Faculty of Aerospace Engineering
Spacecraft Attitude Dynamics and Control: Exercise
AE4313 + AE4313P

Practical Exercise Project Assignment 2

Nonlinear Spacecraft Attitude Control System Design

Responsible Professor: Dr. ir. Q.P. Chu

Author and **Student Number**

Rohan Chotalal, 4746317

(Soft) Deadline: April 19, 2019

2018/2019

Contents

List of Figures

List of Tables

Introduction

Spacecrafts (and satellites) have been vastly increasing in the past decades and most of them possess instruments or antennas that must point (or be pointed) in certain directions. For instance, the Hubble must point its main telescope in search for new stars, planets or other relevant information about the universe or even the Earth. Communication satellites must point their antennas in a direction where signals can be transmitted to ground-based stations.

Taking into account the previous statements, robust attitude control system is necessary to provide control of satellite's orientation in space, so it provides good stabilisation of the attitude vector in the execution of important manoeuvres. But, what does actually define the attitude control of a satellite? This type of control is described by the angular orientation of its body frame relative to a know frame, here named as inertial frame.

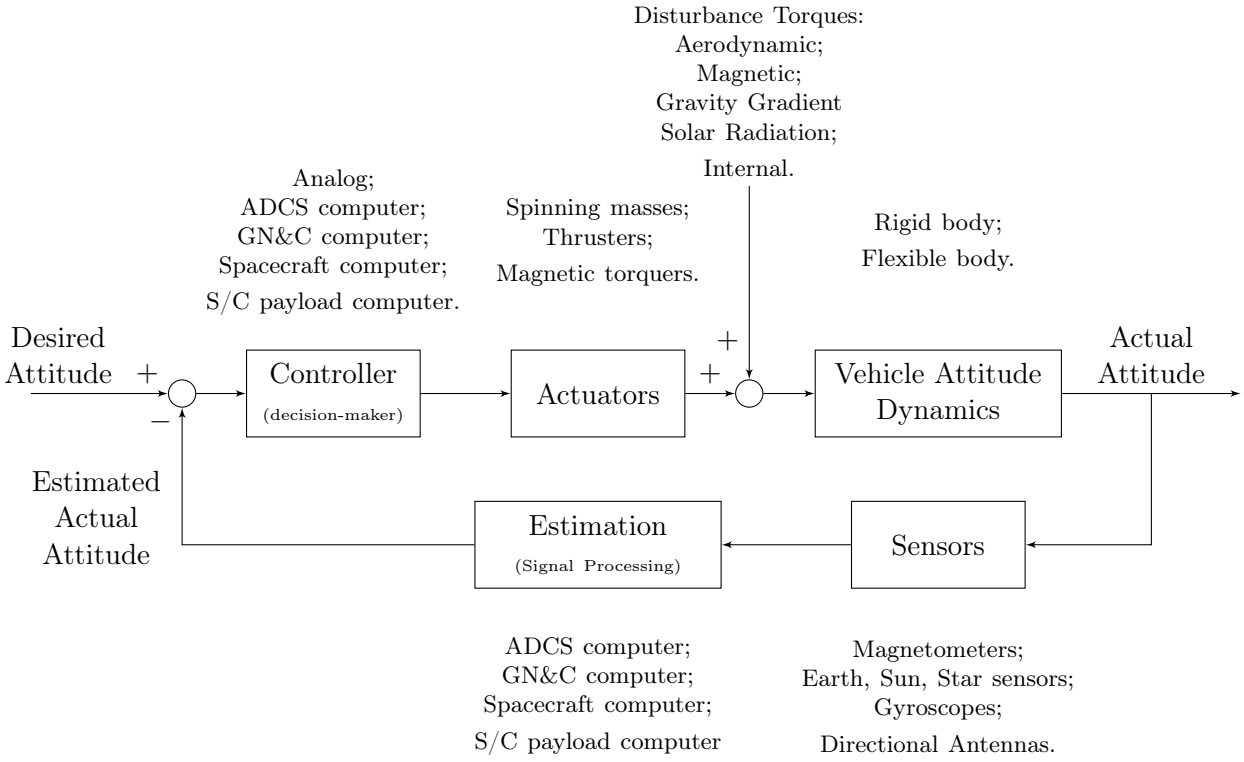


Figure 1: Block Diagram of the Spacecraft Attitude Control System

A schematic of the block diagram for a spacecraft attitude control system is depicted in fig. 1. In order to follow a desired attitude (chosen by the operators working in the ground stations), the current attitude is determined through specific instruments that are sensed with respect to a fixed reference point/frame (such as a star or planet) or using dead-reckoning (derivation from pure integration of certain measurements, such as accelerations or rotational velocity). These measurements are processed through advanced signal processing algorithms to remove the influence of noise or bias. From this point, an estimated value of the attitude is fed to the control algorithm also including information of the desired attitude. The algorithm yields the input laws to the actuators that exert the required torques for the spacecraft to achieve a desired behaviour.

Throughout the years, PID (Proportional-Integral-Derivative) controllers have been widely used in satellite control due to their good performance. However, modern techniques such as Non-linear Dynamic Inversion (known as NDI) ([2]) and the its recently developed incremental version ([4]) have shown better performance than linear controllers in Unmanned Aerial Vehicles (UAV) ([3]), aircraft, spacecraft ([1]) and re-entry vehicles. This report covers the study and implementation of these algorithms in spacecraft attitude control and comparison of the different algorithms.

Considering the outline of the document, it starts with a mathematical description and derivation of the equations relative to spacecraft angular dynamics and kinematics on section 1. This is done in two

representations: Euler angles and quaternions. In section 2, a linear controller (PID) is implemented in the previous models. Model-based NDI control is presented and applied in section 3 for entire system. Sections 4 and 5 present two variants of the NDI control: the first is a model-based approach with the time-scale separation principle and the second is an incremental-based approach (named Incremental NDI, INDI) also applied with the same principle. The principle splits the dynamics and kinematics so that two NDI loops can be applied. The report ends with a comparison of all the methods and overall discussion of the obtained results.

1 Spacecraft Kinematics and Dynamics modelling

In this section, the differential equations of the spacecraft kinematics and dynamics are introduced using the following two approaches:

- Euler Angles;
- Quaternions.

The mathematical model of the spacecraft attitude written in Euler angles accounts for highly non-linear functions (for example, trigonometric functions) which take time to compute results upon working in real-time systems. Moreover, this approach presents numerical accuracy errors when θ or ϕ are $\pm\frac{\pi}{2}$. On the other hand, quaternions are less intuitive than Euler angles, because they operate in an abstract coordinate frame, although they simplify the equations by making the computations faster.

In this report, all the simulations are done offline, which means that the computation time will not be taken into account. The main goal is to compare both approaches in the design of a reliable attitude control system and exploit its limitations.

1.1 Kinematics equation

The kinematics analyse the motion regarding the change of position or orientation of a rigid body when submitted to linear accelerations or angular accelerations. For the attitude control design, a mathematical description of the rotational kinematics of the spacecraft is considered, in order to obtain the differential equations that characterises its rotational movement.

Despite the choice of Euler angles and quaternions, the model could also be presented in other formulations such as direction cosine matrices or MRP (Modified Rodrigues Parameters). However, in this report, they will not be considered, as they can be derived from the previously chosen approaches.

In order to proceed with the derivation of the differential equations, 3 different reference frames should be considered to characterise the spacecraft motion around the the Earth (fig. 2):

- Frame N - Newtonian frame, fixed in the Earth, known as ECI (Earth-Centered Inertial);
- Frame A - Orbital frame - in this case is the Nadir-oriented (spacecraft-centered) frame;
- Frame B - Body frame - attached in the spacecraft. The difference between the orientation of the body frame with respect to the orbital frame yields the spacecraft's attitude (in Euler angles).

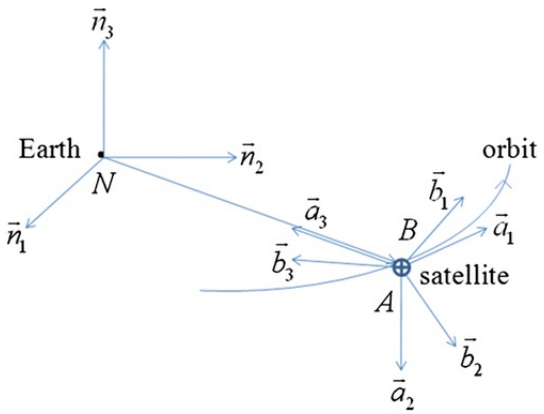


Figure 2: Representation of reference frames used to model the spacecraft

1.1.1 Euler Angles

The Euler angles are considered as a tuple of 3 real numbers:

- roll angle: $\theta_1 = \phi$
- pitch angle: $\theta_2 = \theta$
- yaw angle: $\theta_3 = \psi$

These angles describe the difference in orientation between the body frame B and orbital frame A as a sequence of rotations. To obtain accurate mathematical description, rotational matrices can be defined along x (- axis 1), y (- axis 2) and z (- axis 3) as follows:

- x axis:

$$C_1(\theta_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_1) & \sin(\theta_1) \\ 0 & -\sin(\theta_1) & \cos(\theta_1) \end{bmatrix} \tag{1}$$

- y axis:

$$C_2(\theta_2) = \begin{bmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) \\ 0 & 1 & 0 \\ \sin(\theta_2) & 0 & \cos(\theta_2) \end{bmatrix} \tag{2}$$

- z axis:

$$C_3(\theta_3) = \begin{bmatrix} \cos(\theta_3) & \sin(\theta_3) & 0 \\ -\sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Generically, it is possible to describe a vector between two different reference frames where one is rotated relative to another. This is the case of frames A and B. In order to express a generic vector, $\vec{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T$ from frame A to frame B, yielding $\vec{b} = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}^T$, the sequence of rotations 3-2-1 can be applied mathematically in following manner:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \underbrace{C_1(\theta_1)C_2(\theta_2)C_3(\theta_3)}_{C} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{4}$$

where matrix C is called the direction cossine matrix. Considering $c(\theta) = \cos(\theta)$ and $s(\theta) = \sin(\theta)$, this matrix is defined in eq. (5).

$$\begin{aligned} C_A^B(\vec{\theta}) &= C_1(\theta_1)C_2(\theta_2)C_3(\theta_3) \\ &= \begin{bmatrix} c(\theta_2)c(\theta_3) & c(\theta_2)s(\theta_3) & -s(\theta_2) \\ s(\theta_1)s(\theta_2)c(\theta_3) - c(\theta_1)s(\theta_3) & s(\theta_1)s(\theta_2)c(\theta_3) + c(\theta_1)s(\theta_3) & s(\theta_1)c(\theta_2) \\ c(\theta_1)s(\theta_2)c(\theta_3) + s(\theta_1)s(\theta_3) & c(\theta_1)s(\theta_2)s(\theta_3) - s(\theta_1)c(\theta_3) & c(\theta_1)c(\theta_2) \end{bmatrix} \end{aligned} \tag{5}$$

The rotational velocity of the spacecraft is defined in the Newtonian frame using the following:

$$\vec{\omega}^{B/N} = \vec{\omega}^{B/A} + \vec{\omega}^{A/N} \tag{6}$$

which means that the angular velocity of the spacecraft in the body frame B with respect to the inertial (Newtonian) frame, $\vec{\omega}^{B/N}$, is described as the sum of its angular velocity in the body frame relative to the orbital frame, $\vec{\omega}^{B/A}$, and the angular velocity of the orbital frame relative to the Newtonian frame, $\vec{\omega}^{A/N}$.

However, $\vec{\omega}^{B/A}$ can be written as a sum of vectors whose components of the angular velocity are aligned with one basis vector in the frames A', A'' and B (eq. (7)). In this case, A' obtained from A that rotates θ_3 along the $z - th$ direction. A'' is obtained from A' after rotating θ_2 along the $y' - th$ direction. Frame B is obtained after rotating θ_1 on frame A'' in the $x'' - th$ direction. To express all

vectors in on consistent frame, in this case, the body frame, relevant rotations are applied and shown in eq. (8).

$$\begin{aligned}\vec{\omega}^{B/A} &= \vec{\omega}^{B/A''} + \vec{\omega}^{A''/A'} + \vec{\omega}^{A'/A} = \\ &= \begin{bmatrix}\dot{\theta}_1\\0\\0\end{bmatrix}_B + \begin{bmatrix}0\\\dot{\theta}_2\\0\end{bmatrix}_{A''} + \begin{bmatrix}0\\0\\\dot{\theta}_3\end{bmatrix}_{A'} \end{aligned} \tag{7}$$

$$\vec{\omega}^{B/A''} = \begin{bmatrix}\dot{\theta_1}\\0\\0\end{bmatrix} + C_1(\theta_1)\begin{bmatrix}0\\\dot{\theta_2}\\0\end{bmatrix} + C_2(\theta_2)C_1(\theta_1)\begin{bmatrix}0\\0\\\dot{\theta_3}\end{bmatrix} \tag{8}$$

By definition, $\omega^{A/N}$ is angular velocity of the system in the given orbit influenced by force of gravity. In this case, $\vec{a}_2 = \begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$ is the basis vector along the $y-th$ axis in frame A and n is a constant named the orbital rate which depends on the radius of the orbit (R), mass of the planet (M_T) and gravitational constant (G), as represented in eq. (10).

Therefore, it is possible to express this result from frame A to B, by transforming this vector using the following order of rotations $\theta_3 \to \theta_2 \to \theta_1$, which consists in the same as direction cossine matrix. The result is shown in eq. (9).

$$\omega^{A/N} = -n\vec{a}_2 = -nC\begin{bmatrix}0\\1\\0\end{bmatrix}_A = -n\begin{bmatrix}\cos(\theta_2)\sin(\theta_3)\\ \sin(\theta_1)\sin(\theta_2)\cos(\theta_3) + \cos(\theta_1)\sin(\theta_3)\\ \cos(\theta_1)\sin(\theta_2)\sin(\theta_3) - \sin(\theta_1)\cos(\theta_3)\end{bmatrix}_B \tag{9}$$

$$n = \sqrt{\frac{GM_E}{R^3}} \tag{10}$$

Thus, merging the previous results, eq. (6) can be written as follows:

$$\omega^{B/N} = \underbrace{\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\end{bmatrix}}_{\vec{\omega}^{B/N}=\vec{\omega}} = \underbrace{\begin{bmatrix}1 & 0 & -\sin(\theta_2)\\ 0 & \cos(\theta_1) & \sin(\theta_1)\cos(\theta_2)\\ 0 & -\sin(\theta_1) & \cos(\theta_1)\cos(\theta_2)\end{bmatrix}}_{R}\underbrace{\begin{bmatrix}\dot{\theta_1}\\\dot{\theta_2}\\\dot{\theta_3}\end{bmatrix}}_{\dot{\vec{\theta}}} - n\begin{bmatrix}\cos(\theta_2)\sin(\theta_3)\\ \sin(\theta_1)\sin(\theta_2)\cos(\theta_3) + \cos(\theta_1)\sin(\theta_3)\\ \cos(\theta_1)\sin(\theta_2)\sin(\theta_3) - \sin(\theta_1)\cos(\theta_3)\end{bmatrix} \tag{11}$$

Solving the previous equation for $\dot{\vec{\theta}}$ yields the kinematics equation of the spacecraft/satellite, expressed in Euler angles (eq. (12)), which will be used to estimate the attitude of the satellite with respect to the fixed body frame.

$$\begin{aligned}\begin{bmatrix}\dot{\theta_1}\\\dot{\theta_2}\\\dot{\theta_3}\end{bmatrix} &= \frac{1}{\cos(\theta_2)}\begin{bmatrix}\cos(\theta_2) & \sin(\theta_1)\sin(\theta_2) & \cos(\theta_1)\sin(\theta_2)\\ 0 & \cos(\theta_1)\cos(\theta_2) & -\sin(\theta_1)\cos(\theta_2)\\ 0 & \sin(\theta_1) & \cos(\theta_1)\end{bmatrix}\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\end{bmatrix} + \frac{n}{\cos(\theta_2)}\begin{bmatrix}\sin(\theta_3)\\ \cos(\theta_2)cos(\theta_3)\\ \sin(\theta_2)\sin(\theta_3)\end{bmatrix}\\ &= \underbrace{\begin{bmatrix}1 & \sin(\theta_1)\tan(\theta_2) & \cos(\theta_1)\tan(\theta_2)\\ 0 & \cos(\theta_1)) & -\sin(\theta_1)\\ 0 & \frac{\sin(\theta_1)}{\cos(\theta_2)} & \frac{\cos(\theta_1)}{\cos(\theta_2)}\end{bmatrix}}_{R^{-1}=N(\vec{\theta})}\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\end{bmatrix} + \underbrace{n\begin{bmatrix}\frac{\sin(\theta_3)}{\cos(\theta_2)}\\ cos(\theta_3)\\ \tan(\theta_2)\sin(\theta_3)\end{bmatrix}}_{nR^{-1}C}\end{aligned} \tag{12}$$

1.1.2 Quaternions

Quaternions is a 4-tuple vector that encode rotations in the 3-D space. This is an abstract representation as its geometric meaning is less intuitive than the Euler angles. These numbers are represented

by 3 complex numbers and 1 scalar. However, theoretical derivation of the properties underlying these numbers will be omitted, as it is not the study of this assignment.

Considering the Euler eigenaxis rotation about an arbitrary axis fixed in body frame B and stationary in an inertial reference frame A, each element of the quaternion vector $\mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}$ is defined in eq. (13). The vector $\begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}$ is the direction cosine of the Euler eigenaxis. Thus, looking at the expressions, the first 3 quaternions $\begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}$ encode the information of the rotation and the last quaternion is a normalisation term that ensures the constraint stated in eq. (14), which must always be obeyed. Thus, controlling the first three quaternions is enough to ensure that the fourth one is also controlled.

$$\begin{cases} q_1 = e_1 \sin(\theta/2) \\ q_2 = e_2 \sin(\theta/2) \\ q_3 = e_3 \sin(\theta/2) \\ q_4 = \cos(\theta/2) \end{cases} \tag{13}$$

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \tag{14}$$

Similarly, a result of the direction cosine matrix stated in eq. (5) can be obtained in quaternion form. This is presented in eq. (5).

$$C_A^B(\mathbf{q}) = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1 q_2 + q_3 q_4) & 2(q_1 q_3 - q_2 q_4) \\ 2(q_2 q_1 - q_3 q_4) & 1 - 2(q_1^2 + q_3^2) & 2(q_2 q_3 + q_1 q_4) \\ 2(q_3 q_1 + q_2 q_4) & 2(q_3 q_2 - q_1 q_4) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix} \tag{15}$$

To ensure a relation between the Euler angles and quaternions, a formula presented in eq. (16) can be used. This is important to translate the desired attitude commands in from one notation to another.

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \sin(\frac{\theta_1}{2})\cos(\frac{\theta_2}{2})\cos(\frac{\theta_3}{2}) - \cos(\frac{\theta_1}{2})\sin(\frac{\theta_2}{2})\sin(\frac{\theta_3}{2}) \\ \cos(\frac{\theta_1}{2})\sin(\frac{\theta_2}{2})\cos(\frac{\theta_3}{2}) + \sin(\frac{\theta_1}{2})\cos(\frac{\theta_2}{2})\sin(\frac{\theta_3}{2}) \\ \cos(\frac{\theta_1}{2})\cos(\frac{\theta_2}{2})\sin(\frac{\theta_3}{2}) - \sin(\frac{\theta_1}{2})\sin(\frac{\theta_2}{2})\cos(\frac{\theta_3}{2}) \\ \cos(\frac{\theta_1}{2})\cos(\frac{\theta_2}{2})\cos(\frac{\theta_3}{2}) + \sin(\frac{\theta_1}{2})\sin(\frac{\theta_2}{2})\sin(\frac{\theta_3}{2}) \end{bmatrix} \tag{16}$$

The derivative of the quaternion vector $\dot{\mathbf{q}} = \begin{bmatrix} \dot{q_1} & \dot{q_2} & \dot{q_3} & \dot{q_4} \end{bmatrix}^T$ can be expressed with respect to the angular velocity of the satellite $\vec{\omega}^{B/A}$ in the body frame B with respect to frame A, using eq. (17).

$$\begin{bmatrix} \omega_1^{B/A} \\ \omega_2^{B/A} \\ \omega_3^{B/A} \\ 0 \end{bmatrix} = 2 \underbrace{\begin{bmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \\ q_1 & q_2 & q_3 & q_4 \end{bmatrix}}_{Q} \begin{bmatrix} \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \\ \dot{q_4} \end{bmatrix} \tag{17}$$

Similar to eq. (9), the effect of the gravity acting in the circular orbit described by the satellite can be described in eq. (18) using quaternions.

$$\vec{\omega}^{A/N} = -n\vec{a}_2 = -nC_A^B(\mathbf{q}) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = -n \begin{bmatrix} 2(q_1 q_2 + q_3 q_4) \\ 1 - 2(q_1^2 + q_3^2) \\ 2 - (q_3 q_2 - q_1 q_4) \end{bmatrix} \tag{18}$$

Therefore, to find the rotational velocity of the spacecraft with respect to the inertial (fixed) frame N, the relation eq. (6) can be used again, taking into account the previous two equations.

