rijksuniversiteit
groningen

Integration Project

Faculty of Science and Engineering

The Design of an Non-Linear Satellite Attitude Controller

Student

J.P.H. Hillen | S3217191

First supervisor

M. Muñoz Arias

Second supervisor

G. Larsen

June 21, 2021

Abstract

This thesis is dedicated to the development of a non-linear position based attitude controller for a small satellite. A typical small satellite is subject to numerous non-linear disturbances causing undesired torque exertion on the satellite. This can cause the satellite to abandon its desired attitude and thus malfunction. By implementing a non-linear controller based solely on position, an efficient trajectory following stabilization method with low noise generation is reached. The controller will make use of the port-Hamiltonian framework in combination with the quaternion representation. These respectively have the advantages to be easily compared to other control strategies based on energy usage, and to solve singularity problems encountered with rotational matrices. Thereafter, passivity based control is applied to guarantee asymptotic stability of a developed error system, thus following desired trajectory.

This thesis is written as the final project of BSc Industrial Engineering and Management (IEM), at the University of Groningen, Netherlands, Faculty of Science and Engineering.

Contents

1 Introduction

A satellite in orbit has a specific desired orientation (attitude) for different reasons such as, for example, antennas that need a certain position to communicate with the earth or solar cells that need to face the sun to produce energy [1]. Moreover, the sensors and actuators of the satellite are subject to possible hinders as a result from an undesired attitude [1]. Thus, the attitude is subject to numerous constraints which result in a desired attitude. To maintain this desired attitude, satellites rely on their Attitude Determination and Control Systems. Since research on these (ADCS) is volatile, is there a “maximum” to be reached, or are there still lots of points of improvement to be made? “The sky is not the limit”, would be an unsatisfactory answer, and there is much more improvement on the horizon with respect to for instance power-, propulsion- and guidance systems in the satellite [2]. What all these system improvements on the horizon have in common, is their dependency on a highly efficient ADCS [2]. Moreover, small satellites are often subject to constraints with respect to mass and energy and therefore equipping them with as less attitude sensors as possible, is often seen as a requirement [3]. When a satellite is equipped with numerous sensors, simultaneous use of all sensors is often impossible because of operational limitations [3]. Moreover,
Because of all the different addressed constraints and requirements, the development of a controller that operates with small steady state error to attain the desired trajectory while being faster and therefore consuming less energy, is of interest for both satellite operators and producers that work with state of the art ADCS. Such a controller is useful for not only spacecraft, but also underwater vehicles, ground vehicles, and robotic systems [4].
Currently, the state of the art of ADCS consists of both linear- and non-linear controllers. A large number of ADCS use Kalman filters for attitude estimation [5–7], where after the estimates serve as an input in the Linear Quadratic attitude controllers. The estimation in the latter named linear nature is less accurate than doing so non-linearly, as the separation principle used, does not generally hold for nonlinear systems [8], as is often the nature of the dynamics of satellites [9].

Non-linear state of the art ADCS are found in [4, 10–13]. In [11], an energy shaping control method for the rotational motion of a fully rigid body is used. Reference [12] uses a nonlinear adaptive controller for a satellite with two vector measurements for gyro bias estimation and asymptotic attitude tracking and [13] incorporates adaptive passivity-based controller. In [10,14], a quaternion representation is used to describe the attitude of a rigid body type spacecraft. The controller of the quaternion system is handled with port-Hamiltonian system [14]. For this system, the angular velocity Ω and position measurement is used. Moreover, a simulated environment is developed to test a controller. For the state variable r, the distance from the spacecraft to the center of the earth is chosen to have a value of ($r = 1$) [14]. This makes the simulation highly unrealistic. However, when one simulates the attitude in the order of centimeters, and the distance to the earth in the order of kilometers, the controller will become unstable. Therefore, the system of the satellite rigid body control problem is a fast-slow system, in which some state variables are very slow e.g. large, and other state variables are very fast e.g.

small [15].
What [10–14] have in common, is that they rely on the available attitude and the angular velocity.

However, controlling the satellite without velocity measurement would be desirable, since velocity measurement generates noise [4, 16] and measurement tools to track speed, add on extra weight on the satellite, causing a higher energy usage. Therefore, it is an improvement in efficiency to only track position. In [4, 16], the controller uses only position measurements, by a dynamic extension of the Port-Hamiltonian system used in [14], thus without speed measurement, as is required in the proposed controller of [10–14].
Because of all the different constraints mentioned above, the attitude control based on the measurement of only one sensor would be an advantage. This paper therefore aims to develop a non-linear ADCS for a small satellite, controllable by one input variable, tested for real-world application in a realistic simulated environment by integrating the proposed controller from [14] and [4], by implementing the dynamical extension for the port-Hamiltonian framework of [16] for position-based trajectory tracking into the proposed non-linear controller of [14], and thereafter solving shortcomings in the simulation parameters used in [14] controller using slow-fast system theory, to test this integrated controller in a realistic simulated environment.
The project will start with the required preliminaries about rotational kinematics, rigid body dynamics, the Port-Hamiltonian framework and the control theory, containing the mathematical explanation of the controller of [14]. Thereafter, the problem analysis, containing the system description and the problem statement, is provided, followed by the design goal. Thereafter, the results are be presented, ending with a discussion and future research perspective.

2 Preliminaries

2.1 Rigid Body Dynamics

For determining the attitude, two different reference frames have to be defined. The first one, the inertial frame (I-frame), has its origin in the center of the earth, its z-axis pointing towards the north pole, and its x-axis pointing in the direction of the Vernal Equinox, e.g. when the sun is exactly above the Equator, and its y-axis completing the right handed coordinate system [1] .The body frame (B-frame) is the fixed reference frame that is attached to the satellite body. The axes of the B-frame are chosen as the satellite's principal axes of inertia. The body as observed with respect to the B-frame is thus stationary [1]. This is summarized in Figure. 1 [17], in which the subscript letters i and b belong to the axes of the I-frame and the B-frame, respectively. The I- and B-frames are used to determine the orientation of the satellite. To determine the angle of a spacecraft, both the angle of the B-frame and the angle of the I-frame are required. What follows are the Euler angles, with which a rotation matrix can be constructed [18].

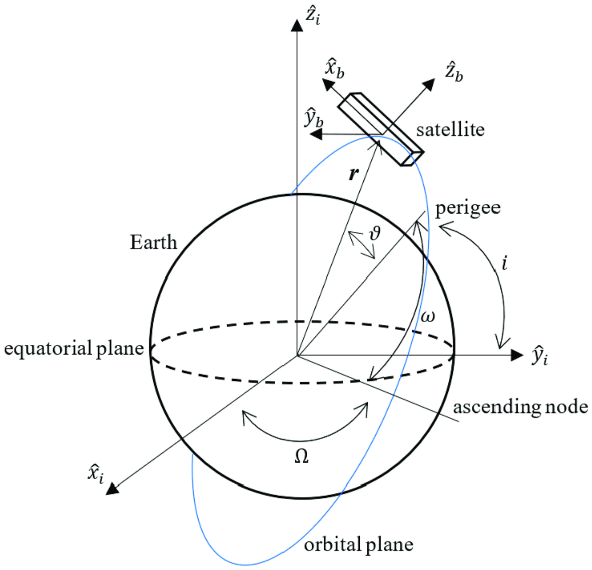

Figure 1: The B-frame and the I-frame visualized

2.2 Disturbances

In this project we consider four different disturbances (perturbations) that contribute to external torques that can lead to an undesired attitude. These include gravity force, atmospheric drag, the J2-effect and solar radiation. A more thorough definition of the disturbances is given below.

2.2.1 Gravity

Gravity is a non-linear disturbance [18]. Its formula is given by

$$T_{gg} = \frac{3\mu}{R_0^3}\hat{o}_3(J \times \hat{o}_3) \tag{1}$$

where μ is the Earth's gravitational coefficient, R_0 is the distance to Earth's center, $\hat{o}_3$ is the unit vector towards nadir, which is the direction of the force of gravity, and J is the spacecraft's inertia tensor [18].

