1016/s1000-9361(11)60035-6. URL https://doi.org/10.1016/s1000-9361(11)60035-6.

[40] M. Wang and Y. Xie. Design of the optimal thruster combinations table for the real time control allocation of spacecraft thrusters. *Conference on Decision and Control*, 12 2009. doi: 10.1109/cdc.2009.5400664.

A | Appendix A

In accordance with the comparison of the allocation strategies presented in Chapter 2 it is important to confirm that the computational efficiency of the optimal algorithms, specifically `linprog` and `fmincon`, remains inferior to the other algorithms even when executed solely on `Matlab`. This condition is required to be verified, since `Simulink` is not able to manage undefined variables, that are a necessary input into these algorithms. To perform these kind of operations in `Simulink`, the user has to force the software, with the command `coder.extrinsic`, to transpose the segment of the code corresponding to the algorithm solving onto `Matlab`. This operation is especially heavy in terms of computational cost. Hence, it is necessary to check whether, computed solely on `Matlab`, the optimal algorithm allocation is not able to find a solution in a faster way with respect the other. If this condition is again met, it corroborates the discussion of the computational efficiency carried out during the comparison. To do this, the allocation procedures are tested using a random initial required torque as input, for a total of 1000 samples, and the computational times are recorded. In Figure A.1, the different computational time spans are presented, for the different strategies, in a semi-logarithmic graph. It is evident that despite having superior computational speed compared to the `Simulink` application, the two optimal algorithms still exhibit slower times in comparison to the other methods. Thus, the conclusions drawn from the comparison remain valid.

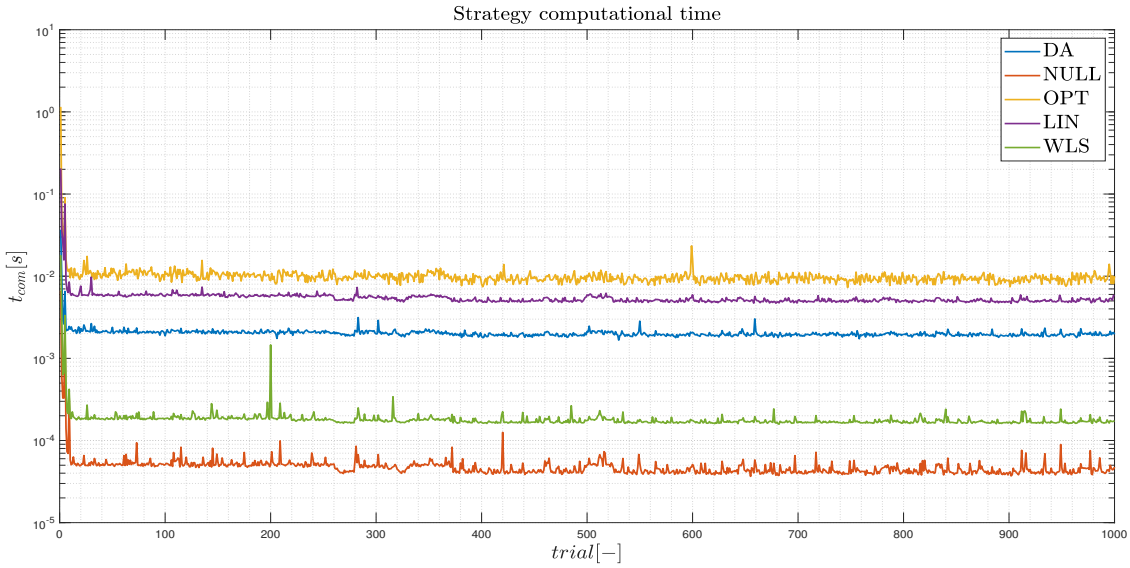

Figure A.1: Strategies computational time comparison

To conclude, the graph clearly displays the same trend as noted in the comparison: the optimal algorithms have the longest computation times, followed by the pseudo direct allocation method which necessitates the inspection of all facets, while the weighted least square approach performs better. Ultimately, the null space algorithm is the fastest due to its minimal number of operations required to execute.