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ 0\end{bmatrix} = 2\begin{bmatrix} q_4 & q_3 & -q_2 & -q_1\\ -q_3 & q_4 & q_1 & -q_2\\ q_2 & -q_1 & q_4 & -q_3\\ q_1 & q_2 & q_3 & q_4\end{bmatrix}\begin{bmatrix}\dot{q_1}\\ \dot{q_2}\\ \dot{q_3}\\ \dot{q_4}\end{bmatrix} - n\begin{bmatrix} 2(q_1q_2+q_3q_4)\\ 1-2(q_1^2+q_3^2)\\ 2-(q_3q_2-q_1q_4)\\ 0\end{bmatrix} \tag{19}$$

Solving for $\dot{\mathbf{q}}$ yields the kinematics equation in quaternion form:

$$\begin{bmatrix}\dot{q_1}\\ \dot{q_2}\\ \dot{q_3}\\ \dot{q_4}\end{bmatrix} = \frac{1}{2}\begin{bmatrix} q_4 & q_3 & -q_2 & -q_1\\ -q_3 & q_4 & q_1 & -q_2\\ q_2 & -q_1 & q_4 & -q_3\\ q_1 & q_2 & q_3 & q_4\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ 0\end{bmatrix} + \frac{n}{2}\begin{bmatrix} q_3\\ q_4\\ -q_1\\ -q_2\end{bmatrix} = \underbrace{\frac{1}{2}\begin{bmatrix} 0 & \omega_3 & -\omega_2+n & \omega_1\\ \omega_3 & 0 & \omega_1 & \omega_2+n\\ \omega_2-n & -\omega_1 & 0 & \omega_2\\ -\omega_1 & -\omega_2-n & \omega_3 & 0\end{bmatrix}}_{N(\mathbf{q})}\begin{bmatrix}q_1\\ q_2\\ q_3\\ q_4\end{bmatrix} \tag{20}$$

The last step is explained through the orthogonality (in this case, orthonomality) property of the Q matrix, as it can be seen below:

$$\begin{bmatrix} q_4 & q_3 & -q_2 & -q_1\\ -q_3 & q_4 & q_1 & -q_2\\ q_2 & -q_1 & q_4 & -q_3\\ q_1 & q_2 & q_3 & q_4\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ 0\end{bmatrix} = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1\\ \omega_3 & 0 & \omega_1 & \omega_2\\ \omega_2 & -\omega_1 & 0 & \omega_2\\ -\omega_1 & -\omega_2 & \omega_3 & 0\end{bmatrix}\begin{bmatrix}q_1\\ q_2\\ q_3\\ q_4\end{bmatrix} \tag{21}$$

1.2 Dynamics equation

The angular dynamics of the spacecraft is obtained through the Euler equations of motion. It relates the kinetics of the system with its motion where the variation of the angular momentum of system ($\frac{d\vec{H}}{dt}$) is equal to the external torques ($\vec{T}$) applied to the system (eq. (23)).

$$\vec{T} = \frac{d\vec{H}}{dt} \tag{22}$$

$$\vec{T} = \hat{J}\cdot\dot{\vec{\omega}} + \underbrace{\vec{\omega}\times}_{\Omega(\vec{\omega})}(\hat{J}\cdot\vec{\omega}) \tag{23}$$

The reader should note that

$$\Omega(\omega) = \begin{bmatrix} 0 & -\omega_3 & \omega_2\\ \omega_3 & 0 & -\omega_1\\ -\omega_2 & \omega_1 & 0\end{bmatrix} \tag{24}$$

and the $\hat{J}$ the inertia matrix is diagonal matrix:

$$\hat{J} = \begin{bmatrix} J_{1,1} & 0 & 0\\ 0 & J_{2,2} & 0\\ 0 & 0 & J_{3,3}\end{bmatrix} \tag{25}$$

where $J_{1,1} = 124.531\ kg\cdot m^2$, $J_{2,2} = 124.586\ kg\cdot m^2$ and $J_{3,3} = 0.704\ kg\cdot m^2$.

The external torques (26) applied to the system are obtained as a sum of three terms. These are the external disturbances ($\vec{T_d}$) applied in the system and are assumed to be constant and equal to 0.0001 $N\cdot m$ in all three axis; the effect of the gravity gradient ($\vec{T_{GG}} = 3n^2\Omega(\vec{a_3})\hat{J}\vec{a_3}$, where $\vec{a_3}$ is expressed in the orbital frame, i.e., frame A) and the inputs given by the actuators T_c. Since the actuators are

assumed to be perfect, the control law obtained from the attitude control system is acted as a direct input of the system.

$$\begin{aligned} \vec{T} &= \vec{T}_d + \vec{T}_{GG} + \vec{T}_c \\ \hat{J} \cdot \dot{\vec{\omega}} + \Omega(\vec{\omega})(\hat{J} \cdot \vec{\omega}) &= \vec{T}_d + \vec{T}_{GG} + \vec{T}_c \end{aligned} \tag{26}$$

Solving the previous equation for $\dot{\omega}$, yields the dynamics of the spacecraft as presented in eq. (27). In matrix notation, it is written in eq. (28).

$$\dot{\vec{\omega}} = -\hat{J}^{-1}\Omega(\vec{\omega})(\hat{J} \cdot \vec{\omega}) + \hat{J}^{-1}(\vec{T}_d + \vec{T}_{GG} + \vec{T}_c) \tag{27}$$

$$\begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} = \hat{J}^{-1} \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & \omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \hat{J} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} + \hat{J}^{-1} \left(\begin{bmatrix} T_{d_1} \\ T_{d_2} \\ T_{d_3} \end{bmatrix} + 3n^2 \begin{bmatrix} 0 & -a_{3,3} & a_{3,2} \\ a_{3,3} & 0 & -a_{3,1} \\ -a_{3,2} & a_{3,1} & 0 \end{bmatrix} \hat{J} \begin{bmatrix} a_{3,1} \\ a_{3,2} \\ a_{3,3} \end{bmatrix} + \begin{bmatrix} T_{c_1} \\ T_{c_2} \\ T_{c_3} \end{bmatrix} \right) \tag{28}$$

Now, only $a_{3,i}$ for $i \in \{1, 2, 3\}$ needs to be defined. Since this vector is in frame A it needs to be converted to frame B using the direction cossine matrix either in Euler angles or in quaternions.

$$\begin{bmatrix} a_{3,1} \\ a_{3,2} \\ a_{3,3} \end{bmatrix} = C_A^B \vec{a_3} = C_A^B \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_A \tag{29}$$

1.2.1 Euler Angles

Using eq. (29) and the direction cosine matrix in eq. (15), vector $\vec{a}_3$ can be expressed as follows in terms of the Euler angles:

$$\begin{bmatrix} a_{3,1} \\ a_{3,2} \\ a_{3,3} \end{bmatrix} = \begin{bmatrix} -\sin(\theta_2) \\ \sin(\theta_1)\cos(\theta_2) \\ \cos(\theta_1)\cos(\theta_2) \end{bmatrix} \tag{30}$$

Thus, the gravity gradient term in the equation can be included in eq. (31) in matrix notation and extended in eq. (32).

$$\begin{aligned} \begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} = &\begin{bmatrix} \frac{1}{J_{1,1}} & 0 & 0 \\ 0 & \frac{1}{J_{2,2}} & 0 \\ 0 & 0 & \frac{1}{J_{3,3}} \end{bmatrix} \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & \omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \begin{bmatrix} J_{1,1} & 0 & 0 \\ 0 & J_{2,2} & 0 \\ 0 & 0 & J_{3,3} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{J_{1,1}} & 0 & 0 \\ 0 & \frac{1}{J_{2,2}} & 0 \\ 0 & 0 & \frac{1}{J_{3,3}} \end{bmatrix} \left(\begin{bmatrix} T_{d_1} \\ T_{d_2} \\ T_{d_3} \end{bmatrix} + \right. \\ &\left. + 3n^2 \begin{bmatrix} 0 & -\cos(\theta_1)\cos(\theta_2) & \sin(\theta_1)\cos(\theta_2) \\ \cos(\theta_1)\cos(\theta_2) & 0 & \sin(\theta_2) \\ -\sin(\theta_1)\cos(\theta_2) & -\sin(\theta_2) & 0 \end{bmatrix} \begin{bmatrix} J_{1,1} & 0 & 0 \\ 0 & J_{2,2} & 0 \\ 0 & 0 & J_{3,3} \end{bmatrix} \begin{bmatrix} -\sin(\theta_2) \\ \sin(\theta_1)\cos(\theta_2) \\ \cos(\theta_1)\cos(\theta_2) \end{bmatrix} + \begin{bmatrix} T_{c_1} \\ T_{c_2} \\ T_{c_3} \end{bmatrix} \right) \end{aligned} \tag{31}$$

$$\begin{cases} \dot{\omega_1} &= \frac{J_3 - J_2}{J_1}\omega_3\omega_2 + \frac{1}{J_1}T_{d_1} + 3n^2\frac{J_3 - J_2}{J_1}\sin(\theta_1)\cos(\theta_1)\cos^2(\theta_2) + \frac{1}{J_1}T_{c_1} \\ \dot{\omega_2} &= \frac{J_1 - J_3}{J_2}\omega_1\omega_3 + \frac{1}{J_2}T_{d_2} + 3n^2\frac{J_3 - J_1}{J_2}\sin(\theta_2)\cos(\theta_1)\cos(\theta_2) + \frac{1}{J_2}T_{c_2} \\ \dot{\omega_3} &= \frac{J_2 - J_1}{J_3}\omega_1\omega_2 + \frac{1}{J_3}T_{d_3} + 3n^2\frac{J_1 - J_2}{J_3}\sin(\theta_1)\sin(\theta_1)\cos(\theta_2) + \frac{1}{J_3}T_{c_3} \end{cases} \tag{32}$$

1.2.2 Quaternions

Similarly to Euler angles, vector $\vec{a}_3$ can be written in quaternion form using the direction cossine matrix given in eq. (15). The result from eq. (29) yields eq. (33).

$$\begin{bmatrix} a_{3,1} \\ a_{3,2} \\ a_{3,3} \end{bmatrix} = \begin{bmatrix} 2(q_1 q_3 - q_2 q_4) \\ 2(q_2 q_3 + q_1 q_4) \\ 1 - 2(q_1^2 + q_2^2) \end{bmatrix} \tag{33}$$

Including this vector in the Gradient Gravity term within the dynamics of the spacecraft, the following expression is obtained:

$$\begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{J_{1,1}} & 0 & 0 \\ 0 & \frac{1}{J_{2,2}} & 0 \\ 0 & 0 & \frac{1}{J_{3,3}} \end{bmatrix} \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & \omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \begin{bmatrix} J_{1,1} & 0 & 0 \\ 0 & J_{2,2} & 0 \\ 0 & 0 & J_{3,3} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{J_{1,1}} & 0 & 0 \\ 0 & \frac{1}{J_{2,2}} & 0 \\ 0 & 0 & \frac{1}{J_{3,3}} \end{bmatrix} \left(\begin{bmatrix} T_{d_1} \\ T_{d_2} \\ T_{d_3} \end{bmatrix} + \right.$$
$$\left. + 3n^2 \begin{bmatrix} 0 & -1 + 2(q_1^2 + q_2^2) & 2(q_2 q_3 + q_1 q_4) \\ 1 - 2(q_1^2 + q_2^2) & 0 & -2(q_1 q_3 - q_2 q_4) \\ -2(q_2 q_3 + q_1 q_4) & 2(q_1 q_3 - q_2 q_4) & 0 \end{bmatrix} \begin{bmatrix} J_{1,1} & 0 & 0 \\ 0 & J_{2,2} & 0 \\ 0 & 0 & J_{3,3} \end{bmatrix} \begin{bmatrix} 2(q_1 q_3 - q_2 q_4) \\ 2(q_2 q_3 + q_1 q_4) \\ 1 - 2(q_1^2 + q_2^2) \end{bmatrix} + \begin{bmatrix} T_{c_1} \\ T_{c_2} \\ T_{c_3} \end{bmatrix} \right) \tag{34}$$

The extended version of the previous equation is given in eq. (35).

$$\begin{cases} \dot{\omega}_1 &= \frac{J_3 - J_2}{J_1} \omega_3 \omega_2 + \frac{1}{J_1} T_{d_1} + 3n^2 \frac{J_3 - J_2}{J_1} (1 - 2(q_1^2 + q_2^2)) 2(q_2 q_3 + q_1 q_4)) + \frac{1}{J_1} T_{c_1} \\ \dot{\omega}_2 &= \frac{J_1 - J_3}{J_2} \omega_1 \omega_3 + \frac{1}{J_2} T_{d_2} + 3n^2 \frac{J_3 - J_1}{J_2} 2(q_1 q_3 - q_2 q_4)(1 - 2(q_1^2 + q_2^2)) + \frac{1}{J_2} T_{c_2} \\ \dot{\omega}_3 &= \frac{J_2 - J_1}{J_3} \omega_1 \omega_2 + \frac{1}{J_3} T_{d_3} + 3n^2 \frac{J_1 - J_2}{J_3} 4(q_1 q_3 - q_2 q_4)(q_2 q_3 + q_1 q_4) + \frac{1}{J_3} T_{c_3} \end{cases} \tag{35}$$

1.3 Final Models

The mathematical equations that establish the dynamics and kinematics of the spacecraft are highly non-linear. Their description can be represented by eq. (36), where $\dot{\mathbf{x}}$ describes the derivative of the state, $\mathbf{y}$ the output of the system and

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}) \end{cases} \tag{36}$$

The computation of the state update is obtained in the following manner:

$$\mathbf{x}_t = \mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{f}(\mathbf{x}, \mathbf{u}) dt \tag{37}$$

In a computer, the data-processing is usually done in discrete-time. Thereafter, using the equations in a discrete-time stamp, the state update is computed as follows:

$$\mathbf{x}_{t_{k+1}} = \mathbf{x}(t_k) + \int_{t_k}^{t_{k+1}} \mathbf{f}(\mathbf{x}(\mathbf{t_k}), \mathbf{u}(\mathbf{t_k})) dt \tag{38}$$

Taking into account that the equations are highly non-linear, numerical Euler method is used to obtain the next state:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{39}$$

where $\Delta t = 0.1$ seconds. Note that other methods could also be used to assure better approximation of the state update. One of the best examples is the Runge-Kutta method.

Additionally, a complete representation of the system dynamics and kinematics is show in both Euler angles and quaternions. This description also includes defining the state vector $\mathbf{x}$ of the system and the input vector $\mathbf{u}$. In this case, for both representations, the input vector is equal to $\vec{T}_c$.

Euler Angles

Taking into account the results and notation given in eqs. (12) and (31), the description of the angular motion of the spacecraft in Euler angles is presented in eq. (40).

$$\begin{bmatrix}\dot{\vec{\theta}}\\ \dot{\vec{\omega}}\end{bmatrix} = \begin{bmatrix}\mathbf{0}_{3\times3} & N(\vec{\theta})\\ \mathbf{0}_{3\times3} & -\hat{J}^{-1}\Omega(\vec{\omega})\hat{J}\end{bmatrix}\begin{bmatrix}\vec{\theta}\\ \vec{\omega}\end{bmatrix} + \begin{bmatrix}\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3}\\ -\hat{J} & -\hat{J} & \mathbf{0}_{3\times3}\end{bmatrix}\begin{bmatrix}\vec{T}_d\\ 3n^2\Omega(a_3)\hat{J}a_3\\ -nR^{-1}C\end{bmatrix} + \begin{bmatrix}\mathbf{0}_{3\times3}\\ -\hat{J}\end{bmatrix}\vec{T}_c \tag{40}$$

The state vector is given by 6 states: $\mathbf{x} = \begin{bmatrix}\theta_1 & \theta_2 & \theta_3 & \omega_1 & \omega_2 & \omega_3\end{bmatrix}^T$, which are the 3 Euler angles and the angular velocity vector in the body frame.

Quaternions Model

A similar description can also be represented in quaternions. Equations 20 and 34 are used to show this description in eq. (41).

$$\begin{bmatrix}\dot{\mathbf{q}}\\ \dot{\vec{\omega}}\end{bmatrix} = \begin{bmatrix}N(\vec{\omega}) & \mathbf{0}_{4\times3}\\ \mathbf{0}_{3\times4} & -\hat{J}^{-1}\Omega(\vec{\omega})\hat{J}\end{bmatrix}\begin{bmatrix}\mathbf{q}\\ \vec{\omega}\end{bmatrix} + \begin{bmatrix}\mathbf{0}_{4\times3} & \mathbf{0}_{4\times3}\\ \hat{J}^{-1} & \hat{J}^{-1}\end{bmatrix}\begin{bmatrix}\vec{T}_d\\ 3n^2\Omega(a_3)\hat{J}a_3\end{bmatrix} + \begin{bmatrix}\mathbf{0}_{4\times3}\\ \hat{J}^{-1}\end{bmatrix}\vec{T}_c \tag{41}$$

In this case, the state vector presents a small change when compared to the previous approach, because instead updating the Euler angles, quaternions are used, i.e, $\mathbf{x} = \begin{bmatrix}q_1 & q_2 & q_3 & q_4 & \omega_1 & \omega_2 & \omega_3\end{bmatrix}^T$.

One common problems to account when updating the quaternions to one time step to another is when the normalization equation (eq. (14)) no longer verifies. Therefore, quaternions must be normalized after the state update. There are two ways of doing it, which will be presented ahead. However, only one of them is chosen.

- Inverting equation eq. (14), yields the following updates:

$$q_1 = \sqrt{1 - q_2 - q_3 - q_4} \tag{42}$$
$$q_2 = \sqrt{1 - q_3 - q_4 - q_1} \tag{43}$$
$$q_3 = \sqrt{1 - q_4 - q_1 - q_2} \tag{44}$$
$$q_4 = \sqrt{1 - q_1 - q_2 - q_3} \tag{45}$$

  These updates are done in order, i.e, after updating q_1, the next step would be to update q_2 with that value, and q_3 with the previous two and in the last step would be to get q_4. The only disadvantage of this method is that it yields complex solutions when either the sum of 3 quaternions or only one of them is bigger than 1. For this reason, the method fails and it is not be used.

- This second approach consisted on dividing all the quaternions by their current norm. This is a method used in linear algebra. Treating the quaternions as basis vectors of a 4-dimensional space, it is possible to normalize them using the current norm in each iteration:

$$||\mathbf{q}|| = \sqrt{q^1 + q^2 + q^3 + q^4} \tag{46}$$

  Then each quaternion is divided by this norm, yielding:

$$q_{i_{\text{normalized}}} = \frac{q_i}{||\mathbf{q}||} \tag{47}$$

  for $i = \{1, 2, 3, 4\}$. These new values are used in the next iteration.

2 Design of a Linear Controller

Now that the mathematical description of the spacecraft model was developed, this section underlies the study and design of a simple linear PD (Proportional-Derivative) controller for tracking the referenced Euler angles (or Quaternions). This controller is described as follows:

$$u = K_p \odot \vec{e} + K_d \odot \dot{\vec{e}} = \begin{bmatrix} K_{p_1} \\ K_{p_2} \\ K_{p_3} \end{bmatrix} \odot \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} K_{d_1} \\ K_{d_2} \\ K_{d_3} \end{bmatrix} \odot \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} K_{p_1} \cdot e_1 \\ K_{p_2} \cdot e_2 \\ K_{p_3} \cdot e_3 \end{bmatrix} + \begin{bmatrix} K_{d_1} \cdot \dot{e}_1 \\ K_{d_2} \cdot \dot{e}_2 \\ K_{d_3} \cdot \dot{e}_3 \end{bmatrix} \tag{48}$$

where K_p and K_p are the gains of the controller and $\odot$ is the hadamart product (or element-wise product). The variables e and $\dot{e}$ are the error and derivative of the static error, respectively, and are given by

$$e_i = \theta_{ref_i} - \theta_i \quad (\text{or } e_i = q_{ref_i} - q_i) \tag{49}$$

$$\dot{e}_i = \dot{\theta}_{ref_i} - \dot{\theta}_i \quad (\text{or } \dot{e} = \dot{q}_{ref_i} - \dot{q}_i) \tag{50}$$

for $i \in \{1, 2, 3\}$. A second way to determine the quaternion error is later used for the NDI control. This is presented in eq. (51).

$$\begin{bmatrix} q_{1_e} \\ q_{2_e} \\ q_{3_e} \\ q_{4_e} \end{bmatrix} = \underbrace{\begin{bmatrix} q_{4_c} & q_{3_c} & -q_{2_c} & -q_{1_c} \\ -q_{3_c} & q_{4_c} & q_{1_c} & -q_{2_c} \\ q_{2_c} & -q_{1_c} & q_{4_c} & -q_{3_c} \\ q_{1_c} & q_{2_c} & -q_{3_c} & q_{4_c} \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}}_{\mathbf{q}} \tag{51}$$

where $q_{i_e} = e_i$ and q_{i_c} are the error and commanded quaternions, respectively. Note that only the first three quaternions are controlled.

