POLITECNICO DI TORINO

Master's degree in Aerospace Engineering

Master's degree thesis

Development of guidance algorithms for attitude maneuvers of small satellites

Supervisors
Elisa Capello
Mauro Mancini

Candidate
Antonio Romano

A.A. 2022-2023

Abstract

One of the most essential tasks of a satellite is to point an instrument along a desired direction in order to achieve mission requirements. In many cases, this operation is complicated due to the presence of sensitive on-board instruments, which cannot be pointed along certain directions, such as the Sun or other bright objects. This problem can be analyzed as a reorientation problem in the presence of attitude constraints. In literature there are not many research papers that address the constraints problem, and its solution has great importance for practical scientific missions. In this thesis work, a guidance algorithm based on the Artificial Potential Field method is analyzed and implemented in MATLAB/Simulink environment, which allows us to describe the desired pointing and the restricted areas through potential fields. A hybrid attractive potential field centered at the desired pointing has been implemented: parabolic close to the objective and conic far away from it, in order to prevent the reaction wheels from saturating and at the same time have a smoother response; to deal with the restricted areas, a hyperbolic repulsive potential field has been implemented, with extra care in designing the gain so that the reference angular velocity remained compatible with the constraint given by the actuation system. The advantage of this method is its analytical approach which allows us not to have to modify either the controller hardware or software, and therefore suitable for on-board calculation and planning. Satellite kinematics and dynamics are described using unit quaternions to avoid the singularity issue which Euler angles are affected by. By combining the guidance with a sliding mode control algorithm, we obtain an attitude control law that guarantees good performance and robustness, allowing flexibility in the presence of obstacles and orbit perturbations.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

GNC (Guidance, Navigation and Control) is one of the main and most important subsystems of a spacecraft. It is responsible for providing it with values of position, angular rates and attitude to track as reference (*guidance*), estimate the true variables from the measurements through state observers or filters (*navigation*), and generate a control action through comparison between current and desired variable (*control*). In figure 1.1 we have the general structure of a GNC subsystem. It can be seen that a set of sensors measure the dynamical variables of the system and they get estimated (cleaned of the process noise) in the navigation block. The guidance block generates, based on the navigation estimate, a reference value for the satellite to track. The information about the "true" and reference values generate an error state for the control, necessary to calculate a control input action, which directly acts on the actuators of the satellite. The feedback cycle closes with the actuators' external action on the satellite dynamics model (indicated here with "plant") causing the time variation of the state variables. Such a subsystem allowed us to reach incredible milestones. Let's think for example of the rendezvous maneuvers, which allowed humanity to land on the Moon, build and operate a manned orbiting space station, even reach asteroids and comets for samples collection far away in the solar system. Specifically in the latter case, human action is limited, and everything relies on the spacecraft ability to orient itself through sensors and precisely calculate the relative position with the target, the attitude to communicate with the deep space network or to track the sun for electricity generation. Terrestrial applications of autonomous systems have also been developed in recent years. Examples are easily found in robotics or aerospace research field, such as Unmanned Autonomous Vehicles (UAV), guided by an onboard autopilot and a set of sensors and able to follow pre-established paths avoiding obstacles. Even for ordinary in-orbit missions, spacecrafts are equipped with a particular payload, which, depending on the type of the mission, could be a high gain antenna for communication and data transfer or an infrared sensor for remote sensing or even a space telescope. The space environment is very insidious, and a certain number of factors and objects can have adverse impacts based on the payload used: a space telescope or a star sensor need to be pointed away from very bright objects such as the Sun, and a high gain antenna, at the same time, needs to be pointed towards the Earth to ensure coverage and thus data transfer. Let's take the

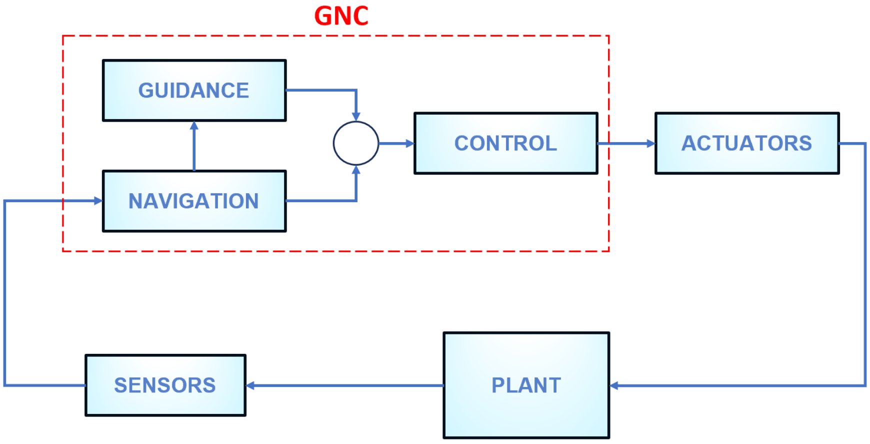

Figure 1.1. General GNC structure.

example of the *James Webb Space Telescope*, whose attitude control subsystem is responsible for high gain antenna pointing, slew maneuvers and ensuring that the observatory remains within the sun avoidance constraints. This clearly introduces a new concept which is the *attitude forbidden constraint*. This introduces a major challenge, both in a practical and theoretical perspective when it comes to the guidance and control subsystems design.

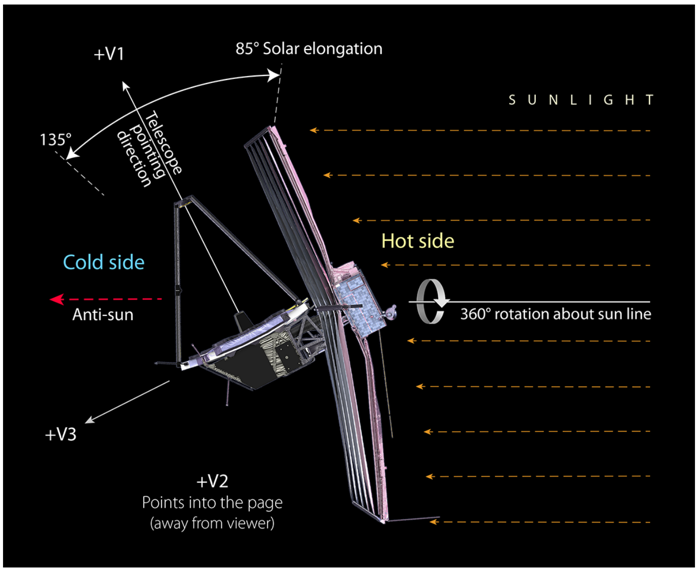

Figure 1.2. JWST Observatory coordinates. The telescope pointing direction lies within the solar elongation, which has to change over time as it orbits the sun.[1]

This thesis work will analyze and simulate a guidance algorithm which, in addition to fulfill the basic attitude orientation control purposes, takes the attitude forbidden constraint into account. This is to all intents a motion planning problem: in the presence of obstacles, it is necessary to plan orientation maneuvers that allow the satellite to fulfill the assigned task without colliding with them. Since the satellite is of course set in space, there is the necessity for the algorithm used to be suited to *on-line planning*, where the geometry of the obstacles can change and is unknown in advance. In fact, the satellite is requested to plan its motion on-line, using information of the workspace gathered on the basis of sensor measurements. This can be obtained with a guidance algorithm based on the *Artificial Potential Field method* (APF), which is also used in robotics to model the position dynamics of the robot taking the obstacles into account through the path. This is achieved through a state-space formulation and the definition of a *potential field*, function of the state-space variable. The variable $\boldsymbol{q}$ that can represent the position or, in this case, the instrument attitude moves in the state space under the influence of a potential field U, obtained as a superposition of an *attractive potential* to the goal and a *repulsive potential* from the obstacle location. At each time step and configuration $\boldsymbol{q}$, the artificial force generated by the potential is defined as the negative gradient $-\nabla U(\boldsymbol{q})$ of the potential, which indicates, as is known, the direction of maximum decrease of the potential function.[17]

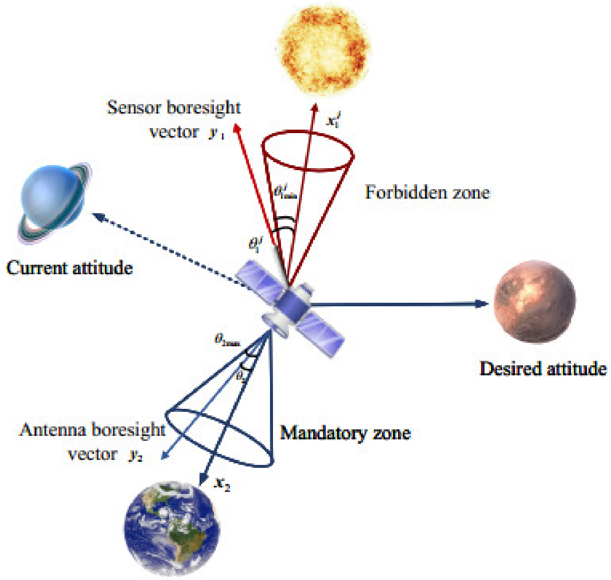

Figure 1.3. Schematization of an orientation maneuver in the presence of attitude forbidden and mandatory constraints.[7]

The challenge that arises from such a problem is the total loss of intuition when the variable of such a state space is not a position (the movement along the $-\nabla U$ direction would be much more intuitive) but a rotation. Indeed, attitude dynamics is described by angles or quaternions, both of which describe a sequence of rotations (respectively 3 and 1). Added to this is the problem, in the control logic, of tracking the separation angle between the instrument pointing and the inertial position of an obstacle, and translate this information into a mathematical entity.

The problem of attitude forbidden constraint is generally not explicitly taken in consideration in classical literature and has been addressed in only a few relatively recent papers. The few works focus on the APF method applied to different representations of attitude, i.e. reduced-attitude [12, 6, 9], modified Rodrigues parameters [4] or unit quaternions. Different mathematical formulations of the potential functions have been proposed. In [12] a conical attractive field and an hyberbolic tangent for the repulsive field have been used whereas in [3] an exponential repulsive field is adopted as a compromise between accuracy and computational complexity for their simplicity. In [19, 11] convex logarithmic barrier potentials are adopted, as well as a logarithmic function to address the *unwinding* problem. The unwinding problem is also taken into account in the formulation of a more classic form of potentials (quadratic and hyperbolic) in [8]. The unwinding is the natural redundancy associated with the quaternion representation of the goal attitude, it's to say the non-uniqueness of the goal quaternion: given an axis and a certain angle of rotation θ to align the body frame to the inertial frame, such goal can be achieved with the same axis, but with the rotation of an angle $2\pi - \theta$. They're equivalent, but in a real situation we want our satellite to perform a *smart rotation*, which is the one who requires the least time and resources in terms of fuel. In [14] the APF is applied with a quadratic attractive potential and a hyperbolic repulsive potential on a flexible spacecraft and a velocity-free control is developed, i.e. without using explicit velocity feedback. The approach used in [10] is to blend the potential fields in one single potential that keeps on tracking the angle separation between the boresight vector and constrained vector and then construct the control action through its gradient.

All the cited papers rely on one thing in common: Lyapunov theorem of stability. Both attractive and repulsive potential fields are grouped in a single potential function $V(\boldsymbol{x})$ (known as a *Lyapunov candidate function*) and, using Lyapunov theorem of stability, by showing the convexity of the function and choosing a control input that makes its time derivative negative, the asymptotic stability is proven, and a mathematical form for the control input $u(\boldsymbol{x})$ is obtained. Our approach will be slighlty different. The potential fields are kept separate and the total potential is given by their sum. Despite this, asymptotic stability is still achieved by setting the gradient of the potential directly as the reference velocity for the control. It will be shown better in the APF chapter that this is one of the possible planning techniques. Once we have a reference goal and a reference velocity, a controller based on the sliding mode control is used. In order to overcome the chattering effect caused by the robust gains being much larger than the external disturbances, an adaptive sliding mode control law is used to adjust the gains online with respect to the external disturbances.

1.1 Overview of the thesis

The thesis will be structured from now on in three more chapters. Chapter 2 will dive into the mathematical modeling of the satellite, with a description of different methods of representing the attitude including listing all the pros and cons of each one. For computational purposes, unit quaternions have been chosen to represent the attitude of the satellite and a brief definition and main properties will be listed. In order to simulate orientation maneuvers, the simulation need to integrate equations of motion. The equations of motion describing the rotational dynamics of rigid bodies are Euler's equations. They will be derived starting from the basic definition of angular momentum and it will be shown how an external torque can give rise to a variation of the angular velocity of the satellite. The connection between the angular velocity and the time evolution of quaternions is given by the kinematic equations, since, as the word says (*kinematics*), it's all about the description of the body motion without wondering about the causes. The integration of the kinematic equations gives rise to the final variation of the quaternions, which tells us at every instant of time the relative orientation between the satellite, described by its body frame of reference, and the inertial frame of reference. The main task of the simulation is of course to model the dynamical system of the satellite and generate an external action given by a certain control input, it's to say the whole plant is equipped with a feedback controller. A model for the actuators is needed as well. The reference for this thesis work will be the *Demeter* satellite, which is equipped with a cluster of 4 reaction wheels with magnetorquers for desaturation. Starting from the principle of operation of a reaction wheel, a mathematical model will be described, based on the filtering of the control input through a low pass filter and a saturation command which takes into account the maximum value of torque that the wheel can generate. After that, all different cluster configurations will be listed, including a general mathematical conversion procedure of torque between the satellite frame and the wheels cluster's one and back. In chapter 3 the Artificial Potential Field (APF) will be introduced. A guidance algorithm known and applied in many fields of robotics and in position dynamics problems in general, it will be applied to the rotational dynamics instead, generating the correct reference values for the controller to drive the satellite orientation towards the correct attitude while avoiding any pointing obstacles along the trajectory. The APF method comes from the simple and intuitive physical concept of a scalar field that gives rise to a force field related to its gradient. Many example in classical mechanics: a charged particle immersed in an electrostatic field will experience a force pointing in or out of the source of the field; same applies to the gravitational potential: a mass immersed in it will experience a force pointing inward the source. Following the latter general meaning, an attractive potential will be defined at the quaternion representing the goal to reach and a repulsive potential at the obstacle, which has to take into account the separation between the instrument boresight vector and the obstacle's inertial vector. A first mathematical formulation of the APF method will be presented, discussing the different mathematical functions for the scalar potential fields and their pros and cons. For the repulsive field, the concept of *attitude constraint* will be introduced and a hyperbolic repulsive field can be derived from it. A total field can be derived by superposing the attractive and repulsive potentials and its gradient will provide the reference value of the angular velocity. Subsequently, the sliding mode control algorithm will be introduced briefly and characterized. Finally, it will be applied to the current case of attitude control, with a slight modification by the introduction of an adaptive control law in the parameter of the sliding variable. In chapter 4 simulation results will be presented. After listing all the satellite parameters, initial condition, guidance and control parameters values and simulation parameters, two simulation will be presented, i.e. the unwinding case and non-unwinding case, both through time evolution of the quaternions, angular velocity and control torque of the reaction wheels. Moreover, the value of the quadratic constraint for different obstacles will be plotted in order to show the effective pointing avoidance performed by the satellite dynamics.

Chapter 2

Mathematical model

In this chapter we will go through the modeling of the spacecraft dynamics. How can attitude be mathematically described? The first section will present two main ways used in literature: Euler's angles and quaternions. Pros and cons will be discussed and why the latter is a better choice for computational purposes. In the second section the spacecraft kinematics will be discussed, i.e. how the spacecraft rotates and evolves over time as a function of angular velocity $\boldsymbol{\omega}$. Next, Euler's equations will be introduced, which describe the rotational dynamics of the spacecraft, i.e. the evolution over time of $\boldsymbol{\omega}$ given an external torque $\boldsymbol{M}$. With the final description of the mathematical model of the satellite actuation system, based on reaction wheels, all the ingredients of the dynamical system are ready to be integrated in a simulation environment such as MATLAB/Simulink.

2.1 Attitude description through quaternions

2.1.1 Euler's angles and the singularity issue

In order to describe the satellite attitude, we can model it as a rigid body, so we can use the main results and laws of rigid body dynamimcs. Introducting a *body frame* $\mathscr{F}_B$ attached to the body identified by the set of unit vectors $(\hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2, \hat{\boldsymbol{e}}_3)$ and an *intertial frame* $\mathscr{F}_I$ identified by the set $(\hat{\boldsymbol{E}}_1, \hat{\boldsymbol{E}}_2, \hat{\boldsymbol{E}}_3)$, it is possible to describe the satellite attitude by analyzing the position of $\mathscr{F}_I$ with respect to $\mathscr{F}_B$. By considering the components of $\hat{\boldsymbol{e}}_i$ in the inertial frame, it is possible to obtain a linear combination between the basis vector sets. Substituting this relation, any vector quantity $\boldsymbol{v}$ can be expressed in both frames through a *transformation matrix* $\boldsymbol{L}_{IB}$ as follows:

$$\boldsymbol{v}_I = \boldsymbol{L}_{IB}\boldsymbol{v}_B \tag{2.1}$$

or, in matrix form:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} e_{1,1} & e_{1,2} & e_{1,3} \\ e_{2,1} & e_{2,2} & e_{2,3} \\ e_{3,1} & e_{3,2} & e_{3,3} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.2}$$

The transformation matrix $\boldsymbol{L}_{IB}$ has the property of being orthogonal. This means that it is fairly easy to write the inverse coordinate transformation:

$$\boldsymbol{v}_B = \boldsymbol{L}_{BI}\boldsymbol{v}_I = (\boldsymbol{L}_{IB})^{-1}\boldsymbol{v}_I = (\boldsymbol{L}_{IB})^T\boldsymbol{v}_I \tag{2.3}$$

Using the transformation matrix it is therefore possible to describe physical quantities in the inertial frame of reference, and 9 parameters are needed (which are the dot product of the basis vectors). However, it must be taken into account that the basis vectors are not free to vary and they have to satisfy the following constraints:

$$\hat{\boldsymbol{e}}_i \cdot \hat{\boldsymbol{e}}_j = \delta_{ij} = \begin{cases} 0 & \text{if} \quad i \neq j \\ 1 & \text{if} \quad i = j \end{cases} \tag{2.4}$$

therefore, only 3 independent parameters are sufficient to describe the attitude of the body frame with respect to the inertial frame.
One of the set of three parameters most used for this purpose are Euler's angles, which consist of an ordered sequence of elementary rotations in order to reach the body frame $\mathscr{F}_B$ starting from the inertial frame $\mathscr{F}_I$.
A sequence widely used in spaceflight dynamics is the 3-2-1, also known as *Bryant's angles*, which brings the orbital frame to the body frame. The orbital frame $\mathscr{F}_O(x_O,y_O,z_O)$, of vectors $\hat{\boldsymbol{i}}_O$, $\hat{\boldsymbol{j}}_O$, $\hat{\boldsymbol{k}}_O$, is the reference frame centered in the satellite center of mass and whose axes are directed such that:

- $\hat{\boldsymbol{k}}_O$ points towards the Earth's center of mass;
- $\hat{\boldsymbol{i}}_O$ is perpendicular to $\hat{\boldsymbol{k}}_O$, points towards the satellite velocity vector and belongs to the orbit plane;
- $\hat{\boldsymbol{j}}_O$ is such that it forms a right-handed coordinate system with the other two unit vectors.

Following this sequence, we can describe the sequence of rotation 3-2-1 as follows[13]:

1. $\mathscr{F}_O$ is rotated around the $\hat{\boldsymbol{k}}_O$ direction by an angle ψ. Respectively, the y_O axis moves in the y' direction and x_O axis in the x' direction. A new set $\mathscr{F}'_O$ is obtained.
2. A second rotation is performed around the y' axis by an angle θ. Respectively, the x_O axis moves in the $x'' \equiv x$ (x axis of body frame) and z_O in the z'' direction, obtaining the new set $\mathscr{F}''_O$.
3. Last rotation is performed around the x axis by an angle ϕ until the orbital frame $\mathscr{F}_O$ coincides with the body frame $\mathscr{F}_B$.