2.2.2 Drag Force

Atmospheric drag can be contributed to atmospheric friction acting in the opposite direction of the orbital motion of the satellite and it is the largest disturbance on the attitude of the satellite next to Gravity [18]. It is given by

$$F_{atmos} = \frac{1}{2}\rho C_D S \|v_{rel}\| v_r el \tag{2}$$

where ρ is total atmospheric mass density, C_D is the dimensionless drag coefficient, S is the spacecraft area, projected along the direction of motion, v_{rel} the velocity of the spacecraft relative to the atmosphere [18]. The torque is consequently given by

$$T_{atmos} = F_{atmos}\left(r_{cp} - r_{CoM}\right) \tag{3}$$

2.2.3 J2 Force

Because of the oblate spheroid shape of the earth, which makes the radius at the Earth's equator 21 km larger than the earth's radius around the poles, the rotation of the earth causes J2 force, another dominant disturbance. The formula of J-2 force is given by

$$T_{J2} = -\frac{3\mu J_2 R_0^2}{2\|r\|^4} r\left(1 - \frac{5r_z^2}{\|r\|}\right) \tag{4}$$

2.2.4 Solar Radiation

Solar radiation is a relatively small disturbance on the attitude of the satellite. Because the photons in solar radiation travel at light speed, they have some momentum which they exert on other objects. This causes a increase in velocity [19]. The formula for solar radiation is as follows

$$T_{sp} = (1 + \rho_s)\frac{S}{c}A\left(r_{ep} - r_{CoM}\right) \tag{5}$$

where ρ_s is the reflecting factor, S is the solar constant, c is the speed of light, A is the surface area and r_{cp} is the location of the center of solar, and r_{CoM} the center of gravity [18].
The formula for the total disturbances is thus given by [18]

$$T_d = T_{gg} + T_{drag} + T_{J2} + T_{sp} \tag{6}$$

2.3 Quaternions

A quaternion is a rotational transformation of a rigid body about a normalized rotating axis. Quaternions represent the attitude of a rigid body globally. Another way of representing the attitude is with a rotational matrix using Euler angles.

However, this method has a disadvantage, in that it has undesirable singular points, which are in which the partial derivative is zero. This could be of harm to the project, when the satellite makes certain angles which correspond to the singular points. Consider the vector $q = [\text{a b c d}]^T$, with a, b, and c as the vector part of the quaternion, about which the rotations are performed, and d can as the scalar part, that specifies the magnitude of rotations. The vector part of the quaternion can then also be described by a rotation about the Euler eigenaxis e ($e = [e1e2e3]^T$) with the rotation angle theta $\theta \in$ R as follows

$$q = [q_1\ q_2\ q_3\ q_4]^T = \left[e_1\ \sin\frac{\theta}{2}\ e_2 \sin\frac{\theta}{2}\ \exp_3 \sin\frac{\theta}{2}\ \cos\frac{\theta}{2}\right]^T \tag{7}$$

Multiplying quaternions to a given vector is equivalent to rotating it, with the following constraint always satisfied

$$\sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} = 1 \tag{8}$$

Furthermore, for $\omega = [\omega_1\, \omega_2\, \omega_3]^T \in \mathbb{R}^3$, the angular velocity of the satellite, the kinematic equation of a rigid body via a quaternion, where $\Omega(q) \in \mathbb{R}^{4\times 3}$, is defined as

$$\frac{d}{dt}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{2}\Omega(q)\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{9}$$

which will be used later for definition in the port-Hamiltonian framework. Moreover, the error quaternion can be calculated, by

$$q_{error} = \Omega(q_d)^T * q = \begin{bmatrix} q_4^d q_1 - q_3^d q_2 + q_2^d q_3 - q_1^d q_4 \\ q_3^d q_1 + q_4^d q_2 - q_1^d q_3 - q_2^d q_4 \\ -q_2^d q_1 + q_1^d q_2 + q_4^d q_3 - q_3^d q_4 \\ q_1^d q_1 + q_2^d q_2 + q_3^d q_3 + q_4^d q_4 - 1 \end{bmatrix} \tag{10}$$

which will be used in the controllers that will be demonstrated.

2.4 port-Hamiltonian framework

The port-Hamiltonian framework provides an efficient tool to describe physical systems in an environment and the interactions between them [20]. It does so in terms of their power ports, the energy variables and their interconnection structure. The transfer of energy between the physical system and the environment is given via energy-dissipating elements, which Hamiltonian systems disregard [21], together with power preserving ports [20]. PH-systems are an extension of the classical Hamiltonian equations and they are formed by the geometric notion of Dirac structure [21], which allow to describe dynamical systems with algebraic constraints [22] that can be useful for the project, with the additional advantage that the PH method preserves the PH structure for the closed-loop system [20] used in this project. The general Hamiltonian system for a physical system is given by

$$\dot{x} = (J(x,t) - R(x,t))\frac{\partial H(x,t)^T}{\partial x} + g(x,t)u \tag{11}$$

$$y = g(x,t)^T \frac{\partial H(x,t)^T}{\partial x} \tag{12}$$

where u and $y \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, where J is a skew symmetric matrix, which represents the energy in the system. R is a symmetric semi-positive definite matrix in which damping faced by the system is represented $H$is the total sum of kinetic and potential energy of the physical system and g is the input matrix for the system [23].

To demonstrate the port-Hamiltonian framework, the example of a magnetic levitation system is considered. Figure 2 resembles a simplified system of a magnetic levitation system.

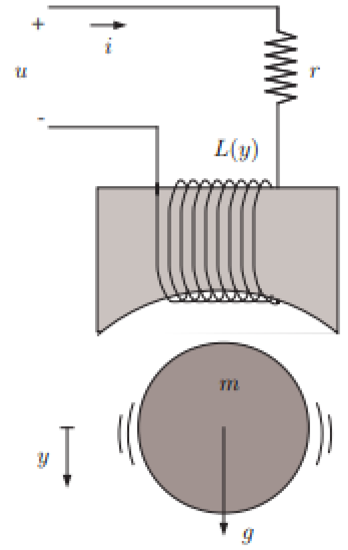

Figure 2: A magnetic Levitation System

The flux, generated by the current through the coil in the middle of the figure, closes through space between the the coil and the iron ball. This air space has variable reluctance and so the system tries to close it. This counteracts the downwards motion caused by gravity force. [24] The equations of motion are

$$\dot{\lambda} = -ri + u$$

$$m\dot{v} = F_m + mg$$

$$\dot{y} = v$$

where $\lambda =$ L(y) is the linkage flux, r is the resistance of the coil, and F_m the magnetic force, given by

$$F_m = \frac{\partial W_c}{\partial y}$$

$$W_c = \frac{1}{2}\frac{\partial L}{\partial y}i^2$$

is the magnetic co-energy L(y) is a function of the air gap, y. An approximation for L, only for systems with a with small y is

$$L(y) = \frac{k}{a+y}$$

with k a constant. The system can be expressed as a port-Hamiltonian system with the state variables

$$x = [\lambda, p = mv, y]^T$$

as Hamiltonian variables, and the Hamiltonian function given by

$$H(x) = \frac{1}{2k}(a+p)\lambda^2 + \frac{1}{2m}p^2 - mgy$$

such that

$$\dot{x} = \left(\underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}}_{\text{J}} - \underbrace{\begin{bmatrix} r & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\text{R}} \right) \frac{\partial H}{\partial x} + \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}}_{\text{g}} u$$

$$y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial x} \end{bmatrix} \tag{13}$$

2.5 Modelling

The modelling of the non-linear controller will be done in MATLAB. The model uses a trajectory for the desired values, and not a set point, implying that the desired state values keep changing with respect to time. The error between the desired trajectory and the current state therefore also changes with respect to time. The schematic overview of the closed loop system that is applicable to the project, with a non-linear control algorithm, is given in Fig. 3

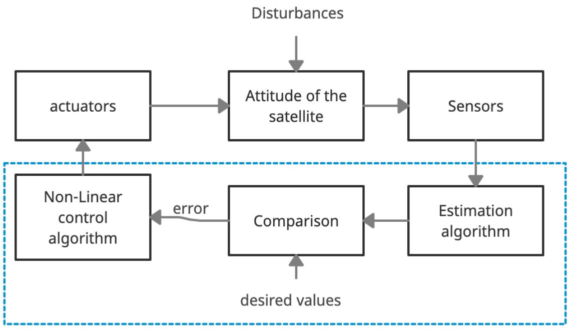

Figure 3: the closed loop feedback system for attitude control used in MATLAB

3 Problem Analysis

3.1 System description

In Figure. 3, the system of the project is shown. The red-dotted line represents the technical sub-system that is the controller proposed in [14]. The blue dotted line represents the scope of the project and the opportunity that is created when integrating the dynamic extension for the Port-Hamiltonian system proposed by [2] into the improved controller of [14]. Out of this integration, a novel trajectory control system for the rigid body satellite could be developed. In Figure. 3, the interrelation of the different stakeholders, that are outside the scope of this project, with respect to the technical part of the project is also shown. As is also shown in Figure. 3, the project will use quaternions to represent the attitude of the rigid body, in combination with the Port-Hamiltonian framework, with a dynamic extension provided by [2] for position based control. Thereafter, a passivity based control strategy will be used to develop a novel controller for the rigid body attitude.