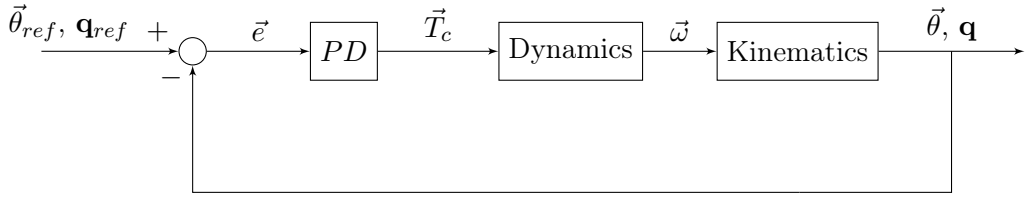


Figure 3: Block diagram of the Spacecraft Attitude Control using PID control

The block diagram of the closed loop system for both approaches is presented in figure fig. 3. The following requirements are considered in the design of these controllers:

- small error - the difference between the desired and current values of attitude should be tend to zero;
- small overshoot;
- fast response - it achieves the desired value quickly and stabilises;
- few (or preferably no) oscillations.

The desired attitude commands for the Euler angles are depicted in fig. 4, 0 degrees from 0 to 99.9 seconds, 70 degrees from 100 seconds to 500 seconds, −70 degrees from 500.1 seconds to 900 seconds and again 0 degrees from 900.1 seconds to 1500 seconds. All these commands are given in all axes as asked in the assignment and they and the controller has to cope with these "bang-bang" maneuver. For quaternions, the desired commands are obtained through eq. (16) from the Euler angles and shown in fig. 5. Using this information, the derivative of the attitude command ($\dot{\theta}_i$ and $\dot{q_{i_e}}$) in the error derivative

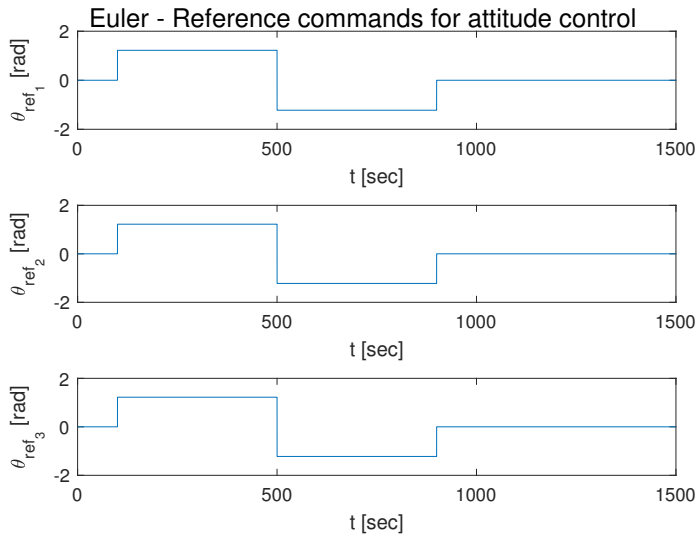
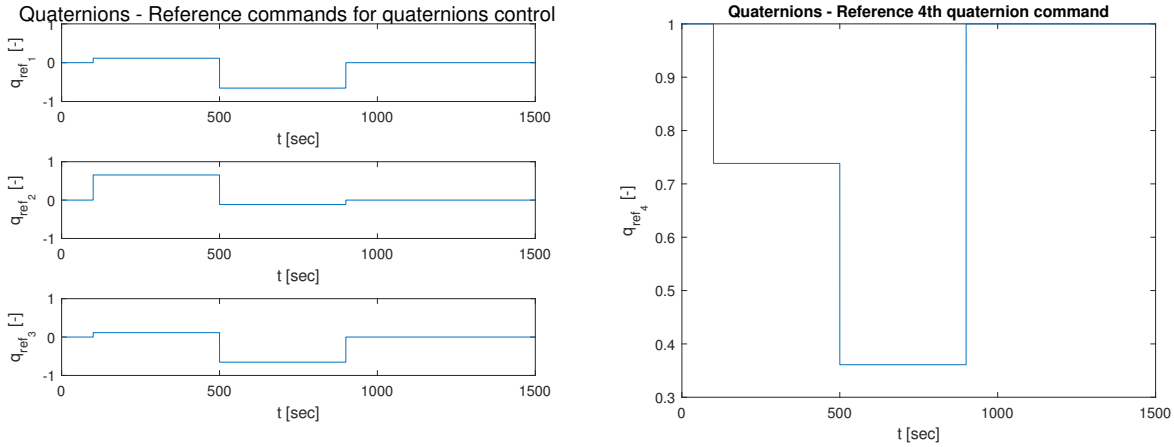


Figure 4: Commanded attitude commands expressed in Euler angles



(a) First 3 quaternions

(b) 4th quaternion command

Figure 5: Commanded attitude commands expressed in quaternions

$\dot{\vec{e}}$ for both approaches is zero during the entire simulation. The effect of the infinite derivative when the step command occurs is negleted. Thus, to compute this error derivative the following holds:

$$\frac{de_i}{dt} = \underbrace{\frac{d\theta_{ref_i}}{dt}}_{=0} - \frac{d\theta_i}{dt} = -\frac{d\theta_i}{dt} = -\dot{\theta}_i \tag{52}$$

$$\frac{de_i}{dt} = \frac{d([Q\mathbf{q}]_i)}{dt} = \left[\underbrace{\frac{dQ}{dt}}_{=0}\mathbf{q}\right]_i - \left[Q\frac{\mathbf{q}}{dt}\right]_i = -\left[Q\frac{\mathbf{q}}{dt}\right]_i = -[Q\dot{\mathbf{q}}]_i \tag{53}$$

Note that $\frac{dQ}{dt} = 0$ because it is a matrix with constant entries.

The initial attitude of the spacecraft is 30 degrees in all axes and a value of 30 deg/s is attributed to the initial angular velocity. In the end, a simulation of the 1500 seconds "course" is generated with this controller.

2.1 Euler angles

To obtain the desired responses for the system described in Euler angles, after some tweaks, the gains presented in eq. (54) are chosen so it best fulfils the previous requirements. The attitude responses

are presented in fig. 6 alongside with the information of the angular velocity in fig. 7.

$$K_p = \begin{bmatrix} 200 \\ 200 \\ 0.5 \end{bmatrix} \qquad K_d = \begin{bmatrix} 700 \\ 700 \\ 1 \end{bmatrix} \tag{54}$$

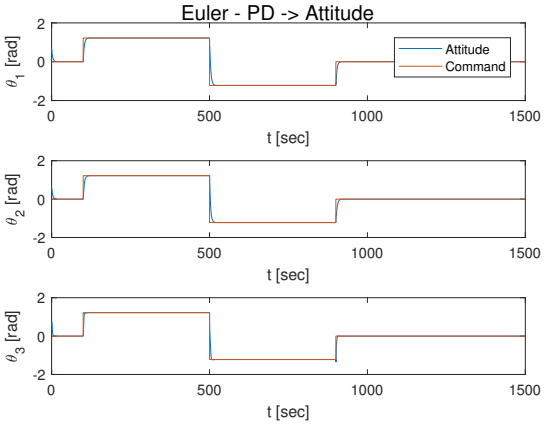


Figure 6: Euler angles of the satellite with PD controller

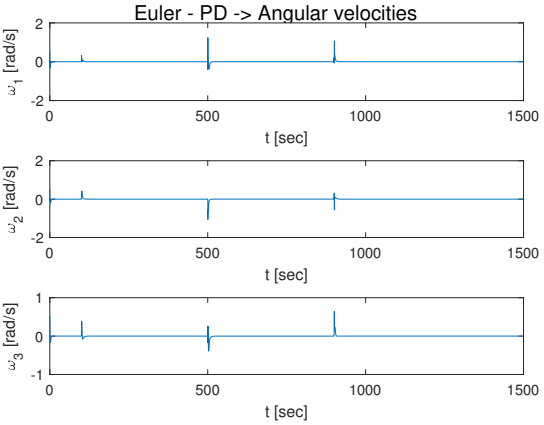


Figure 7: Angular velocities of the satellite with PD controller - Euler angles

All three Euler angles track the desired commands when they are constant with no oscillations. Moreover, the respective angular velocities are zero. When a step input of the attitude occurs, the angular velocities quickly change to trigger a variation of the system attitude. From the plot, it can be seen that θ_3 presents a small overshoot in the last manoeuvre when it restores its attitude back to zero degrees. But its response is faster than the previous two angles as it quickly reaches the steady-state response.

Two additional plots of the static error and its derivative are given in figs. 8 and 9, which show that the error always goes to zero after a quick manoeuvre. The control inputs given to the system are depicted in fig. 10 for all three axes. The plots show that the inputs along x and y axes require more energy to cope with the sudden change in attitude compared to the input in the $z - th$, as their amplitudes are almost a thousand times higher. Moreover, the system is able to cope with these high-frequency control laws. This also explains the robustness of the PD controller on rejecting the influence of the disturbances T_{d_i} introduced in the system. These disturbances are small compared to actual inputs T_{c_i}. Additionally, if actuator dynamics is modelled and included in simulation, saturation and delays could influence the final response of the system.

2.2 Quaternions

For the system written in quaternion form, the gains are presented in eq. (55). The results for the quaternion tracking are depicted in fig. 11. The first three quaternions follow the commanded values and just like in the Euler angles, no oscillations occur. Moreover, all four quaternions are between 0 and 1, which takes into account the normalisation rule in eq. (14).

$$K_p = \begin{bmatrix} 200 \\ 200 \\ 0.5 \end{bmatrix} \qquad K_d = \begin{bmatrix} 700 \\ 500 \\ 1 \end{bmatrix} \tag{55}$$

The error and derivative of the error quaternions is presented in figs. 12 and 13. When a step input of the desired quaternions occur, the error quaternion suffers a drastic difference. However, the controller is able to reduce this error to zero in all three axes. The angular velocities also follow the same behaviour whenever this change happens. Moreover, the amplitude of the control torques is

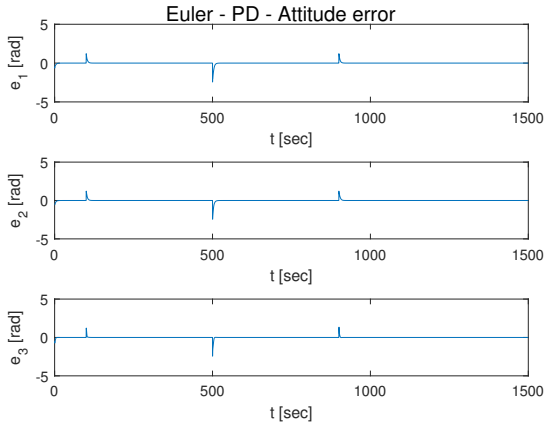


Figure 8: Attitude error of the satellite with PD controller - Euler angles

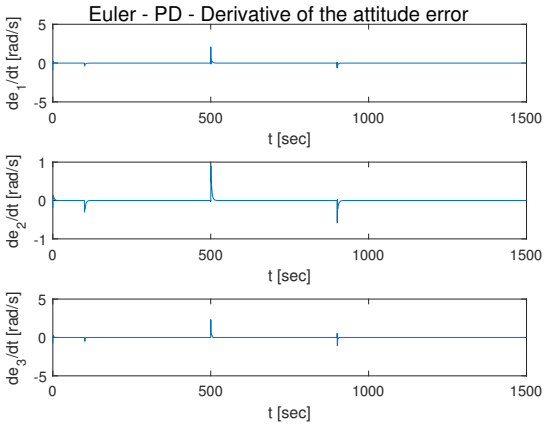


Figure 9: Derivative of attitude error of the satellite with PD controller - Euler angles

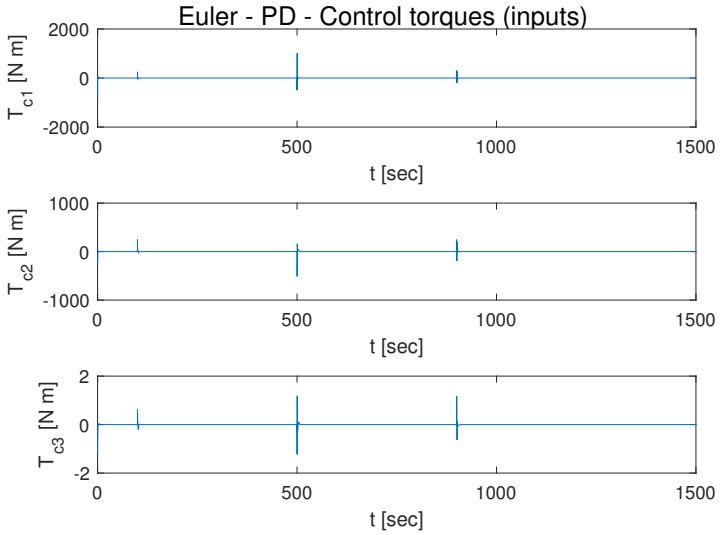
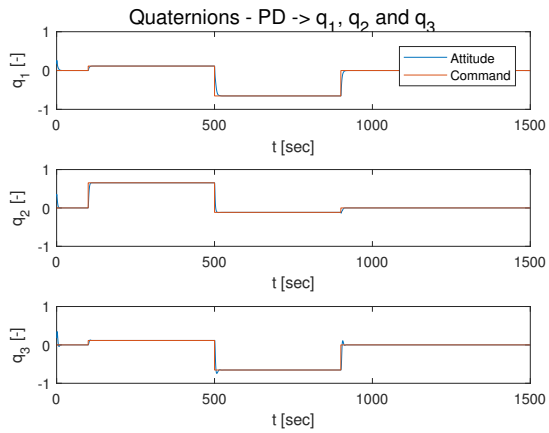
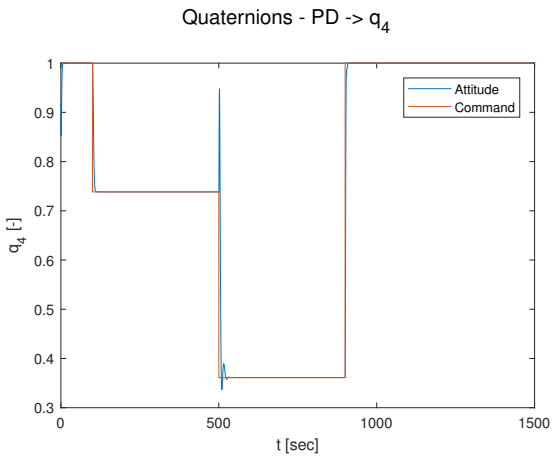


Figure 10: Control inputs of the satellite with PD controller - Euler angles

higher to command the first two angular velocities and lower to control the third. However, they are smaller when compared to the control with Euler angles, which is already an advantage as it requires less energy from the actuators to perform such extreme maneuvers.



(a) First 3 quaternions



(b) 4th quaternion

Figure 11: Quaternions of the satellite with PD controller

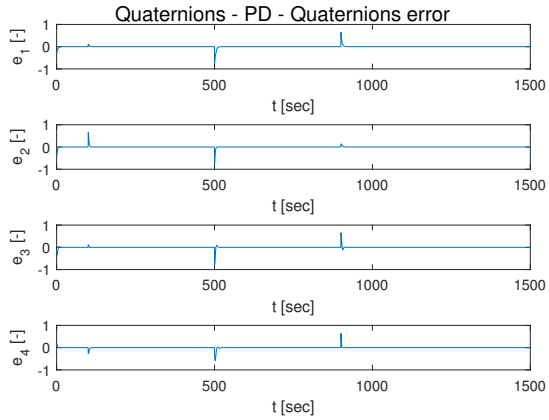


Figure 12: Attitude error of the satellite with PD controller - Quaternions

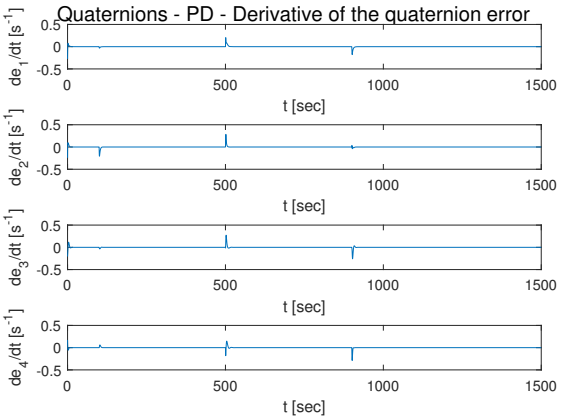


Figure 13: Derivative of the attitude error of the satellite with PD controller - Quaternions

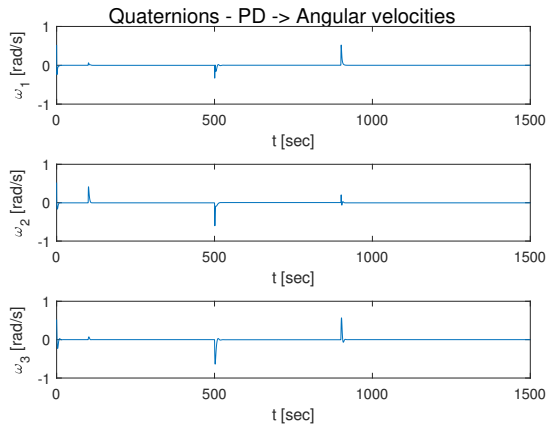


Figure 14: Angular velocity of the satellite with PD controller - Quaternions

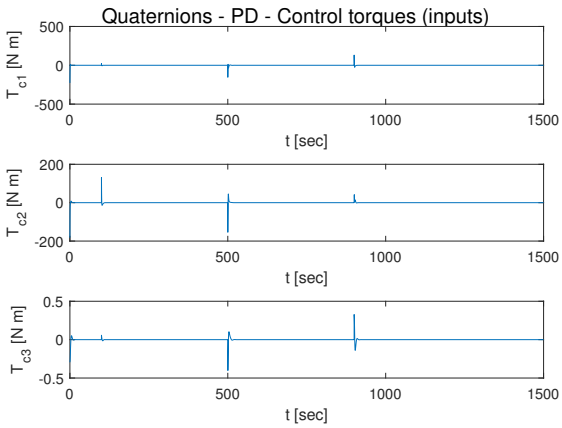


Figure 15: Control inputs of the satellite with PD controller - Quaternions

3 Non-Linear Dynamic Inversion (NDI)

The limitations of linear controllers on non-linear systems lead to the study and development of more robust control algorithms that can reach wider regions of the system's state-space. The following sections describe the design of non-linear controllers applied in spacecraft dynamics and attitude control.

Non-Linear Dynamic Inversion or Feedback Linearisation (FBL) is a type non-linear control approach that attempts to remove the non-linearities of a system written eq. (56), where the inputs are considered to be linear. One of the assumptions on this method is equality of the number of inputs and outputs of the system, i.e., the number of outputs is afine with number of inputs. Hereafter, the control variables ($\vec{cv}$) are the outputs of the system. The reader should note that the state vector has dimension n and the number of inputs/outputs is equal to m.

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + G(\mathbf{x})\mathbf{u} \\ \mathbf{y} = \mathbf{h}(\mathbf{x}) \end{cases} \tag{56}$$

The first step of this control design is to determine the relative degree of the system r_i, where $i \in \{1, ..., m\}$ is the index that represents the control variable. The goal is to compute the time derivative of these variables recursively r_i times until an explicit dependency with the input is obtained. This step is denoted as the input-output linearization. An example of this procedure for the equations derived in section 1 is presented in appendix appendix A. The reader should note that $\dot{\vec{\omega}}$ already depends in the inputs $\vec{T_c}$ directly. The sum of all the relative degrees of the system r (hereby stated as the total relative degree) must be lower or equal than the dimension of the state vector n, i.e, $r = \sum_{i=1}^{m} r_i \leq n$. If $r < n$, then $n - r$ new coordinates need to be constructed to cope with the "unobservable" internal dynamics of the system.