Therefore, taken a generic vector quantity $\boldsymbol{v}$, we obtain:

$$[\boldsymbol{v}]_{\mathscr{F}_B} = \boldsymbol{L}\,[\boldsymbol{v}]_{\mathscr{F}_O} \tag{2.5}$$

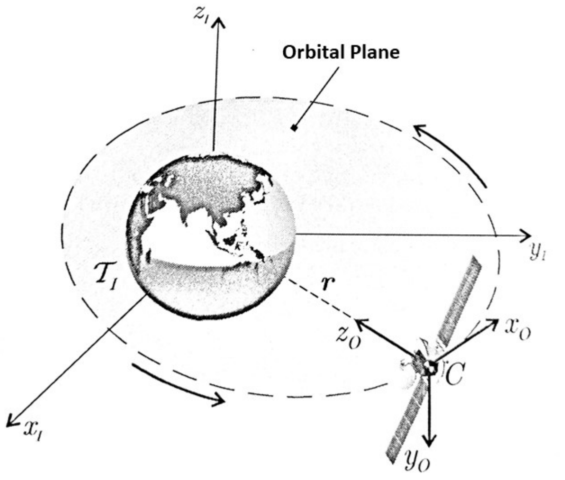

Figure 2.1. Orbital reference frame.[13]

where $\boldsymbol{L} = \boldsymbol{L}_x(\phi)\,\boldsymbol{L}_y(\theta)\,\boldsymbol{L}_z(\psi)$, or in a more explicit form:

$$\boldsymbol{L} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{2.6}$$

The transformation matrix in terms of the Bryant's angles $\boldsymbol{L}(\phi,\theta,\psi)$ is affected by a *singularity*, which can be seen by the fact that, for $\theta = \pm\pi/2$, the matrix does not depend on ϕ and ψ separately, but only on their difference $\phi - \psi$. The singularity causes more appreciable issues later with the kinematics equations.

2.1.2 Quaternion definition

Generally speaking, a *quaternion* is a mathematical entity of the following form:

$$\boldsymbol{Q} = a + b\,\boldsymbol{i} + c\,\boldsymbol{j} + d\,\boldsymbol{k} \tag{2.7}$$

where a, b, c, d are real numbers and $\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{k}$ are unit vectors. As Hamilton pointed out, a quaternion can be described as consisting of a *scalar part*, a, and a *vector part*, $b\,\boldsymbol{i}+c\,\boldsymbol{j}+d\,\boldsymbol{k}$. We can use the following notation:

$$\boldsymbol{Q} = q_0 + k\boldsymbol{q} \qquad \textit{or} \qquad \boldsymbol{Q} = (q_0, \boldsymbol{q}^T)^T \tag{2.8}$$

We can then observe that the set of quaternions form a 4-dimensional vector space with $\{1, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$ as basis; this allows us to define a component-wise addition:

$$\begin{aligned}&(a_1 + b_1\,\boldsymbol{i} + c_1\,\boldsymbol{j} + d_1\,\boldsymbol{k}) + (a_2 + b_2\,\boldsymbol{i} + c_2\,\boldsymbol{j} + d_2\,\boldsymbol{k})\\ &= (a_1 + a_2) + (b_1 + b_2)\,\boldsymbol{i} + (c_1 + c_2)\,\boldsymbol{j} + (d_1 + d_2)\,\boldsymbol{k}\end{aligned} \tag{2.9}$$

and a component-wise scalar multiplication:

$$\lambda(a + b\,\boldsymbol{i} + c\,\boldsymbol{j} + d\,\boldsymbol{k}) = \lambda a + (\lambda b)\,\boldsymbol{i} + (\lambda c)\,\boldsymbol{j} + (\lambda d)\,\boldsymbol{k} \tag{2.10}$$

How can this new formulation help describe the satellite attitude?

We can recall *Euler's eigenaxis rotation theorem*, which states that it is possible to rotate an inertial frame $\mathscr{F}_I$ onto a body frame $\mathscr{F}_B$ with a simple rotation by an angle α around an axis $\hat{\boldsymbol{a}}$ that is fixed in both frames called *eigenaxis.* This can be demonstrated recalling that the eigenvalues of any orthogonal matrix have unit modulus and at least one is exactly $\lambda = 1$. This means that, by the definition of eigenvector:

$$\boldsymbol{L}\hat{\boldsymbol{a}} = 1 \cdot \hat{\boldsymbol{a}} \tag{2.11}$$

there exist a direction $\hat{\boldsymbol{a}}$ which does not change under the action of the transformation matrix $\boldsymbol{L}$. This also means that the components of the vector $\hat{\boldsymbol{a}}$ are exactly the same in both frames:

$$\hat{\boldsymbol{a}} = a_1\hat{\boldsymbol{e}}_1 + a_2\hat{\boldsymbol{e}}_2 + a_3\hat{\boldsymbol{e}}_3 = a_1\hat{\boldsymbol{E}}_1 + a_2\hat{\boldsymbol{E}}_2 + a_3\hat{\boldsymbol{E}}_3 \tag{2.12}$$

Using some algebra, it is possible to express the transformation matrix $\boldsymbol{L}_{BI}$ as a function of the angle α and the Euler eigenaxis $\hat{\boldsymbol{a}}$, obtaining the following relation:

$$\boldsymbol{L}_{BI} = \cos\alpha\mathbf{1} + (1 - \cos\alpha)\boldsymbol{a}\boldsymbol{a}^T - \sin\alpha[\boldsymbol{a}]_\times \tag{2.13}$$

where $\mathbf{1}$ is the identity matrix and $[\boldsymbol{a}]_\times$ is the skew-symmetric matrix that represent the cross product:

$$[\boldsymbol{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \implies \boldsymbol{a} \times \boldsymbol{b} = [\boldsymbol{a}]_\times\boldsymbol{b} \tag{2.14}$$

Therefore, a rotation is represented by the following quaternion:

$$\begin{aligned} q_0 &= \cos(\alpha/2)\\ q_1 &= a_1\sin(\alpha/2)\\ q_2 &= a_2\sin(\alpha/2)\\ q_3 &= a_3\sin(\alpha/2) \end{aligned} \tag{2.15}$$

Equation 2.13 can be expressed then as:

$$\boldsymbol{L}_{BI} = (q_0^2 - \boldsymbol{q}\cdot\boldsymbol{q})\mathbf{1} + 2\boldsymbol{q}\boldsymbol{q}^T - 2q_0[\boldsymbol{q}]_\times \tag{2.16}$$

On the contrary, it is possible to calculate the corresponding quaternion from the elements c_{ij} of transformation matrix $\boldsymbol{L}_{BI}$:

$$\boldsymbol{Q} = \frac{1}{2}\begin{pmatrix} \sqrt{1 + c_{11} + c_{22} + c_{33}} \\ sign(c_{32} - c_{23})\sqrt{1 + c_{11} - c_{22} - c_{33}} \\ sign(c_{13} - c_{31})\sqrt{1 - c_{11} + c_{22} - c_{33}} \\ sign(c_{21} - c_{12})\sqrt{1 - c_{11} - c_{22} + c_{33}} \end{pmatrix} \tag{2.17}$$

2.1.3 Quaternion properties

Some useful quaternion properties are listed here.

Conjugate Let $\boldsymbol{Q} = (q_0, \boldsymbol{q}^T)^T$. The *conjugate* of $\boldsymbol{Q}$ is defined as:

$$\boldsymbol{Q}^* = (q_0, -\boldsymbol{q}^T)^T \tag{2.18}$$

From this we can observe that $(\boldsymbol{Q}^*)^* = \boldsymbol{Q}$. It represent a rotation by the same angle α around the same eigenaxis but in the opposite direction.

Norm The *norm* of a quaternion is defined as follows:

$$||\boldsymbol{Q}|| = \sqrt{q_0^2 + \boldsymbol{q}^T\boldsymbol{q}} \tag{2.19}$$

A quaternion and its conjugate have the same norm. A quaternion whose norm is 1 is called *unit quaternion.*

Inverse The *inverse* of a quaternion is defined as follows:

$$\boldsymbol{Q}^{-1} = \frac{\boldsymbol{Q}^*}{||\boldsymbol{Q}||^2} \tag{2.20}$$

In the case of a unit quaternion, from the definition, the inverse coincides with the conjugate. Moreover, a unit quaternion fulfills the following constraint:

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \tag{2.21}$$

Reference quaternion Later on, we will associate the *unity quaternion* to the inertial fixed frame, which will be our reference quaternion during the development of the guidance system and simulations:

$$\boldsymbol{Q}_I = (1, 0, 0, 0)^T \tag{2.22}$$

Observation. Since the final attitude achieved by a rotation of $-\alpha$ around the axis $2\pi - \hat{\boldsymbol{a}}$ is the same of the regular rotation, the reference quaternion can also be represented by $-\boldsymbol{Q}_I = (-1, 0, 0, 0)^T$.

2.1.4 Quaternion multiplication

The quaternion multiplication satisfies the following rules introduced by Hamilton:

$$\begin{aligned} &\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1 \\ &\boldsymbol{ij} = \boldsymbol{k} = -\boldsymbol{ji} \\ &\boldsymbol{jk} = \boldsymbol{i} = -\boldsymbol{kj} \\ &\boldsymbol{ki} = \boldsymbol{j} = -\boldsymbol{ik} \end{aligned} \tag{2.23}$$

Denoting the product with the symbol $\otimes$, we can calculate the product of two quaternions $\boldsymbol{P}$ and $\boldsymbol{Q}$. In order to make everything as compact as possible, it is possible to simplify the expression by using the dot and cross product as follows:

$$\boldsymbol{P} \otimes \boldsymbol{Q} = p_0 q_0 - \boldsymbol{p} \cdot \boldsymbol{q} + p_0 \boldsymbol{q} + q_0 \boldsymbol{p} + \boldsymbol{p} \times \boldsymbol{q} \tag{2.24}$$

The multiplication thus defined has a specific geometric meaning: the resulting quaternion represents the final orientation of a frame after two subsequent rotations described precisely by the quaternions $\boldsymbol{P}$ and $\boldsymbol{Q}$. Given:

$$\begin{aligned} \boldsymbol{P} &= (\cos(\alpha/2), \hat{\boldsymbol{a}} \sin(\alpha/2)) \\ \boldsymbol{Q} &= (\cos(\beta/2), \hat{\boldsymbol{b}} \sin(\beta/2)) \end{aligned} \quad \Longrightarrow \quad \boldsymbol{R} = (r_0, \boldsymbol{r}^T)^T = \boldsymbol{P} \otimes \boldsymbol{Q} \tag{2.25}$$

where r_0 and $\boldsymbol{r}$ are given by equation (2.24). This means that the cumulative rotation occurs around the axis $\hat{\boldsymbol{c}} = \boldsymbol{r}/||\boldsymbol{r}||$ by an angle of $\gamma = 2\cos^{-1}(r_0)$.

Observation. *It is easy to prove the following properties:*

$$\boldsymbol{Q}^* \otimes \boldsymbol{Q} = \boldsymbol{Q} \otimes \boldsymbol{Q}^* \tag{2.26}$$

$$(\boldsymbol{P} \otimes \boldsymbol{Q})^* = \boldsymbol{Q}^* \otimes \boldsymbol{P}^* \tag{2.27}$$

$$\boldsymbol{Q}^{-1} \otimes \boldsymbol{Q} = \boldsymbol{Q} \otimes \boldsymbol{Q}^{-1} = 1 \tag{2.28}$$

2.1.5 Quaternion error vector

In order to design a feedback control law for our satellite, we need to have a physical quantity that represents the *error* between our current vector state and the desired vector state. Since the vector state in attitude dynamics represents geometric orientations, a way to express the difference between the two frames is needed. For this purpose, let's consider two reference frames: the desired attitude $\mathscr{F}_D$ and the current attitude $\mathscr{F}_B$, respectively described by the quaternions $\boldsymbol{P}$ and $\boldsymbol{Q}$. Since the quaternions describe the direction and amplitude of the rotation needed to bring the inertial frame to the desired or current fixed frame, we can define a *quaternion error* $\boldsymbol{E} = (\epsilon_0, \boldsymbol{\epsilon}^T)^T$ which describes the rotation needed to bring the desired attitude to the current attitude and thus a measure of the *error* between the state vectors, suitable for a control law later. With a chain of rotations, we can observe that bringing the inertial frame to the desired attitude frame and then rotate towards the current frame is equivalent to rotate the inertial towards the current from the start, which can be described in terms of quaternion multiplication as follows:

$$\boldsymbol{P} \otimes \boldsymbol{E} = \boldsymbol{Q} \tag{2.29}$$

we can therefore calculate the quaternion error vector as:

$$\boldsymbol{E} = \boldsymbol{P}^* \otimes \boldsymbol{Q} \tag{2.30}$$

with components given by equation (2.24).

2.2 Attitude kinematics

2.2.1 Evolution of Euler's angles

In subsection 2.1.1, page 8, the Euler sequence 3-2-1, also known as Bryant's angles, was introduced and used in order to calculate the transformation matrix $\boldsymbol{L}_{BI}$. The purpose of this section is to derive the differential equations that describe the evolution of the Euler's angles as a function of time and the angular velocity $\boldsymbol{\omega}$, also known as *kinematic equations*. The angular velocity in body frame is given by:

$$\boldsymbol{\omega} = \omega_1\hat{\boldsymbol{e}}_1 + \omega_2\hat{\boldsymbol{e}}_2 + \omega_3\hat{\boldsymbol{e}}_3 \tag{2.31}$$

but, using the sequence 3-2-1, we also know that the following is true:

$$\boldsymbol{\omega} = \dot{\psi}\hat{\boldsymbol{E}}_3 + \dot{\theta}\hat{\boldsymbol{e}}'_2 + \dot{\phi}\hat{\boldsymbol{e}}_1 \tag{2.32}$$

Next, let's represent the basis vectors in equation (2.32) in the body frame:

$$(\hat{\boldsymbol{E}}_3)_B = \boldsymbol{L}_{BI}\hat{\boldsymbol{E}}_3 = \begin{pmatrix} -\sin\theta \\ \sin\phi\cos\theta \\ \cos\phi\cos\theta \end{pmatrix} \tag{2.33}$$

$$(\hat{\boldsymbol{e}}'_2)_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \cos\phi \\ -\sin\phi \end{pmatrix} \tag{2.34}$$

Substituting we obtain:

$$\boldsymbol{\omega} = (\dot{\phi} - \dot{\psi}\sin\theta)\hat{\boldsymbol{e}}_1 + (\dot{\psi}\sin\phi\cos\theta + \dot{\theta}\cos\phi)\hat{\boldsymbol{e}}_2 + (\dot{\psi}\cos\phi\cos\theta - \dot{\theta}\sin\phi)\hat{\boldsymbol{e}}_3 \tag{2.35}$$

or in compact matrix form:

$$\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \tag{2.36}$$

Inverting the relation, we obtain the *kinematic equations*:

$$\begin{cases} \dot{\phi} = \omega_1 + (\omega_2\sin\phi + \omega_2\cos\phi)\tan\theta \\ \dot{\theta} = \omega_2\cos\phi - \omega_3\sin\phi \\ \dot{\psi} = (\omega_2\sin\phi + \omega_3\cos\phi)/\cos\theta \end{cases} \tag{2.37}$$

which, once integrated, give the evolution of the bryant's angles as a function of the angular velocity. The geometric singularity is noticeable in equation (2.37), since, for $\theta = \pm\pi/2$, ϕ and ψ are not defined, hence the need to introduce a new way of representing attitude such as quaternions which don't suffer from singularity issues.

2.2.2 Evolution of quaternions

Since the attitude is being represented through unit quaternions, we need to rewrite the kinematic equations in terms of quaternions, it's to say an equation whose solution is the time evolution of the quaternion given the angular velocity $\boldsymbol{\omega}$ in the body frame or inertial frame. Such a problem is basically to find, given an unit quaternion function $\boldsymbol{Q}(t)$, its time derivative. Depending on which frame we describe the angular velocity, $\boldsymbol{\omega_B}$ in body frame and $\boldsymbol{\omega_I}$ in inertial frame, the evolution of the quaternion follows the equation:

$$\dot{\boldsymbol{Q}} = \frac{1}{2}\boldsymbol{\omega}_I \otimes \boldsymbol{Q} = \frac{1}{2}\boldsymbol{Q} \otimes \boldsymbol{\omega}_B \tag{2.38}$$

Proof. [18] Suppose a time interval Δt. The attitude at time $t + \Delta t$ is described by the quaternion $\boldsymbol{Q}(t + \Delta t)$. The difference between the two quaternions is an extra rotation around the instantaneous axis $\hat{\boldsymbol{\omega}}_B = \boldsymbol{\omega}_B/||\boldsymbol{\omega}_B||$ in body frame by a small angle $\Delta\theta_B = ||\boldsymbol{\omega}_B||\Delta t$. We can associate a new quaternion with this extra rotation:

$$\Delta\boldsymbol{Q}_B = \cos\frac{\Delta\theta_B}{2} + \hat{\boldsymbol{\omega}}_B \sin\frac{\Delta\theta_B}{2} \tag{2.39}$$

Since we are dealing with a small perturbation, we can assume:

$$\boldsymbol{\Delta Q}_B \approx \begin{bmatrix} 1 \\ \Delta\boldsymbol{\theta}_B/2 \end{bmatrix} \tag{2.40}$$

with:

$$\boldsymbol{\omega}_B \triangleq \frac{d\boldsymbol{\theta}_B(t)}{dt} \triangleq \lim_{\Delta t \to 0} \frac{\Delta\boldsymbol{\theta}_B}{\Delta t} \tag{2.41}$$

Therefore, starting from time t we can reach the attitude at $t + \Delta t$ with the extra rotation, which means through the sequence of rotation:

$$\boldsymbol{Q}(t + \Delta t) = \boldsymbol{Q}(t) \otimes \Delta\boldsymbol{Q}_B \tag{2.42}$$

Calculating the derivative of the quaternion as the limit of the difference quotient:

$$\begin{aligned}
\dot{\boldsymbol{Q}}(t) &\triangleq \lim_{\Delta t \to 0} \frac{\boldsymbol{Q}(t + \Delta t) - \boldsymbol{Q}(t)}{\Delta t} \\
&= \lim_{\Delta t \to 0} \frac{\boldsymbol{Q} \otimes \Delta\boldsymbol{Q}_B - \boldsymbol{Q}}{\Delta t} \\
&= \lim_{\Delta t \to 0} \frac{\boldsymbol{Q} \otimes \left(\begin{bmatrix} 1 \\ \Delta\boldsymbol{\theta}_B/2 \end{bmatrix} - \begin{bmatrix} 1 \\ \boldsymbol{0} \end{bmatrix} \right)}{\Delta t} \\
&= \lim_{\Delta t \to 0} \frac{\boldsymbol{Q} \otimes \begin{bmatrix} 0 \\ \Delta\boldsymbol{\theta}_B/2 \end{bmatrix}}{\Delta t} \\
&= \frac{1}{2}\boldsymbol{Q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_B \end{bmatrix}
\end{aligned} \tag{2.43}$$

□

By defining:

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) \triangleq \begin{bmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & -[\boldsymbol{\omega}]_\times \end{bmatrix} \tag{2.44}$$

we can write (2.38) in a matrix form as follows:

$$\dot{\boldsymbol{Q}} = \frac{1}{2}\boldsymbol{Q} \otimes \boldsymbol{\omega}_B = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_B)\boldsymbol{Q} \tag{2.45}$$

Observation. The angular velocity vector $\boldsymbol{\omega}_B$ is expressed with respect to the body frame. If we want to express the perturbation and hence the angular velocity with respect to the intertial frame at each time, we can obtain so through a rotation with the quaternion $\boldsymbol{Q}$ as follows:

$$\boldsymbol{\omega}_I = \boldsymbol{Q} \otimes \boldsymbol{\omega}_B \otimes \boldsymbol{Q}^* \tag{2.46}$$

By multiplying at both right sides of (2.46) by $\boldsymbol{Q}$ we obtain the other version of the derivative showed in (2.38):

$$\frac{1}{2}\boldsymbol{\omega}_I \otimes \boldsymbol{Q} = \dot{\boldsymbol{Q}} = \frac{1}{2}\boldsymbol{Q} \otimes \boldsymbol{\omega}_B \tag{2.47}$$

Summary

Throughout the discussion above, we were able to appreciate the far more intuitive geometric meaning of Euler's angles representation of attitude and the minimum number of parameters needed. Nevertheless, a singularity problem occured, which is dangerous when it comes to calculation. The definition of quaternions solves the singularity problem, but with the cost of adding a 4th parameter and respecting the constraint in (2.21). Despite this, the kinematic equation with quaternion is less computationally expensive.