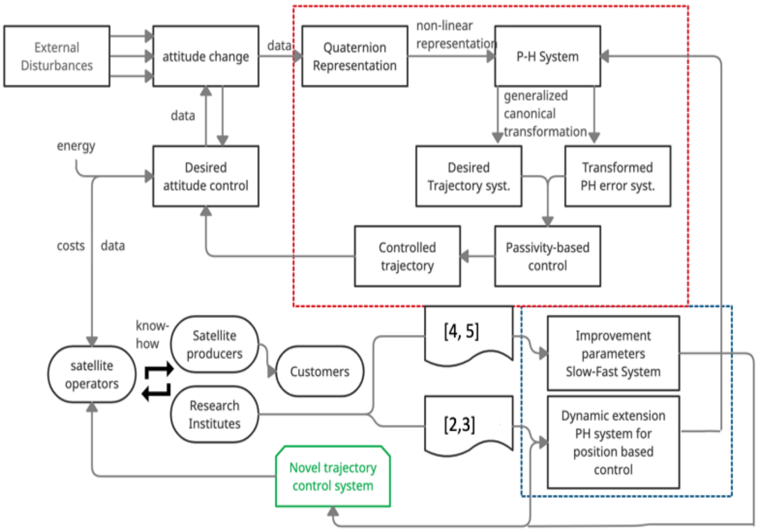

Figure 4: System description of the project and the scope of the research

3.2 Problem analysis

Small satellites are often subject to constraints with respect to energy and mass. Therefore, equipping them with as less attitude sensors as possible is often seen as a requirement [10]. Moreover, simultaneous use of all sensors is often impossible because of operational limitations. A smaller amount of attitude sensors also means a smaller amount of inputs in the ADCS. A controller with only one input, meaning only one sensor, would therefore be an improvement for a satellite in orbit. Currently, the state of the art ADCS include both linear- and non-linear controllers. Linear controllers have an estimation profile of which its linear nature is less accurate than non-linear controller. If this profile were to be improved with a non-linear controller, less energy would be consumed, and thus the satellite would be more efficient. A non-linear controller could also provide a more accurate way of estimating the non-linear disturbances provided in the preliminaries. However, the non-linear controllers addressed all rely on two or more inputs, meaning two or more sensors. A clear and well-functioning non-linear controller is proposed in [4]. This paper uses quaternions to represent the orientation, in combination with the port-Hamiltonian framework. The desired trajectory is fed into a closed loop feedback system transforming the port-Hamiltonian system in an error system that describes the behavior of the tracking error of the original port-Hamiltonian system. In this paper, there is a problem. A simulated environment is developed to test a controller, for which the state variable r, the distance from the spacecraft to the centre of the earth, is said to have a value of $r = 1$ [4]. This makes the simulation highly unrealistic. However, when one simulates the attitude in matter of centimetres, and the distance to the earth in kilometres, the controller will become unstable. Therefore, the system of the satellite rigid body control problem is a fast-slow system, in which some state variables are very slow e.g. large, and other state variables are very fast e.g. small [7]. To prove proper functioning in real-life scenarios, this simulated environment has to be enhanced to a realistic form. A second problem is also embedded in the nature of the proposed controller of [4], being that it depends on both the angular velocity w- and the position measurement of the rigid body. When the angular velocity is required, not only is the satellite heavier and more energy-consuming because of extra measurement tools, but also speed measurements create noise that requires extra filtering- and estimation algorithms [6]. A non-linear controller that uses only position measurements, by a dynamic extension of the Port-Hamiltonian system used in [14], thus without speed measurement, is given in [4]. However, this technique is applied to rigid joint robots, and not to ADCS of a satellite.

3.3 Problem statement

After a thorough problem analysis and system description, a clear idea of the problem statement is developed. The problem statement is given by

Small satellites are subject to energy- and mass constraints, demanding a minimum of on-board sensors and weight. Currently, both linear- and non-linear algorithms are often used for the control of rigid bodies in orbit. Linear controllers estimate the disturbances gravity, atmospheric drag, J-2 and solar radiation, less accurate than non-linear controllers, and are thus more energy consuming, making non-linear controllers preferred. However, investigated non-linear controllers rely on multiple input variables, causing the need for more measurement sensors and therefore more energy consumption and mass on-board. Small satellite control is therefore in need of a non-linear controller, controllable with only one input variable, tested in a realistic environment to proof real-world application.

4 Theoretical Background

To get greater sense of understanding of the controllers and frameworks that are used, some mathematical background is required. What is important to get familiar with, is the mathematics that translate the dynamical equations of the satellite into the port-Hamiltonian framework, as this framework will also serve as the core framework of which the transformation will lead to the non-linear controller. After first introducing the translation from the dynamical equations to the port-Hamiltonian framework, the operations that lead to the non-linear controller are presented.

4.1 Dynamical Equations to port-Hamiltonian framework

Similar to the previous example about the magnetic levitation system, the physical system of the satellite can be summarized by four dynamical equations, given by

$$\dot{r} = -\omega \times r + v \tag{14}$$

$$m\dot{v} + m\omega \times v = f \tag{15}$$

$$I\dot{\omega} + \omega \times I\omega = \rho \times f + u \tag{16}$$

$$\dot{q} = \frac{1}{2}\Omega(q)\omega, \tag{17}$$

where $r = [r_1\, r_2\, r_3]^T \in \mathbb{R}^3$ is the position, $\omega = [\omega_1\, \omega_2\, \omega_3]^T \in \mathbb{R}^3$ is the angular velocity, $v = [v_1\, v_2\, v_3]^T \in \mathbb{R}^3$ is the tangential velocity, $m \in \mathbb{R}$ is the mass, $f = [f_1\, f_2\, f_3]^T \in \mathbb{R}^3$ is the external force, $I \in \mathbb{R}^{3\times3}$ denotes the inertia tensor, $u = [u_1\, u_2\, u_3]^T \in \mathbb{R}^3$ is the external torque, and $\rho = [\rho_1\, \rho_2\, \rho_3]^T \in \mathbb{R}^3$ is the vector from the center of mass to the action point. Moreover, we also define the angular momentum and linear momentum, given by

$$p = I\omega \tag{18}$$

$$s = mv \tag{19}$$

where p is defined as $p = [p_1\, p_2\, p_3]^T \in \mathbb{R}^3$ and s, is defined as $s = [s_1\, s_2\, s_3]^T \in \mathbb{R}^3$.

The state vector is then defined as $x = [r^T,\, q^T,\, s^T,\, p^T]^T$, which gives us

$$x = \begin{bmatrix} r_1 \\ r_2 \\ \ldots \\ p_2 \\ p_3 \end{bmatrix} \tag{20}$$

In (20), x is a vector consisting of 13 elements.
From the momenta, (18) and (19), the port-Hamiltonian is derived, which is equal to the total energy, being the sum of the kinetic energy and potential energy. In this case, no potential energy is present, which means

$$H(p, s) = \frac{1}{2}p^T I^{-1} p + \frac{1}{2}s^T m^{-1} s \tag{21}$$

Now, the partial derivative of $H(p, s)$ is differentiated with respect to the previously defined state vector $x = [r^T, q^T, s^T, p^T]^T$, which results in

$$\frac{\partial H(p,s)}{\partial x} = [0_{1\times3} \quad 0_{1\times4} \quad \frac{1}{2}s^T((m^{-1})^T + m^{-1}) \quad \frac{1}{2}p^T((I^{-1})^T + I^{-1})] \tag{22}$$

$$= [0_{1\times3} \quad 0_{1\times4} \quad v \quad \omega], \tag{23}$$

Next, the four dynamical equations of motion (14) - (17) can be rewritten in terms of $\frac{\partial H(p,s)}{\partial x}$. Here, $E_{i\times i} \in \mathbb{R}^{i\times i}$ is defined as the identity matrix, $0_{i\times j} \in \mathbb{R}^{i\times j}$ is the zero matrix of which its dimensions are given in the framework, and the $\tilde{a}$, for a vector $\tilde{a} = [a_1\, a_2\, a_3] \in \mathbb{R}^3$, stands for the cross multiplication matrix given by