List of Figures

List of Tables

List of Symbols

Variable	Description	SI unit
$\boldsymbol{\beta}$	Euler parameters	-
γ	WLS weighting factor	-
$\boldsymbol{\epsilon}$	Pointing error vector	rad
θ	Orbit true anomaly	rad
ρ	Atmosphere density	kg/m^3
ρ_s	Reflected radiation coefficient	-
ρ_d	Diffused radiation coefficient	-
$\boldsymbol{\lambda}$	Lagrangian multipliers	Variable
μ	Earth constant	km^3/s^2
$\boldsymbol{\mu}$	Control fraction vector	-
$\boldsymbol{\sigma}$	Rodrigues parameters vector	rad
ϕ	Euler angle	rad
Φ	Attainable moment space	-
$\boldsymbol{\omega}$	Spacecraft angular velocity vector	rad/s
$\dot{\boldsymbol{\omega}}$	Spacecraft angular acceleration vector	rad/s^2
Ω	Feasible control space	-
a	Orbit semi-major axis	km
A_{ref}	Direct cosine matrix	-
A_{net}	Net surface exposed to air stream	m^2
B	Effectiveness	Variable
B_{RCT}	Effectiveness matrix for RCT	Variable
B_{RW}	Effectiveness matrix for RW	Variable
$\boldsymbol{d}$	WLS residual	Variable
C_d	Spacecraft drag coefficient	-
e	Orbit eccentricity	-
$\boldsymbol{e}$	Euler axis	-

Variable	Description	SI unit
$\boldsymbol{F}$	Force due to drag	N
$\boldsymbol{F_{ext}}$	External force vector	N
$\boldsymbol{h}$	Orbit angular momentum vector	m^2/s
g_0	Earth acceleration	m/s^2
H	Spacecraft altitude	km
I	Spacecraft inertia matrix	kgm^2
I_n	Identity matrix	-
I_{sp}	Specific impulse	s
$\boldsymbol{K_p}$	Proportional gain	-
$\boldsymbol{K_d}$	Derivative gain	-
$\boldsymbol{K_i}$	Integral gain	-
MA	Module A	-
MB	Module B	-
$\boldsymbol{m}$	Required torque	Nm
$\boldsymbol{m_{residual}}$	Residual torque	Nm
$\boldsymbol{m_{RCT}}$	Required torque for RCT	Nm
$\boldsymbol{m_{RW}}$	Required torque for RW	Nm
$\hat{\boldsymbol{n}}$	Nadir direction	-
$\hat{\boldsymbol{n}}_{surf}$	Normal direction of a surface	-
N	Null Space	-
p	Cost function	Variable
$\boldsymbol{p}$	WLS perturbation	Variable
P	Generalized inverse	-
P_w	Average pressure due to radiation	Pa
$\boldsymbol{r_{CoG}}$	Centre of gravity position in BRF	m
$\boldsymbol{r_{CoP}}$	Centre of pressure position in BRF	m
$\boldsymbol{r}$	Nadir vector	m
$\boldsymbol{R}$	Spacecraft position vector in IRF	m
$\boldsymbol{s}$	Sun position in body frame	m
$\boldsymbol{t}$	Perpendicular vector for DA	-
T	Thrust	N
$\boldsymbol{T_{drag}}$	Drag torque vector	Nm
$\boldsymbol{T_{ext}}$	External torque vector	Nm

Variable	Description	SI unit
$\boldsymbol{T_{GGT}}$	Gravity gradient torque vector	Nm
$\boldsymbol{u}$	Control effort vector	Variable
$\boldsymbol{u_f}$	Feasible control effort vector	Variable
$\boldsymbol{u_{lim}}$	Bounded control effort vector	Variable
$\boldsymbol{u_{max}}$	Control effort vector upper bound	Variable
$\boldsymbol{u_{min}}$	Control effort vector lower bound	Variable
$\boldsymbol{u_s}$	Saturated control effort vector	Variable
$\boldsymbol{u_{RCT}}$	Control vector for RCT	-
$\boldsymbol{u_{RW}}$	Control vector for RW	N
$\boldsymbol{v}$	Spacecraft velocity in BRF	m/s
$\boldsymbol{V}$	Spacecraft velocity in IRF	m/s
$\boldsymbol{w}$	Weighting vector	-
W	Weighting matrix	-
WS	Working Set	-
$\boldsymbol{x}$	System position vector	m
$\dot{\boldsymbol{x}}$	System velocity vector	m/s
$\ddot{\boldsymbol{x}}$	System acceleration vector	m/s^2

Acknowledgements

I express my gratitude to everyone who supported and guided me through my academic journey. Without your enduring patience and support through my highs and lows, this work would not have been possible. I would like to extend special thanks to my family, friends, and my advisor P. Di Lizia, whose guidance was instrumental in the development of this thesis, G. Scirè, who not only answered my questions and addressed my doubts on a weekly basis for several months, but introduced me in the company and to a lot of intelligent, caring people.