2.3 Attitude dynamics

So far we've explored the attitude kinematics of the spacecraft and we've seen how quaternions evolve over time given the angular velocity vector $\boldsymbol{\omega}$. But where does it come from? As is well known, kinematics only deals with the description of the different ways a body can move. Dynamics, on the other hand, deals with the *causes* of motion. The backbone of dynamics is the well known *Newton second law of motion* which states that if external forces act on a body, they cause a variation of its momentum. In the same way, in rigid body rotational dynamics, an external *torque* $\boldsymbol{M}$ causes a variation of *angular momentum* $\boldsymbol{h}$:

$$\frac{d\boldsymbol{h}}{dt} = \boldsymbol{M} \tag{2.48}$$

From equation (2.48), we will show that an external torque $\boldsymbol{M}$ causes a variation of angular velocity $\boldsymbol{\omega}$, whose integration over time provides the input function for the kinematics equation which, in turn, provide the time evolution of quaternions.
For this purpose, let's consider the inertial frame $\mathscr{F}_I(\hat{\boldsymbol{E}}_1, \hat{\boldsymbol{E}}_2, \hat{\boldsymbol{E}}_3)$ and a body frame $\mathscr{F}_B(\hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2, \hat{\boldsymbol{e}}_3)$ attached to the spacecraft, represented as a rigid body, and centered in

its center of mass CM. Discretizing the body, each small mass element dm moves, during the rotation, with a velocity $\boldsymbol{v}$ and is identified in the body frame by the position vector $\boldsymbol{r}$. The infinitesimal angular momentum of the element is defined as:

$$d\boldsymbol{h} = \boldsymbol{r} \times dm\boldsymbol{v} = (\boldsymbol{r} \times \boldsymbol{v})\,\rho(\boldsymbol{r})dV \tag{2.49}$$

and integrating over the volume of the entire body gives the total angular momentum:

$$\boldsymbol{h} = \int_V (\boldsymbol{r} \times \boldsymbol{v})\,\rho(\boldsymbol{r})dV \tag{2.50}$$

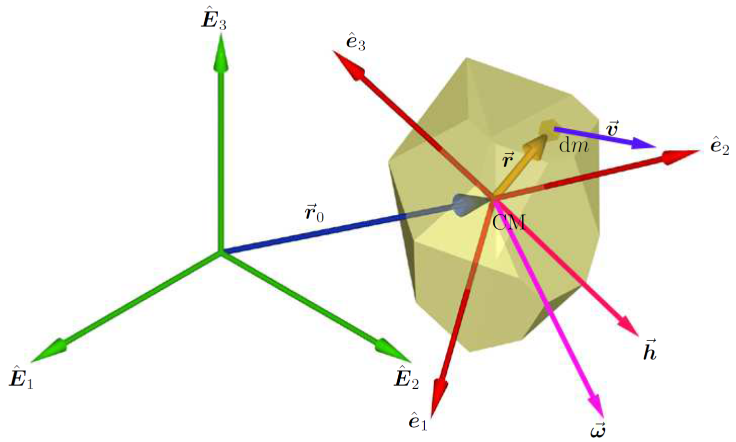

Figure 2.2. Spacecraft as a rotating rigid body.[2]

As shown in figure 2.2, since the body rotates around the center of mass CM, we can rewrite equation (2.50) as follows:

$$\boldsymbol{h} = \int_V [\boldsymbol{r} \times (\boldsymbol{\omega} \times \boldsymbol{v})]\,\rho(\boldsymbol{r})dV \tag{2.51}$$

Using vector properties we can extend the triple product in (2.51) as follows:

$$\boldsymbol{r} \times (\boldsymbol{\omega} \times \boldsymbol{r}) = (\boldsymbol{r} \cdot \boldsymbol{r})\,\boldsymbol{\omega} - (\boldsymbol{r} \cdot \boldsymbol{\omega})\,\boldsymbol{r} \tag{2.52}$$

Lastly, recalling the definition of *dyadic tensor* by two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ as:

$$\boldsymbol{ab} \equiv \boldsymbol{ab}^T = \sum_{i=1}^{3}\sum_{j=1}^{3} a_i b_j \boldsymbol{e}_i \boldsymbol{e}_j \tag{2.53}$$

we can express the right dot product as follows[1]:

$$(\boldsymbol{ab}) \cdot \boldsymbol{c} = a_i(b_j c^j)\boldsymbol{e}_i = (\boldsymbol{b} \cdot \boldsymbol{c})\boldsymbol{a} \tag{2.54}$$

[1] The Einstein summation convention is used.

Applying this rule to equation (2.52) we obtain the following expression of the angular momentum:

$$\boldsymbol{h} = \int_V [\boldsymbol{r} \times (\boldsymbol{\omega} \times \boldsymbol{v})] \, \rho(\boldsymbol{r}) dV = \left(\int_V [(\boldsymbol{r} \cdot \boldsymbol{r}) - (\boldsymbol{r}\boldsymbol{r})] \, \rho(\boldsymbol{r}) dV \right) \boldsymbol{\omega} = \boldsymbol{I}\boldsymbol{\omega} \tag{2.55}$$

We recognize in the parenthesis the mathematical expression of the *inertia tensor*, whose components are:

$$I_{ij} = \int_V [\delta_{ij} ||\boldsymbol{r}||^2 - x_i x_j] \, \rho(\boldsymbol{r}) dV \tag{2.56}$$

indeed, we can extract some well known components using the definition of the *Kronecker delta* given in equation (2.4):

$$\begin{aligned} I_{xx} &= \int_V [\delta_{11}(x^2 + y^2 + z^2) - x^2] \, \rho(\boldsymbol{r}) dV = \int_V (y^2 + z^2) \, \rho(\boldsymbol{r}) dV \\ I_{xy} &= \int_V [\delta_{12}(x^2 + y^2 + z^2) - xy] \, \rho(\boldsymbol{r}) dV = \int_V -xy \, \rho(\boldsymbol{r}) dV \end{aligned} \tag{2.57}$$

and so forth. The inertia tensor is symmetric and made of real components, therefore its eigenvalues are real and its eigenvectors are mutually orthogonal. This of course allows us to identify a frame of reference in which the inertia matrix is diagonal. The eigenvectors are called *principal axex*, whereas the respective eigenvalues are called *principal moments of inertia*:

$$\boldsymbol{I} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{2.58}$$

In order to derive Euler's equations, let's evaluate the derivative of $\boldsymbol{h}$ in equation (2.48) in body frame as follows:

$$\dot{\boldsymbol{h}}_B + \boldsymbol{\omega}_B \times \boldsymbol{h}_B = \boldsymbol{M}_B \tag{2.59}$$

assuming that the inertia matrix $\boldsymbol{I}$ is constant, we obtain:

$$\boldsymbol{I}\dot{\boldsymbol{\omega}}_B + \boldsymbol{\omega}_B \times (\boldsymbol{I}\boldsymbol{\omega}_B) = \boldsymbol{M}_B \tag{2.60}$$

which leads us, in a set of principal axes, to the *Euler's equations*:

$$\begin{cases} I_x\dot{\omega}_1 + (I_z - I_y)\omega_2\omega_3 = M_1 \\ I_y\dot{\omega}_2 + (I_x - I_z)\omega_3\omega_1 = M_2 \\ I_z\dot{\omega}_3 + (I_y - I_x)\omega_1\omega_2 = M_3 \end{cases} \tag{2.61}$$

In our case, with a satellite equipped with reaction wheels as the actuation system, equation (2.60) becomes:

$$\boldsymbol{I}\dot{\boldsymbol{\omega}}_B + \boldsymbol{\omega}_B \times (\boldsymbol{I}\boldsymbol{\omega}_B + \boldsymbol{I}_{RW}\boldsymbol{\omega}_{RW}) = \boldsymbol{M}_B \tag{2.62}$$

where $\boldsymbol{\omega}_{RW}$ is the angular velocity of the reaction wheel. In $\boldsymbol{M}_B$ the effect of the reaction wheels $\boldsymbol{M}_{RW}$ and external disturbances $\Delta\boldsymbol{M}_{ex}$ are included.

Summary

We finally closed the problem: given an external torque $\boldsymbol{M}_B$ of the actuation system plus external disturbances, we can integrate Euler's equation and derive the time evolution of the angular velocity $\boldsymbol{\omega}$, which, in turn, can be used to determine the time evolution of the attitude quaternion, which can ultimately be compared with a reference and enter a guidance and control logic, closing the loop.

2.4 Actuation system

The last ingredient of the mathematical model needed to accomplish the attitude control is of course the actuation system. The satellite needs to provide an external force in order to orient itself performing the control signal generated by the guidance and control subsystem. In our case study, the satellite is equipped with 4 *reaction wheels*, which provide a high pointing accuracy and are particularly useful when the spacecraft must be rotated by very small amounts, for instance space telescopes.
A reaction wheel is a type of flywheel used primarly for three axis attitude control. It is a brushless motor attached to a high-inertia flywheel which is free to spin along a fixed spacecraft axis. The operating principle is simple: *conservation of angular momentum*, which states that, in absence of external torque, the total angular momentum of the spacecraft remains constant:

$$\frac{d\boldsymbol{h}}{dt} = 0 \implies \boldsymbol{h} = constant \tag{2.63}$$

For the system of satellite plus its reaction wheel, we obtain:

$$\boldsymbol{I}\boldsymbol{\omega}_B + \boldsymbol{I}_{RW}\boldsymbol{\omega}_{RW} = constant \tag{2.64}$$

So, in order to mantain the total angular momentum constant, by accelerating the wheel in one direction about the wheel spin axis, it appies a reaction torque to the spacecraft in the opposite direction.

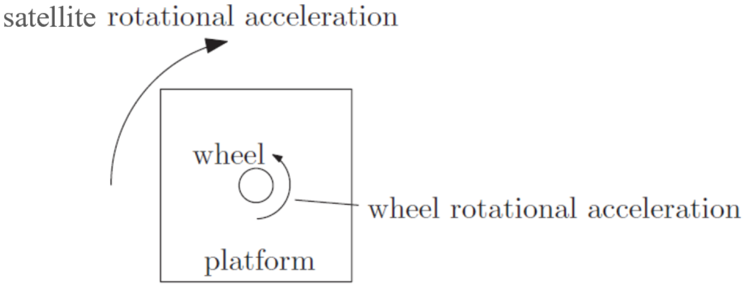
satellite rotational acceleration
wheel
wheel rotational acceleration
platform

Figure 2.3. Operating principle of a reaction wheel.[5]

For this reason a reaction wheel is considered a *momentum exchange device*, since angular momentum is transferred between wheels and satellite to control its attitude. But equation (2.48) clearly states that in case of an external torque, there's a change in overall spacecraft angular momentum. Therefore, if the control objective is to keep the spacecraft attitude inertially fixed (the satellite platform keeps zero angular momentum), all the angular momentum change due to the external disrturbance torque is absorbed by the reaction wheels, which will start spinning faster. This rises a crucial point, which is the wheel *saturation*. As disturbances keep on acting, they result in a build-up of wheel angular momentum, causing the wheel to face the following limitations:

- *torque saturation*: due to reaching the maximum current that can flow in motor wires, the wheel is not able to provide a torque higher than the one for which it has been designed;
- *momentum saturation*: due to mechanical limitations, the wheel, once the maximum angular velocity is reached, it is not able to further accelerate.

Thus it is necessary to de-saturate the wheel, and it can be achieved by applying an external torque to de-load the wheel; it can be provided by reaction control thrusters or magnetorquers. A reliable three-axis stabilization attitude control system, therefore, needs a minimum of three reaction wheel, an additional one for redundancy and a set of thrusters and/or magnetorquers for the de-saturation.

2.4.1 Reaction wheels modeling

With the introduction of the reaction wheels, Euler's equations (2.60) becomes:

$$\dot{\boldsymbol{\omega}}_B = \boldsymbol{I}^{-1}(\boldsymbol{M}_{ex} - \boxed{\boldsymbol{M}_{RW}} - \boldsymbol{\omega}_B \times (\boldsymbol{I}\boldsymbol{\omega}_B + \boxed{\boldsymbol{h}_{RW}})) \tag{2.65}$$

In the boxes, we recall the torque from the reaction wheels, which enters Euler's equations with a "minus" sign, and the reaction wheel angular momentum. One advantage of reaction wheels is that they apply a *continuous* and variable torque to the system. Therefore, the control torque only needs to undergo a "minus sign" gain, a possible geometrical transformation due to the relative positioning of the wheels, a low-pass filter to filter out high frequencies of the control torque for each wheel and a saturation action which limits the value of the actual torque for the dynamics. What figure 2.4 shows are the following equations in Laplace domain:

$$\begin{aligned} M_{RW} &= -\frac{1}{\tau_{RW}s + 1}M_c \\ h_{RW} &= \frac{1}{s}M_{RW} \end{aligned} \tag{2.66}$$

As we already said previously, a satellite needs a cluster of reaction wheels, to fully control the three-axis attitude, made of 3 wheels plus 1 for redundancy: in this way if a fail occurs, a three-axis attitude control is always ensured, even though pointing accuracy may be degraded. Hence, we can find different configurations of clusters. The most used are:

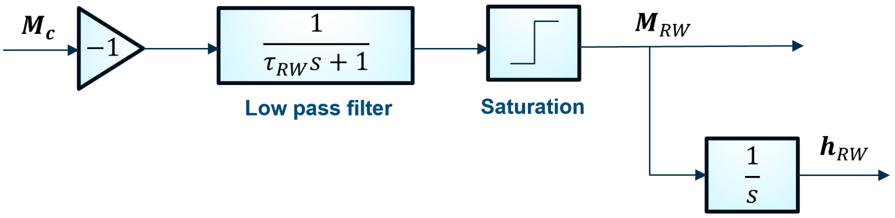

Figure 2.4. Block scheme of a simple reaction wheel model.

- *NASA 3+1*: three wheels are aligned with the satellite main inertia axis whereas the fourth is inclined by a certain angle with respect to all the planes;
- *pyramidal*: the wheels spin axis are pointed towards the faces of a pyramid with squared base;
- *tetrahedral*: just like the pyramidal one, here the spin axes are pointed towards the faces of a tetrahedron.

We note that, before applying equations (2.66), the control torque, which is a 3 × 1 column vector, needs to be allocated to the four wheels assembly, it's to say to check how each reaction wheel axis contributes to the control torque along the three body axes. In our case, a pyramidal configuration will be used in the simulation. In order to describe how the allocation process works and derive the transformation matrix, let's focus on figure 2.5. Let's consider the case with $\alpha = 0°$. The red vectors are the torque component given by

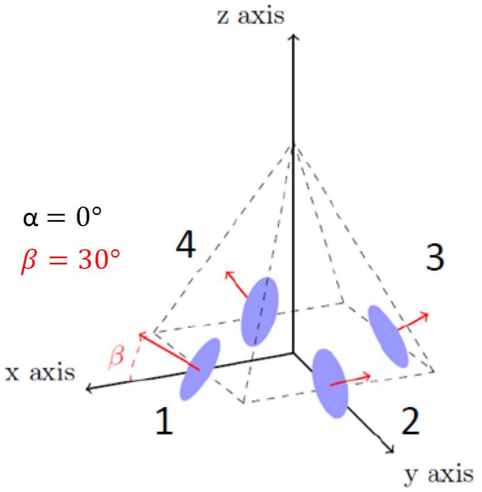

Figure 2.5. Pyramidal configuration with mounting angles $\beta = 30°$ and $\alpha = 0°$.[5]

each wheel. How can we go back to the 3-components vectors? We can notice that, fixing one axis at a time, for instance the x axis, only the wheel number 1 and 3 contribute, respectively with a coefficient $\cos\beta$ and $-\cos\beta$, whereas the number 2 and 4 do not contribute. Same thing happens with y axis, but here only 2 and 4 contribute. Finally, along the z axis all the wheels contribute with a coefficient $\sin\beta$. This transformation is then represented by the following matrix relation:

$$\begin{bmatrix} M_{RW_1} \\ M_{RW_2} \\ M_{RW_3} \end{bmatrix} = \begin{bmatrix} \cos\beta & 0 & -\cos\beta & 0 \\ 0 & \cos\beta & 0 & -\cos\beta \\ \sin\beta & \sin\beta & \sin\beta & \sin\beta \end{bmatrix} \begin{bmatrix} M_{W_1} \\ M_{W_2} \\ M_{W_3} \\ M_{W_4} \end{bmatrix} \tag{2.67}$$

where M_{W_i} is the component of torque given by the i-th wheel in the cluster. What if $\alpha \neq 0$? In that case we can simply multiply the vector obtained in equation (2.67) with a rotation matrix along the z axis by an angle of α. We can group the two matrices in the $\boldsymbol{Z}$ matrix, which transforms the torque in the cluster to the 3-components torque needed by the dynamics:

$$\underbrace{\begin{bmatrix} M_{RW_1} \\ M_{RW_2} \\ M_{RW_3} \end{bmatrix}}_{\boldsymbol{M}_{RW}} = \underbrace{\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & -\cos\beta & 0 \\ 0 & \cos\beta & 0 & -\cos\beta \\ \sin\beta & \sin\beta & \sin\beta & \sin\beta \end{bmatrix}}_{\boldsymbol{Z}} \underbrace{\begin{bmatrix} M_{W_1} \\ M_{W_2} \\ M_{W_3} \\ M_{W_4} \end{bmatrix}}_{\boldsymbol{M}_W} \tag{2.68}$$

where we define:

$$\boldsymbol{Z} = \begin{bmatrix} \cos\beta\cos\alpha & -\cos\beta\sin\alpha & -\cos\beta\cos\alpha & \cos\beta\sin\alpha \\ \cos\beta\sin\alpha & \cos\beta\cos\alpha & -\cos\beta\sin\alpha & -\cos\beta\cos\alpha \\ \sin\beta & \sin\beta & \sin\beta & \sin\beta \end{bmatrix} \tag{2.69}$$

To summarise, with a reaction wheels cluster, the control torque, which is a 3 × 1 vector, needs to be transformed in the cluster frame of reference and this can be achieved inverting equation (2.68) through the definition of the *pseudo-inverse matrix* (or *generalized inverse*):

$$\boldsymbol{M}_C^* = \boldsymbol{Z}^+ \boldsymbol{M}_C \implies \boldsymbol{M}_{RW} = \boldsymbol{Z}\boldsymbol{M}_W \tag{2.70}$$

where $\boldsymbol{M}_C^*$ is the control torque in the cluster frame before the filtering.