$$\tilde{a} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

The four equations of motion are now translated into the port-Hamiltonian framework by

$$\dot{r} = -\omega \times r + v \tag{14}$$

$$= \begin{bmatrix} -r_3 w_2 + r_2 w_3 + s_1 m^{-1} \\ r_3 w_1 - r_1 w_3 + s_2 m^{-1} \\ -r_2 w_1 + r_1 w_2 + s_3 m^{-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_1 m^{-1} \\ s_2 m^{-1} \\ s_3 m^{-1} \end{bmatrix} + \begin{bmatrix} 0 & -r_3 & a_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

$$= E_{3\times3} \frac{\partial H(p,s)^T}{\partial s} + \tilde{r} \frac{\partial H(p,s)^T}{\partial p} \tag{24}$$

$$m\dot{v} + m\omega \times v = f \tag{15}$$

$$\dot{s} = m\dot{v} = f - m\omega \times v = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} - m \begin{bmatrix} \omega_2 v_3 - \omega_3 v_2 \\ \omega_3 v_1 - \omega_1 v_3 \\ \omega_1 v_2 - \omega_2 v_1 \end{bmatrix} = E_{3\times3} f - m\tilde{w} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$$= E_{3\times3} f - m\tilde{w} \frac{\partial H(p,s)^T}{\partial s} = -E_{3\times3} \frac{\partial H(p,s)^T}{\partial r} + E_{3\times3} f - m\tilde{w} \frac{\partial H(p,s)^T}{\partial s} \tag{25}$$

$$I\dot{\omega} + \omega \times I\omega = \rho \times f + u \tag{16}$$

$$\dot{p} = I\dot{\omega} = \rho \times f + u - \omega \times I\omega = \begin{bmatrix} \rho_2 f_3 - \rho_3 f_2 \\ \rho_3 f_1 - \rho_1 f_3 \\ \rho_1 f_2 - \rho_2 f_1 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} - \omega \times p$$

$$= -\tilde{\rho}f + E_{3\times3}u - \begin{bmatrix}\omega_2p_3 - \omega_3p_2\\ \omega_3p_1 - \omega_1p_3\\ \omega_1p_2 - \omega_2p_1\end{bmatrix} = \tilde{\rho}f + E_{3\times3}u + \tilde{p}\frac{\partial H(p,s)^T}{\partial p}$$

$$= \tilde{r}\frac{\partial H(p,s)^T}{\partial r} - \frac{\Omega(q)^T}{2}\frac{\partial H(p,s)^T}{\partial q} + \tilde{p}\frac{\partial H(p,s)^T}{\partial p} + \tilde{\rho}f + E_{3\times3}u \tag{26}$$

$$\dot{q} = \frac{1}{2}\Omega(q)\omega, \tag{17}$$

$$\dot{q} = \frac{1}{2}\begin{bmatrix} q_4 & -q_3 & q_2\\ q_3 & q_4 & -q_1\\ -q_2 & q_1 & q_4\\ -q_1 & -q_2 & -q_3\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix} = \frac{1}{2}\Omega(q)\frac{\partial H(p,s)^T}{\partial p} \tag{27}$$

Now, (24) - (27) can be captured in matrix form, leading to the Port-Hamiltonian framework given by

$$\dot{x} = \begin{bmatrix}\dot{r}\\ \dot{q}\\ \dot{s}\\ \dot{p}\end{bmatrix} = \begin{bmatrix} 0_{3\times3} & 0_{3\times4} & E_{3\times3} & \tilde{r}\\ 0_{4\times3} & 0_{4\times4} & 0_{4\times3} & \frac{\Omega(q)}{2}\\ -E_{3\times3} & 0_{3\times4} & -m\tilde{\omega} & 0_{3\times3}\\ \tilde{r} & -\frac{\Omega(q)^T}{2} & 0_{3\times3} & \tilde{\rho}\end{bmatrix}\frac{\partial H(p,s)^T}{\partial x} + \begin{bmatrix}0_{3\times3} & 0_{3\times3}\\ 0_{4\times3} & 0_{4\times3}\\ E_{3\times3} & 0_{3\times3}\\ \tilde{\rho} & E_{3\times3}\end{bmatrix}\begin{bmatrix}f\\ u\end{bmatrix} \tag{28}$$

$$y = g(p,s)^T\frac{\partial H(p,s)^T}{\partial x} = \begin{bmatrix}0_{3\times3} & 0_{4\times3} & E_{3\times3} & -\tilde{\rho}\\ 0_{3\times3} & 0_{4\times3} & 0_{3\times3} & E_{3\times3}\end{bmatrix}\frac{\partial H(p,s)^T}{\partial x} \tag{29}$$

Now the system is of the form

$$\dot{x} = (J(x,t) - R(x,t))\frac{\partial H(x,t)^T}{\partial x} + g(x,t)u \tag{30}$$

$$y = g(x,t)^T\frac{\partial H(x,t)^T}{\partial x} \tag{31}$$

From which the properties can be found in Chapter 2.4, In this port-Hamiltonian framework, dissipation matrix R is not present, because no damping factors are included yet.

4.2 Trajectory transformation

A crucial point both the non-linear controlled situation and the PD-controlled situation have in common, is that the scenario that is simulated is unrealistic. Therefore, a solution and physical interpretation for the position vector for a realistic simulated scenario is now presented, to be able to have a more valuable comparison between the linear PD controller and the non-linear controller. To scale up the simulation to realistic values, is fairly complex. This has to do with two factors; the first factor that causes trouble when trying to implement a realistic scenario is the position vector r_d, is the current Cartesian coordinate system, which has to be changed to a spherical coordinate system. This is done by transformations given by [25] as shown in section

$$\begin{aligned} x &= rcos(\phi)sin(\theta) \\ y &= rsin(\phi)sin(\theta) \\ z &= rcos(\theta) \end{aligned} \tag{32}$$

To understand how the new trajectory for the realistic scenario was formed, and also why a scale up of R_0 in position vector v has effect on the velocity and the acceleration of the system, first it is important to understand how the position vector r is related to the velocity v and acceleration $\dot{v}$. In [14], when choosing a trajectory for r, the trajectories of r, v and $\dot{v}$, are interconnected by

$$r^d = \begin{bmatrix} r_1^d \\ r_2^d \\ r_3^d \end{bmatrix} = \begin{bmatrix} cost \\ sint \\ 1 \end{bmatrix} \tag{33}$$

$$\dot{r}^d = \begin{bmatrix} \dot{r}_1^d \\ \dot{r}_2^d \\ \dot{r}_3^d \end{bmatrix} = \begin{bmatrix} -sint \\ cost \\ 0 \end{bmatrix} \tag{34}$$

$$v^d = \dot{r}^d + \omega^d \times r^d \tag{35}$$

$$= \begin{bmatrix} \dot{r}_1^d \\ \dot{r}_2^d \\ \dot{r}_3^d \end{bmatrix} + \begin{bmatrix} \omega_2^d r_3^d - \omega_3^d r_2^d \\ \omega_3^d r_1^d - \omega_1^d r_3^d \\ \omega_1^d r_2^d - \omega_2^d r_1^d \end{bmatrix} \tag{36}$$

$$\begin{bmatrix} \sin(t) \\ \cos(t) \\ 0 \end{bmatrix} + \frac{1}{4\sqrt{3}} \begin{bmatrix} 1 - \sin(t) \\ \cos(t) - 1 \\ \sin(t) - \cos(t) \end{bmatrix} \tag{37}$$

$$= +\frac{1}{4\sqrt{3}} \begin{bmatrix} 1 - (1 + 4\sqrt{3})\sin(t) \\ -1 + (1 + 4\sqrt{3})\cos(t) \\ \sin(t) - \cos(t) \end{bmatrix} \tag{38}$$

$$\dot{v}^d = \begin{bmatrix} \dot{v}_1^d \\ \dot{v}_2^d \\ \dot{v}_3^d \end{bmatrix} = \frac{1}{4\sqrt{3}} \begin{bmatrix} (1 + 4\sqrt{3})\cos(t) \\ -(1 + 4\sqrt{3})\sin(t) \\ \cos(t) + \sin(t) \end{bmatrix} \tag{39}$$