Thus, for each equation i, the new coordinates $\phi_j^i = cv_j^i$, where $j \in \{1, ..., r_i\}$, can be written as follows in eq. (57). The index j yields the number of times equation was differentiated in time.

$$\begin{cases} \phi_1^i(\mathbf{x}) & = L_f^0 h_i(\mathbf{x}) = h_i(\mathbf{x}) \\ \phi_2^i(\mathbf{x}) & = L_f^1 h_i(\mathbf{x}) \\ \vdots & \\ \phi_{r_i}^i(\mathbf{x}) & = L_f^{r_i} h_i(\mathbf{x}) \end{cases} \tag{57}$$

where $L_f^k h_i$ is the r-th order Lie derivative of h_i. Writing the previous coordinates into a companion form, yields:

$$\begin{cases} \dot{\phi}_1^i = \phi_2^i(\mathbf{x}) \\ \vdots \\ \dot{\phi}_{r_i-1}^i = \phi_{r_i}^i(\mathbf{x}) \\ \dot{\phi}_{r_i}^i = L_f^{r_i} h_i(\mathbf{x}) + \sum_{j=1}^{m} L_{g_j} L_f^{r_i-1} h_i(\mathbf{x}) u_j \end{cases} \tag{58}$$

Note that up until now, each output $h_i = y_i = cv_i$ is treated as a sub-system of the system eq. (56). Collecting the last equation from all control variables, yields the following set of equations:

$$\begin{cases} \dot{\phi}_{r_1}^1 & = L_f^{r_1} h_1(\mathbf{x}) + \sum_{j=1}^{m} L_{g_j} L_f^{r_1-1} h_1(\mathbf{x}) u_j \\ \dot{\phi}_{r_2}^2 & = L_f^{r_2} h_2(\mathbf{x}) + \sum_{j=1}^{m} L_{g_j} L_f^{r_2-1} h_2(\mathbf{x}) u_j \\ & \vdots \\ \dot{\phi}_{r_m}^m & = L_f^{r_m} h_m(\mathbf{x}) + \sum_{j=1}^{m} L_{g_j} L_f^{r_m-1} h_m(\mathbf{x}) u_j \end{cases} \tag{59}$$

which in vector form is given by $\mathbf{v}(\mathbf{x}) = \mathbf{b}(\mathbf{x}) + A(\mathbf{x})\mathbf{u}$ where:

$$\mathbf{v} = \begin{bmatrix} \dot{\phi}^1_{r_1} \\ \dot{\phi}^2_{r_2} \\ \vdots \\ \dot{\phi}^m_{r_m} \end{bmatrix} = \begin{bmatrix} \frac{d^{r_1} h_1(\mathbf{x})}{dt^{r_1}} \\ \frac{d^{r_2} h_2(\mathbf{x})}{dt^{r_2}} \\ \vdots \\ \frac{d^{r_m} h_m(\mathbf{x})}{dt^{r_m}} \end{bmatrix} \tag{60}$$

$$\mathbf{b}(\mathbf{x}) = \begin{bmatrix} L_f^{r_1} h_1(\mathbf{x}) \\ L_f^{r_2} h_2(\mathbf{x}) \\ \vdots \\ L_f^{r_m} h_m(\mathbf{x}) \end{bmatrix} \tag{61}$$

$$A(\mathbf{x}) = \begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(\mathbf{x}) & L_{g_2} L_f^{r_1-1} h_1(\mathbf{x}) & \dots & L_{g_m} L_f^{r_1-1} h_1(\mathbf{x}) \\ L_{g_1} L_f^{r_2-1} h_2(\mathbf{x}) & L_{g_2} L_f^{r_2-1} h_2(\mathbf{x}) & \dots & L_{g_m} L_f^{r_2-1} h_2(\mathbf{x}) \\ \vdots & \vdots & \vdots & \vdots \\ L_{g_1} L_f^{r_m-1} h_m(\mathbf{x}) & L_{g_2} L_f^{r_m-1} h_m(\mathbf{x}) & \dots & L_{g_m} L_f^{r_m-1} h_m(\mathbf{x}) \end{bmatrix} \tag{62}$$

Inverting the equation, the control law of $\mathbf{u}$ is solved:

$$\mathbf{u} = A^{-1}(\mathbf{x})[\mathbf{v}(\mathbf{x}) - \mathbf{b}(\mathbf{x})] \tag{63}$$

Using this controller, the system is completely linearized and the m equations are decoupled from each other, i.e.,

$$\frac{d^{r_i}(cv_i)}{dt^{r_i}} = v_i \tag{64}$$

The virtual control, denoted by $\mathbf{v}$, establishes the control law that can be used in a stability or tracking problem. In this case, since it is a tracking problem using a PD controller:

$$v_i = K_{p_i} e_i + K_{d_i} \frac{de_i}{dt} \tag{65}$$

where K_{p_i} and K_{d_i} are the proportional and derivative gains. Moreover, e_i is the static error, just as presented in section 2. For the NDI loop and its variants presented in next two sections, the quaternion tracking problem is done by means of eq. (51). A diagram of this method is presented in fig. 16.

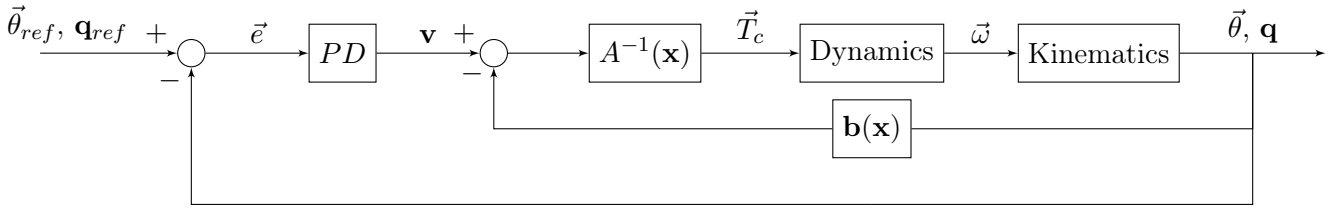


Figure 16: Block diagram of the Spacecraft Attitude Control using NDI control

3.1 Euler Angles

To design the NDI loop, control variables (cv) must be chosen. Since the goal is to control the attitude angles of the spacecraft, the chosen variables are the Euler angles $\vec{cv} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}$. Thus, these variables are the output of the system, defined as follows:

$$\vec{cv} = \mathbf{y} = H\mathbf{x} = \underbrace{\begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}}_{H} \mathbf{x} \tag{66}$$

where $\mathbf{I}_{3\times3}$ is the 3×3 identity matrix and $\mathbf{0}_{3\times3}$ is another 3×3 matrix with zero entries.

These variables do not present a direct dependence in the inputs. Thus, higher order derivatives with respect to time must be taken. By computing the first derivative yields:

$$\frac{d(\vec{cv})}{dt} = \frac{d(H\mathbf{x})}{dt} = H \underbrace{\frac{d\mathbf{x}}{dt}}_{\dot{\mathbf{x}}} = Hf(\mathbf{x}) + HG(\mathbf{x})\mathbf{u} \tag{67}$$

In order to interpret the above result, an assessment of the terms $Hf(\mathbf{x})$ and $HG(\mathbf{x})$ can be done in order to make a bridge with the teoretical results. In the scalar case, the following properties of the Lie derivatives are verified in the scalar case:

$$L_f h = \nabla h \cdot f \tag{68}$$
$$L_G L_f h = \nabla(L_f \cdot h)G \tag{69}$$

An more generalised result of the first property can be obtained as follows:

$$L_f^k h = L_f(L_f^{k-1}h) \tag{70}$$

Extending to the vector case, it gives:

$$H\mathbf{f}(\mathbf{x}) = L_{\mathbf{f}}\mathbf{h} = L_{\mathbf{f}}\mathbf{y} = D\mathbf{y} \cdot \mathbf{f}(\mathbf{x}) \tag{71}$$

where $\mathbf{h} = \mathbf{y}$ and $D\mathbf{y}$ is the matrix derivative of the output variables with respect to the state variables, since:

$$D\mathbf{y} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \nabla y_1 \\ \nabla y_2 \\ \nabla y_3 \end{bmatrix} = \begin{bmatrix} \nabla \theta_1 \\ \nabla \theta_2 \\ \nabla \theta_3 \end{bmatrix} = H \tag{72}$$

with $\nabla = \frac{\partial}{\partial \mathbf{x}}$.

Similarly, the same extension can be made for the term $HG(\mathbf{x})$:

$$HG(\mathbf{x}) = L_g\mathbf{h} = L_g(L_{\mathbf{f}}^0\mathbf{h}) = D\mathbf{y} \cdot G(\mathbf{x}) \tag{73}$$

Recalling from eq. (67), the following results of those terms are:

$$\begin{aligned} H\mathbf{f}(\mathbf{x}) &= \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \mathbf{f}(\mathbf{x}) \\ &= \underbrace{\begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{3\times3} & N(\vec{\theta}) \\ \mathbf{0}_{3\times3} & -\hat{J}^{-1}\Omega(\vec{\omega})\hat{J} \end{bmatrix}}_{N(\vec{(}\theta))\cdot\vec{\omega}} \begin{bmatrix} \vec{\theta} \\ \vec{\omega} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ -\hat{J} & -\hat{J} & \mathbf{0}_{3\times3} \end{bmatrix}}_{\mathbf{0}_{3\times9}} \begin{bmatrix} \vec{T}_d \\ 3n^2\Omega(a_3)\hat{J}a_3 \\ -nR^{-1}C \end{bmatrix} \\ &= N(\vec{\theta}) \cdot \vec{\omega} \end{aligned} \tag{74}$$

$$H\mathbf{g}(\mathbf{x}) = \underbrace{\begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{3\times3} \\ -\hat{J} \end{bmatrix}}_{\mathbf{0}_{3\times3}} \vec{T}_c = \mathbf{0}_{3\times1} \tag{75}$$

Thus, the first derivative of the control variable is given by:

$$\frac{d(\vec{cv})}{dt} = N(\vec{\theta}) \cdot \vec{\omega} \tag{76}$$

which shows no dependency on the inputs. Thus, a second derivative with respect to time must be taken:

$$\frac{d^2(\vec{cv})}{dt^2} = \frac{d}{dt}[H\dot{\mathbf{x}}] = \frac{d}{dt}[H\mathbf{f}(\mathbf{x}) + \underbrace{HG(\mathbf{x})}_{=\mathbf{0}_{3\times 3}}\mathbf{u}] = \frac{d}{dt}[H\mathbf{f}(\mathbf{x})] \tag{77}$$

Because H is a matrix with scalar entries, it can be taken outside the time derivative, which yields the following step:

$$\frac{d}{dt}[H\mathbf{f}(\mathbf{x})] = H\frac{d}{dt}[\mathbf{f}(\mathbf{x})] \tag{78}$$

Using the chain rule, the time derivative of $\mathbf{f}(\mathbf{x})$ can be written as:

$$\frac{d}{dt}[\mathbf{f}(\mathbf{x})] = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \underbrace{\frac{d\mathbf{x}}{dt}}_{\dot{x}} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\dot{\mathbf{x}} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}[\mathbf{f}(\mathbf{x}) + G(\mathbf{x})\mathbf{u}] \tag{79}$$

Substituting the previous two results in eq. (77), the second derivative of the control variables can written as function of Lie derivatives:

$$\frac{d^2(\vec{cv})}{dt^2} = H\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}[\mathbf{f}(\mathbf{x}) + G(\mathbf{x})\mathbf{u}] = \underbrace{\frac{\partial (H\mathbf{f}(\mathbf{x}))}{\partial \mathbf{x}}}_{=\frac{\partial}{\partial \mathbf{x}}(L_{\mathbf{f}}\mathbf{y})}[\mathbf{f}(\mathbf{x}) + G(\mathbf{x})\mathbf{u}] \tag{80}$$

$$= \frac{\partial (L_{\mathbf{f}}\mathbf{y})}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}) + \frac{\partial (L_{\mathbf{f}}\mathbf{y})}{\partial \mathbf{x}}G(\mathbf{x})\mathbf{u} \tag{81}$$

$$= L_{\mathbf{f}}^2\mathbf{y} + L_G(L_{\mathbf{f}}\mathbf{y})\mathbf{u} \tag{82}$$

because

$$\frac{\partial (L_{\mathbf{f}}\mathbf{y})}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}) = D(L_{\mathbf{f}}\mathbf{y}) = L_{\mathbf{f}}^2\mathbf{y} \tag{83}$$

$$\frac{\partial (L_{\mathbf{f}}\mathbf{y})}{\partial \mathbf{x}}G(\mathbf{x}) = D(L_{\mathbf{f}}\mathbf{y})G(\mathbf{x}) = L_G(L_{\mathbf{f}}\mathbf{y}) \tag{84}$$

In order to compute the second time derivative of the control variables, the above Lie derivatives must be determined. For Euler angles, they are given as follows:

$$L_{\mathbf{f}}^2 h = \frac{\partial}{\partial \mathbf{x}}[N(\vec{\theta})\cdot\vec{\omega}]\mathbf{f}(\mathbf{x}) \tag{85}$$

$$L_{\mathbf{g}}L_{\mathbf{f}}\mathbf{y} = \frac{\partial}{\partial \mathbf{x}}[N(\vec{\theta})\cdot\vec{\omega}]\mathbf{g}(\mathbf{x}) \tag{86}$$

where

$$\frac{\partial}{\partial \mathbf{x}}[N(\vec{\theta})\cdot\vec{\omega}] = \frac{\partial}{\partial \mathbf{x}}\left\{\begin{bmatrix} 1 & \sin(\theta_1)\tan(\theta_2) & \cos(\theta_1)\tan(\theta_2) \\ 0 & \cos(\theta_1) & -\sin(\theta_1) \\ 0 & \frac{\sin(\theta_1)}{\cos(\theta_2)} & \frac{\cos(\theta_1)}{\cos(\theta_2)} \end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}\right\}$$

$$= \begin{bmatrix} (\cos(\theta_1)\omega_2 - \sin(\theta_1)\omega_3)\tan(\theta_2) & \frac{\sin(\theta_1)\omega_2+\cos(\theta_1)\omega_3}{\cos^2(\theta_2)} & 0 & 1 & \sin(\theta_1)\tan(\theta_2) & \cos(\theta_1)\tan(\theta_2) \\ -\sin(\theta_1)\omega_2 - \cos(\theta_1)\omega_3 & 0 & 0 & 0 & \cos(\theta_1) & -\sin(\theta_1) \\ \frac{\cos(\theta_1)\omega_2-\sin(\theta_1)\omega_3}{\cos(\theta_2)} & \frac{(\sin(\theta_1)\omega_2+\cos(\theta_1)\omega_3)\tan(\theta_2)}{\cos(\theta_2)} & 0 & 0 & \frac{\sin(\theta_1)}{\cos(\theta_2)} & \frac{\cos(\theta_1)}{\cos(\theta_2)} \end{bmatrix} \tag{87}$$

Therefore, the second derivative of the control variables can be written in the following manner:

$$\frac{d^2(\vec{cv})}{dt^2} = \frac{\partial}{\partial \mathbf{x}}[N(\vec{\theta})\cdot\vec{\omega}]\left\{\begin{bmatrix}\mathbf{0}_{3\times 3} & N(\vec{\theta}) \\ \mathbf{0}_{3\times 3} & -\hat{J}^{-1}\Omega(\vec{\omega})\hat{J}\end{bmatrix}\begin{bmatrix}\vec{\theta}\\ \vec{\omega}\end{bmatrix} + \begin{bmatrix}\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_{3\times 3} \\ -\hat{J} & -\hat{J} & \mathbf{0}_{3\times 3}\end{bmatrix}\begin{bmatrix}\vec{T}_d \\ 3n^2\Omega(a_3)\hat{J}a_3 \\ -nR^{-1}C\end{bmatrix}\right\} +$$

$$+ \frac{\partial}{\partial \mathbf{x}}[N(\vec{\theta})\cdot\vec{\omega}]\begin{bmatrix}\mathbf{0}_{3\times 3} \\ -\hat{J}\end{bmatrix}\vec{T}_c \tag{88}$$

which explicitly depends on control input $\mathbf{u} = \vec{T}_c$. Considering $\frac{d^2(c\vec{v})}{dt^2} = \mathbf{v}$, i.e., the virtual control and solving with respect to the control input, yields:

$$\mathbf{u} = (L_{\mathbf{g}}(L_{\mathbf{f}}\mathbf{y}))^{-1}\left(\frac{d^2(c\vec{v})}{dt^2} - L_{\mathbf{f}}^2\mathbf{y}\right) \tag{89}$$

which is the same result as the one obtained in eq. (63), where:

$$A^{-1} = (L_{\mathbf{g}}(L_{\mathbf{f}}\mathbf{y}))^{-1} = \left\{\frac{\partial}{\partial \mathbf{x}}[N(\vec{\theta})\cdot\vec{\omega}]\begin{bmatrix}\mathbf{0}_{3\times3}\\ -\hat{J}\end{bmatrix}\right\}^{-1} \tag{90}$$

$$\mathbf{b} = L_{\mathbf{f}}^2\mathbf{y} = \frac{\partial}{\partial \mathbf{x}}[N(\vec{\theta})\cdot\vec{\omega}]\left(\begin{bmatrix}\mathbf{0}_{3\times3} & N(\vec{\theta})\\ \mathbf{0}_{3\times3} & -\hat{J}^{-1}\Omega(\vec{\omega})\hat{J}\end{bmatrix}\begin{bmatrix}\vec{\theta}\\ \vec{\omega}\end{bmatrix} + \begin{bmatrix}\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3}\\ -\hat{J} & -\hat{J} & \mathbf{0}_{3\times3}\end{bmatrix}\begin{bmatrix}\vec{T}_d\\ 3n^2\Omega(a_3)\hat{J}a_3\\ -nR^{-1}C\end{bmatrix}\right) \tag{91}$$

The whole expression is written as follows:

$$\mathbf{u} = \left\{\frac{\partial}{\partial \mathbf{x}}[N(\vec{\theta})\cdot\vec{\omega}]\begin{bmatrix}\mathbf{0}_{3\times3}\\ -\hat{J}\end{bmatrix}\right\}^{-1}\left\{\frac{d^2c\vec{v}}{dt^2} - \frac{\partial}{\partial \mathbf{x}}[N(\vec{\theta})\cdot\vec{\omega}]\left(\begin{bmatrix}\mathbf{0}_{3\times3} & N(\vec{\theta})\\ \mathbf{0}_{3\times3} & -\hat{J}^{-1}\Omega(\vec{\omega})\hat{J}\end{bmatrix}\begin{bmatrix}\vec{\theta}\\ \vec{\omega}\end{bmatrix} + \right.\right.$$
$$\left.\left. + \begin{bmatrix}\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3}\\ -\hat{J} & -\hat{J} & \mathbf{0}_{3\times3}\end{bmatrix}\begin{bmatrix}\vec{T}_d\\ 3n^2\Omega(a_3)\hat{J}a_3\\ -nR^{-1}C\end{bmatrix}\right)\right\} \tag{92}$$

The results of the attitude tracking problem with NDI control is presented in figure fig. 17. In this case, the NDI shows an increase of performance as the response seems faster and with small overshot. The steady state error quickly converges to zero as seen in fig. 19. Additionally, the controller responds to the high frequency commands by requesting control torques with lower amplitude when compared with the PD controller, in the x and y body axis.