Chapter 3

Guidance and Control Algorithms

In the following chapter the guidance algorithm will be presented. It's based on the Artificial Potential Field method (APF), whose mathematical formulation will be presented first, and subsequently applied to the case in question.
The APF method goes through the mathematical concept of the *gradient descent*, where a generalized force is generated by taking the gradient of the potential field at each point in state-space and the state variable evolves accordingly. Attitude constraint and quaternions are taken into account in a particular manner exploiting quaternion properties and a global potential field, function of the quaternion error state vector, can be formulated, providing a reference value for the control system and dynamics to follow.

3.1 Mathematical formulation of APF

In the introduction we explained the importance of on-line motion planning and it's mandatory for a satellite to orient itself with help of partial information provided by the on-board sensors. The APF method is suitable for on-line planning, since the satellite and obstacle attitudes can be measured and therefore it is possible to evaluate a potential field of a predefined mathematical form; thus, a promising direction of local motion is indicated by the negative gradient of the potential $-\nabla U$. In this section we won't work with attitudes yet, but introduce the general formalism with a general configuration $\boldsymbol{q}$.

3.1.1 Attractive Potential

The attractive potential is designed to guide the system to its goal configuration $\boldsymbol{q}_g$. It is possible to think of two types of potential fields: a *conical* attractive field and a *parabolic* field. A parabolic attractive field has the following form[17]:

$$U_{a1}(\boldsymbol{q}) = \frac{1}{2}k_a\boldsymbol{e}^T(\boldsymbol{q})\boldsymbol{e}(\boldsymbol{q}) = \frac{1}{2}k_a||\boldsymbol{e}(\boldsymbol{q})||^2 \tag{3.1}$$

where $k_a > 0$ is a constant used to shape the field and $\boldsymbol{e} = \boldsymbol{q}_g - \boldsymbol{q}$ is the error vector with respect to a configuration goal $\boldsymbol{q}_g$. It's been defined so that it's always positive and has a global minimum at the origin (in $\boldsymbol{q}_g$). Calculating the gradient:

$$\boldsymbol{f}_{a1} = -\nabla U_{a1}(\boldsymbol{q}) = k_a \boldsymbol{e}(\boldsymbol{q}) \tag{3.2}$$

This tells us that the resulting force is linear and converges linearly to zero. Notice that the apparent lack of a minus sign is due to the definition of the error vector, which, working the chain rule, leads to $\nabla(\boldsymbol{q}_g - \boldsymbol{q}) = -1$.

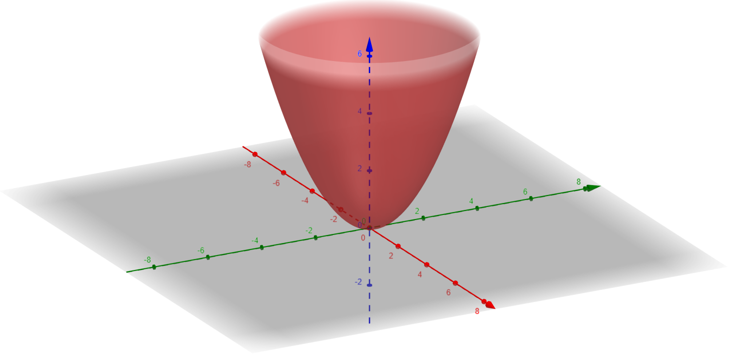

Figure 3.1. Parabolic attractive potential.

On the other hand, a conical attractive field has the following form:

$$U_{a2}(\boldsymbol{q}) = k_b ||\boldsymbol{e}(\boldsymbol{q})|| \tag{3.3}$$

Just like the parabolic field, it's always positive except at the origin where it is zero. The resulting force is:

$$\boldsymbol{f}_{a2} = -\nabla U_{a2}(\boldsymbol{q}) = k_b \frac{\boldsymbol{e}(\boldsymbol{q})}{||\boldsymbol{e}(\boldsymbol{q})||} \tag{3.4}$$

we can notice that the force is constant in modulus. Why is it even considered? Having a constant force is an advantage when the error vector is very large. This is the case of having, at the beginning of the simulation, an obstacle kind of far away from the current position and, with a parabolic field, the resulting force would be too large (it grows linearly with the error) and the actuation system may not be able to achieve the reference value; a constant force when the error is large avoids an unfortunate saturation of the actuation system. On the other hand, the conical field has a discontinuity in the derivative in $\boldsymbol{q}_g$ and the resulting force is indefinite.

We can combine the advantages of the two potential fields by defining a hybrid attractive potential made of a conical surface away from the goal and a parabolic surface in the vicinity of the goal. In order for the combination to produce a continuous force, an extra care of the mathematical form must be taken. Let's indicate with ρ the separation region

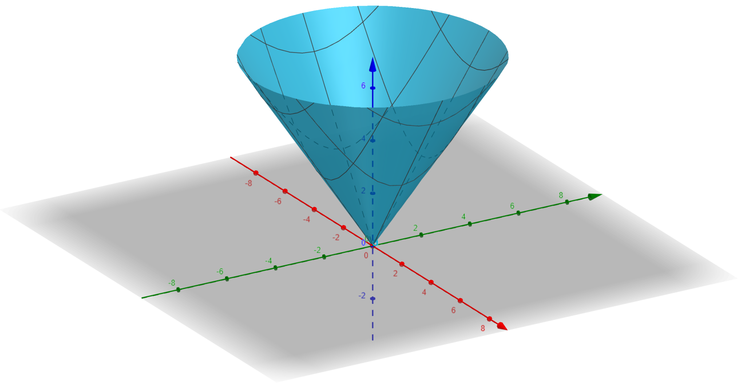

Figure 3.2. Conical attractive potential.

of the two forms of field. Looking at figures 3.1 and 3.2 we notice that for the combination to be continuous at $||\boldsymbol{e}(\boldsymbol{q})|| = \rho$, the vertex of the cone must translate, and then be non zero at $\boldsymbol{q}_g$. This can also be derived by noticing that at that point the "conic line" must have the same slope as the parabola at that point, which equals to the derivative of the parabolic field at that point; subsequently, one line from the sheaf of parallel lines must be chosen, in particular the one passing through that same point. We express the attractive potential as follows:

$$U_a(\boldsymbol{q}) = \begin{cases} \frac{1}{2}k_a||\boldsymbol{e}(\boldsymbol{q})||^2 & \text{if } ||\boldsymbol{e}(\boldsymbol{q})|| \leq \rho \\ k_b||\boldsymbol{e}(\boldsymbol{q})|| + K & \text{if } ||\boldsymbol{e}(\boldsymbol{q})|| > \rho \end{cases} \tag{3.5}$$

Keeping the constant k_a fixed and known, we have two constant to determine (k_b and K). We can do so by imposing the condition of continuity and differentiability to the field.

Differentiability

We impose that at $||\boldsymbol{e}(\boldsymbol{q})|| = \rho$ the derivatives of the two pieces in (3.5) are equal:

$$k_a\boldsymbol{e}(\boldsymbol{q})\Big|_\rho = k_b\frac{\boldsymbol{e}(\boldsymbol{q})}{||\boldsymbol{e}(\boldsymbol{q})||}\Bigg|_\rho \implies k_a = \frac{k_b}{\rho} \implies k_b = \rho k_a \tag{3.6}$$

Continuity

We impose the same condition directly at the potential in (3.5) to find out the value of K:

$$\frac{1}{2}k_a\rho^2 = k_a\rho^2 + K \implies K = -\frac{1}{2}k_a\rho^2 \tag{3.7}$$

The combined attractive potential is of the following form:

$$U_a(\boldsymbol{q}) = \begin{cases} \dfrac{1}{2}k_a||\boldsymbol{e}(\boldsymbol{q})||^2 & \text{if } ||\boldsymbol{e}(\boldsymbol{q})|| \leq \rho \\ \rho k_a||\boldsymbol{e}(\boldsymbol{q})|| - \dfrac{1}{2}k_a\rho^2 & \text{if } ||\boldsymbol{e}(\boldsymbol{q})|| > \rho \end{cases} \tag{3.8}$$

and the resulting force:

$$\boldsymbol{f}_a = \begin{cases} k_a\boldsymbol{e}(\boldsymbol{q}) & \text{if } ||\boldsymbol{e}(\boldsymbol{q})|| \leq \rho \\ \rho k_a \dfrac{\boldsymbol{e}(\boldsymbol{q})}{||\boldsymbol{e}(\boldsymbol{q})||} & \text{if } ||\boldsymbol{e}(\boldsymbol{q})|| > \rho \end{cases} \tag{3.9}$$

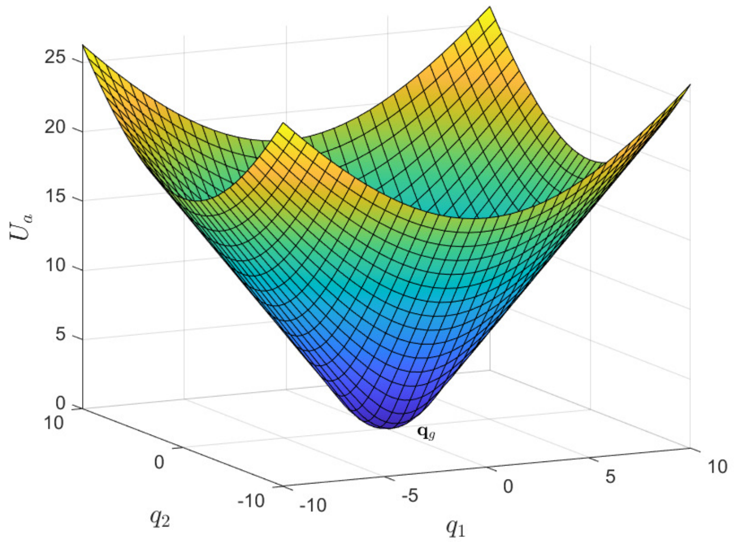

Figure 3.3. Final attractive potential.

3.1.2 Repulsive Potential

A repulsive potential field must be added to account for the presence of obstacles as the attractive field is followed. The idea is to add a barrier potential in the vicinity of the obstacle, which is a region where the field, from zero, start to rise and keeps growing as close as we get to the obstacle, so that the resulting force always points outwards of the obstacle. Indicating with $\eta(\boldsymbol{q})$ the relative distance from the obstacle and with η_0 the *range of influence*, for each obstacle we can build the following repulsive potential:

$$U_{r,i}(\boldsymbol{q}) = \begin{cases} \dfrac{1}{2}k_{r,i}\left(\dfrac{1}{\eta_i(\boldsymbol{q})} - \dfrac{1}{\eta_{0,i}}\right)^2 & \text{if } \eta_i(\boldsymbol{q}) \leq \eta_{0,i} \\ 0 & \text{if } \eta_i(\boldsymbol{q}) > \eta_{0,i} \end{cases} \tag{3.10}$$

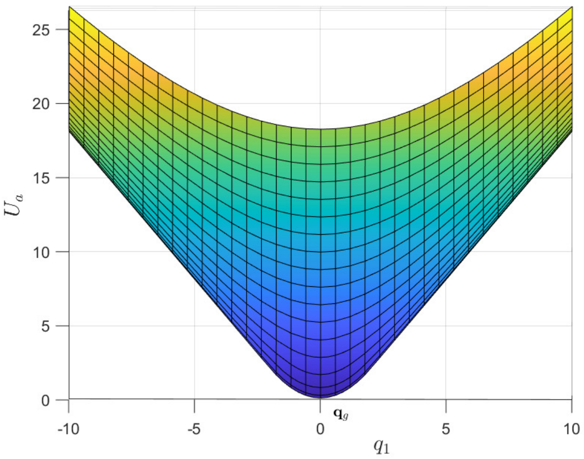

Figure 3.4. Profile detail of the attractive potential.

The resulting repulsive force is:

$$\boldsymbol{f}_{r,i}(\boldsymbol{q}) = -\nabla U_{r,i}(\boldsymbol{q}) = \begin{cases} \dfrac{k_{r,i}}{\eta_i^2(\boldsymbol{q})}\left(\dfrac{1}{\eta_i(\boldsymbol{q})} - \dfrac{1}{\eta_{0,i}}\right)\nabla\eta_i(\boldsymbol{q}) & \text{if } \eta_i(\boldsymbol{q}) \leq \eta_{0,i} \\ 0 & \text{if } \eta_i(\boldsymbol{q}) > \eta_{0,i} \end{cases} \tag{3.11}$$

For n obstacles, the aggregate repulsive is obtained by summing up the individual potentials:

$$U_r(\boldsymbol{q}) = \sum_{i=1}^{n} U_{r,i}(\boldsymbol{q}) \tag{3.12}$$

For the sake of simplicity, we'll assume throughout the thesis work and simulation that the goal is placed far away from the obstacles, and thus outside of the range of influence of any obstacle.

3.1.3 Total Potential

The total potential, as mentioned before, can be obtained by superposition of the attractive and repulsive potentials:

$$U_t(\boldsymbol{q}) = U_a(\boldsymbol{q}) + U_r(\boldsymbol{q}) \tag{3.13}$$

which results in the force field:

$$\boldsymbol{f}_t(\boldsymbol{q}) = -\nabla U_t(\boldsymbol{q}) = \boldsymbol{f}_a(\boldsymbol{q}) + \sum_{i=1}^{n} \boldsymbol{f}_{r,i}(\boldsymbol{q}) \tag{3.14}$$

The resulting potential fiel will then have a global minimum in $\boldsymbol{q}_g$ which will attract the configuration vector while avoiding the obstacles thanks to the repulsive forces pointing along the negative gradient (for an obstacle this is *away* from it).

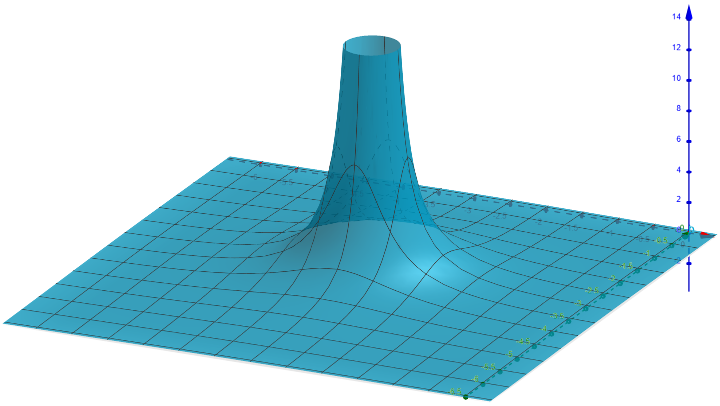

Figure 3.5. Repulsive potential field.

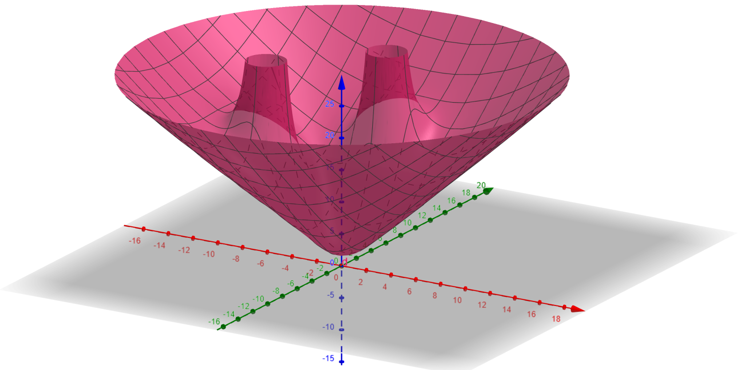

Figure 3.6. Graphic example of the total potential field as a superposition of attractive and repulsive fields.

3.1.4 Planning Techniques

So far we've been calling the gradient of the potential a generalized *force*, from the formalism of classical mechanics. But can it be considered an actual force? The answer is yes... but also other things. Based on the physical meaning given to the gradient of the potential, we can have different planning techniques. The three main used are listed below[17]:

1. We let the gradient be:

$$\boldsymbol{\tau} = \boldsymbol{f}_t(\boldsymbol{q}) \tag{3.15}$$

   which is an actual set of generalized forces that act on the system according to its dynamic model.

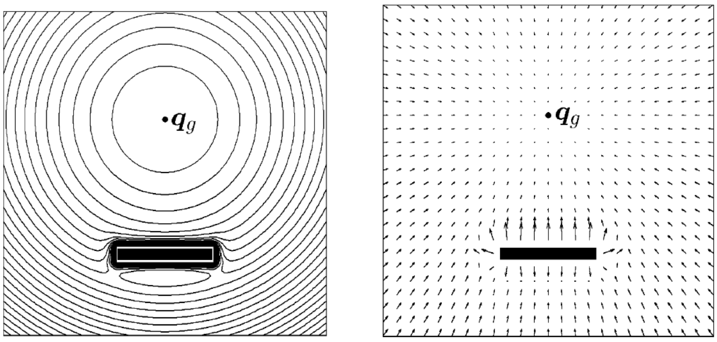

Figure 3.7. Total potential field. *Left*: equipotential contours, *right*: resulting force field.[17]

2. We consider the system as a point mass, hence the generalized forces are indeed an acceleration vector:
$$\ddot{\boldsymbol{q}} = \boldsymbol{f}_t(\boldsymbol{q}) \tag{3.16}$$

3. Third possibility is to consider the force as a reference velocity:
$$\dot{\boldsymbol{q}} = \boldsymbol{f}_t(\boldsymbol{q}) \tag{3.17}$$

Let's discuss briefly on their role. In the first case, (3.15) directly represents control input for the system (the force goes directly into the pant dynamics), whereas (3.16) requires the solution of the inverse dynamics problem, i.e. substituting into the dynamic system $\ddot{\boldsymbol{q}}$ (i.e., $\dot{\boldsymbol{\omega}}$) to solve for the generalized force $\boldsymbol{\tau}$ that realize such acceleration. Finally, (3.17) can provide a reference value ($\boldsymbol{\omega}$) for the controller to follow. What are the pros and cons of these techniques so that a suitable one can be chosen? The use of (3.15) generates smoother paths, since the reactions of the presence of obstacles are filtered by the system dynamics, but, just like (3.16), in order to achieve asymptotic stability a dumping term proportional to $\dot{\boldsymbol{q}}$ must be added to the gradient of the potential. On the other hand (3.17) is safer because it's faster in executing the motion corrections given by the potential gradient. In addition, it automatically guarantees the asymptotic stability of the goal. The third choice (3.17) is typically used. Generally at this point, equation (3.17) is numerically integrated to obtain, at each time step, the next system configuration:

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \Delta t \boldsymbol{f}_t(\boldsymbol{q}_k) \tag{3.18}$$

with Δt as the time step. Since $\boldsymbol{f}_t(\boldsymbol{q}) = -\nabla U_t(\boldsymbol{q})$, equation (3.18) is also known as *algorithm of steepest descent.*

3.2 Attitude guidance design with APF

In this section the application of APF method to the problem in exam is presented. We will move from general system with general state variable (or configuration) $\boldsymbol{q}$ to an attitude reorientation problem, with the rotational dynamics and kinematics represented through a unit quaternion $\boldsymbol{Q}$. A way to link the quaternion configuration at each time step to the angular separation between instrument and obstacle is needed, and the APF method will be used, with the technique given by (3.17). After that, both the reference values for the quaternion and the angular velocity (the latter as the output of the guidance algorithm) will generate error state vectors, which in turn will enter the control law and provide the dynamic system with a control torque.