And so, in the same way as the above trajectories are interconnected, for a transformation from the unrealistic Cartesian coordinate system of [14], to a realistic spherical coordinate system, the transformed trajectories become

$$r_d = [R_0 cos(\phi)sin(\theta); R_0 sin(\phi)sin(\theta); R_0 cos(\theta)] \tag{40}$$

$$\begin{aligned} v_d = (R_0 * \pi/phase)[((cos((\theta)).^2 - (sin((\phi)).^2); \\ 2((cos((\phi)sin(\theta); -sin(\theta)] + (R_0/(4\sqrt{3})))[cos(\theta) - sin(\phi)sin(\theta); \\ cos(\phi)sin(\theta) - cos(\theta); sin(\theta)sin(\phi) - cos(\phi)sin(\theta)] \end{aligned} \tag{41}$$

$$\begin{aligned}\dot{v}_d = (R_0(\pi^2)/(phase^2))[-2cos(\phi)sin(\theta)2((cos(phi)).^2) - 2(sin(\theta).^2)\\ - cos(\theta)] + ((R_0pi)/(4\sqrt{3}phase))[-2(cos(\phi)sin(\theta)) - sin(\phi)\\ (-((sin(\phi)).^2) + sin(\theta) + (cos(\phi)).^2)((sin(\theta)).^2 + 2cos(\phi).sin(\theta) - (cos(\phi)).^2)]\end{aligned} \tag{42}$$

where in addition the angular velocity ω and the quaternions q trajectory is kept at the trajectory of [14] and thus remains unchanged. The desired position r, desired velocity v and desired acceleration $\dot{v}$ are now dependent on θ and ϕ, both given by

$$\phi = \theta = (t\pi)/phase \tag{43}$$

Note that since in this research, only a symmetrical spherical trajectory is used and so θ and ϕ are identical.

4.3 PD Controller

A PD-controller, or a proportional derivative controller, is used as a first test controller to see if the dynamics can be controlled to some degree. However, in order to compare the both controllers later on in this thesis, this PD-controller will be used for following a trajectory. For the satellite to be controlled entirely, both the orientation (attitude) and the orbit need to be controlled [ref to paper of Mauricio]. the attitude PD controller will control the quaternions q and angular velocity ω, while the orbital PD controller will control position vector r and tangential velocity v. The attitude PD-controller has predefined gains that are tuned beforehand and are given by $K_p 0.6$ and $K_d = 0.1$. The formula is given by

$$u(t) = -K_p q_{error} - K_d \omega, \tag{44}$$

with K_p being the gain of the proportional term, K_d as the derivative term and q_{error} is the error quaternion, given by [26]

$$q_{error} = \begin{bmatrix} q_4^d & -q_3^d & q_2^d \\ q_3^d & q_4^d & -q_1^d \\ -q_2^d & q_1^d & q_4^d \\ -q_1^d & -q_2^d & -q_3^d \end{bmatrix}^T q, \tag{45}$$

for $q = [q_1 \quad q_2 \quad q_3 \quad q_4]^T$. Giving us the full formula for the PD controller by

$$u(t) = 0.6 \begin{bmatrix} q_4^d & -q_3^d & q_2^d \\ q_3^d & q_4^d & -q_1^d \\ -q_2^d & q_1^d & q_4^d \\ -q_1^d & -q_2^d & -q_3^d \end{bmatrix}^T (q - 0.1) I^{-1} p, \tag{46}$$

In which I is the inertia matrix, p is the angular momentum and gains K_p and K_d are given by 0.6 and 0.1, respectively

The orbital PD controller is found in [27]. It is similar to the attitude PD controller and its formula is given by

$$f_r = -\underbrace{K_r(r - R_0)}_{\text{spring force}} - \underbrace{D_r M^{-1} p}_{\text{damping force}} \tag{47}$$

After tuning the gains, they are given by $K_r = 4$ and $D_t = 20$.

4.4 Non-Linear Controller

The non-linear controller that is used, comes from [14] and developing it involves three distinctive steps. These steps are now provided in detail, in order to gain a good understanding of the mathematics behind the non-linear controller. The first step in the development of the controller starts with the derivation port-Hamiltonian system, in which the dynamics of the attitude of the satellite are captured. This is further explained in section 5.1 and ends with the system given in the form

$$\dot{x} = (J(x,t) - R(x,t))\,\frac{\partial H(x,t)^T}{\partial x} + g(x,t)u \tag{28}$$

$$y = g(x,t)^T \frac{\partial H(x,t)^T}{\partial x} \tag{31}$$

Thereafter, the port-Hamiltonian framework is transformed by a generalized canonical transformation to embed trajectory tracking. In other words, the port-Hamiltonian system is changed to an error system, according to transformation given by

$$\begin{aligned} \bar{x} &= \Phi(x,t) \\ \bar{H} &= H(x,t) + U(x,t) \\ \bar{y} &= y + \alpha(x,t) \\ \bar{u} &= u + \beta(x,t) \end{aligned} \tag{48}$$

which transforms the system to the form

$$\begin{cases} \dot{\bar{x}} = \left(\bar{J}(\bar{x},t) - \bar{R}(\bar{x},t)\right)\dfrac{\partial \bar{H}(\bar{x},t)^T}{\partial \bar{x}} + \bar{g}(\bar{x},t)\bar{u} \\ \bar{y} = \bar{g}(\bar{x},t)^T \dfrac{\partial \bar{H}(\bar{x},t)^T}{\partial \bar{x}} \end{cases} \tag{49}$$

In which the new states are given by $\bar{x}$, the new Hamiltonian is given by $\bar{H}$, the new output is given by $\bar{y}$ and the new input is given by $\bar{u}$. This described transformation has to satisfy a total of three lemmas. The underlying reasons of why these three lemmas are essential for a working non-linear controller will be excluded here. Alternatively, we will provide what these lemmas mathematically entail. Lemma 1 entails that

$$\frac{\partial \Phi}{\partial(x,t)}\begin{pmatrix} (J-R)\dfrac{\partial U}{\partial x}^T + (K-S)\dfrac{\partial (H+U)}{\partial x}^T + g\beta \\ -1 \end{pmatrix} = 0 \tag{50}$$

which means that we have to find U, β, α and Φ such that (50) holds. We will assume that the found U, β, α and Φ in [14] are true and after defining them, continue directly with the mathematical operations that lead to the final controller. To start, α is defined as

$$\alpha(t) \equiv -y^d(t) = \begin{bmatrix} -(v_1^d + \rho_3\omega_2^d - \rho_2\omega_3^d) \\ -(v_2^d + \rho_1\omega_3^d - \rho_3\omega_1^d) \\ -(v_3^d + \rho_2\omega_1^d - \rho_1\omega_2^d) \\ -\omega_1^d \\ -\omega_2^d \\ -\omega_3^d \end{bmatrix} \tag{51}$$

Thereafter, U is defined as

$$U = -p^T\omega^d - s^Tv^d + \frac{1}{2}\omega^{dT}I\omega^d + \frac{1}{2}v^{dT}mv^d \tag{52}$$

However, Lemma 2 states that

$$(H+U)(x,t) \geq (H+U)(x^d(t),t) = 0 \tag{53}$$

and if we recall (21), then $(H+U)$ is then given by

$$(H+U) = \frac{1}{2}p^TI^{-1}p + \frac{1}{2}s^Tm^{-1}s - p^T\omega^d - s^Tv^d \tag{54}$$

which contradicts Lemma 2. Therefore, we have to add another term to U, so that Lemma 2 is satisfied, giving us

$$U = -p^T\omega^d - s^Tv^d + \frac{1}{2}\omega^{dT}I\omega^d + \frac{1}{2}v^{dT}mv^d \tag{55}$$

Now that we found U, α, β is defined as

$$\beta = \begin{bmatrix} m\omega_3v_2^d - m\omega_2v_3^d - m\dot{v}_1^d \\ m\omega_1v_2^d - m\omega_3v_1^d - m\dot{v}_2^d \\ m\omega_2v_1^d - m\omega_1v_2^d - m\dot{v}_3^d \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{bmatrix} \tag{56}$$

$$\beta_4 = p_2\omega_3^d - p_3\omega_2^d + p_3(m\omega_1v_2^d - m\omega_3v_1^d - m\dot{v}_2^d) - p_2(m\omega_2v_1^d - m\omega_1v_2^d - m\dot{v}_3^d) - I_{11}\dot{\omega}_1^d - I_{12}\dot{\omega}_2^d - I_{13}\dot{\omega}_3^d$$

$$\beta_5 = p_3\omega_1^d - p_1\omega_3^d + p_1(m\omega_2v_1^d - m\omega_1v_2^d - m\dot{v}_3^d) - p_3(m\omega_3v_2^d - m\omega_2v_3^d - m\dot{v}_1^d) - I_{12}\dot{\omega}_1^d - I_{22}\dot{\omega}_2^d - I_{23}\dot{\omega}_3^d$$

$$\beta_6 = p_1\omega_2^d - p_2\omega_1^d + p_2(m\omega_3v_2^d - m\omega_2v_3^d - m\dot{v}_1^d) - p_1m\omega_1v_2^d - m\omega_3v_1^d - m\dot{v}_2^d) - I_{13}\dot{\omega}_1^d - I_{23}\dot{\omega}_2^d - I_{33}\dot{\omega}_3^d$$