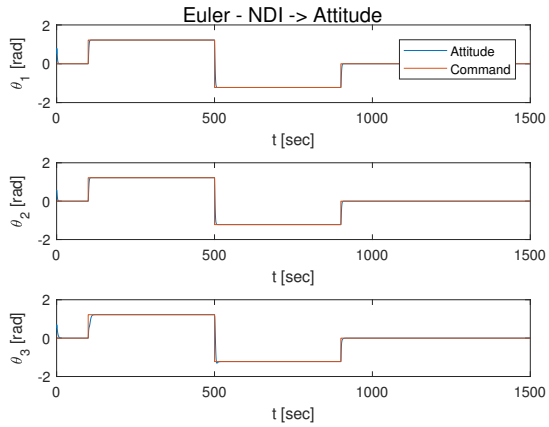


Figure 17: Euler angles of the satellite with NDI controller

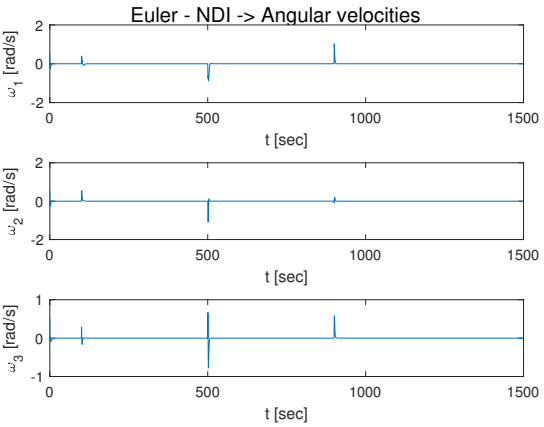


Figure 18: Angular velocities of the satellite with NDI controller - Euler angles

3.2 Quaternions

For the quaternion case, the derivation is the same as for the Euler angles. In this approach, the control variables are given by the first 3 quaternions ($c\vec{v} = \begin{bmatrix}q_1 & q_2 & q_3\end{bmatrix}^T$) and the goal is to control them. Since the fourth one is linear combination of the other two, it will not matter in the NDI control design. Thus, taking into account the states of the system defined in eq. (41):

$$c\vec{v} = \mathbf{y} = H\mathbf{x} = \begin{bmatrix}\mathbf{I}_{3\times4} & \mathbf{0}_{3\times3}\end{bmatrix} \tag{93}$$

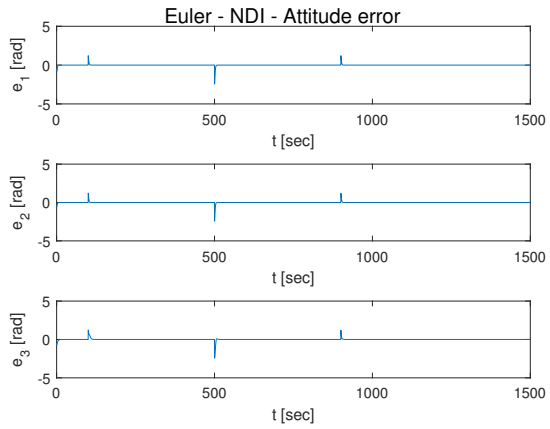


Figure 19: Attitude error of the satellite with NDI controller - Euler angles

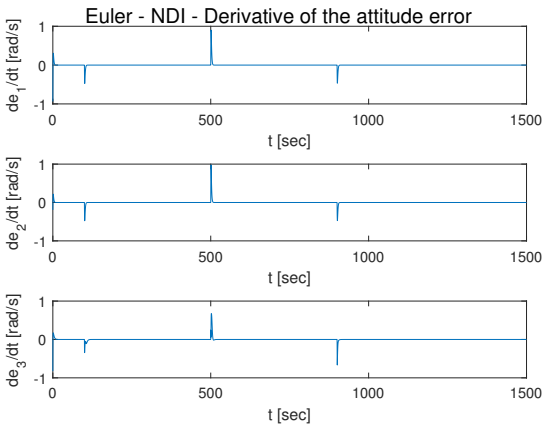


Figure 20: Derivative of attitude error of the satellite with NDI controller - Euler angles

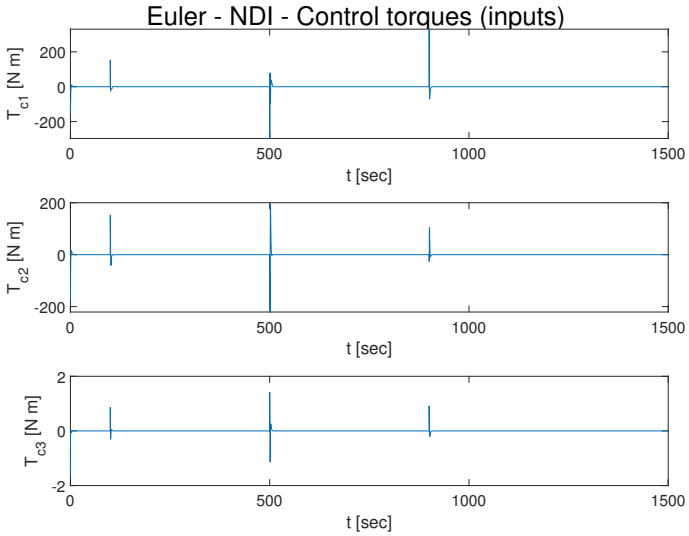


Figure 21: Control inputs of the satellite with NDI controller - Euler angles

where $\mathbf{I}_{3\times 4}$ is given by

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{94}$$

Again, the control variables do not show direct dependence in the inputs. This means that higher order time derivatives must be taken. Like in the Euler angles, similar derivations are presented. By computing the first derivative, yields eq. (67). In this case, the results of the terms $H\mathbf{f}(\mathbf{x})$ and $H\mathbf{g}(\mathbf{x})$

are determined as follows:

$$
\begin{aligned}
H\mathbf{f}(\mathbf{x}) &= \begin{bmatrix}\mathbf{I}_{3\times4} & \mathbf{0}_{4\times3}\end{bmatrix}\mathbf{f}(\mathbf{x}) \\
&= \underbrace{\begin{bmatrix}\mathbf{I}_{3\times4} & \mathbf{0}_{3\times3}\end{bmatrix}\begin{bmatrix}N(\vec{\omega}) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\hat{J}^{-1}\Omega(\vec{\omega})\hat{J}\end{bmatrix}\begin{bmatrix}\mathbf{q} \\ \vec{\omega}\end{bmatrix}}_{[N(\vec{\omega})]_{3\times4}\cdot\mathbf{q}} + \underbrace{\begin{bmatrix}\mathbf{I}_{3\times4} & \mathbf{0}_{3\times3}\end{bmatrix}\begin{bmatrix}\mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} \\ -\hat{J} & -\hat{J}\end{bmatrix}\begin{bmatrix}\vec{T}_d \\ 3n^2\Omega(a_3)\hat{J}a_3\end{bmatrix}}_{\mathbf{0}_{3\times3}} \\
&= [N(\vec{\omega})]_{3\times4}\cdot\mathbf{q} \qquad (95)\\
H\mathbf{g}(\mathbf{x}) &= \underbrace{\begin{bmatrix}\mathbf{I}_{3\times4} & \mathbf{0}_{3\times3}\end{bmatrix}\begin{bmatrix}\mathbf{0}_{4\times3} \\ -\hat{J}\end{bmatrix}}_{\mathbf{0}_{3\times3}}\vec{T}_c = \mathbf{0}_{3\times1} \qquad (96)
\end{aligned}
$$

Note that the matrix $[N(\vec{\omega})]_{3\times4}$ only includes the first 3 rows and the 4 columns of $[N(\vec{\omega})]$. Therefore, the first derivative of the control variable is given by:

$$
\frac{d(\vec{cv})}{dt} = [N(\vec{\omega})]_{3\times4}\cdot\mathbf{q} = \frac{1}{2}\begin{bmatrix}0 & \omega_3 & -\omega_2+n & \omega_1 \\ \omega_3 & 0 & \omega_1 & \omega_2+n \\ \omega_2+n & -\omega_1 & 0 & \omega_2\end{bmatrix}\begin{bmatrix}q_1 \\ q_2 \\ q_3 \\ q_4\end{bmatrix} \qquad (97)
$$

which shows no dependency on the inputs. Thus, a second derivative with respect to time is taken and the result is already derived for Euler angles in eq. (82). Following the same procedure, the Lie derivatives are given by:

$$
L_{\mathbf{f}}^2 h = \frac{\partial}{\partial\mathbf{x}}([N(\vec{\omega})]_{3\times4}\cdot\mathbf{q})\mathbf{f}(\mathbf{x}) \qquad (98)
$$

$$
L_G L_{\mathbf{f}}\mathbf{y} = \frac{\partial}{\partial\mathbf{x}}([N(\vec{\omega})]_{3\times4}\cdot\mathbf{q})G(\mathbf{x}) \qquad (99)
$$

where

$$
\begin{aligned}
\frac{\partial}{\partial\mathbf{x}}([N(\vec{\omega})]_{3\times4}\cdot\mathbf{q}) &= \frac{\partial}{\partial\mathbf{x}}\left\{\frac{1}{2}\begin{bmatrix}0 & \omega_3 & -\omega_2+n & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2+n \\ \omega_2+n & -\omega_1 & 0 & \omega_2\end{bmatrix}\begin{bmatrix}q_1 \\ q_2 \\ q_3 \\ q_4\end{bmatrix}\right\} \\
&= \begin{bmatrix}0 & \omega_3 & n-\omega_2 & 0 & 0 & -1 & 1 \\ -\omega_3 & 0 & \omega_1 & \omega_2+n & 1 & 1 & -1 \\ \omega_2-n & -\omega_1 & 0 & \omega_3 & -1 & 1 & 1\end{bmatrix} \qquad (100)
\end{aligned}
$$

Therefore, the second derivative of the control variables can be written in the following manner:

$$
\begin{aligned}
\frac{d^2(\vec{cv})}{dt^2} = \frac{\partial}{\partial\mathbf{x}}([N(\vec{\omega})]_{3\times4}\cdot\mathbf{q})\left\{\begin{bmatrix}N(\vec{\omega}) & \mathbf{0}_{4\times3} \\ \mathbf{0}_{4\times3} & -\hat{J}^{-1}\Omega(\vec{\omega})\hat{J}\end{bmatrix}\begin{bmatrix}\mathbf{q} \\ \vec{\omega}\end{bmatrix} + \begin{bmatrix}\mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} \\ -\hat{J} & -\hat{J}\end{bmatrix}\begin{bmatrix}\vec{T}_d \\ 3n^2\Omega(a_3)\hat{J}a_3\end{bmatrix}\right\} + \\
+ \frac{\partial}{\partial\mathbf{x}}([N(\vec{\omega})]_{3\times4}\cdot\mathbf{q})\begin{bmatrix}\mathbf{0}_{4\times3} \\ -\hat{J}\end{bmatrix}\vec{T}_c \qquad (101)
\end{aligned}
$$

which explicitly depends on control input $\mathbf{u} = \vec{T}_c$. Solving for this control input, yields:

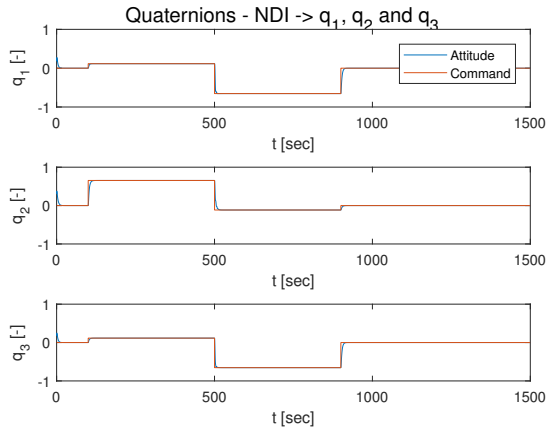
$$
\begin{aligned}
\mathbf{u} = \left\{\frac{\partial}{\partial\mathbf{x}}([N(\vec{\omega})]_{3\times4}\cdot\mathbf{q})\begin{bmatrix}\mathbf{0}_{4\times3} \\ -\hat{J}\end{bmatrix}\right\}^{-1}\left\{\frac{d^2\vec{cv}}{dt^2} - \frac{\partial}{\partial\mathbf{x}}([N(\vec{\omega})]_{3\times4}\cdot\mathbf{q})\left(\begin{bmatrix}N(\vec{\omega}) & \mathbf{0}_{4\times3} \\ \mathbf{0}_{3\times3} & -\hat{J}^{-1}\Omega(\vec{\omega})\hat{J}\end{bmatrix}\begin{bmatrix}\mathbf{q} \\ \vec{\omega}\end{bmatrix} + \right. \\
\left. + \begin{bmatrix}\mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} \\ -\hat{J} & -\hat{J}\end{bmatrix}\begin{bmatrix}\vec{T}_d \\ 3n^2\Omega(a_3)\hat{J}a_3\end{bmatrix}\right)\right\} \qquad (102)
\end{aligned}
$$

where

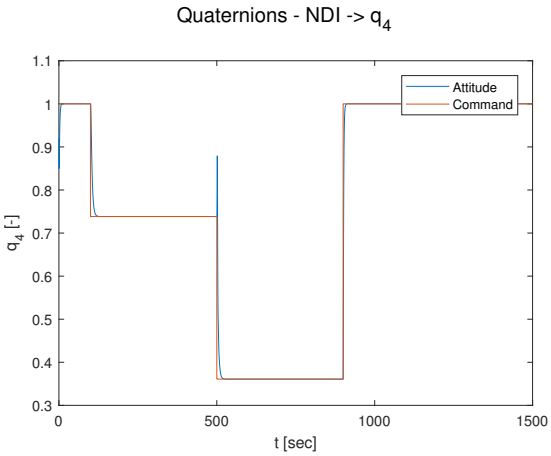
$$A^{-1} = \left\{ \frac{\partial}{\partial \mathbf{x}} ([N(\vec{\omega})]_{3\times 4} \cdot \mathbf{q}) \begin{bmatrix} \mathbf{0}_{4\times 3} \\ -\hat{J} \end{bmatrix} \right\}^{-1} \tag{103}$$

$$\mathbf{b} = \frac{\partial}{\partial \mathbf{x}} ([N(\vec{\omega})]_{3\times 4} \cdot \mathbf{q}) \left(\begin{bmatrix} N(\vec{\omega}) & \mathbf{0}_{4\times 3} \\ \mathbf{0}_{4\times 3} & -\hat{J}^{-1}\Omega(\vec{\omega})\hat{J} \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \vec{\omega} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{4\times 3} & \mathbf{0}_{4\times 3} \\ -\hat{J} & -\hat{J} \end{bmatrix} \begin{bmatrix} \vec{T}_d \\ 3n^2\Omega(a_3)\hat{J}a_3 \end{bmatrix} \right) \tag{104}$$

The derivation of the controller is similar to the one obtained with Euler angles. Again, the attitude tracking problem shows better performance than only using a PD controller. The response of this closed loop system is faster as the steady state error also converges quickly to zero and presents no oscillations. The request for control inputs (fig. 26) is still smaller than in Euler angles and PD controller applied to quaternions when high frequency commands occur. The results are shown in figure fig. 26.



(a) First 3 quaternions



(b) 4th quaternion

Figure 22: Quaternions of the satellite with NDI controller

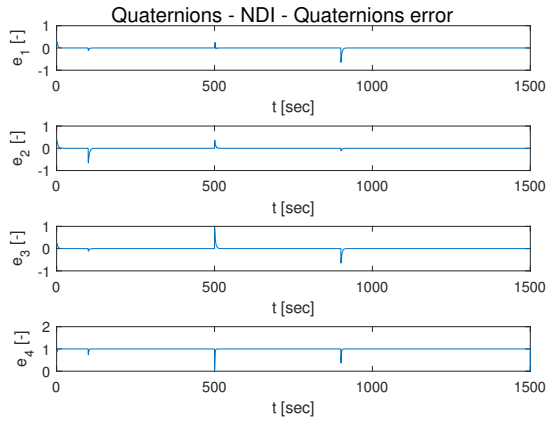


Figure 23: Attitude error of the satellite with NDI controller - Quaternions

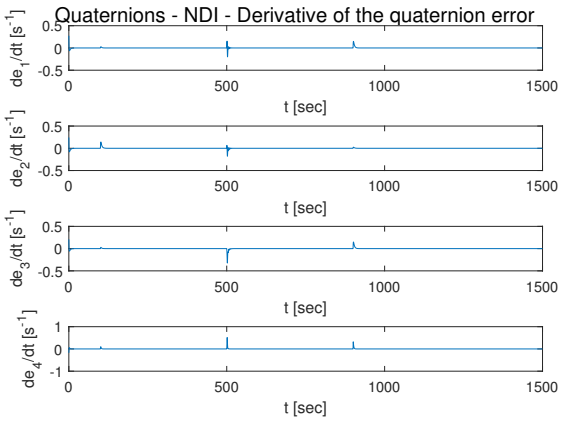


Figure 24: Derivative of the attitude error of the satellite with NDI controller - Quaternions

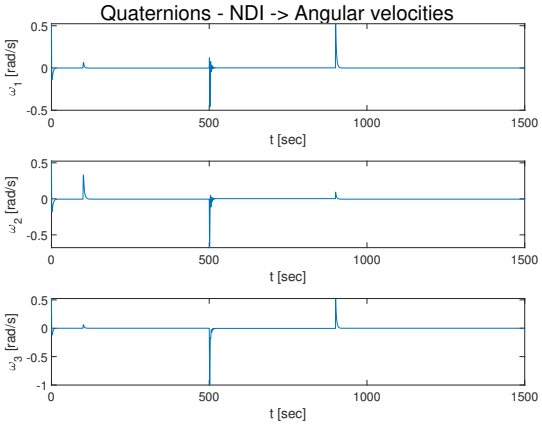


Figure 25: Angular velocity of the satellite with NDI controller - Quaternions

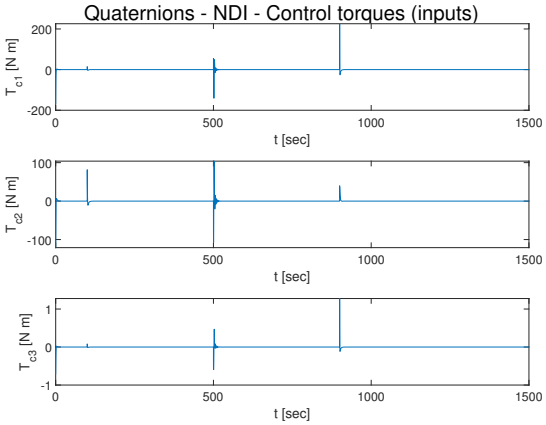


Figure 26: Control inputs of the satellite with NDI controller - Quaternions

4 NDI with time scale separation principle

Taking into account the previous examples, the attitude NDI control has different relative degree when compared to the rotational rate control. The attitude control loop has six first order differential equations (three for the rotational rates obtained in the orbital frame and another three for the coordinate transformation from this frame to the body frame). To meet full controllability of the system, i.e, such that there is no influence of any internal dynamics, the total relative degree should be equal to 6. On the other hand, the rotational rate NDI control is given by 3 differential equations, which means that the total relative degree should be 3.

The previous motivation leads to the statement of the time scale separation principle, which introduces the concept of slow and fast dynamics to control the system. Variables are said to have slow dynamics when the control effectiveness on the dynamics is low. When the control effectiveness of the dynamics is said to be high, then the dynamics are said to be fast. Usually, this principle is used when the ratio of the control effectiveness matrix is negligible, i.e., when the two differ at least by a factor of 10. This is an experimental value, although larger values are of course preferred.

The main idea of this principle is to separate the dynamics and kinematics of the system, treated as two sub-systems that work independently. The NDI control is applied twice. The outputs of the kinematic inversion, the rates, are used as inputs to the dynamic inversion. The dynamic inversion yields the inputs obtained from the actuators and feeds to the spacecraft. The block diagram of this approach is presented in figure fig. 27. The inner loop in the block diagram is given in fig. 28.

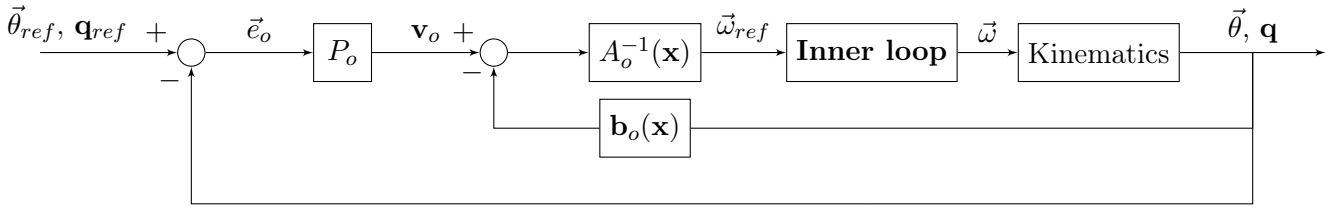


Figure 27: Block diagram of the Spacecraft Attitude Control using NDI control with time-scale separation

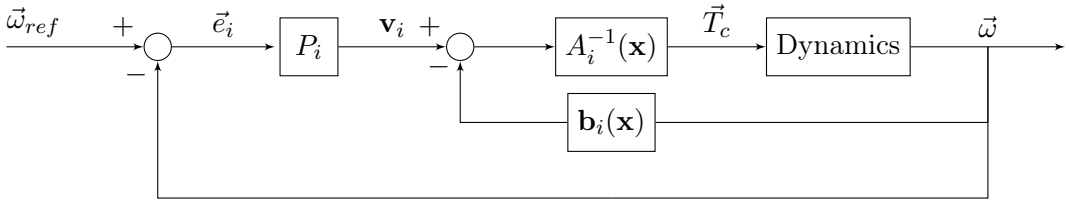


Figure 28: Inner loop of of the Spacecraft Attitude Control using NDI control with time-scale separation

Starting with the inner loop, the equations that represent the dynamics of the system are the same for both Euler angles and quaternion, except for the gravity gradient term. However, this term is not a direct input of the system. Herein, the control variables chosen to control in this loop are the angular velocities expressed in the orbital frame $\vec{cv} = \begin{bmatrix}\omega_1 & \omega_2 & \omega_3\end{bmatrix}$ and they do not depend on any input. Taking the first time derivative yields the dynamics equation presented in eq. (27), where it appears the term dependent on $\vec{T}_c$. In this case, expressing this system as in eq. (56), it yields the following:

$$\mathbf{f}(\mathbf{x}) = -\hat{J}^{-1}\Omega(\vec{\omega})(\hat{J}\cdot\vec{\omega}) + \hat{J}^{-1}(\vec{T}_d + \vec{T}_{GG}) \tag{105}$$

$$G(\mathbf{x}) = \hat{J}^{-1} \tag{106}$$

Therefore, solving for the control input $\mathbf{u} = \vec{T}_c$, the following law is obtained:

$$\mathbf{u} = (G(\mathbf{x}))^{-1}\left(\frac{d(c\vec{v}_{in})}{dt} - \mathbf{f}(\mathbf{x})\right) \tag{107}$$

where $(G(\mathbf{x}))^{-1} = A_i^{-1} = \hat{J}$ is the control effectiveness matrix and $\mathbf{f}(\mathbf{x}) = \mathbf{b}_i$. The subscript i refers to the inner loop. The virtual control defined by $\frac{d(\vec{cv})}{dt} = \mathbf{v}_i$ is obtained as output of a simple P controller:

$$\mathbf{v}_i = \vec{K}_{p_{in}} \odot \vec{e}_i \tag{108}$$

where $\vec{e}_i = \vec{\omega}_{ref} - \vec{\omega}$.