3.2.1 Attitude constraints

For the attitude constraints model, let's identify the pointing direction of the on-board instrument with a normalized boresight vector $\boldsymbol{y}$ in $\mathscr{F}_B$ and the direction of the obstacle with a normalized vector $\boldsymbol{x}$ in $\mathscr{F}_I$ and suppose a half-cone angle greater than θ must be maintained between the two vectors. As shown in figure 3.8, during the reorientation maneuver the cone around the instrument vector should avoid the unwanted objects. How can we make sure this information includes the system state variable, the unit quaternion, for an APF implementation? Let's note first that the vector $\boldsymbol{y}$ is expressed in the body frame whereas the obstacle $\boldsymbol{x}$ is expressed in the inertial frame. From the mathematical model chapter we've seen how we can calculate the rotational matrix $\boldsymbol{L}_{BI}$ knowing the quaternion. Using equation (2.16), we can express the boresight instrument vector in the inertial frame as follows:

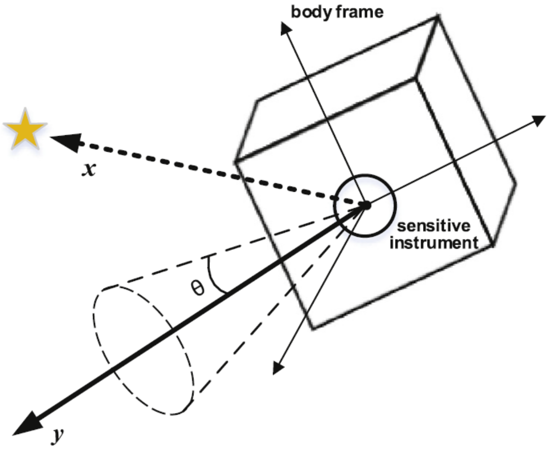

Figure 3.8. Attitude constraint model.[15]

$$\boldsymbol{y}_I = \boldsymbol{L}_{IB}\boldsymbol{y} = \boldsymbol{L}_{BI}^T\boldsymbol{y} = (q_0^2 - \boldsymbol{q}^T\boldsymbol{q})\boldsymbol{y} + 2(\boldsymbol{q}^T\boldsymbol{y})\boldsymbol{q} + 2q_0(\boldsymbol{q} \times \boldsymbol{y}) \tag{3.19}$$

The attitude constraint can be expressed as follows:

$$\boldsymbol{x} \cdot \boldsymbol{y}_I < \cos\theta \tag{3.20}$$

it's to say, the cosine of the angle between the two normalized vectors must be less than the cosine of the angle θ, which means that the angle between the vectors must be greater than the angle θ throughout the maneuver. It follows from eq. (3.20) that:

$$q_0^2\boldsymbol{x}^T\boldsymbol{y} - \boldsymbol{q}^T\boldsymbol{q}\boldsymbol{x}^T\boldsymbol{y} + 2(\boldsymbol{q}^T\boldsymbol{y})\boldsymbol{x}^T\boldsymbol{q} + 2q_0\boldsymbol{q}^T(\boldsymbol{y}\times\boldsymbol{x}) < \cos\theta \tag{3.21}$$

using some algebra, we can highlight the entire attitude unit quaternion $\boldsymbol{Q}$ as follows:

$$\boldsymbol{Q}^T \begin{bmatrix} \boldsymbol{x}\boldsymbol{y}^T + \boldsymbol{y}\boldsymbol{x}^T - (\boldsymbol{x}^T\boldsymbol{y})\boldsymbol{I}_3 & \boldsymbol{y}\times\boldsymbol{x} \\ (\boldsymbol{y}\times\boldsymbol{x})^T & \boldsymbol{x}^T\boldsymbol{y} \end{bmatrix} \boldsymbol{Q} < \cos\theta \tag{3.22}$$

We can generalize equation (3.22) supposing there are m on-board sensitive instruments and, for the jth instrument, n constrained obstacles are associated with it. Therefore, the attitude unit quaternion $\boldsymbol{Q}$ for which the boresight vector $\boldsymbol{y}_j$ is associated with the ith obstacle should satisfy the following equation:

$$\boldsymbol{Q}^T\boldsymbol{M}_i^j\boldsymbol{Q} < \cos\theta_i^j \tag{3.23}$$

where

$$\boldsymbol{M}_i^j = \begin{bmatrix} A_i^j & b_i^j \\ b_i^{jT} & d_i^j \end{bmatrix} \tag{3.24}$$

with

$$A_i^j = \boldsymbol{x}_i\boldsymbol{y}_j^T + \boldsymbol{y}_j\boldsymbol{x}_i^T - (\boldsymbol{x}_i^T\boldsymbol{y}_j)\boldsymbol{I}_3, \quad b_i^j = \boldsymbol{y}_j \times \boldsymbol{x}_i, \quad d_i^j = \boldsymbol{x}_i^T\boldsymbol{y}_j$$
$$i = 1,2,\dots,n, \quad j = 1,2,\dots,m$$

The angle θ_i^j is the constraint angle for the jth instrument boresight vector $\boldsymbol{y}_j$ about the direction of the ith object specified by $\boldsymbol{x}_i$. We can assume its domain to be $(0,\pi)$.
By defining $\bar{\boldsymbol{M}}_i^j = \boldsymbol{M}_i^j - \cos\theta_i^j\boldsymbol{I}_4$, equation (3.23) can be written as follows:

$$\boldsymbol{Q}^T\bar{\boldsymbol{M}}_i^j\boldsymbol{Q} < 0 \tag{3.25}$$

we will refer to the term in equation (3.25) by *quadratic constraint* in the results chapter: it tells us numerically the difference value between the cosine of the separation angle and the cosine of the constraint angle of instrument and obstacle throughout the simulation and its being negative reflects the starting constraint stated in eq. (3.20) and guarantees us that during the maneuver the instrument actually avoids the obstacle.

3.2.2 Attractive potential

The attractive potential used for our attitude problem will be of the form of eq. (3.8), which is parabolic close to the goal and conical far away from it. As the *error* state variable in this case we will use the vectorial part of the *error quaternion* $\boldsymbol{Q}_e$ defined in eq. (2.30). Therefore, defining the goal quaternion $\boldsymbol{Q}_g$, we will express the potential fields as a function of:

$$\boldsymbol{q}_e = \mathrm{Vec}[\boldsymbol{Q}_e] = \mathrm{Vec}[\boldsymbol{Q}_g^* \otimes \boldsymbol{Q}] \tag{3.26}$$

which is the vectorial part of the error quaternion $\boldsymbol{Q}_e$. The resulting attractive potential is:

$$U_a(\boldsymbol{q}_e) = \begin{cases} \frac{1}{2}k_a||\boldsymbol{q}_e||^2 & \text{if } ||\boldsymbol{q}_e|| \leq \rho \\ \rho k_a||\boldsymbol{q}_e|| - \frac{1}{2}k_a\rho^2 & \text{if } ||\boldsymbol{q}_e|| > \rho \end{cases} \tag{3.27}$$

Once the value of ρ is chosen, in our case the value k_a needs to fulfill a crucial point. When the satellite is far away from the goal the conical part of the attractive potential field is on and therefore a constant reference angular velocity ($-\nabla U_a$) is followed. But clearly this value needs to comply with the actual performance of the actuation system: a reference value greater than the maximum obtainable angular velocity of the system can't be... obtained! The actuation system would saturate and not be able to follow the reference value which comprise of the path to avoid the obstacle. Hence, k_a needs to be chosen accordingly in order to generate a constant reference value which is equal to the maximum value given by the actuation system: we'll call it Ω. As we did in the mathematical formulation of APF, by taking care of continuity and differentiability, we can obtain the reference value of angular velocity by taking the gradient of the attractive potential field:

$$\boldsymbol{\omega}_a = \begin{cases} -\frac{\Omega}{\rho}\boldsymbol{q}_e & \text{if } ||\boldsymbol{q}_e|| \leq \rho \\ -\Omega\frac{\boldsymbol{q}_e}{||\boldsymbol{q}_e||} & \text{if } ||\boldsymbol{q}_e|| > \rho \end{cases} \tag{3.28}$$

In order to account for the unwinding problem, simulation shows that it can be taken into the account by multiplying $\boldsymbol{\omega}_a$ with the sign of the scalar part of the error quaternion, obtaining:

$$\boldsymbol{\omega}_a = \begin{cases} -\frac{\Omega}{\rho}sgn(q_{e,0})\,\boldsymbol{q}_e & \text{if } ||\boldsymbol{q}_e|| \leq \rho \\ -\Omega\, sgn(q_{e,0})\frac{\boldsymbol{q}_e}{||\boldsymbol{q}_e||} & \text{if } ||\boldsymbol{q}_e|| > \rho \end{cases} \tag{3.29}$$

The values of the constants will be presented in the results chapter.

3.2.3 Repulsive potential

A repulsive potential for attitude constraints needs to take into account the modeling of the latter. We have done it by introducing the whole attitude unit quaternion into the angle separation inequality through the quadratic constraint in eq. (3.25). We can construct a repulsive potential field that is hyperbolic through the function in eq. (3.23) such that if the quadratic constraint (which indicates the cosine of the separation angle) approaches the cosine of the constraint angle, it generates a higher and higher value of the repulsive reference angular velocity which of course deviates the path from the obstacles. Such a potential has the following form:

$$U_r(\boldsymbol{Q}) = \frac{\alpha}{(\boldsymbol{Q}^T\boldsymbol{M}\boldsymbol{Q} - \cos\theta)^2} \tag{3.30}$$

Supposing the presence of m on-board sensitive instruments and n constrained obstacles, a repulsive potential field can be constructed as the sum of the contributions:

$$U_r(\boldsymbol{Q}) = \sum_{j=1}^{m}\sum_{i=1}^{n} \frac{\alpha}{(\boldsymbol{Q}^T \boldsymbol{M}_i^j \boldsymbol{Q} - \cos\theta_i^j)^2} \tag{3.31}$$

or equivalently:

$$U_r(\boldsymbol{Q}) = \sum_{j=1}^{m}\sum_{i=1}^{n} \frac{\alpha}{(\boldsymbol{Q}^T \bar{\boldsymbol{M}}_i^j \boldsymbol{Q})^2} \tag{3.32}$$

where α is a design parameter used to adjust the relative weighting between the attractive and repulsive potential, usually chosen to be a small value in this case in order to maintain a reasonable value for the control input. By taking the gradient, we obtain the reference value for the repulsive angular velocity. Since the repulsive potential is constructed with unit quaternions, the gradient would give back a 4×1 vector. We can construct a 3×1 reference angular velocity by extracting the vectorial part as follows:

$$\boldsymbol{\omega}_r = \sum_{j=1}^{m}\sum_{i=1}^{n} \frac{4\alpha \, \mathrm{Vec}[\boldsymbol{Q}^T \boldsymbol{M}_i^j]}{(\boldsymbol{Q}^T \boldsymbol{M}_i^j \boldsymbol{Q} - \cos\theta_i^j)^3} \tag{3.33}$$

Exactly as the attractive potential, we can insert the sign of the scalar part of the quaternion in order to take into account the unwinding problem:

$$\boldsymbol{\omega}_r = \sum_{j=1}^{m}\sum_{i=1}^{n} \frac{4\alpha \, sgn(q_0) \mathrm{Vec}[\boldsymbol{Q}^T \boldsymbol{M}_i^j]}{(\boldsymbol{Q}^T \boldsymbol{M}_i^j \boldsymbol{Q} - \cos\theta_i^j)^3} \tag{3.34}$$

3.2.4 Total potential

Superposing the contributions of the attractive and repulsive potentials, we can build the total potential field acting on the satellite which guides it towards the goal avoiding obstacles. We will assume the obstacles are far away from the goal position, so that only the function for $||\boldsymbol{q}_e|| > \rho$ in eq. (3.28) affects the repulsive part. We can then sum the contributions:

$$\boldsymbol{\omega}_{ref} = \boldsymbol{\omega}_a + \boldsymbol{\omega}_r = -\Omega \, sgn(q_{e,0}) \frac{\boldsymbol{q}_e}{||\boldsymbol{q}_e||} + \sum_{j=1}^{m}\sum_{i=1}^{n} \frac{4\alpha \, sgn(q_0) \mathrm{Vec}[\boldsymbol{Q}^T \boldsymbol{M}_i^j]}{(\boldsymbol{Q}^T \boldsymbol{M}_i^j \boldsymbol{Q} - \cos\theta_i^j)^3} \tag{3.35}$$

After defining all the parameters in eq. (3.35), given the value of the reference quaternion $\boldsymbol{Q}_g$ an error quaternion $\boldsymbol{Q}_e$ can be defined and eq. (3.35) gives, step by step, the reference value for the angular velocity which guides the satellite to the reference quaternion by avoiding obstacle along the path.

We then have the two main ingredients for a control law, which are (e.g. with the purpose of aligning the body frame on the inertial frame):

$$\boldsymbol{Q}_{ref} = [1{,}0{,}0{,}0] \quad \text{and} \quad \boldsymbol{\omega}_{ref} \text{ given by (3.35)}$$

3.3 Sliding mode control

We will analize now the control law used in this thesis work. A *sliding mode control* has been chosen: because of the nonlinear nature of the system and the need for robustness against perturbation for an online planning, it is suitable and provides us with asymptotic stability. Following, the general mathematical formulation for the sliding mode control will be introduced and, after that, a slightly modified version including an adaptation law will be applied to our case study.

3.3.1 Mathematical formulation

Sliding mode control is a non-linear variable structure control method. It is one particular approach to robust controller design. Like all robust controllers, it solves the problem of providing the desired performance in the presence of disturbances and/or uncertainties, whose action creates the discrepancy between the real plant and its mathematical model. We'll start by giving the general definition of the methodology and apply it to a basic example to understand better its power. It follows the following approach. Given a generic nonlinear system:

$$\dot{x}(t) = f(x(t), u(t)), \quad x(t) \in \mathbb{R}^n \tag{3.36}$$

the sliding mode control method consists of the two following steps:

- a *sliding variable* is defined, function of the system state-variables:

  $$\sigma(x(t)) \in \mathbb{R}^m \tag{3.37}$$

  and the relative *sliding surface* or *sliding manifold* follows, which is a subset of the state-space of dimention $n - m$ and on which the trajectory of the system in state-space is desired to lie:

  $$\sigma(x(t)) = 0 \tag{3.38}$$

- a *control law* is designed to force the trajectory towards the sliding surface and, once there, keep it close to it until the desired position.

The control law design generates thus two main phases: a *reaching phase* where, starting from the initial condition, the trajectory is brought towards the sliding surface in a finite time and a *sliding phase* where the trajectory, by means of the control action, is forced to *slide* along the sliding surface, well designed with the purpose of crossing the origin of the state-space, providing asymptotic convergence. The phases are well shown in figure 3.9. It is noted that the way the sliding mode is defined and the control law is designed really depends on the structure of the system and eventually on the bounded values of the uncertainties. Moreover, the type of mathematical function that the control action assumes characterizes the type of the sliding mode controller. We have in fact, to cite a couple, a *first order sliding mode control* (*FOSM*), the *super-twisting sliding mode control* (*STW*), etc. The difference consists on the ability of dealing with disturbances which are not bounded or to solve the *chattering effect* coming out of the discontinuous nature of the control law. In order to appreciate how the method works, how it tackles the

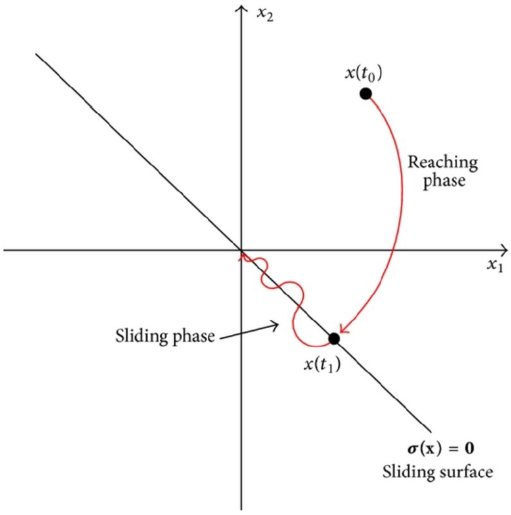

Figure 3.9. Sliding mode control phases.[5]

disturbances and what's the procedure of choosing a suitable sliding variable and control action, let's consider the basic example of a system that describes the single-dimensional motion of a unit mass, also known as a *double integrator*, and apply in the end a control law for a FOSM. Introducing the position and velocity of the mass $x_1 = x$ and $x_2 = \dot{x}$, a state-variable description is[16]:

$$\begin{cases} \dot{x}_1 = x_2 & x_1(0) = x_{10} \\ \dot{x}_2 = u + f(x_1, x_2, t) & x_2(0) = x_{20} \end{cases} \tag{3.39}$$

where u is the control force and $f(x_1, x_2, t)$ the disturbance term, assumed to be bounded, i.e. $|f(x_1, x_2, t)| \leq L > 0$. The problem is to design a feedback control law u that drives the mass to the origin asymptotically, i.e.:

$$u = u(x_1, x_2) : \lim_{t \to \infty} x_1, x_2 = 0$$

The challenge is in the presence of the unknown bounded disturbance $f(x_1, x_2, t)$. What we want is the dynamics to behave, for example, as a linear time-invariant differential equation for the position of the unit mass, i.e., reaching the origin in a finite time:

$$\dot{x}_1 + cx_1 = 0, \;\; c > 0 \tag{3.40}$$

which has the following solution:

$$x_1(t) = x_1(0)e^{-ct} \tag{3.41}$$

of course since $x_2 = \dot{x}_1$, the velocity as well will converge to the origin. In this compensated desired dynamics we observe no effect of the disturbance. In order to achieve this, we can define a sliding variable σ, function of the state-space variables of the system:

$$\sigma(x_1, x_2) = x_2 + cx_1, \;\; c > 0 \tag{3.42}$$

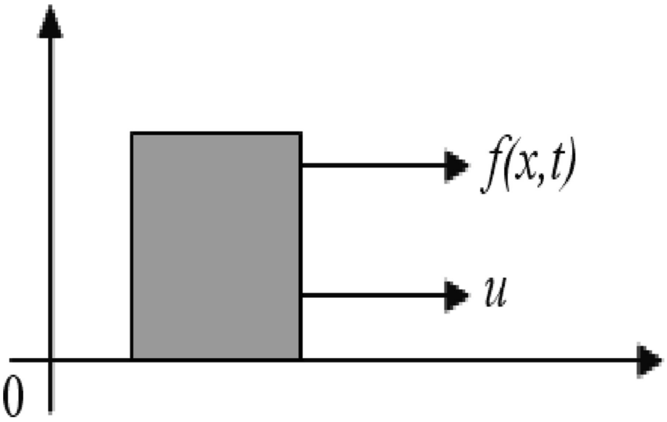

Figure 3.10. Single-dimensional motion of a unit mass.[16]

The sliding manifold or sliding surface is described by the equation:

$$\sigma(x_1, x_2) = 0 \tag{3.43}$$

and it has been designed so that the system in sliding mode reaches the origin; in fact we have from eq. (3.42) a line that crosses the origin in the state-space:

$$\sigma = 0 \implies x_2 = -cx_1$$

After the definition of a suitable sliding variable and sliding surface, a feedback law is designed to bring the trajectory towards the sliding surface in a finite time and, once there, to stay close to it. To make sure than the original system (3.39) reaches asymptotically the origin despite the disturbance, we have to ensure it reaches, moving along the sliding variable, the sliding surface in finite time by means of the control u. We can derive the σ-dynamics using eqs. (3.39) and (3.42):

$$\dot{\sigma} = cx_2 + f(x_1, x_2, t) + u, \quad \sigma(0) = \sigma_0 \tag{3.44}$$

We can use the Lyapunov function method in order to ensure we have asymptotic stability towards the sliding surface $\sigma = 0$. Let's define a candidate Lyapunov function:

$$V = \frac{1}{2}\sigma^2 \tag{3.45}$$

For $\sigma = 0$ to be an equilibrium point about which there's asymptotic stability, the function in eq. (3.45) must satisfy the following conditions:

1. $V(\sigma) = 0, \text{ for } \sigma = 0$
2. $V(\sigma) > 0, \ \forall \ \sigma \neq 0$
3. $\lim_{|\sigma| \to \infty} V = \infty$

4. $\dot{V} < 0, \quad \text{for } \sigma \neq 0$

The first 3 conditions are already satisfied by eq. (3.45). In order to achieve finite-time convergence, the fourth condition can be modified as follows[16]:

$$\dot{V} \leq -\alpha\sqrt{V}, \quad \alpha > 0 \tag{3.46}$$

integrating the differential equation we obtain:

$$V^{1/2}(t) = -\frac{1}{2}\alpha t + V^{1/2}(0) \tag{3.47}$$

and hence $V(t)$ reaches zero in a finite time t_r bounded by:

$$t_r \leq \frac{2V^{1/2}(0)}{\alpha} \tag{3.48}$$

Therefore, a well defined control u can drive the sliding variable σ to zero (to the sliding surface) in a finite time and will keep it a zero thereafter. Let's now calculate the derivative of V:

$$\dot{V} = \frac{\partial V}{\partial \sigma}\frac{\partial \sigma}{\partial t} = \sigma\dot{\sigma} = \sigma(cx_2 + f(x_1, x_2, t) + u) \tag{3.49}$$

Assuming $u = -cx_2 + \nu$:

$$\dot{V} = \sigma(f(x_1, x_2, t) + \nu) = \sigma f(x_1, x_2, t) + \sigma\nu \leq |\sigma|L + \sigma\nu \tag{3.50}$$

by choosing $\nu = -\rho\, sgn(\sigma)$ with $\rho > 0$, we obtain:

$$\dot{V} \leq |\sigma|L - |\sigma|\rho = -|\sigma|(\rho - L) \tag{3.51}$$

Condition (3.46) can be rewritten, taking into account the function in eq. (3.45) as:

$$\dot{V} \leq -\alpha V^{1/2} = -\frac{\alpha}{\sqrt{2}}|\sigma| \tag{3.52}$$

Combining:

$$\dot{V} \leq -|\sigma|(\rho - L) = -\frac{\alpha}{\sqrt{2}}|\sigma| \tag{3.53}$$

This lets us compute the control gain ρ:

$$\rho = L + \frac{\alpha}{\sqrt{2}} \tag{3.54}$$

Finally, a control input u that drives the sliding variable to the sliding surface in a finite time is:

$$u = -cx_2 - \rho\, sgn(\sigma) \tag{3.55}$$

Observation. The condition (3.46) can be written as:

$$\sigma\dot{\sigma} \leq -\frac{\alpha}{\sqrt{2}}|\sigma| \tag{3.56}$$

which is known as the *reachability condition*, which ensures that the trajectory of the system (3.39) is driven towards the sliding surface in a finite time and remains on it thereafter.