In addition, in (35) K and S are defined as

$$S = 0 \tag{57}$$

$$K = \begin{bmatrix} 0_{3\times3} & 0_{3\times4} & 0_{3\times3} & K_\alpha \\ 0_{4\times3} & 0_{4\times4} & 0_{4\times3} & 0_{4\times3} \\ 0_{3\times3} & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \\ -K_\alpha^T & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} K_\alpha = \begin{bmatrix} 0 & \frac{\omega_2^d}{\omega_2-\omega_2^d}(r_3-r_3^d) & \frac{\omega_3^d}{\omega_3-\omega_3^d}(r_2-r_2^d) \\ \frac{\omega_1^d}{\omega_1-\omega_1^d}(r_3-r_3^d) & 0 & \frac{\omega_3^d}{\omega_3-\omega_3^d}(r_1-r_1^d) \\ \frac{\omega_1^d}{\omega_1-\omega_1^d}(r_2-r_2^d) & \frac{\omega_2^d}{\omega_2-\omega_2^d}(r_1-r_1^d) & 0 \end{bmatrix} \tag{58}$$

Now finally, the error system can be derived by solving (35), which then gives us

$$\bar{x} = \begin{bmatrix} \bar{r} \\ \bar{q} \\ \bar{s} \\ \bar{p} \end{bmatrix} = \begin{bmatrix} r_1 - r_1^d \\ r_2 - r_2^d \\ r_3 - r_3^d \\ q_4^d q_1 - q_3^d q_2 + q_2^d q_3 - q_1^d q_4 \\ q_3^d q_1 + q_4^d q_2 - q_1^d q_3 - q_2^d q_4 \\ -q_2^d q_1 + q_1^d q_2 + q_4^d q_3 - q_3^d q_4 \\ q_1^d q_1 + q_2^d q_2 + q_3^d q_3 + q_4^d q_4 - 1 \\ s_1 - s_1^d \\ s_2 - s_2^d \\ s_3 - s_3^d \\ p_1 - p_1^d \\ p_2 - p_2^d \\ p_3 - p_3^d \end{bmatrix} = \Phi(x) \tag{59}$$

Now the system is in the form

$$\begin{cases} \dot{\bar{x}} = \left(\bar{J}(\bar{x},t) - \bar{R}(\bar{x},t)\right) \dfrac{\partial \bar{H}(\bar{x},t)^T}{\partial \bar{x}} + \bar{g}(\bar{x},t)\bar{u} \\ \bar{y} = \bar{g}(\bar{x},t)^T \dfrac{\partial \bar{H}(\bar{x},t)^T}{\partial \bar{x}} \end{cases} \tag{60}$$

where

$$\bar{g} = \frac{\partial \Phi}{\partial x} g, \;\; \bar{J} = \frac{\partial \Phi}{\partial x}(J+K)\frac{\partial \Phi}{\partial x}^T, \;\; \bar{R} = \frac{\partial \Phi}{\partial x}(R+S)\frac{\partial \Phi}{\partial x}^T \tag{61}$$

Now that the error system is defined, it is time to apply passivity based control to guarantee asymptotic stability of error system, causing it to following trajectory. However at this stage, $\bar{H}$ is not positive definite [14], which is lemma 3 we need to satisfy. Therefore, we will add a virtual potential function $\bar{U}$ so that the transformed Hamiltonian $\hat{H} = \bar{H} + \bar{U}$ will satisfy lemma 3 and thus is positive definite. $\bar{U}$ is a function of ($\bar{r}$, $\bar{q}$), being both positive definite. Then, the transformed Hamiltonian is given by

$$\hat{H} = \frac{1}{2}\bar{p}^T I^{-1}\bar{p} + \frac{1}{2}\bar{s}^T m^{-1}\bar{s} + \bar{U}(\bar{r},\bar{q}) \tag{62}$$

which gives us the new system

$$\dot{\bar{x}} = \bar{J}(\bar{x},t)\frac{\partial \hat{H}(\bar{x})^T}{\partial}\bar{x} + \bar{g}(\bar{x},t)\hat{u} \tag{63a}$$

$$\hat{y} = \bar{g}(\bar{x},t)^T \frac{\partial \hat{H}(\bar{x})}{\partial \bar{x}} \tag{63b}$$

Now that we have the old transformed system equation (61) and the new system equation (63), we can obtain the relation of the control inputs, which is given by

$$\bar{u} = \hat{u} - \bar{G}^{-1}\bar{J}_{12}^{T}\begin{bmatrix} \dfrac{\partial \bar{U}}{\partial \bar{r}}^{T} \\ \dfrac{\partial \bar{U}}{\partial \bar{q}}^{T} \end{bmatrix} \tag{64}$$

From which $\hat{u}$ is derived to be

$$\hat{u} = -C\hat{y} = -C\bar{g}(\bar{x},t)^{T}\frac{\partial \hat{H}(\bar{x})^{T}}{\partial \bar{x}} = -C\frac{\partial \Phi}{\partial x}\begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ 0_{4\times3} & 0_{4\times3} \\ E_{3\times3} & 0_{3\times3} \\ \tilde{\rho} & E_{3\times3} \end{bmatrix}^{T}\begin{bmatrix} k*\bar{r} \\ l*\bar{q} \\ \bar{v} \\ \bar{\omega} \end{bmatrix} \tag{65}$$

where $\hat{H} = \frac{1}{2}\bar{p}^{T}I^{-1}\bar{p} + \frac{1}{2}\bar{s}^{T}m^{-1}\bar{s} + \bar{U}(\bar{r},\bar{q})$. Where $\bar{U}$ is defined to be

$$\bar{U}(\bar{r},\bar{q}) = \frac{1}{2}(k\bar{r}^{T}\bar{r} + l\bar{q}^{T}\bar{q}) \tag{66}$$

and $\dfrac{\partial \Phi}{\partial(x)}$ is defined to be

$$\frac{\partial \Phi}{\partial x} = \begin{bmatrix} E_{3\times3} & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \\ 0_{4\times3} & \dfrac{\partial \Phi}{\partial q} & 0_{4\times3} & 0_{4\times3} \\ 0_{3\times3} & 0_{3\times4} & E_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times4} & 0_{3\times3} & E_{3\times3} \end{bmatrix} \tag{67}$$

where

$$\frac{\partial \Phi}{\partial q} = \begin{bmatrix} q_4^d & -q_3^d & q_2^d & -q_1^d \\ q_3^d & q_4^d & -q_1^d & -q_2^d \\ -q_2^d & q_1^d & q_4^d & -q_3^d \\ q_1^d & q_2^d & q_3^d & q_4^d \end{bmatrix} \tag{68}$$

Now it is time to solve for $\bar{G}^{-1}$, $\bar{J}^{T_{12}}$ and $\begin{bmatrix} \dfrac{\partial \bar{U}}{\partial \bar{r}}^{T} \\ \dfrac{\partial \bar{U}}{\partial \bar{q}}^{T} \end{bmatrix}$ from (42). Consequently, $\bar{G}$ is found to be

$$\bar{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -\rho_3 & \rho_2 & 1 & 0 & 0 \\ \rho_3 & 0 & -\rho_1 & 0 & 1 & 0 \\ -\rho_2 & \rho_1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{69}$$

and $\bar{J}_{12}^{T}$ is taken from the left bottom of matrix $\bar{J} = \begin{pmatrix} \bar{J}_{11} & \bar{J}_{12} \\ -\bar{J}_{12}^{T} & \bar{J}_{22} \end{pmatrix}$ where $\bar{J} =$