For the outer loop control, the procedure is similar for both Euler angles and quaternions, however, the equations slightly change. Starting with Euler angles, the control variables are $\vec{cv} = \begin{bmatrix}\theta_1 & \theta_2 & \theta_3\end{bmatrix}$. Since these do not depend on the input directly, they must be differentiated with respect to time, which yields the kinematics equation given ahead:

$$\frac{d(\vec{cv_{out}})}{dt} = \dot{\vec{\theta}} = N(\vec{\theta}) \cdot \vec{\omega} + nN(\vec{\theta})Ca_2 \tag{109}$$

In this case, comparing again to equation eq. (56), the following considerations can be made:

$$\mathbf{f}(\mathbf{x}) = nN(\vec{\theta})C\vec{a}_2 = A_o \tag{110}$$

$$G(\mathbf{x}) = N(\vec{\theta}) = \mathbf{b}_o \tag{111}$$

Solving for the control input $\mathbf{u} = \vec{\omega}$, it yields the following input, which is used as a reference value for rate controller:

$$\vec{\omega}_{ref} = (N(\vec{\theta}))^{-1}\left(\frac{d(\vec{cv_{out}})}{dt} - nN(\vec{\theta})C\vec{a}_2\right) \tag{112}$$

For quaternions, the procedure is similar. The control variables are the first three quaternions $\vec{cv} = \begin{bmatrix}q_1 & q_2 & q_3\end{bmatrix}^T$. These, per se, do not depend on the angular velocities. Hereafter, a first order derivative must be computed:

$$\frac{d(\vec{cv}_{out})}{dt} = [\dot{\mathbf{q}}]_{3\times 1} = [N(\vec{\omega})]_{3\times 4} \cdot \mathbf{q} \tag{113}$$

$$= \frac{1}{2}\begin{bmatrix} 0 & \omega_3 & -\omega_2 + n & \omega_1 \\ \omega_3 & 0 & \omega_1 & \omega_2 + n \\ \omega_2 - n & -\omega_1 & 0 & \omega_2 \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{114}$$

Using the left-most expression from eq. (20), the previous expression can be rewriten again in the following manner:

$$\mathbf{v}_o = \frac{d(\vec{cv}_{out})}{dt} = [\dot{\mathbf{q}}]_{3\times 1} = \begin{bmatrix}\dot{q_1} \\ \dot{q_2} \\ \dot{q_3}\end{bmatrix} = \frac{1}{2}\begin{bmatrix} q_4 & q_3 & -q_2 \\ -q_3 & q_4 & q_1 \\ q_2 & -q_1 & q_4 \end{bmatrix}\begin{bmatrix}\omega_1 \\ \omega_2 \\ \omega_3\end{bmatrix} + \frac{n}{2}\begin{bmatrix} q_3 \\ q_4 \\ -q_1 \end{bmatrix} \tag{115}$$

where for a system written in the form eq. (56) yields:

$$\mathbf{f}(\mathbf{x}) = \frac{1}{2}\begin{bmatrix} q_4 & q_3 & -q_2 \\ -q_3 & q_4 & q_1 \\ q_2 & -q_1 & q_4 \end{bmatrix} = A_o \tag{116}$$

$$G(\mathbf{x}) = \frac{n}{2}\begin{bmatrix} q_3 \\ q_4 \\ -q_1 \end{bmatrix} = \mathbf{b}_o \tag{117}$$

Therefore, the NDI control for the quaternion control law can be written in the following manner to yield the reference value of the angular velocity:

$$\vec{\omega}_{ref} = \begin{bmatrix} q_4 & q_3 & -q_2 \\ -q_3 & q_4 & q_1 \\ q_2 & -q_1 & q_4 \end{bmatrix}^{-1}\left(\frac{d(\vec{cv}_{out})}{dt} - \frac{n}{2}\begin{bmatrix} q_3 \\ q_4 \\ -q_1 \end{bmatrix}\right) \tag{118}$$

The virtual control $\frac{d(\vec{cv}_{out})}{dt}$ comes as an output of a simple P controller written in the following manner:

$$\mathbf{v} = \frac{d\vec{\omega}_{ref}}{dt} + K_{p_{out}} \odot \vec{e} \tag{119}$$

4.1 Euler angles

The current approach with Euler angles, the gains chosen for the inner and outer loops using the P controllers are given in eq. (121).

$$K_{p_{out}} = \begin{bmatrix} 1 \\ 1 \\ 0.5 \end{bmatrix} \qquad K_{p_{in}} = \begin{bmatrix} 10 \\ 10 \\ 5 \end{bmatrix} \tag{120}$$

The response of the attitude control is presented in fig. 29. Similar to the simple NDI controller, the reference attitude is correctly tracked. Whenever a quick input is requested, the control system is able to give a fast response on reducing static error. Moreover, the amplitude of the inputs T_c (fig. 33) to the system are higher when compared to the previous two approaches, requiring more energy from the actuators.

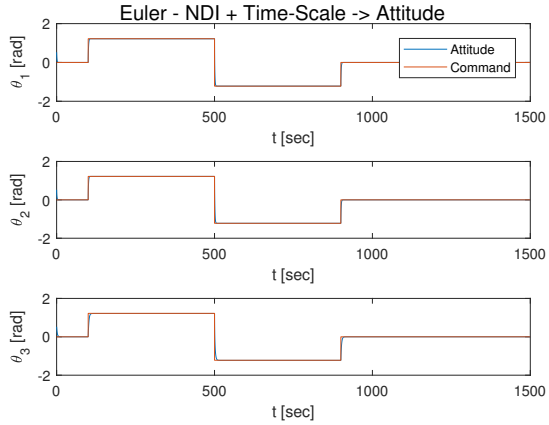


Figure 29: Euler angles of the satellite with time scale NDI controller

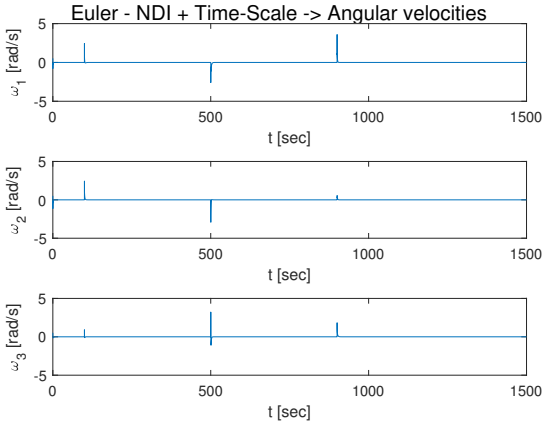


Figure 30: Angular velocities of the satellite with time scale NDI controller - Euler angles

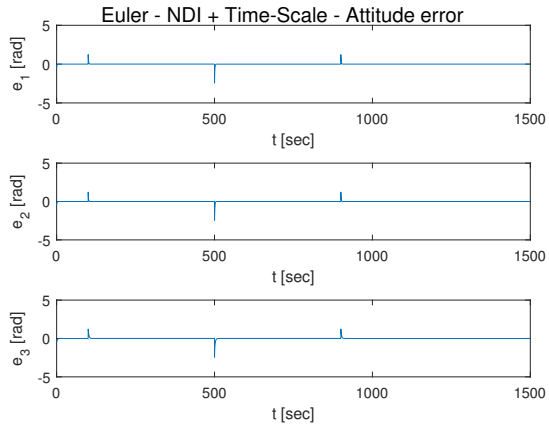


Figure 31: Attitude error of the satellite with time scale NDI controller - Euler angles

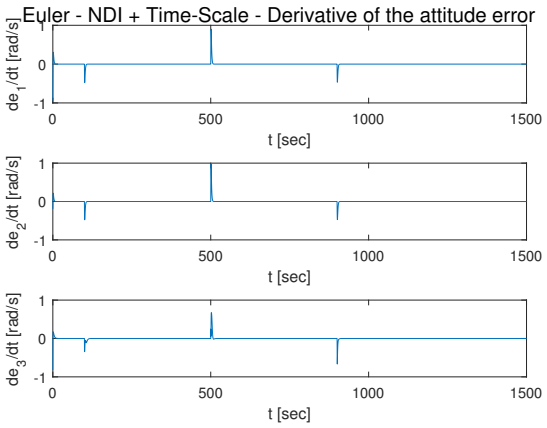


Figure 32: Derivative of attitude error of the satellite with time scale NDI controller - Euler angles

4.2 Quaternions

For the quaternion-based approach, the following gains where set:

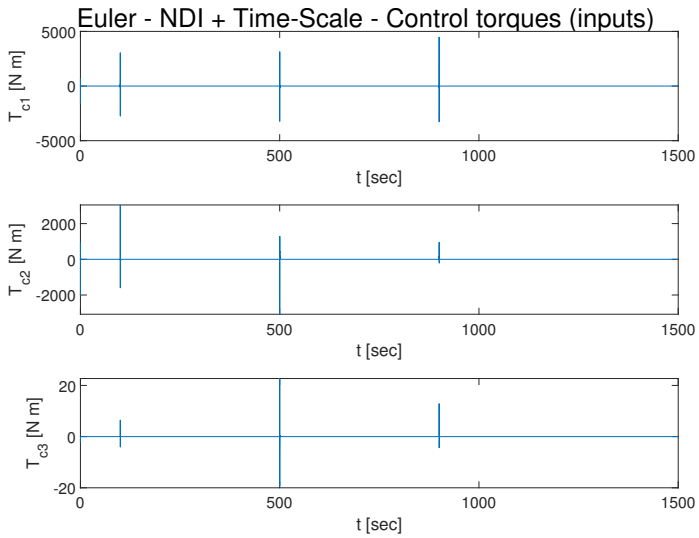
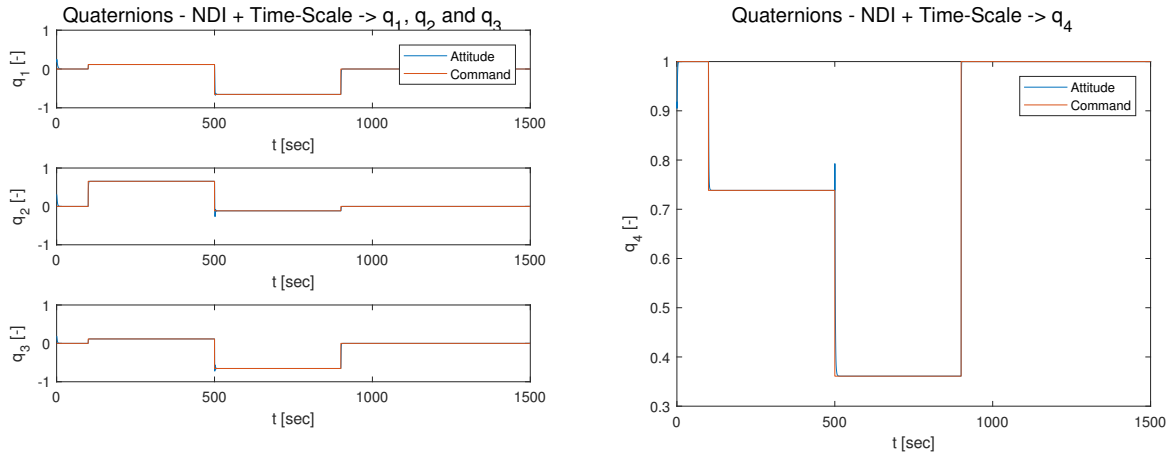


Figure 33: Control inputs of the satellite with time scale NDI controller - Euler angles

$$K_{p_{out}} = \begin{bmatrix} -0.9 \\ -0.8 \\ -1.0 \end{bmatrix} \qquad K_{p_{in}} = \begin{bmatrix} 0.4 \\ 0.6 \\ 0.7 \end{bmatrix} \tag{121}$$

The results are similar to the Euler angles approach. The quaternions track the reference values (fig. 34) as the static error goes to zero (fig. 35). Moreover, whenever a step input of a commanded quaternion is requested, large torques (fig. 33) are needed to stabilise the system, more than the previous two control approaches.



(a) First 3 quaternions

(b) 4th quaternion

Figure 34: Quaternions of the satellite with time scale NDI controller

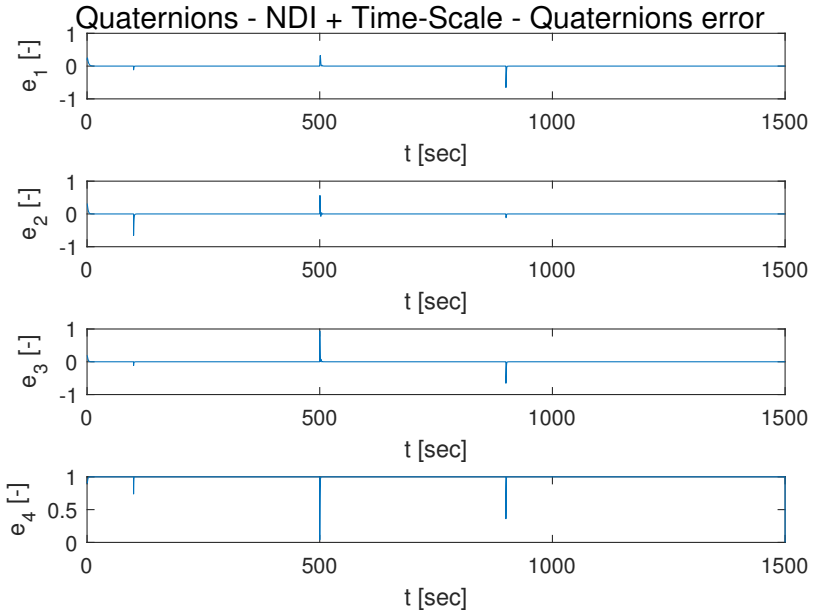


Figure 35: Attitude error of the satellite with time scale NDI controller - Quaternions

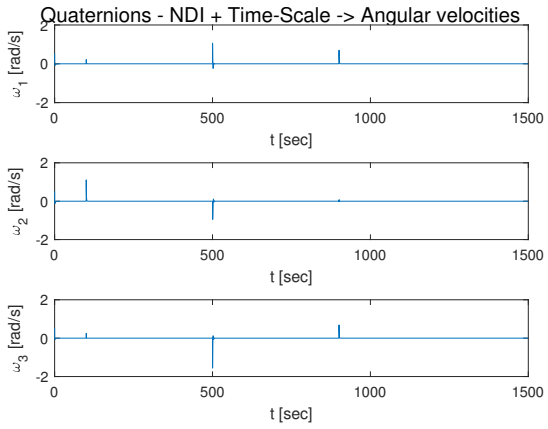


Figure 36: Angular velocity of the satellite with time scale NDI controller - Quaternions

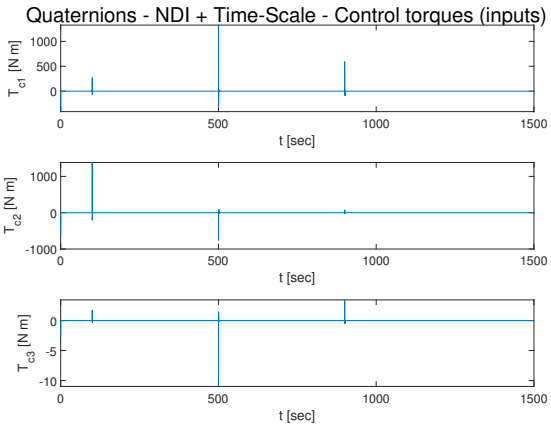


Figure 37: Control inputs of the satellite with time scale NDI controller - Quaternions

5 Incremental Non-Linear Dynamic Inversion (INDI)

The advantage of NDI is the fact that it operates accurately when a mathematical model of the system is explicitly described. Nonetheless, this is not the case in the real world, where several non-linearities are difficult to model.

The next variant of the NDI control method is called INDI (Incremental Non-Linear Dynamic Inversion). This approach takes into account a small part/subset of the system in the control design. However, this controller requires the use of incremental steps of the input.

INDI can be applied even when considers a generic non-linear system described eq. (56). By means of Taylor series expansion, the system is linearised at the current time point $t = t_0$ as follows in equation eq. (122), where $x_0 = x(t = t_0)$ and $u_0 = u(t = t_0)$.

$$\begin{aligned}\dot{\mathbf{x}} &= \underbrace{\mathbf{f}(\mathbf{x}_0, \mathbf{u}_0)}_{\dot{\mathbf{x}}_0} + \underbrace{\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_0,\mathbf{u}=\mathbf{u}_0}}_{F(\mathbf{x}_0,\mathbf{u}_0)}(\mathbf{x}-\mathbf{x}_0) + \underbrace{\left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\mathbf{x}=\mathbf{x}_0,\mathbf{u}=\mathbf{u}_0}}_{G(\mathbf{x}_0,\mathbf{u}_0)}(\mathbf{u}-\mathbf{u}_0) \\ &= \dot{\mathbf{x}}_0 + F(\mathbf{x}_0,\mathbf{u}_0)(\mathbf{x}-\mathbf{x}_0) + G(\mathbf{x}_0,\mathbf{u}_0)(\mathbf{u}-\mathbf{u}_0)\end{aligned} \tag{122}$$

Using the time-scale separation principle stated in section 4 and considering the hypothesis that the sampling rate (f_s) is sufficiently high, the next state is assumed to approximately equal to the previous one ($\mathbf{x}_0 \approx \mathbf{x}$). Therefore, the previous equation can be further simplified in the following manner:

$$\begin{aligned}\dot{\mathbf{x}} &= \dot{\mathbf{x}}_0 + F(\mathbf{x}_0,\mathbf{u}_0)\underbrace{(\mathbf{x}-\mathbf{x}_0)}_{\approx 0} + G(\mathbf{x}_0,\mathbf{u}_0)\underbrace{(\mathbf{u}-\mathbf{u}_0)}_{\Delta\mathbf{u}} \\ &\approx \dot{\mathbf{x}}_0 + G(\mathbf{x}_0,\mathbf{u}_0)\Delta\mathbf{u}\end{aligned} \tag{123}$$

where $\Delta\mathbf{u}$ is the incremental input given to the system. A choice of NDI controller for this incremental method is designed as follows

$$\Delta\mathbf{u} = G^{-1}(\mathbf{x}_0,\mathbf{u}_0)(\mathbf{v}-\dot{\mathbf{x}}_0) \tag{124}$$

where v is the virtual control input. When $\dot{\mathbf{x}}_0$ is (directly) measurable from the sensors, the only uncertainty within the INDI controller is the control effectiveness matrix $G(\mathbf{x}_0,\mathbf{u}_0)$. Therefore, the control input $\mathbf{u}_{t+1}$ in the next iteration, $k = t+1$, is updated as indicated in eq. (125), where $\mathbf{u}_t$ and $\Delta\mathbf{u}_t$ are the input and incremental control input at time $t = k$.

$$\mathbf{u}_{t+1} = \mathbf{u}_t + \Delta\mathbf{u}_t = \mathbf{u}_t + G^{-1}(\mathbf{x}_t,\mathbf{u}_t)(\mathbf{v}_t - \dot{\mathbf{x}}_t) \tag{125}$$

In order to use this incremental control approach, time-scale separation is considered again. The INDI is applied only to the dynamics, because it consists of a direct relationship between the system input (actuators $\vec{T}_c$) and the angular velocity of the spacecraft. Taking into account eq. (124), for this case, the incremental control is written as follows:

$$\Delta\vec{T}_{c_t} = G^{-1}(\vec{\omega}_t,\vec{T}_{c_t})(\mathbf{v}_t - \dot{\vec{w}}_t) \tag{126}$$

To determine the INDI control input, the control effectiveness matrix must be computed. From eq. (106), one can conclude that it is the same matrix for both Euler angles and quaternions approach, and is given by the inverse of the inertia matrix (eq. (127)).

$$G(\vec{\omega}_t,\vec{T}_{c_t}) = J^{-1} = \begin{bmatrix}\frac{1}{J_{11}} & 0 & 0\\ 0 & \frac{1}{J_{22}} & 0\\ 0 & 0 & \frac{1}{J_{33}}\end{bmatrix} \tag{127}$$

Moreover, since $\dot{\vec{\omega}}_t$ is not directly measured from a real system, simulated data of the dynamical model is obtained by means of eqs. (32) and (35) in both Euler angles and quaternions. The virtual control is

obtained as a result of a simple P controller used to track the desired angular velocities of the system as given in eq. (119).