The mathematical structure for the control input (3.55) is what characterizes a first-order sliding mode control FOSM. We can notice the presence of the sign function which is discontinuous.

Equivalent control and sliding mode equations

The effect of the sign function in the input control (3.55) is to "average" the effect of the perturbation: the bounded value of the latter is in fact used to the design the control gain ρ. In a pure theorical way though, we could compute the so called *equivalent control* u_{eq}. Once the sliding surface is reached at time t_r, the trajectory remains on it by means of (3.55); this means that[16]:

$$\sigma = \dot{\sigma} = 0 \implies cx_2 + f(x_1, x_2, t) + u = 0 \iff u_{eq} = -cx_2 - f(x_1, x_2, t) \tag{3.57}$$

It ensures that the trajectory stays on the sliding surface ($\dot{\sigma} = 0$) but of course it's not the actual control applied to the system; nevertheless, as mentioned, it averages the effect of the high frequency of the switching control. This procedure is useful to derive the system's compensated dynamics in the sliding mode. The equivalent control u_{eq} drives the system's dynamics in sliding mode. Substituting we obtain[16]:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \underbrace{(-cx_2 - f(x_1, x_2, t))}_{u_{eq}} + f(x_1, x_2, t) \end{cases} \implies \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -cx_2 \end{cases} \implies \begin{cases} \dot{x}_1 = -cx_1 \\ \dot{x}_2 = -cx_2 \end{cases} \tag{3.58}$$

The system's compensated dynamics in the sliding mode is:

$$\begin{cases} \dot{x}_1 = -cx_1 \\ x_2 = -cx_1 \end{cases} \tag{3.59}$$

The fact that the compensated dynamics is of first-order, less than the system's one, is due to the fact that the dynamics in (3.59) describes the "slow" motion only, and it disregards the effect of the high-frequency switching control. As already mentioned, the compensated dynamics do not depend on the bounded disturbance $f(x_1, x_2, t)$; however, the design of the gain control ρ takes into account its upper limit L.[16]

Chattering elimination

We previously mentioned a high-frequency switching control. It is indeed correct because of the presence of the sign function in (3.55). What this provokes is a "zigzag" motion of small amplitude and high frequency in the state-space. This is due to the high-frequency switch of the input control, which, based on the sign of the σ variable, changes sign, switching from $-\rho$ to ρ. This effect given by the imperfection of the sign function implementation is called *chattering*. In many practical control systems, including our case study, it is important to avoid the chattering since the actuation system cannot act back and forth with high frequency. To avoid this, it is possible to approximate the sign function by some continuous function. A function that approximate it well is:

$$sign(\sigma) \approx \frac{\sigma}{|\sigma| + \varepsilon} \tag{3.60}$$

It can be observed that:

$$\lim_{\varepsilon \to 0} \frac{\sigma}{|\sigma| + \varepsilon} = sign(\sigma)$$

the parameter ε is chosen as a compromise between mantaining robustness and ensuring a smooth control action. The input control can then be expressed as:

$$u = -c\,x_2 - \rho\frac{\sigma}{|\sigma| + \varepsilon} \tag{3.61}$$

This function is part of the so-called *saturation functions*, since they bound the input function with finite upper and lower values, like the sign function, but providing a smooth transition at zero. Another saturation function used, for reference, is the hyperbolic tangent:

$$sign(\sigma) \approx \tanh(k_\sigma \sigma) \tag{3.62}$$

As with the sigmoid function, it can be observed that:

$$\lim_{k_\sigma \to \infty} \tanh(k_\sigma \sigma) = sign(\sigma)$$

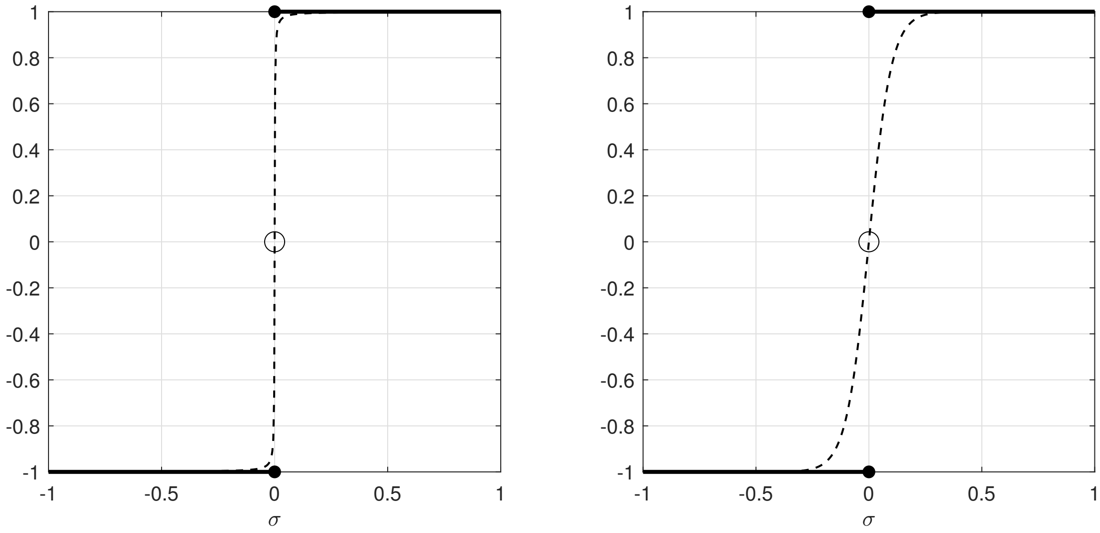

Figure 3.11. Approximation of the sign function. *Left*: sigmoid, *right*: hyperbolic tangent.

Super-twisting sliding mode control

In many cases a high frequency switching control is impractical and continuous control is a necessity. A different form of the sliding variable can be chosen. It can be shown that the following form:

$$\begin{cases} u = c|\sigma|^{1/2} sign(\sigma) + w \\ \dot{w} = b\, sign(\sigma) \end{cases} \tag{3.63}$$

is continuous since both $c|\sigma|^{1/2}sign(\sigma)$ and the term $w = b\int sign(\sigma)dt$ are continuous, since the high-frequency switching term $sign(\sigma)$ is "hidden" under the integral. This

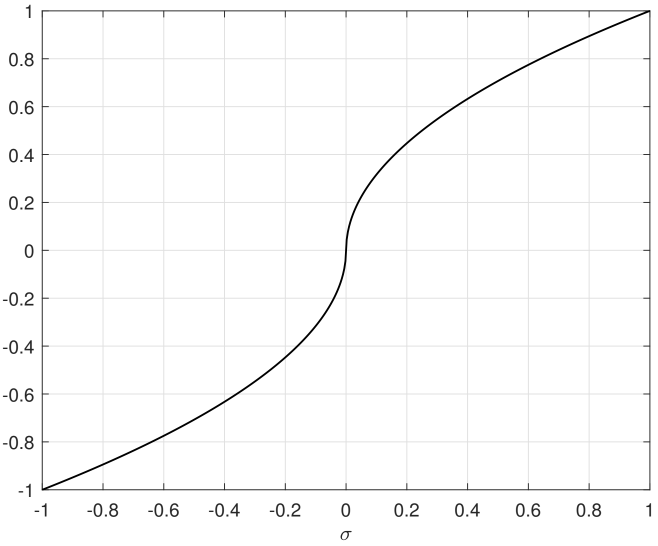

Figure 3.12. Continuity of the term $c|\sigma|^{1/2}sign(\sigma)$.

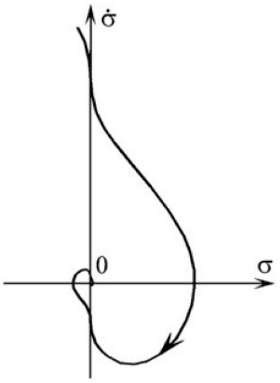

Figure 3.13. Typical phase portrait with a super-twisting sliding mode control.[16]

is the *super-twisting sliding mode control* which is a *second-order sliding mode control* method.[16] Figure 3.12 shows the absence of discontinuity in the first term of equation (3.63). In figure 3.13 a typical phase portrait with a super-twisting sliding mode control is shown. As a regular SMC, it can be shown it drives the state onto the sliding surface in finite time and forces it to lay onto it during the sliding phase. The advantage of not having a discontinuous control action is reflected in its capability to deal with the unbounded nature of external disturbances and, more importantly, to make sure the stationary state errors go to zero in finite time instead of asymptotically.

3.3.2 Attitude control with adaptive sliding mode

Let's go back to our case study. In the previous subsection, we've introduced and described the sliding mode control and its effect on a system affected by unknown bounded disturbances, being able to provide asymptotic stability nevertheless, proving its characteristic of robustness. Through a basic example, we introduced the methodology for the implementation of a first-order sliding mode control (FOSM) and also described briefly the super-twisting sliding mode control (STW) and listed its advantages. In this simulation, however, a FOSM has been used for a simpler computational demand but, even so, it gives us good results nevertheless thanks to the coupling with the APF algorithm and the planning technique (3.17). In addition, we will implement a variable form of the first-order sliding mode control: we'll call it *adaptive sliding mode*. In order to understand it, let's first go through the steps of choosing the sliding variable and the input control.

Sliding variable

The algorithm to implement needs to be of course general: it has to work with different values of reference quaternion and angular velocity; hence, the system does not need to be driven to zero always. The goal of course is, once defined the reference values, to drive to zero the *errors* of both state-variables, namely:

$$\boldsymbol{q}_e = \mathrm{Vec}[\boldsymbol{Q}^*_{ref} \otimes \boldsymbol{Q}], \quad \boldsymbol{\omega}_e = \boldsymbol{\omega} - \boldsymbol{\omega}_{ref}$$

Therefore, we can build up a sliding variable as a linear combination of the error quaternion and error angular velocity:

$$\sigma = \lambda\, \boldsymbol{q}_e + \boldsymbol{\omega}_e \tag{3.64}$$

it has the general form of a manifold in the state-space $(\boldsymbol{q}_e, \boldsymbol{\omega}_e)$ and the corresponding sliding surface $\sigma = 0$ corresponds to a straight line in the same space which drives the system's dynamics to null values for both the errors.

Observation. Recall that, from the kinematics equation (2.38), the angular velocity $\boldsymbol{\omega}$ does in fact contribute to the determination of the derivative of the quaternion $\boldsymbol{Q}$. Equation (3.64) can then be thought of as a conventional sliding variable with the error dynamics e:

$$\sigma = \dot{e} + \lambda e$$

whose sliding surface corresponds to the stable dynamics of a first-order system, just like the compensated dynamics for the unit mass system introduced in the general mathematical formulation in (3.40).

Input control

As mentioned, the coupling with the APF algorithm ends up giving good results even with the implementation of a first-order sliding mode control with chattering elimination. The input control will then be of the form:

$$\boldsymbol{u} = -Ksat(\sigma) = -Ksat(\lambda\boldsymbol{q}_e + \boldsymbol{\omega}_e) \tag{3.65}$$

Lambda adaptation

What about the *adaptive*? It is in the parameter λ. An adaptive control law will be implemented so that, when the error on the quaternion is large, the input control will only act on the reference angular velocity: it will be to all intents a guidance law releasing some computational demand as well; once the error quaternion gets smaller and smaller, the input control will change its form and act on both variables, guaranteeing a smoother response at the vicinity of the goal. This method can be implemented acting on the form of the parameter λ, which can't be a constant value anymore but a function of time and error quaternion. The values of λ are bounded in the interval $[\lambda_l, \lambda_u]$, which is chosen as $[0, \lambda_u]$, since what we want the form of σ to be :

$$\begin{cases} \sigma = \boldsymbol{\omega}_e & \text{if } \lambda = 0 \\ \sigma = \lambda_u \boldsymbol{q}_e + \boldsymbol{\omega}_e & \text{if } \lambda = \lambda_u \end{cases} \tag{3.66}$$

The differential equation that describes the time evolution of lambda for our case is chosen as follows:

$$\begin{cases} \dot{\lambda} = -\alpha^*(\lambda - \lambda_u) + \beta ||\boldsymbol{q}_e|| \\ \lambda(0) = \lambda_u \end{cases} \tag{3.67}$$

where, we will see in the results chapter, $\beta = f(\alpha^*)$, $\alpha^* > 0$ and $\beta < 0$. The general solution has the form:

$$\lambda(t) = e^{-\alpha^* t} \left[\int e^{\alpha^* t} (\beta ||\boldsymbol{q}_e|| + \alpha^* \lambda_u) dt \right] \tag{3.68}$$

Of course, eq. (3.67) must be solved numerically since $||\boldsymbol{q}_e||$ does not have a pre-defined dependence on time. Imposing the bounded values for lambda $[0, \lambda_u]$ in the integration procedure in a numerical calculus environment, we obtain the evolution wanted for lambda, which is zero when $||\boldsymbol{q}_e||$ is large (at the beginning of the simulation) and a defined value λ_u when $||\boldsymbol{q}_e||$ is close to zero (closer to the goal). Finally, $\boldsymbol{u}$ is the control torque that enters the actuation system model, which we called $\boldsymbol{M}_C$ in eq. (2.70).

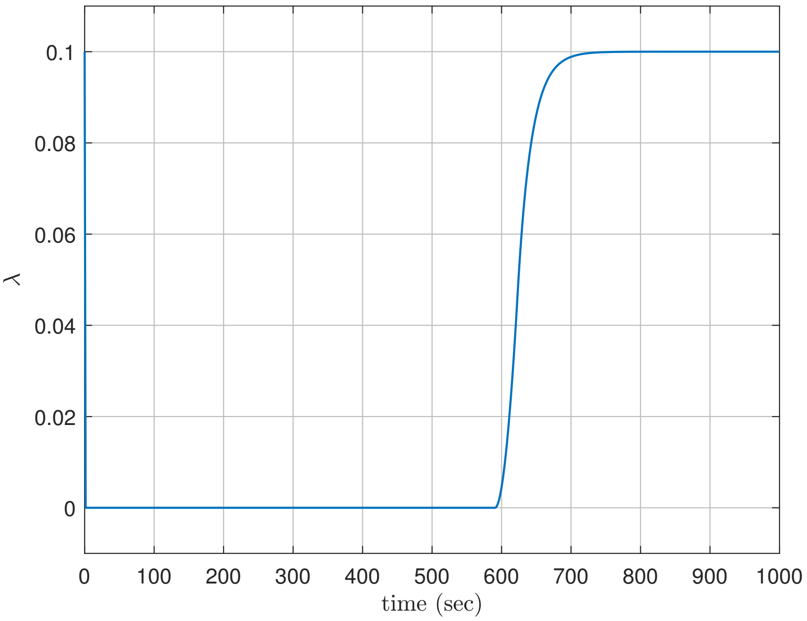

Figure 3.14. Time evolution of $\lambda(t)$ obtained in the simulation.

Chapter 4

Simulation and results

In this chapter simulation parameters will be defined and results of two simulations presented: one with the unwinding phenomenon and one without. Geometrical data for the satellite are chosen, i.e. inertia data and all the parameters needed from the actuation system. Later on, guidance and control parameters are chosen, i.e. the parameters of the potential fields and the ones defining the sliding mode controller. After defining initial condition for both simulations, results will be produced and subsequently presented and commented.

4.1 Satellite parameters

The satellite reference of the simulation is the *Demeter* satellite, which was a micro-satellite devoted to the investigation of the ionospheric disturbances due to seismic and volcanic activity. It's a small satellite, and its actuation system model is used to simulate the attitude dynamics in the presence of obstacles. The satellite model is characterized by defining its inertial matrix I, used for the dynamics modeling, and the actuation system performance, i.e. the maximum torque and maximum angular momentum achievable. The values T_{max} and h_{max} in table 4.1 are used in the actuation system model as limit values of

Parameters	**Value**	**Units**	
I_{xx}	30.012	$kg \cdot m^2$	Inertia moment around x
I_{yy}	30.012	$kg \cdot m^2$	Inertia moment around y
I_{zz}	40.012	$kg \cdot m^2$	Inertia moment around z
I_{xy}	-3	$kg \cdot m^2$	Cross inertia moment x-y
I_{xz}	0	$kg \cdot m^2$	Cross inertia moment x-z
I_{yz}	-2	$kg \cdot m^2$	Cross inertia moment y-z
T_{max}	0.005	$N \cdot m$	Maximum torque produced by reaction wheel
h_{max}	0.12	$kg \cdot m^2/s$	Maximum angular momentum of reaction wheel

Table 4.1. Satellite parameters used in the simulator.

respectively the satellite torque and angular momentum commands. The implementation of such limits in Simulink environment can be done using the *saturation block*. The actuation system of the Demeter satellite is modeled as a second-order low-pass filter, with the following transfer function:

$$G(s) = \frac{1.214s + 0.7625}{s^2 + 2.4s + 0.7625} \tag{4.1}$$

The actuation control input (torque), after being expressed in the cluster reference frame by eq. (2.70), is fed to the saturation block with T_{max} and $-T_{max}$ as limits. Afterwards, it is filtered through the transfer function in eq. (4.1). By integrating the torque, we get the angular momentum of the reaction wheel, which can be in turn saturated with h_{max} and $-h_{max}$ and finally return to the satellite frame with eq. (2.70) again.