$\dfrac{\partial \Phi}{\partial x}(J+K)\dfrac{\partial \Phi}{\partial x}^T$ and is given by

$$\bar{J} = \begin{bmatrix} E_{3\times3} & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \\ 0_{4\times3} & \dfrac{\partial \Phi}{\partial q} & 0_{4\times3} & 0_{4\times3} \\ 0_{3\times3} & 0_{3\times4} & E_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times4} & 0_{3\times3} & E_{3\times3} \end{bmatrix} \begin{bmatrix} 0_{3\times3} & 0_{3\times4} & E_{3\times3} & \tilde{r}+K_\alpha \\ 0_{4\times3} & 0_{4\times4} & 0_{4\times3} & \dfrac{\Omega(q)}{2} \\ -E_{3\times3} & 0_{3\times4} & -m\tilde{\omega} & 0_{3\times3} \\ \tilde{r}-K_\alpha^T & -\dfrac{\Omega(q)^T}{2} & 0_{3\times3} & \tilde{\rho} \end{bmatrix} \tag{70}$$

$$\begin{bmatrix} E_{3\times3} & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \\ 0_{4\times3} & \dfrac{\partial \Phi}{\partial q} & 0_{4\times3} & 0_{4\times3} \\ 0_{3\times3} & 0_{3\times4} & E_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times4} & 0_{3\times3} & E_{3\times3} \end{bmatrix} \tag{71}$$

which gives us $\bar{J}_{12}$ as

$$\bar{J}_{12} = \begin{bmatrix} E_{3\times3} & \tilde{r}+K_\alpha \\ 0_{4\times3} & \dfrac{1}{2}\Omega\dfrac{\partial \Phi}{\partial q} \end{bmatrix} \tag{72}$$

in which K_α is given by

$$K_\alpha = \begin{bmatrix} 0 & \dfrac{\omega_2^d}{\omega_2-\omega_2^d}(r_3-r_3^d) & \dfrac{\omega_3^d}{\omega_3-\omega_3^d}(r_2-r_2^d) \\ \dfrac{\omega_1^d}{\omega_1-\omega_1^d}(r_3-r_3^d) & 0 & \dfrac{\omega_3^d}{\omega_3-\omega_3^d}(r_1-r_1^d) \\ \dfrac{\omega_1^d}{\omega_1-\omega_1^d}(r_2-r_2^d) & \dfrac{\omega_2^d}{\omega_2-\omega_2^d}(r_1-r_1^d) & 0 \end{bmatrix}$$

Finally, $\text{col}\begin{bmatrix} \dfrac{\partial \bar{U}}{\partial \bar{r}}^T & \dfrac{\partial \bar{U}}{\partial \bar{q}}^T \end{bmatrix}$ is derived to be

$$\begin{bmatrix} \dfrac{\partial \bar{U}}{\partial \bar{r}}^T \\ \dfrac{\partial \bar{U}}{\partial \bar{q}}^T \end{bmatrix} = \begin{bmatrix} k\bar{r} \\ l\bar{q} \end{bmatrix} \tag{73}$$

When we now combine everything by filling in (43), we obtain the final non-linear controller as proposed by [14], which is given by

$$\hat{u} = -C \begin{bmatrix} E_{3\times3} & 0_{3\times4} & 0_{3\times3} & 0_{3\times3} \\ 0_{4\times3} & \dfrac{\partial \Phi}{\partial q} & 0_{4\times3} & 0_{4\times3} \\ 0_{3\times3} & 0_{3\times4} & E_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times4} & 0_{3\times3} & E_{3\times3} \end{bmatrix} \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ 0_{4\times3} & 0_{4\times3} \\ E_{3\times3} & 0_{3\times3} \\ \tilde{\rho} & E_{3\times3} \end{bmatrix})^T \begin{bmatrix} k*\bar{r} \\ l*\bar{q} \\ \bar{v} \\ \bar{\omega} \end{bmatrix} = \begin{bmatrix} \bar{v}_1-\rho_3\bar{\omega}_2+\rho_2\bar{\omega}_3 \\ \bar{v}_2-\rho_3\bar{\omega}_1+\rho_1\bar{\omega}_3 \\ \bar{v}_3-\rho_2\bar{\omega}_1+\rho_1\bar{\omega}_2 \\ \bar{\omega}_1 \\ \bar{\omega}_2 \\ \bar{\omega}_3 \end{bmatrix} \tag{74}$$

$$\bar{u} = -C(\begin{bmatrix} \bar{v}_1 - \rho_3\bar{\omega}_2 + \rho_2\bar{\omega}_3 \\ \bar{v}_2 - \rho_3\bar{\omega}_1 + \rho_1\bar{\omega}_3 \\ \bar{v}_3 - \rho_2\bar{\omega}_1 + \rho_1\bar{\omega}_2 \\ \bar{\omega}_1 \\ \bar{\omega}_2 \\ \bar{\omega}_3 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -\rho_3 & \rho_2 & 1 & 0 & 0 \\ \rho_3 & 0 & -\rho_1 & 0 & 1 & 0 \\ -\rho_2 & \rho_1 & 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} E_{3\times 3} & 0_{3\times 4} \\ \tilde{r} + K_\alpha & \frac{1}{2}\Omega\frac{\partial \Phi}{\partial q} \end{bmatrix} \begin{bmatrix} k\bar{r} \\ l\bar{q} \end{bmatrix}$$

$$= -C \begin{bmatrix} \bar{v}_1 - \rho_3\bar{\omega}_2 + \rho_2\bar{\omega}_3 \\ \bar{v}_2 - \rho_3\bar{\omega}_1 + \rho_1\bar{\omega}_3 \\ \bar{v}_3 - \rho_2\bar{\omega}_1 + \rho_1\bar{\omega}_2 \\ \bar{\omega}_1 \\ \bar{\omega}_2 \\ \bar{\omega}_3 \end{bmatrix} - \begin{bmatrix} E_{3\times 3} & 0_{3\times 4} \\ (-\tilde{\rho} + (\tilde{r} + K_\alpha))^T & (\frac{\partial \Phi}{\partial q}\Omega)^T \end{bmatrix} \begin{bmatrix} k\bar{r} \\ l\bar{q} \end{bmatrix} \tag{75}$$

Finally, gains k, l are given in [14] as

$$k > m(\frac{M_2}{I_3} + \frac{M_3}{I_2} + \frac{M_3}{I_1} + \frac{M_1}{I_3} + \frac{M_1}{I_2} + \frac{M_2}{I_1}) + \frac{\sqrt{3}}{\mu}(\frac{1}{I_1}(N_2 + N_3) + \frac{1}{I_2}(N_3 + N_1)$$
$$+\frac{1}{I_3}(N_1 + N_2)) + \frac{2\sqrt{3}}{\mu}k(M_1 + M_2 + M_3) + \frac{1}{\mu}v(\mu)^T C v(\mu)$$

$$l > 2m\mu(\frac{M_2}{I_3} + \frac{M_3}{I_2} + \frac{M_3}{I_1} + \frac{M_1}{I_3} + \frac{M_1}{I_2} + \frac{M_2}{I_1}) + 2\sqrt{3}(\frac{1}{I_1}(N_2 + N_3) + \frac{2}{I_2}(N_3 + N_1)$$
$$+\frac{2}{I_3}(N_1 + N_2)) + 4\sqrt{3}k(M_1 + M_2 + M_3) + 2v(\mu)^T C v(\mu)$$

$$\mu > 2\sqrt{3}(M_1 + M_2 + M_3)$$

where $v(\mu) = [\sqrt{\mu}, \sqrt{\mu}, \sqrt{\mu}, \sqrt[4]{3}, \sqrt[4]{3}, \sqrt[4]{3}]^T$

For simplicity, the gains are initially chosen to be k=25 and l= 45, after testing multiple values in the simulated environment of [14].

5 Results

Now that all preliminaries and the theoretical background are introduced, the simulations are run, which includes the system with disturbance T_{gg} as in (1) and with PD orbital input control signal f as in (15). For structure, first the control of the attitude with PD controller and the non-linear controller is presented in the non-realistic scenario. For this scenario, the scenario provided by [14] is used. This way, both controllers are tested and prove successful replication of the results of [14]. Thereafter the orbital control with both the PD- and the non-linear is provided in the non-realistic environment given by [14]. Thereafter, the results of the simulation in the realistic environment are provided. Initiated by background information about the logic behind the realistic trajectories, the attitude control of the PD- and the non-linear controller is again first provided, followed by the orbital control of the PD- and the non-linear controller.