The kinematics equation consists of a coordinate transformation between the orbital and body frames. Therefore, following the same procedure as in the previous section, the same model-based NDI is computed in the outer loop as presented in fig. 27. The virtual control is also obtained as a result of a P controller used to track the Euler angles or quaternions. The block diagram in fig. 38 summarises the inner loop of control system the includes these loops: a fast inner loop corresponds to the INDI rate control and a outer loop NDI attitude control.

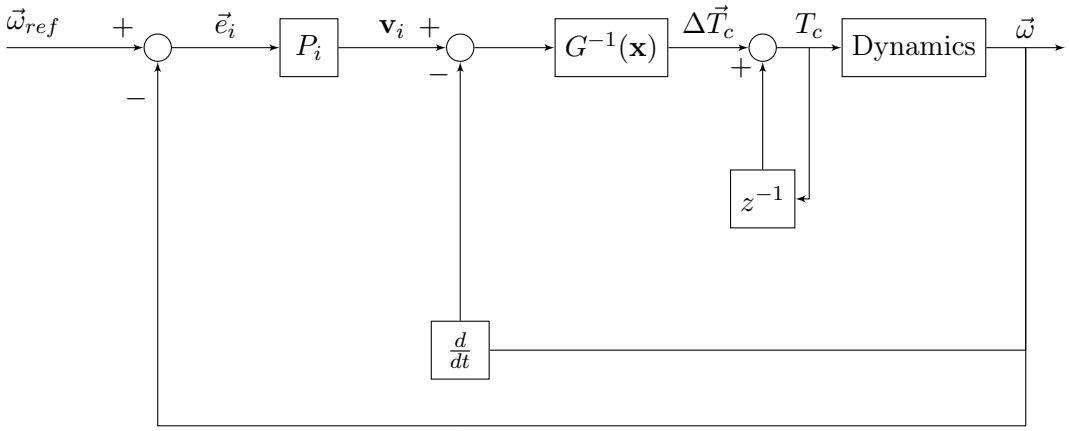


Figure 38: Inner loop of of the Spacecraft Attitude Control using INDI control with time-scale separation

5.1 Euler Angles

For Euler angles, the following gains were chosen for the inner and outer loop control:

$$K_{p_{in}} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} \qquad K_{p_{out}} = \begin{bmatrix} 0.5 \\ 0.5 \\ 1 \end{bmatrix} \tag{128}$$

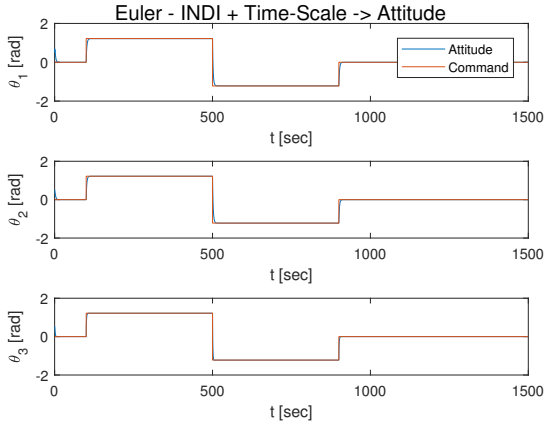


Figure 39: Euler angles of the satellite with INDI controller

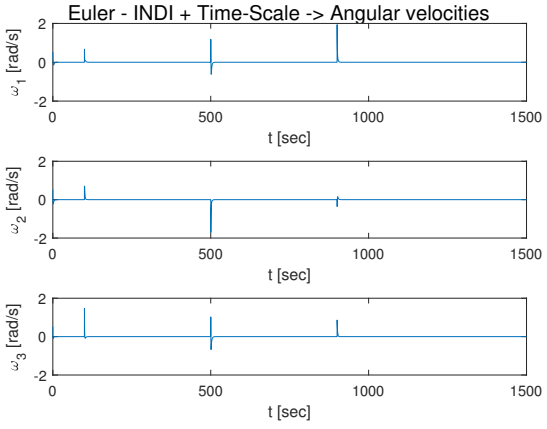


Figure 40: Angular velocities of the satellite with INDI controller - Euler angles

The results of the attitude tracking control are presented in fig. 39. The attitude follows the commanded values without oscillations. Taking into account fig. 41, the steady state error quickly goes to zero, showing good convergence properties. The INDI loop control presents quick responses to the high frequency desired attitude commands, as presented in figs. 40 and 42. The amplitude of the control inputs is still higher when using Euler angles.

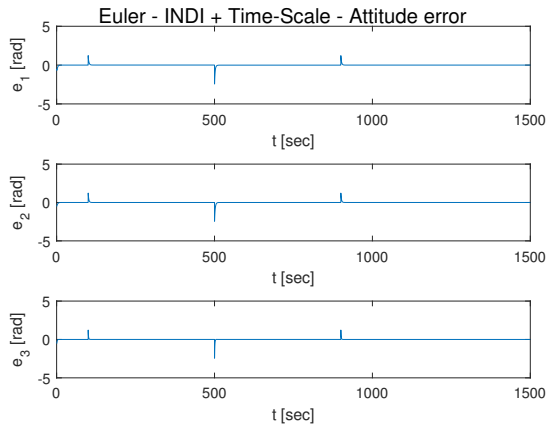


Figure 41: Attitude error of the satellite with INDI controller - Euler angles

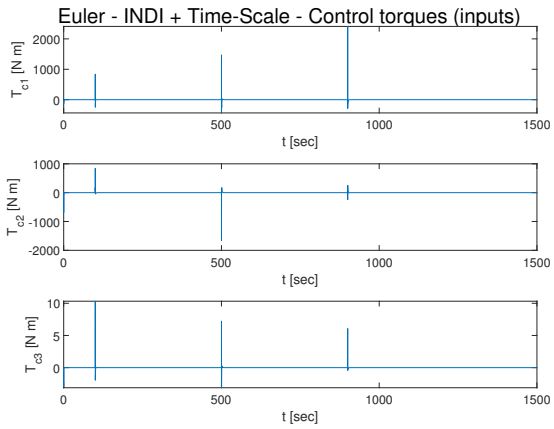
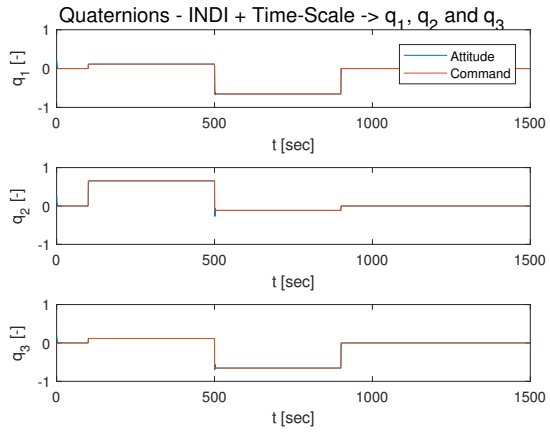


Figure 42: Control inputs of the satellite with INDI controller - Euler angles

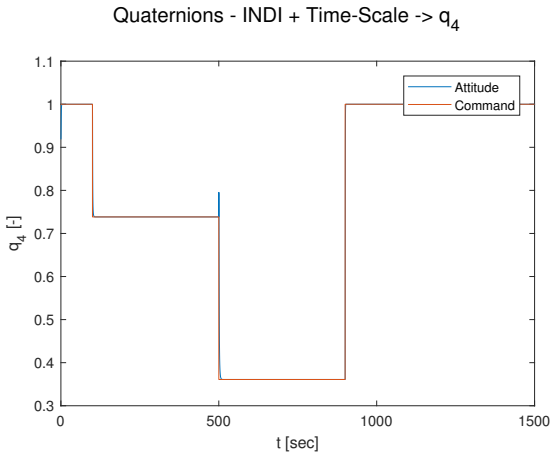
5.2 Quaternions

In the quaternion based approach description, the following gains were chosen for the inner and outer loop control:

$$K_{p_{in}} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad K_{p_{out}} = -\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{129}$$



(a) First 3 quaternions



(b) 4th quaternion

Figure 43: Quaternions of the satellite with INDI controller

The results of the quaternion tracking control are presented in fig. 43. The quaternions follows the commanded values without oscillations. Taking into account fig. 44, the steady state error quickly converges to zero. The INDI loop control presents quick responses when fast transitions of the desired attitude commands occur, as presented in figs. 45 and 46. The amplitude of the control inputs is also considerable, however, less than obtained with the Euler angles.

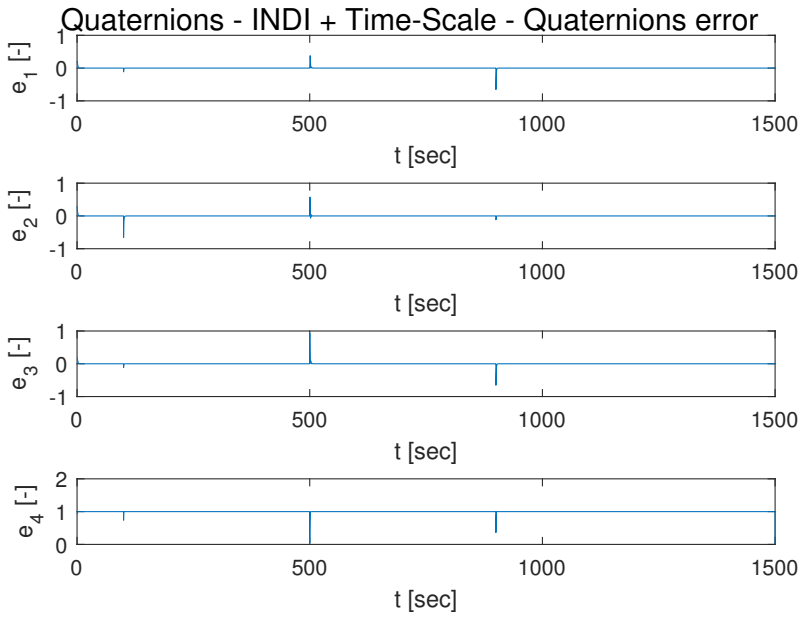


Figure 44: Attitude error of the satellite with INDI controller - Quaternions

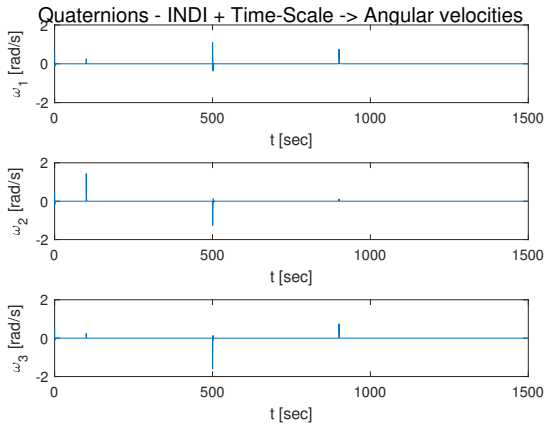


Figure 45: Angular velocity of the satellite with INDI controller - Quaternions

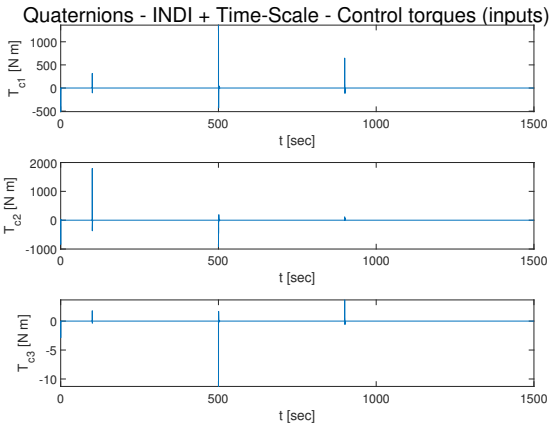


Figure 46: Control inputs of the satellite with INDI controller - Quaternions

Discussion of results and conclusions

In this assignment, linear (PD) and non-linear (NDI and its variants) control methods for spacecraft attitude control were studied. All these methods are, in theory, suitable to control a spacecraft. NDI control linearises the system and decouples its eigenmotions so that a PID controller can be applied on top of it. Considering the results obtained in the previous sections, an overall analysis of the all the methods is done by considering a small interval where a quick change in the commanded attitude angle (or quaternion) occurs. In this case, such variation occurs in $t = 500$ seconds when it is requested to go from $+70$ degrees to -70 degrees. The chosen interval is $[495, 550]$ seconds.

Taking into account the system described in Euler angles, all the NDI control algorithms show better performance than the PD controller, specially those with the time-scale separation. However, observing the plots on figs. 47 and 52, the NDI with time-scale separation controller presented quick and better attitude tracking compared to other controllers for θ_1 and θ_2 tracking, but not for θ_3, where INDI converged faster. The PD controller was the slowest. Additionally, no oscillation or overshoot is presented in all methods and the amplitude of the control inputs (fig. 50) is the smallest for the INDI control, which means that the energy required by this controller to cope with quick changes in attitude is also smaller not to mention that this controller is also able to reject better the external disturbances. The inner loop, characterised by a fast dynamics is also robust as it can be seen through the quick response of the angular velocities in fig. 49 . Therefore, this controller shows better results than the others.

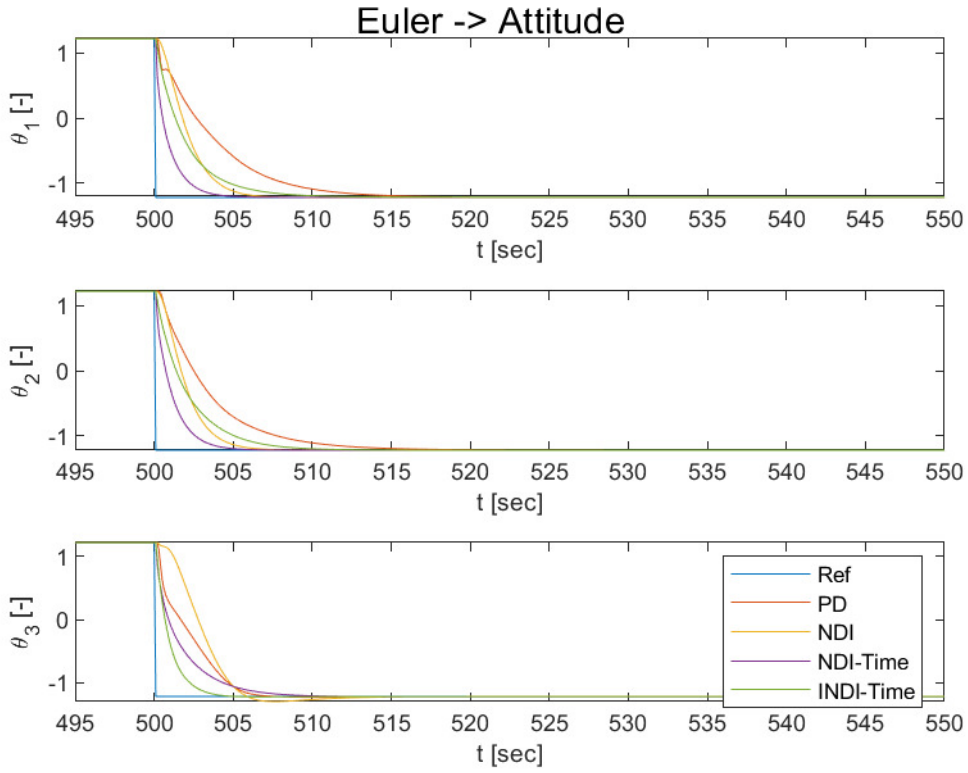


Figure 47: Comparison of attitude responses obtained with different controllers using Euler angles

For quaternions, the system with an INDI controller was able to quickly track the reference quaternions in less than 5 seconds (figs. 51 and 52). The same happens for the system controlled with the NDI using the time-scale approach. The PD control is the slowest as the transient of the attitude response takes almost 20 seconds to achieve a steady state quaternion. Compared to the Euler angles, the amplitude of the inputs using the NDI control approach is larger, therefore requiring more energy from the actuators to compute the external torques necessary to control the spacecraft to the desired attitude (fig. 54). The INDI controller also showed to be more robust on rejecting constant

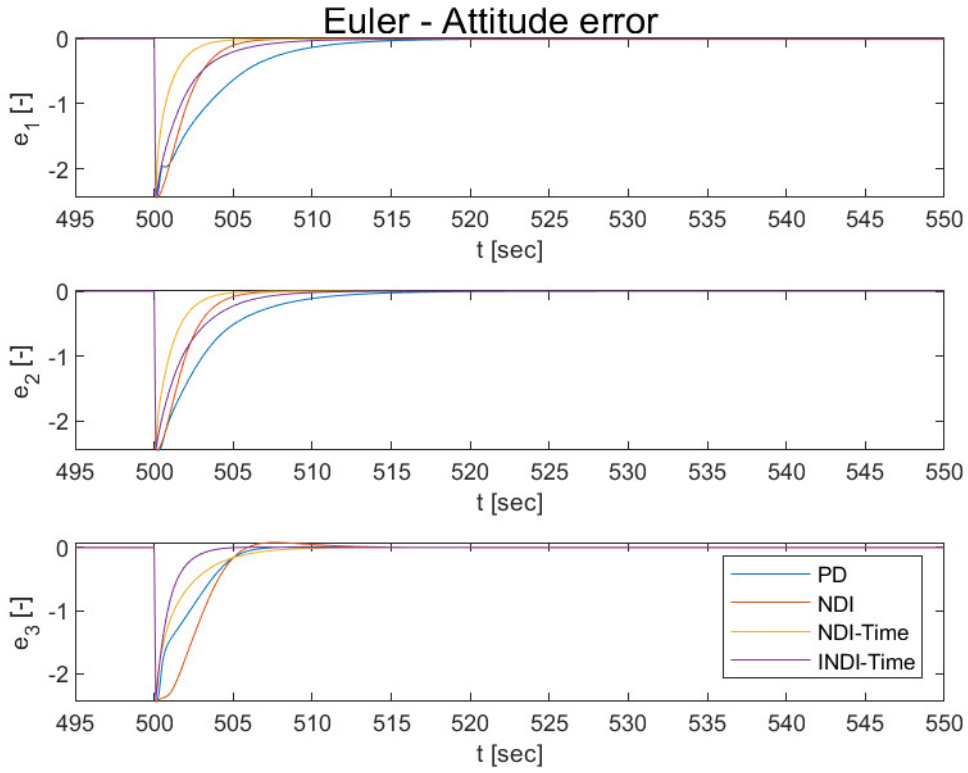


Figure 48: Comparison of attitude error obtained with different controllers using Euler angles

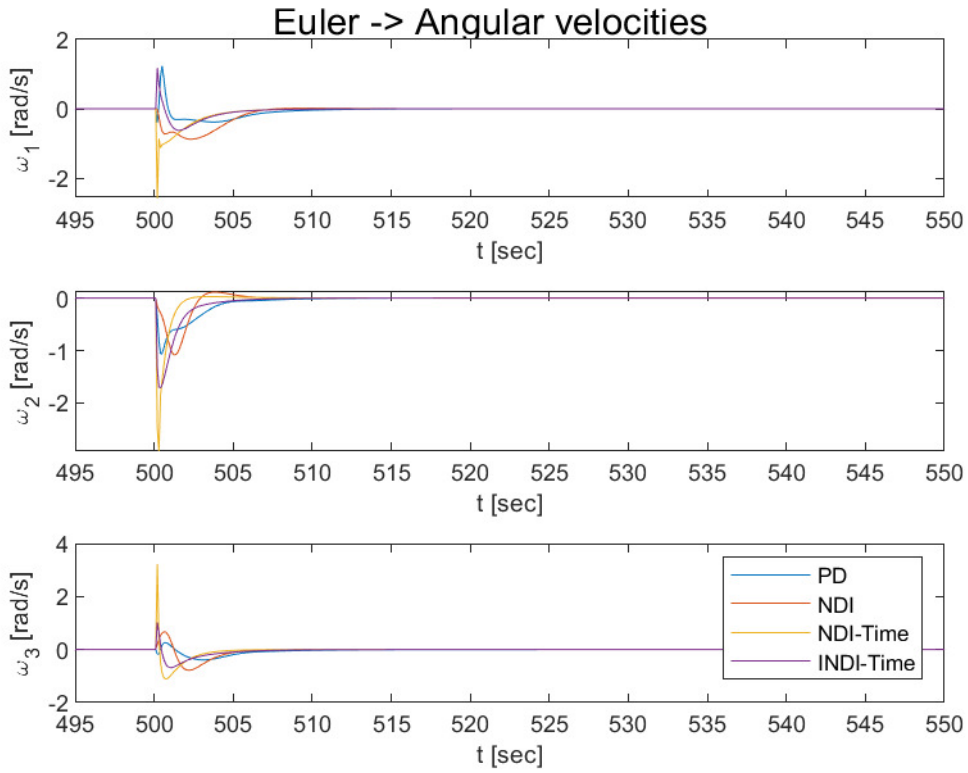


Figure 49: Comparison of angular velocity responses obtained with different controllers using Euler angles