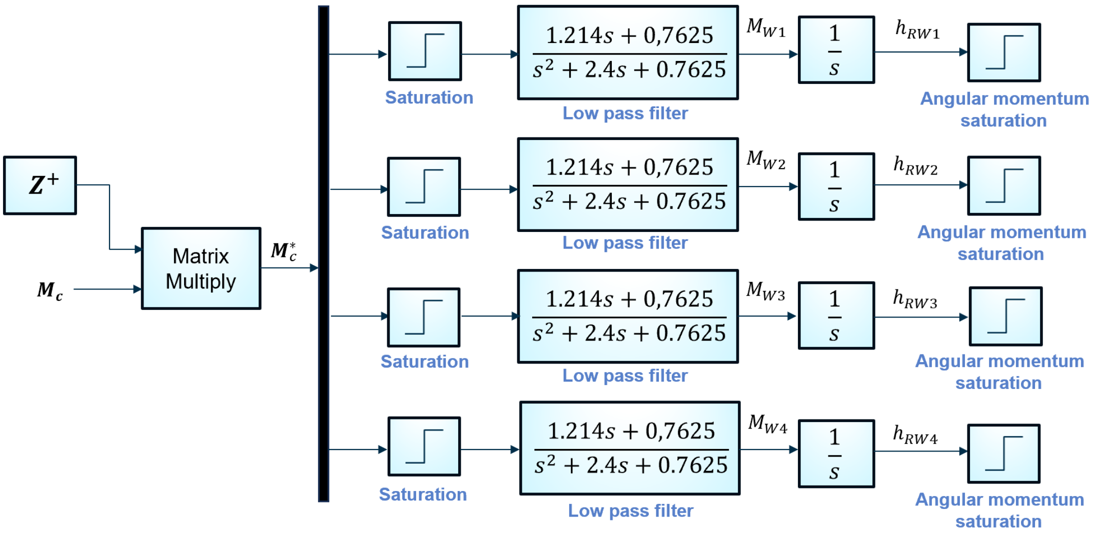

Figure 4.1. Saturation and filtering of the reaction wheels cluster in the simulator.

4.2 Simulation without unwinding

Let's first list all the parameters needed for the first simulation, which doesn't account for the unwinding phenomenon. These are a set of initial conditions and all the parameters for the guidance and control algorithm, whose references like obstacles position and initial quaternion vary between the simulations.

4.2.1 Initial conditions

Initial conditions have to be specified for the simulation to work. As specified in the mathematical model of attitude representation, we will use quaternions for computational reasons, but, for the sake of simplicity, we will choose a set of initial Euler's angles and from there calculate the initial unit quaternion for the calculations. Hence, an initial angular velocity vector is needed for the dynamics integration, namely eq. (2.62) which we recall here for a better reading:

$$\boldsymbol{I}\dot{\boldsymbol{\omega}}_B + \boldsymbol{\omega}_B \times (\boldsymbol{I}\boldsymbol{\omega}_B + \boldsymbol{I}_{RW}\boldsymbol{\omega}_{RW}) = \boldsymbol{M}_B$$

and, after that, with the initial quaternion, the kinematic equations can be integrated, namely eq. (2.38) which we recall:

$$\dot{\boldsymbol{Q}} = \frac{1}{2}\boldsymbol{Q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_B \end{bmatrix}$$

Also initial value for the integrator in fig. 4.1 representing the initial reaction wheel's angular momentum is shown in table 4.2.

Parameters	**Value**	**Units**	
ϕ_0	80	deg	Initial roll angle
θ_0	120	deg	Initial pitch angle
ψ_0	-100	deg	Initial yaw angle
ω_{x_0}	0	rad/s	Initial roll rate
ω_{y_0}	0	rad/s	Initial pitch rate
ω_{z_0}	0	rad/s	Initial yaw rate
q_0	0.6726	–	Scalar component of initial quaternion
q_1	-0.3016	–	First vector component of initial quaternion
q_2	0.6726	–	Second vector component of initial quaternion
q_3	0.0644	–	Third vector component of initial quaternion
$\boldsymbol{Q}_{ref}$	[1,0,0,0]	–	Reference quaternion
h_{rw_0}	0	$kg \cdot m^2/s$	Initial angular momentum of the reaction wheels

Table 4.2. Initial conditions for the simulation without unwinding.

4.2.2 Guidance and control algorithm parameters

All the algorithms of guidance and control contain parameters and gains in order to solve their purpose. The tuning of such parameters has been accomplished by trial and error. As regards the guidance algorithm, we need to specify all the parameters of the attractive potential field, such as Ω and ρ and the ones of the repulsive potential, which are the inertial position of the obstacles and the position in the body frame of the instrument, as well as the gain parameter α and the constraint angle θ_m. For the sliding mode controller, in order to integrate the sliding gain λ, we need to specify the parameters α^*, β, λ_u and the controller gain K. All the parameters for the guidance algorithm used in the simulation are presented in table 4.3. In table 4.4 we can find all the parameters needed for the sliding mode controller tuning. In particular, in the sliding mode control section we mentioned that $\beta = f(\alpha^*)$; it has in fact been chosen to be $\beta = -\alpha\lambda_u/\rho$.

Parameters	Value	Units	
Ω	$3 \cdot 10^{-3}$	rad/s	Upper limit of reference attractive angular velocity
ρ	$5 \cdot 10^{-2}$	–	Attractive field transition value
x_1	0.6634	m	x-component of first obstacle's position vector
y_1	0.7297	m	y-component of first obstacle's position vector
z_1	0.1658	m	z-component of first obstacle's position vector
x_2	0.7036	m	x-component of second obstacle's position vector
y_2	0.6920	m	y-component of second obstacle's position vector
z_2	-0.1615	m	z-component of second obstacle's position vector
x_3	0.5774	m	x-component of instrument's position vector
y_3	0.5774	m	y-component of instrument's position vector
z_3	0.5774	m	z-component of instrument's position vector
α	$6 \cdot 10^{-4}$	–	Gain of repulsive potential function
θ_m	9	deg	Constraint angle

Table 4.3. Guidance algorithm parameters for the simulation without unwinding.

Parameters	Value	Units	
α^*	0.05	–	Parameter for the adaptive mode of λ
β	-0.1	–	Parameter for the adaptive mode of λ
λ_u	0.1	–	Initial value of λ
K	$5 \cdot 10^{-3}$	–	Control gain

Table 4.4. Sliding mode control algorithm parameters for the simulation without unwinding.

4.3 Simulation with unwinding

Here all the parameters will be listed for the case with the unwinding. Let's recall that only the initial attitude (and thus initial quaternion) and the inertial position vectors for the obstacles change. All the gains of the guidance and control algorithms are unchanged.

4.3.1 Initial conditions

As we said, we will present here a similar table for initial conditions, with the only difference in the initial attitude; the initial angular velocity and reference quaternion stay the same. Note in table 4.5 the minus sign of the component q_0 of the quaternion. Even though the reference stays [1,0,0,0], the guidance and control algorithm deal with it and make the simulation and final attitude converge to the quaternion $[-1,0,0,0]$, which physically represent the same attitude as [1,0,0,0], converging then to the right attitude but in a faster and "smarter" way.

Parameters	**Value**	**Units**	
ϕ_0	40	deg	Initial roll angle
θ_0	60	deg	Initial pitch angle
ψ_0	60	deg	Initial yaw angle
ω_{x_0}	0	rad/s	Initial roll rate
ω_{y_0}	0	rad/s	Initial pitch rate
ω_{z_0}	0	rad/s	Initial yaw rate
q_0	-0.6193	–	Scalar component of initial quaternion
q_1	0.4914	–	First vector component of initial quaternion
q_2	0.2588	–	Second vector component of initial quaternion
q_3	0.5550	–	Third vector component of initial quaternion
$\boldsymbol{Q}_{ref}$	[1,0,0,0]	–	Reference quaternion
h_{rw_0}	0	$kg \cdot m^2/s$	Initial angular momentum of the reaction wheels

Table 4.5. Initial conditions for the simulation with unwinding.

4.3.2 Guidance and control algorithm parameters

For the guidance algorithm different positions for the obstacles have been chosen in order to place it along the free trajectory to activate the guidance effect. The control parameters are the same as the case without the unwinding phenomenon, shown in table 4.4.

Parameters	**Value**	**Units**	
Ω	$3 \cdot 10^{-3}$	rad/s	Upper limit of reference attractive angular velocity
ρ	$5 \cdot 10^{-2}$	–	Attractive field transition value
x_1	0.7771	m	x-component of first obstacle's position vector
y_1	0.4989	m	y-component of first obstacle's position vector
z_1	0.3837	m	z-component of first obstacle's position vector
x_2	0.6914	m	x-component of second obstacle's position vector
y_2	0.5416	m	y-component of second obstacle's position vector
z_2	0.4782	m	z-component of second obstacle's position vector
x_3	0.5774	m	x-component of instrument's position vector
y_3	0.5774	m	y-component of instrument's position vector
z_3	0.5774	m	z-component of instrument's position vector
α	$6 \cdot 10^{-4}$	–	Gain of repulsive potential function
θ_m	9	deg	Constraint angle

Table 4.6. Guidance algorithm parameters for the simulation with unwinding.

4.4 Simulation parameters

In the end, we specify all the simulation parameters, such as the simulation stop time T_f, the step size Δt and the type of solver. In table 4.7 basic simulation parameters are shown. A fixed-step solver has been chosen, in particular a 4-th order Runge-Kutta method has been used. Let's recall briefly the Runge-Kutta method. It is a family of numerical methods for integrating ordinary differential equations (also known as ode4 in various numerical calculus softwares). Here the 4-th order method has been used. Given an initial value differential problem:

Parameters	Value	Units	
T_f	1000	s	Stop time of the simulation
Δt	0.01	s	Time step of the simulation
Solver type	Runge-Kutta 4	–	Type of solver used in the simulation.

Table 4.7. Simulation parameters.

$$\dot{y} = f(t, y), \quad y(t_0) = y_0 \tag{4.2}$$

with t_0 and y_0 known values, defining $t_{n+1} = t_n + \Delta t$ and $y_{n+1} = y(t_{n+1})$, the general discretization of the Runge-Kutta method is the following:

$$y_{n+1} = y_n + \sum_{i=1}^{s} \Delta t \, b_i k_i \tag{4.3}$$

where:

$$\begin{cases} k_1 = f(t_n, y_n) \\ k_2 = f(t_n + c_2\Delta t, y_n + \Delta t(a_{21}k_1)) \\ k_3 = f(t_n + c_3\Delta t, y_n + \Delta t(a_{31}k_1 + a_{32}k_2)) \\ \vdots \\ k_s = f(t_n + c_s\Delta t, y_n + \Delta t(a_{s1}k_1 + a_{s2}k_2 + \cdots + a_{s,s-1}k_{s-1})) \end{cases} \tag{4.4}$$

In order to choose a particular type of Runge-Kutta method, the number s of terms and the parameters a_{ij} and b_i have to be chosen. All the parameters can be grouped in a tableau, known as *Butcher array*:

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & & \ddots & & \\ c_s & a_{s1} & a_{s2} & \dots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \dots & b_{s-1} & b_s \end{array}$$

A 4-th order Runge-Kutta method has the following parameters:

$$\begin{cases} k_1 = f(t_n, y_n) \\ k_2 = f(t_n + \dfrac{\Delta t}{2}, y_n + \dfrac{1}{2}k_1\Delta t) \\ k_3 = f(t_n + \dfrac{\Delta t}{2}, y_n + \dfrac{1}{2}k_2\Delta t) \\ k_4 = f(t_n + \Delta t, y_n + k_3\Delta t) \end{cases} \tag{4.5}$$

with the following Butcher array:

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

so that:

$$y_{n+1} = y_n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{4.6}$$

Once set, the simulation runs successfully providing us with the main results which will be presented in the next section.

4.5 Results

In this section the main results of the simulations are shown. In order to appreciate the satellite dynamics, we need to have a look at the time evolution of the unit quaternion and check that it reaches the reference quaternion $\boldsymbol{Q}_{ref}$ = [1,0,0,0] or [−1,0,0,0] for the unwinding case, and the time evolution of the angular velocity checking that it eventually goes to zero for both cases.

4.5.1 Case without unwinding

Since we chose such a value for the reference quaternion, we can look at the time evolution of the quaternion error $\boldsymbol{Q}_e$ and check that it reaches the same value, *equivalent* to the zero. Figure 4.2 shows that the satellite successfully reaches the desired attitude given by the inertial reference of frame: it starts from the initial values of the quaternion and reaches the reference quaternion, both specified in table 4.2. We can notice from figure 4.2 the presence of the obstacles; in fact the vector components modify their heading to zero at around $t = 400\,s$ until overshooting a bit at around $t = 650\,s$. This is given by the control action wanting to reduce to zero the norm of the error quaternion, which recall is the main ingredient of the attractive field and as well the sliding mode control once close to the objective. Figure 4.3 shows the time evolution of the norm of the error quaternion. We notice the overshoot in order to avoid the obstacle and the final convergence to zero. Let's see the angular velocity. Figure 4.4 shows the time evolution of the angular velocity

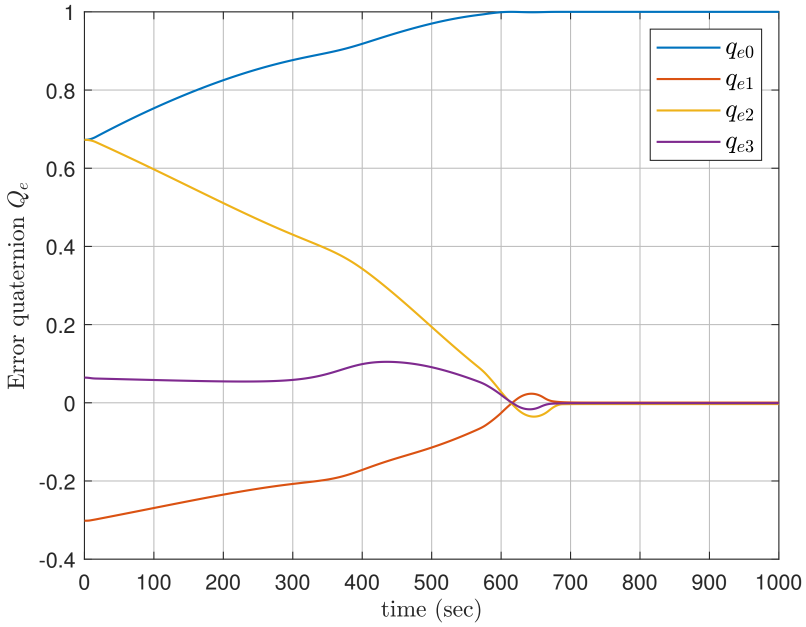

Figure 4.2. Time evolution of the error quaternion Q_e without unwinding.

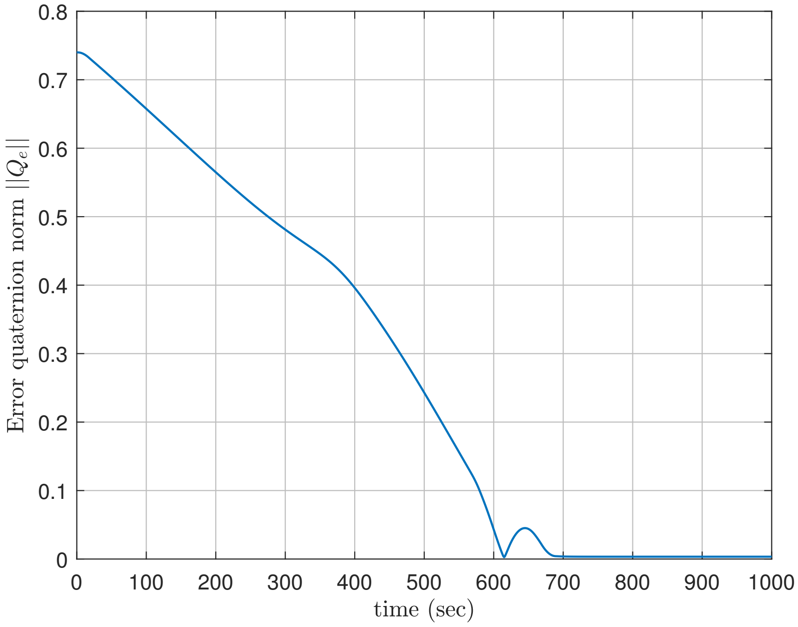

Figure 4.3. Time evolution of the norm of the error quaternion q_e without unwinding.

vector. It follows the evolution dictated by the reference value given by the guidance algorithm until converging to zero once reached the desired attitude. Here as well we can

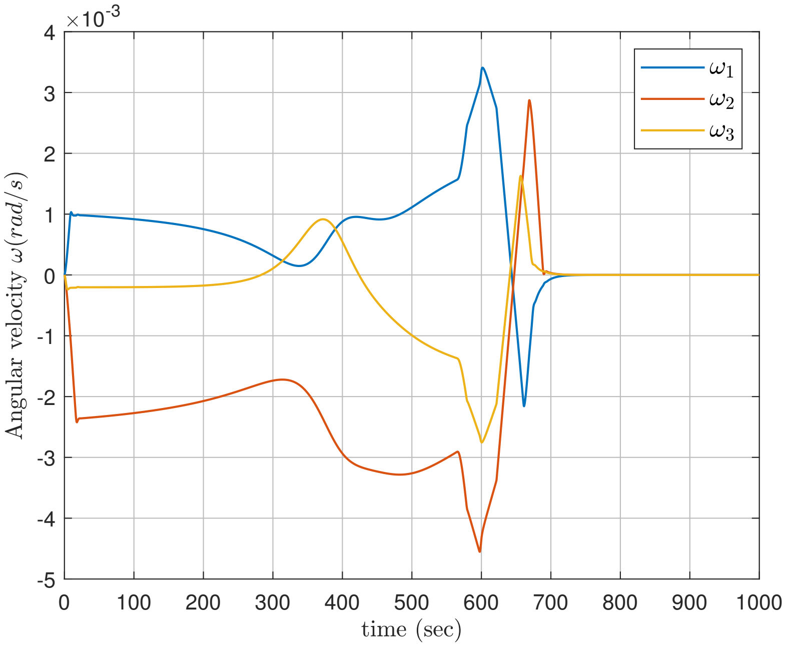

Figure 4.4. Time evolution of the angular velocity ω without unwinding.

notice the slight overshoot but, overall, the saturation phenomenon is never encountered for the absence of constant values of $\boldsymbol{\omega}$ as time goes by. What about the actuation system? Figure 4.5 shows the time evolution of the control torque acting on the satellite by means of the actuation system. The values plotted are those indicated by $\boldsymbol{M}_{RW}$ in eq. (2.70), which is the control action filtered after being represented in the cluster reference of frame. At the same time instances we find again the action at the overshoot which makes the satellite stick to the reference values despite the obstacles' presence. Moreover, we can notice the torque values stay below the maximum value given by T_{max} in table 4.1 (and above $-T_{max}$ of course), without saturation.

How can we know that the attitude constraint has actually been respected? We can recall the definition of the quadratic constraint expressed in eq. (3.25); let's recall it here for simplicity:

$$\boldsymbol{Q}^T \bar{\boldsymbol{M}}_i^j \boldsymbol{Q} < 0$$

which embodies the constraint stated in eq. (3.20):

$$\boldsymbol{x} \cdot \boldsymbol{y}_I < \cos \theta_m$$

with θ_m specified in table 4.3, in such a way that the quaternion $\boldsymbol{Q}$ comes in and can be used in the definition of the repulsive potential field and its gradient (3.34). Hence, by plotting the time evolution of the quadratic constraints for each obstacle, we can check and make sure they stay negative throughout the whole simulation. Figure 4.6 shows the time evolution of the quadratic constraints. They both start from negative and higher absolute values meaning that the angle separating the instrument and the obstacle at the

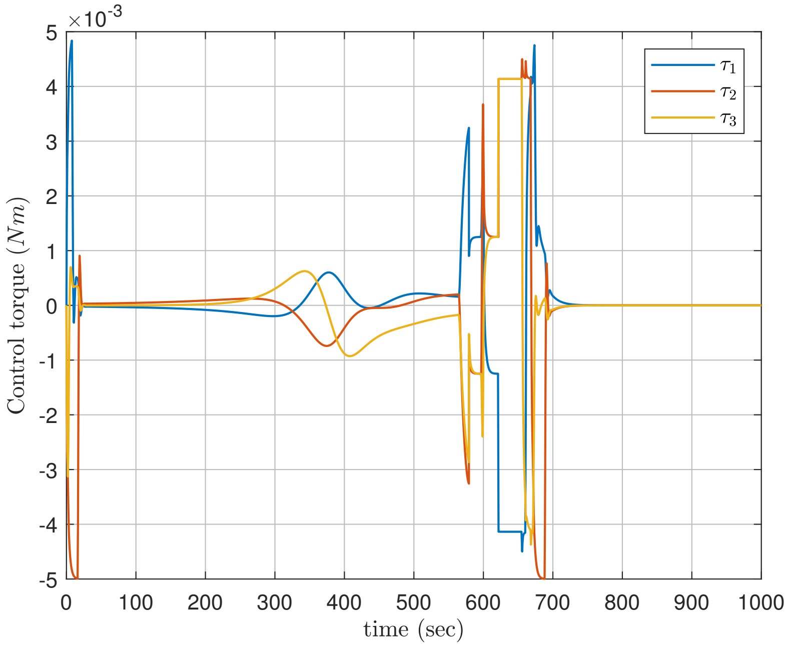

Figure 4.5. Time evolution of the control torque M_{RW} without unwinding.

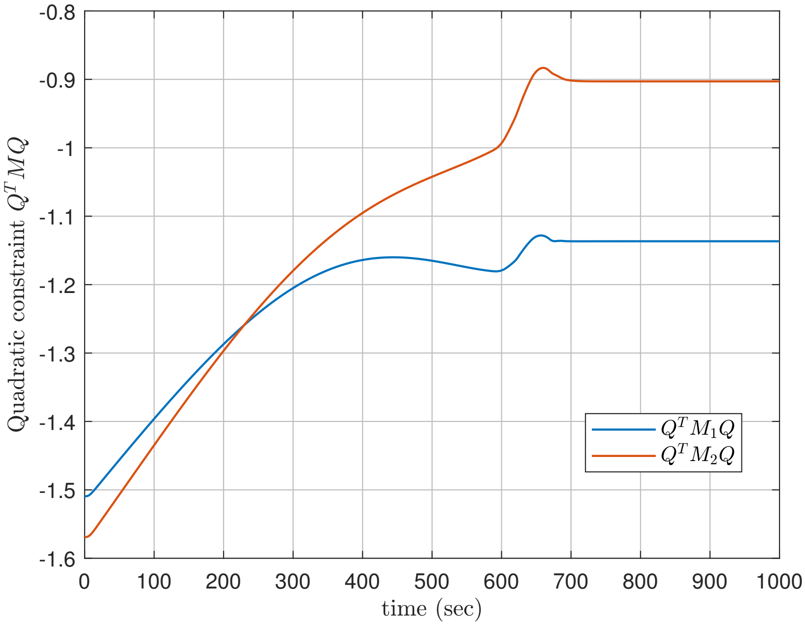

Figure 4.6. Time evolution of the quadratic constraints $Q^T \bar{M}_i Q$ without unwinding.

start is greater than the constraint angle. As time evolves and the satellite attitude gets closer to the obstacles, the separation angle decreases and its cosine takes on higher and higher values. Of course it's not a linear growth since the avoidance maneuver modifies the separation angle, as figure 4.6 shows: it grows again for a bit during the avoidance maneuver until stabilizing at a constant separation once reached the desired attitude. The different values for the two obstacles depend on their position in the inertial space, the constraint angle θ_m and the guidance gain α^*. By working on the gain parameter, a less "aggressive" algorithm can be found, allowing the instrument to get closer to the obstacle; this of course depends on the instrument itself and its capability of tolerating small separation angles.

4.5.2 Case with unwinding

The second simulation deals with the unwinding problem. The relative initial attitude with respect to the reference attitude is such that a standard guidance and control algorithm based on APF would strickly make the quaternion to converge to [1,0,0,0], even though the "smart", most efficient rotation is the one obtained along the same direction but opposite verse, taking less time and less fuel; this case corresponds to the convergence to the final quaternion $[-1,0,0,0]$. This is done, as we saw in the mathematical formulation of APF, by considering the sign of the scalar part of the error quaternion since the start. We'll now list the same physical quantities and show that the right quaternion and the right angular velocity are reached, as well as the quadratic constraint is respected.

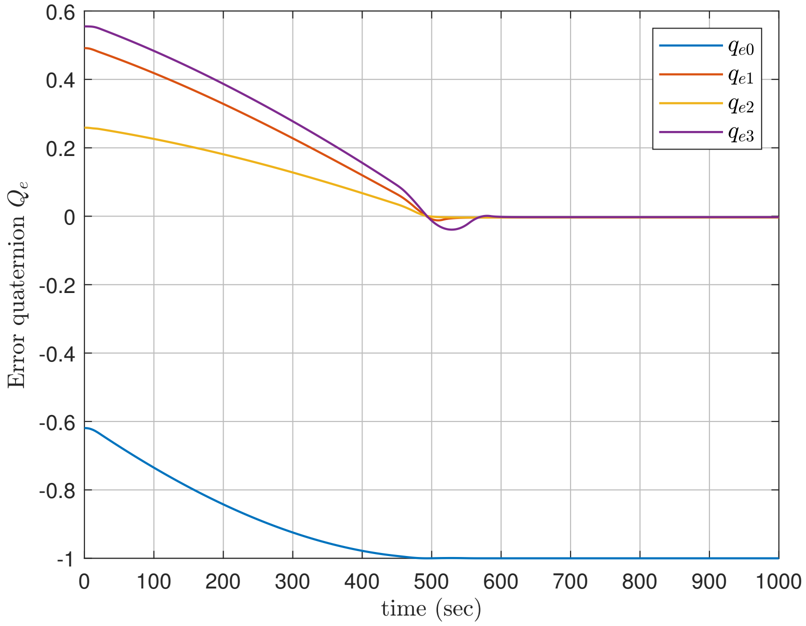

Figure 4.7. Time evolution of the error quaternion Q_e with unwinding.

In figure 4.8 we can notice a small but noticeable stationary state error. This happens

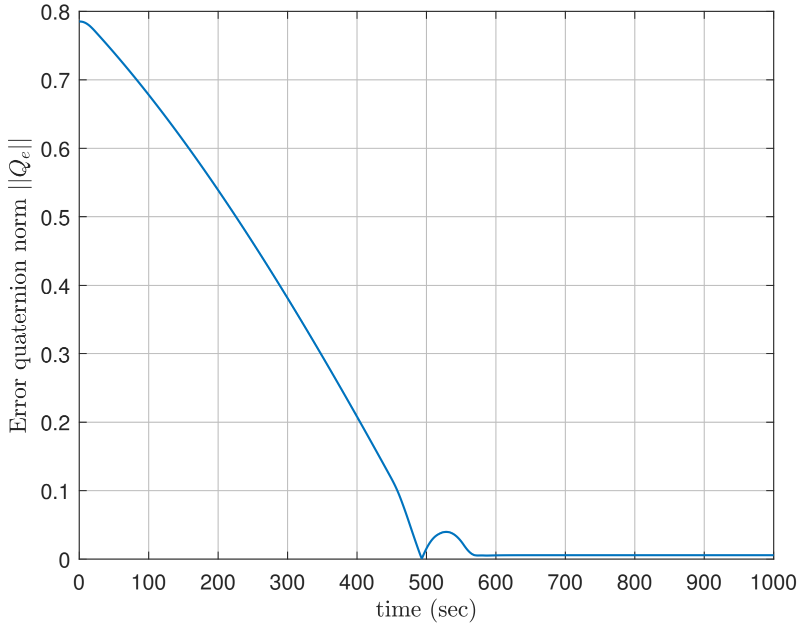

Figure 4.8. Time evolution of the norm of the error quaternion q_e with unwinding.

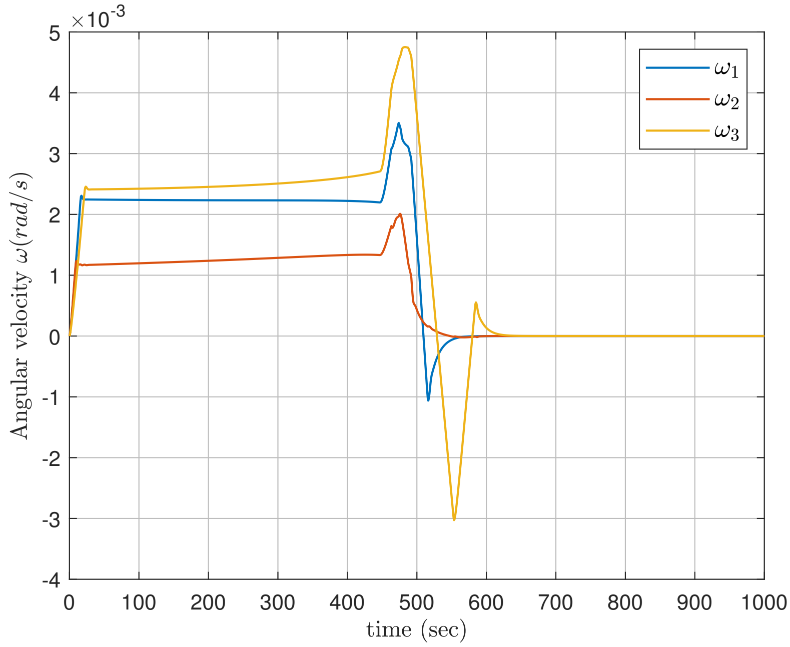

Figure 4.9. Time evolution of the angular velocity ω with unwinding.

because of the order of the sliding mode control we chose, which we remind is a FOSM, in order to keep the computational demand low. It can be adjusted by raising the order of the control model, for example with a super-twisting algorithm, in future works.

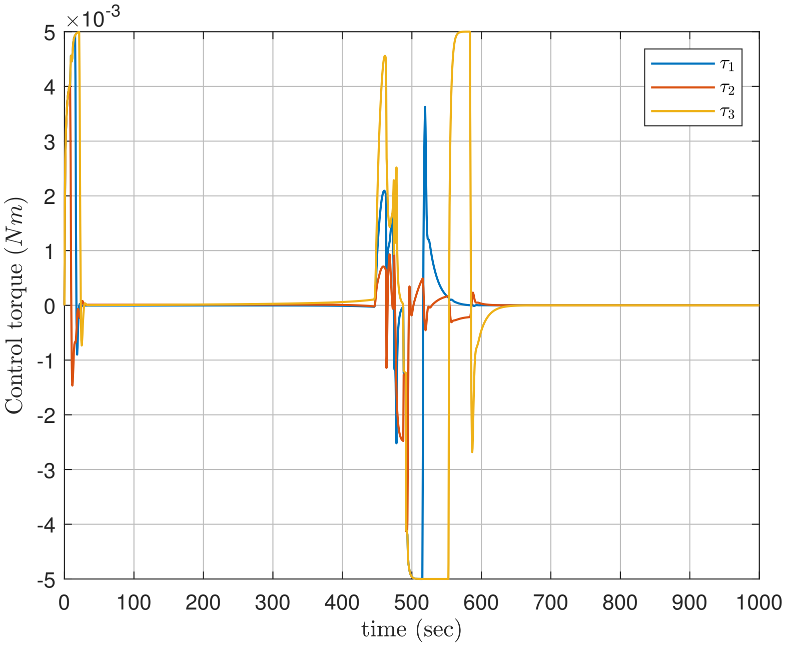

Figure 4.10. Time evolution of the control torque M_{RW} with unwinding.

The different evolution in figure 4.11 depends on the relative position of instrument and obstacles. Since the obstacles' positions had to be changed because of the different initial attitude, the evolution of the separation angle can take different trajectory; the main important fact is that it stays below the zero as the attitude constraint described by the quadratic constraint says. Again, different values for the guidance and control gains and/or the constraint angle can be chosen based on the payload requirements.

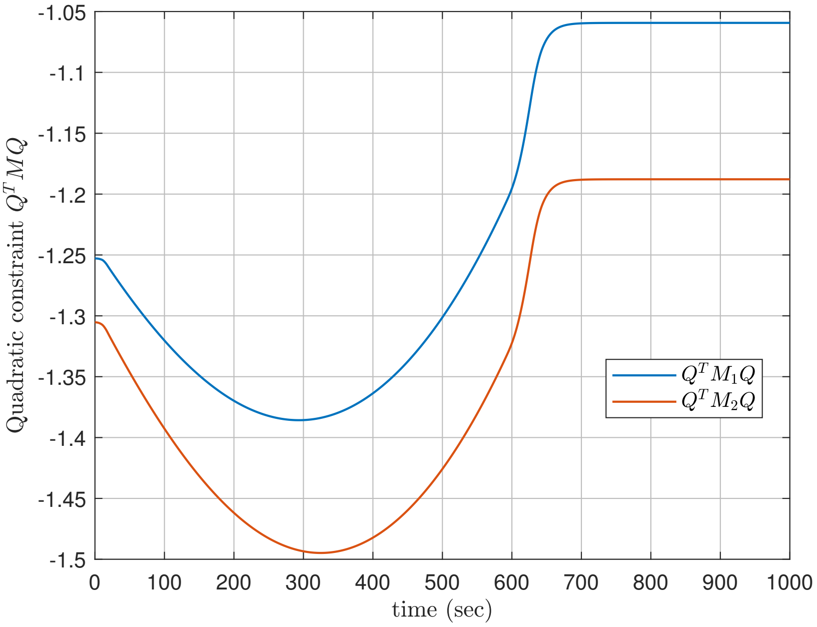

Figure 4.11. Time evolution of the quadratic constraints $Q^T \bar{M}_i Q$ with unwinding.

Chapter 5

Conclusions and future works

The simulations have shown that a guidance algorithm based on the Artificial Potential Field (APF) method can successfully solve the main problem of attitude constraints during a reorientation maneuver, avoiding damage to sensitive payloads. The method is used a lot to deal with problems concerning positional dynamics, not only terrestrial applications but also space ones: APF can be used as a guidance algorithm for on-orbit rendezvous and docking maneuver. A mathematical model describing the relative position dynamics is used by integration of the *Clohessy-Wiltshire equations* and an APF method can work fine by placing at the docking goal an attractive potential field. With an extra care on the mathematics, we have shown that translating the APF to the attitude dynamics is possible and it generates the correct value for the reference angular velocity. The difficulty arises only with the usage of quaternions to represent the satellite attitude, both for their low capacity of description of what is actually going on and for their slightly more complex algebra and non-linearity. Nevertheless, it is possible to deal with the state-space of unit quaternions and construct scalar potential fields functions of $\boldsymbol{Q}$. Focusing on the requirement of keeping a boresight vector of a sensible instrument away from an inertial obstacle it was possible to take advantage of the quaternion algebraic properties and construct a quadratic constraint, which encapsulates the dependence on the quaternion and the positions of obstacle and instrument. From it, a repulsive potential field could be constructed and implemented. The simulations show us that the coupling of APF and SMC works and allows the satellite's attitude to converge. Nevertheless, slight modifications could be implemented in order to correct some stationary state errors, e.g. implementing a STW sliding mode control, providing a continuous control action and thus converging the state in a finite time instead of only asymptotically; moreover, it is better suited for all kinds of disturbances, included unbounded ones. Different types of potential fields as well can be implemented, with different functions based on the specific requirement of the on-board instrument, as well as the control parameters. Therefore, the next step could be the simulation of a higher number of obstacles and on-board instruments, with a different repulsive potential field that can provide stability and keep the satellite away from all the obstacles. Moreover, it could be modified to account for possible problems that could occur like the local minima or a too close to the goal obstacle's position, all coupled with a super-twisting sliding mode control to secure the spacecraft from perturbations.

Bibliography

[1] JWST User Documentation (JDox). JWST User Documentation Website, January 2016. Provided by the SAO/NASA Astrophysics Data System.

[2] G. Avanzini. *Spacecraft Attitude Dynamics and Control.* Dispense del corso di Dinamica e Controllo di Assetto di Satelliti, Politecnico di Torino, 2008-2009.

[3] G. Avanzini, G. Radice, and I. Ali. Potential approach for constrained autonomous manoeuvres of a spacecraft equipped with a cluster of control moment gyroscopes. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 223(3):285–296, 2009.

[4] Riccardo Calaon and Hanspeter Schaub. Constrained attitude maneuvering via modified-rodrigues-parameter-based motion planning algorithms. *Journal of Spacecraft and Rockets*, 59(4):1342–1356, 2022.

[5] E. Capello. *Dispense del corso di Dinamica e Controllo di Veicoli Spaziali.* Politecnico di Torino.

[6] Abhijit Dongare, Reza Hamrah, and Amit K. Sanyal. Attitude pointing control using artificial potentials with control input constraints. pages 1–6, 2021.

[7] Tao Guan and Bin Li. Output feedback attitude control for rigid spacecraft under attitude constraints. *Journal of Industrial and Management Optimization*, 19(7), 2023.

[8] Qinglei Hu, Biru Chi, and Maruthi R. Akella. Anti-unwinding attitude control of spacecraft with forbidden pointing constraints. *Journal of Guidance, Control, and Dynamics*, 42(4):822–835, 2019.

[9] Qinglei Hu, Biru Chi, and Maruthi R. Akella. Reduced attitude control for boresight alignment with dynamic pointing constraints. *IEEE/ASME Transactions on Mechatronics*, 24(6):2942–2952, 2019.

[10] Qinglei Hu, Yueyang Liu, Hongyang Dong, and Youmin Zhang. Saturated attitude control for rigid spacecraft under attitude constraints. *Journal of Guidance, Control, and Dynamics*, 43(4):790–805, 2020.

[11] Unsik Lee and Mehran Mesbahi. Feedback control for spacecraft reorientation under attitude constraints via convex potentials. *IEEE Transactions on Aerospace and Electronic Systems*, 50:2578–2592, 10 2014.

[12] Jiakun Lei, Tao Meng, Weijia Wang, Shujian Sun, Heng Li, and Zhonghe Jin. Spacecraft attitude pointing control under pointing forbidden constraints with guaranteed accuracy. 2022.

[13] G. Mengali and A.A. Quarta. *Fondamenti di meccanica del volo spaziale*. Didattica e Ricerca. Manuali. Pisa University Press, 2013.

[14] Qiang Shen, Chengfei Yue, and Cher Hiang Goh. Velocity-free attitude reorientation of a flexible spacecraft with attitude constraints. *Journal of Guidance, Control, and Dynamics*, 40:1–7, 04 2017.

[15] Qiang Shen, Chengfei Yue, Cher Hiang Goh, Baolin Wu, and Danwei Wang. Rigid-body attitude stabilization with attitude and angular rate constraints. *Automatica*, 90:157–163, 2018.

[16] Yuri Shtessel, C. Edwards, Leonid Fridman, and Arie Levant. *Sliding Mode Control and Observation*. 2014.

[17] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer London, 2009.

[18] Joan Solà. Quaternion kinematics for the error-state kalman filter. `http://www.iri.upc.edu/people/jsola/JoanSola/objectes/notes/kinematics.pdf`, 2017.

[19] Yuan Tian, Qinglei Hu, and Xiaodong Shao. Adaptive fault-tolerant control for attitude reorientation under complex attitude constraints. *Aerospace Science and Technology*, 121:107332, 2022.