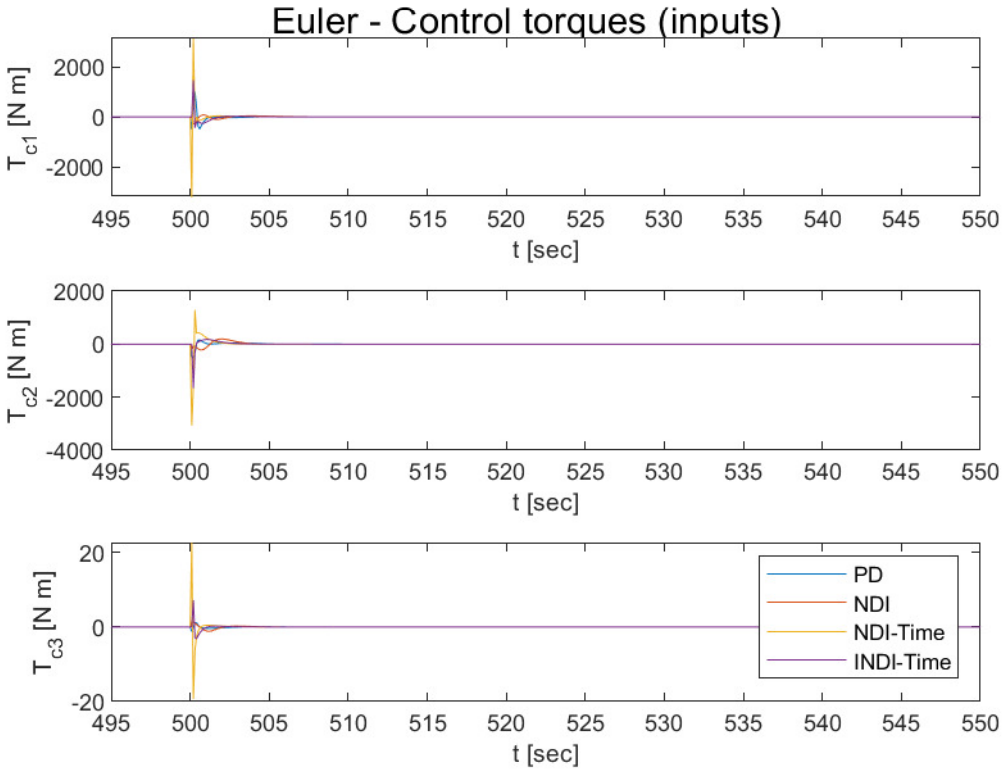


Figure 50: Comparison of control input responses obtained with different controllers using Euler angles disturbances.

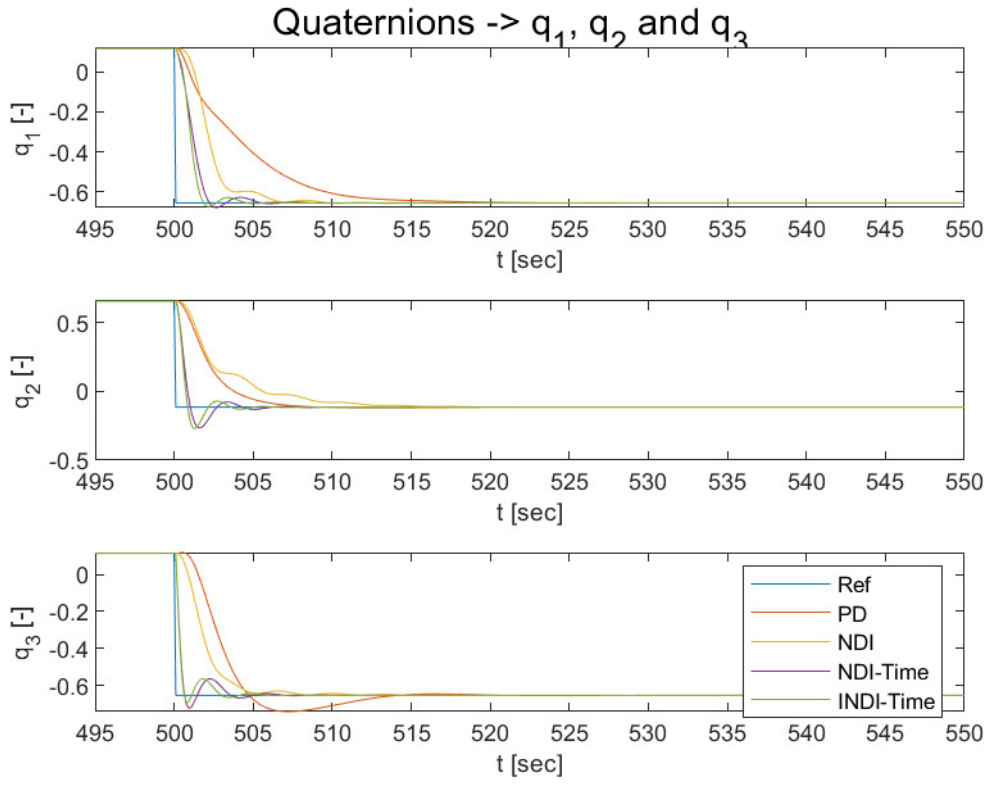


Figure 51: Comparison of quaternion responses obtained with different controllers

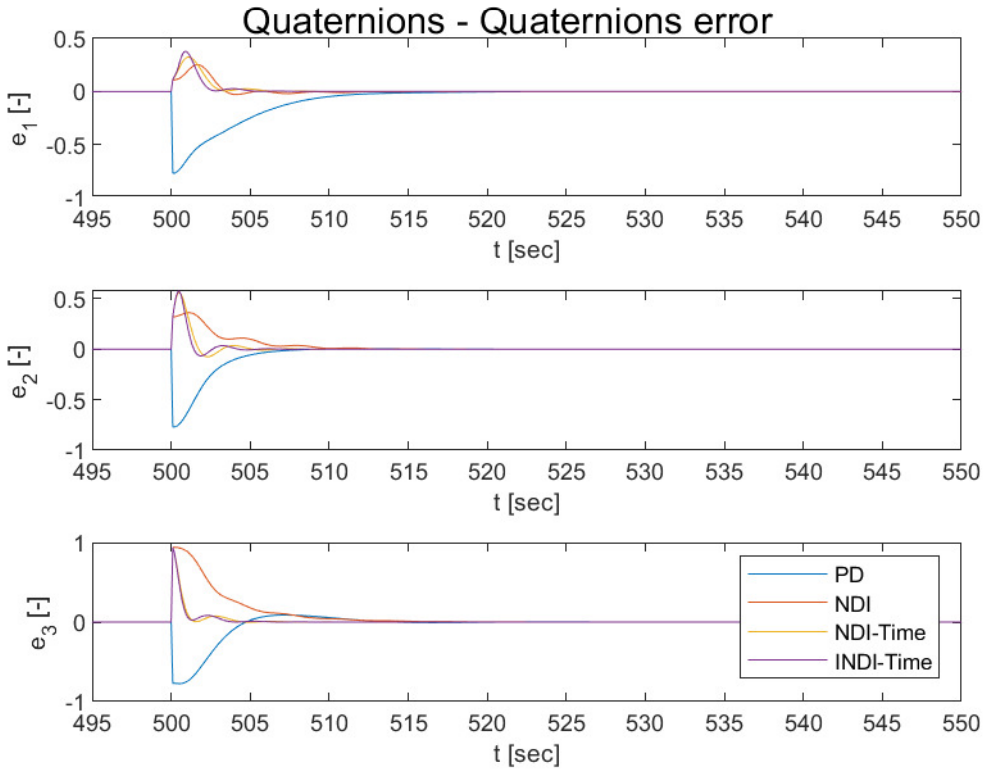


Figure 52: Comparison of quaternion error obtained with different controllers

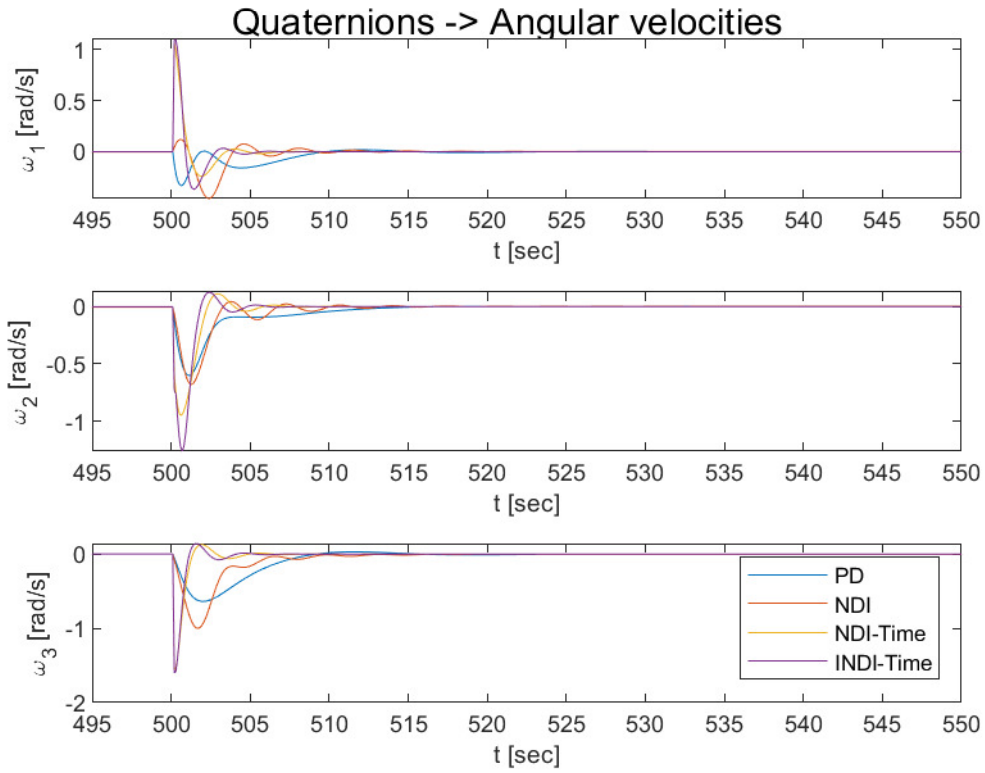


Figure 53: Comparison of angular velocity responses obtained with different controllers using Quaternions

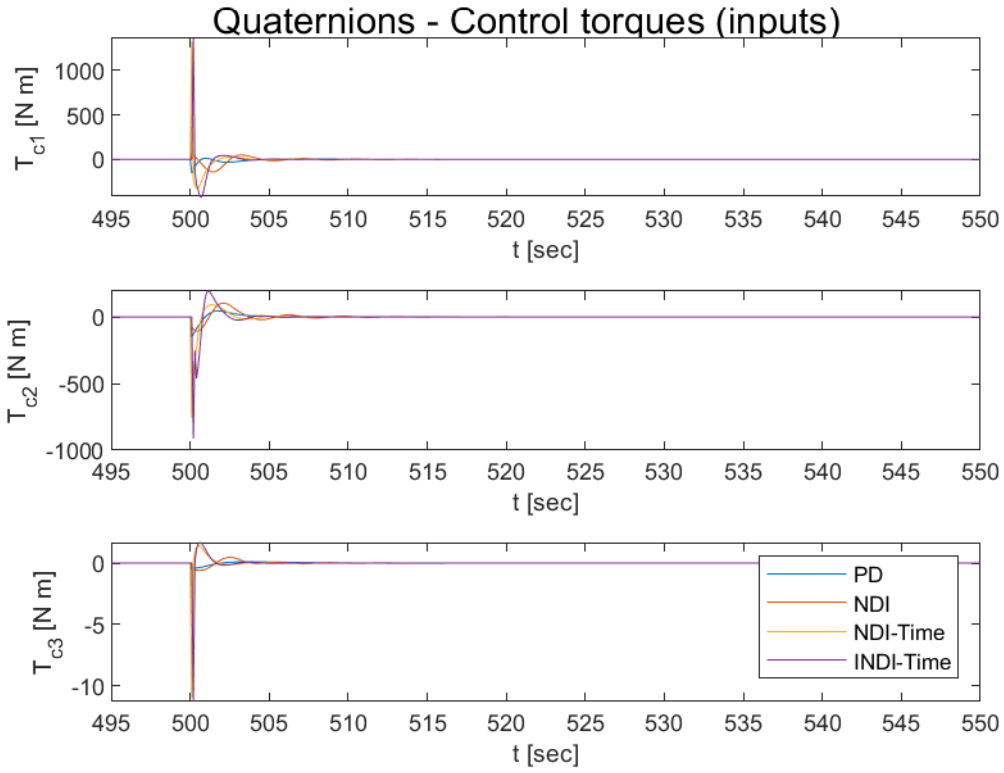


Figure 54: Comparison of control input responses obtained with different controllers using Quaternions

The current assignment is important to get a preliminary study on the influence of different control methods to obtain the desired attitude responses. Moreover, it is important to take into account that the amplitude of disturbances added in the system are small when compared to control inputs given by the actuators. This means that other disturbances should have been modelled and implemented to verify the robustness of this controller in its rejection. Moreover, actuators and sensor measurements are assumed to be perfect. Thus, in order to obtain a more realistic simulation, implementation of actuator and sensor models should also be considered as the first mainly introduces delays in whole closed loop system and the second introduces noise and bias in the measurements. For future work, it would be interesting to combined the 3 assignments proposed in the course.

Overall, taking into account the given results, the INDI control shows good tracking performance for both Euler angles and quaternions and can be a substitute to the traditional PD controllers. This controller also successfully rejected the disturbances imposed in the system. In the end, this method enforces great potential for spacecraft attitude control systems.

References

[1] P Acquatella, W. Falkena, E.J. van Kampen, Q. P. Chu., "Robust Nonlinear Spacecraft Attitude Control using Incremental Nonlinear Dynamic Inversion" (2012)

[2] R. R. da Costa, Q. P. Chu, and J. A. Mulder. "Reentry Flight Controller Design Using Nonlinear Dynamic Inversion", Journal of Spacecraft and Rockets, Vol. 40, No. 1 (2003), pp. 64-71.

[3] E. J. J. Smeur, Q. P. Chu, G. C. H. E. de Croon. "Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles", Journal of Guidance, Control, and Dynamics, Vol. 39, No. 3 (2016), pp. 450-461.

[4] S. Sieberling, Q. P. Chu, and J. A. Mulder. "Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction", Journal of Guidance, Control, and Dynamics, Vol. 33, No. 6 (2010), pp. 1732-1742.

A NDI - Computation of the system relative degree

Euler angles

$$y_1 = \theta_1 \tag{130}$$

$$\dot{y}_1 = \dot{\theta}_1 = \omega_1 + \sin(\theta_1)\tan(\theta_2)\omega_2 + \cos(\theta_1)\tan(\theta_2)\omega_3 + n\frac{\sin(\theta_3)}{\cos(\theta_2)} \tag{131}$$

$$\ddot{y}_1 = \dot{\theta}_1(\cos(\theta_1)\tan(\theta_2)\omega_2 - \sin(\theta_1)\tan(\theta_2)\omega_3) + \dot{\theta}_2\left(\frac{\sin(\theta_1)}{\cos^2(\theta_2)}\omega_2 + \frac{\cos(\theta_1)}{\cos^2(\theta_2)}\omega_3 + n\frac{\sin(\theta_3)\sin(\theta_2)}{(\cos(\theta_2))^2}\right) + \\ + \dot{\theta}_3\left(n\frac{\sin(\theta_3)}{\cos(\theta_2)}\right) + \dot{\omega}_1 + \dot{\omega}_2(\sin(\theta_1)\tan(\theta_2)) + \dot{\omega}_3(\cos(\theta_1)\tan(\theta_2)) \tag{132}$$

$$y_2 = \theta_2 \tag{133}$$

$$\dot{y}_2 = \dot{\theta}_2 = \cos(\theta_1)\omega_2 - \sin(\theta_1)\omega_3 + n\cos(\theta_3) \tag{134}$$

$$\ddot{y}_2 = \ddot{\theta}_2 = \dot{\theta}_1(-\sin(\theta_1)\omega_2 - \cos(\theta_1)\omega_3) + \dot{\theta}_3(n\cos(\theta_3)) + \dot{\omega}_2(\cos(\theta_1)) + \dot{\omega}_3(-\sin(\theta_1)) \tag{135}$$

$$y_3 = \theta_3 \tag{136}$$

$$\dot{y}_3 = \dot{\theta}_3 = \frac{\sin(\theta_1)}{\cos(\theta_2)}\omega_2 + \frac{\cos(\theta_1)}{\cos(\theta_2)}\omega_3 + n\tan(\theta_2)\sin(\theta_3) \tag{137}$$

$$\ddot{y}_3 = \ddot{\theta}_3 = \dot{\theta}_1\left(\frac{\cos(\theta_1)}{\cos^2(\theta_2)} - \frac{\sin(\theta_1)}{\cos(\theta_2)}\right) + \dot{\theta}_2\left(-\frac{\sin(\theta_1)}{\cos(\theta_2)} - \frac{\cos(\theta_1)\sin\theta_2}{\cos^2(\theta_2)} + n\frac{\sin(\theta_3)}{\cos(\theta_2)}\right) + \\ + \dot{\theta}_3(n\tan(\theta_2)\cos(\theta_3)) + \dot{\omega}_2\left(\frac{\sin(\theta_2)}{\cos(\theta_2)}\right) + \dot{\omega}_3\left(\frac{\cos(\theta_1)}{\cos(\theta_2)}\right) \tag{138}$$

The total relative degree of the system is 6. The $\dot{\omega}_i$ term directly depends on the input $\dot{T}_{c_i}$.

Quaternions

$$y_1 = q_1 \tag{139}$$

$$\dot{y}_1 = \dot{q}_1 = \frac{1}{2}(\omega_3 q_2 + (n - \omega_2)q_3 + \omega_1 q_4) \tag{140}$$

$$\ddot{y}_1 = \ddot{q}_1 = \frac{1}{2}(\dot{q}_2\omega_3 + \dot{q}_3(n - \omega_2) + \dot{q}_4\omega_1 + \dot{q}_4) \tag{141}$$

$$y_2 = q_2 \tag{142}$$

$$\dot{y}_2 = \dot{q}_2 = \frac{1}{2}(\omega_3 q_1 + \omega_1 q_3 + (\omega_2 + n)q_4) \tag{143}$$

$$\ddot{y}_2 = \ddot{q}_2 = \frac{1}{2}(\dot{q}_1(-\omega_3) + \dot{q}_3\omega_1 + \dot{q}_4(\omega_2 + n) + \dot{\omega}_1 q_3 + \dot{\omega}_2 q_4 - \dot{\omega}_3 q_1) \tag{144}$$

$$y_3 = q_3 \tag{145}$$

$$\dot{y}_3 = \dot{q}_3 = \frac{1}{2}((\omega_2 - n)q_1 - \omega_1 q_2 + \omega_3 q_4) \tag{146}$$

$$\ddot{y}_3 = \ddot{q}_3 = \frac{1}{2}(\dot{q}_1(\omega_2 - n) + \dot{q}_2\omega_1 + \dot{q}_4\omega_3 + \dot{\omega}_1 q_2 + \dot{\omega}_2 q_1 + \dot{\omega}_3 q_4) \tag{147}$$

$$y_4 = q_4 \tag{148}$$

$$\dot{y}_4 = \dot{q}_4 = -\frac{1}{2}(-\omega_1 q_1 - (\omega_2 + n)q_2 - \omega_3 q_3) \tag{149}$$

$$\ddot{y}_4 = \ddot{q}_4 = -\frac{1}{2}(\dot{q}_1\omega_1 + \dot{q}_2(\omega_2 + n) + \dot{q}_3\omega_3 + \dot{\omega}_1 q_1 + \dot{\omega}_2 q_2 + \dot{\omega}_3 q_3) \tag{150}$$

Again, the total relative degree of the system is 6. The $\dot{\omega}_i$ term directly depends on the input $\dot{T}_{c_i}$.