5.1 Attitude control of non-realistic scenario with PD-controller

The PD attitude control with control output u is given by eq. (34). The simulation shows strong converging behaviour, which is proven by the error in every four dimensions of q of eq. (17) converges to a zero steady state error (Figure 5). Figure 6 shows that the angular velocity ω as found in (16) is indeed stabilized as it converges to a steady state value of zero within a 5 sec time frame, which means that the attitude remains constant.

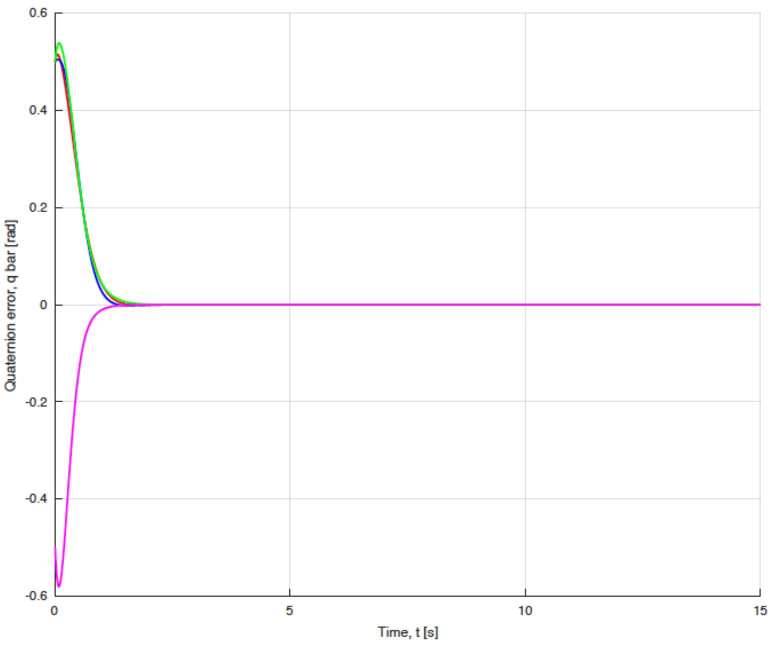

Figure 5: The error with respect to the desired trajectory of the quaternions, controlled by eq. (34). There can be clearly seen a steady state error as it converges to zero. However it converges less fast and less accurate then the non-linear controlled system

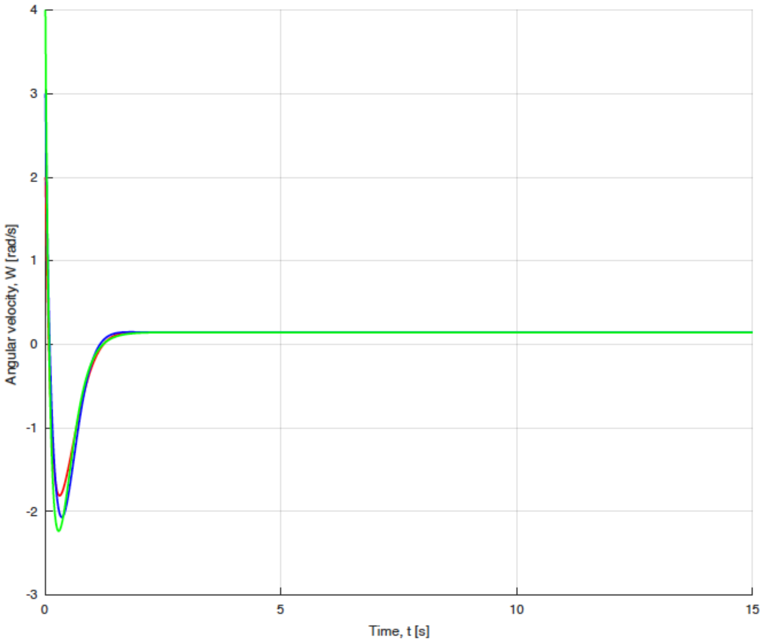

Figure 6: Angular velocity ω as found in (15) vs Time, where it can be clearly seen the all dimensions of the angular velocity converge to zero, implying a steady state attitude. This means that eq. (34) successfully controls the attitude

5.2 Attitude control of non-realistic scenario with non-linear controller

The non-linear controller is a stronger controller, as can be read in section [4.4] by looking at the gains of the final controller, that is expected to converge faster and more accurate to the zero steady state error. Consequently, the quaternions show proper trajectory following. However, the desired trajectory is not necessarily faster than with the help of the PD controller. Nonetheless, the zero steady state error the quaternion error $\bar{q}$ converges too, is reached fast and accurately. It is proven by (Figure 7). Moreover, the non-linear controller again shows very strong controlled behaviour with respect to the angular velocity ω. The angular velocity ω converges much faster then with the PD controller, namely under 1 second (Figure 8).

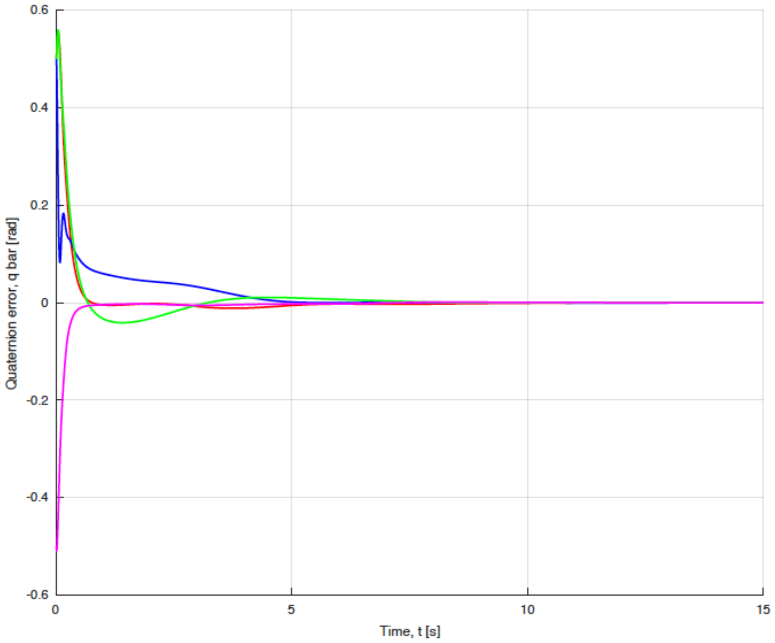

Figure 7: The error with respect to the desired trajectory of the quaternions, with the non-linear controller. There can be clearly seen a steady state error as it converges to zero.

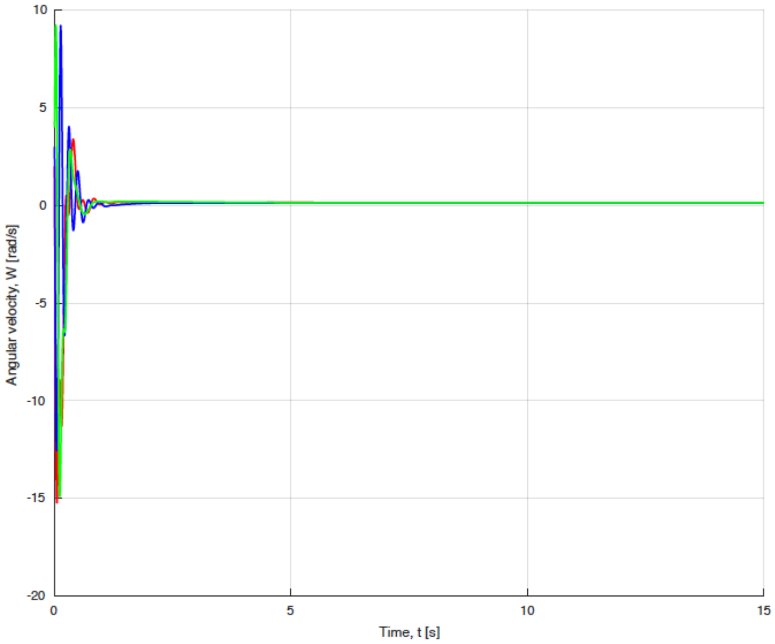

Figure 8: Angular Velocity vs Time. It can be clearly seen that all dimensions of angular velocity ω converge to zero, meaning that the satellite undergoes a steady attitude.

5.3 Orbital control with the PD controller in the non-realistic environment

We also see fairly accurately controlled behaviour with respect to position vector r, controlled by eq. (35). The three dimensions of r all seem to approach the desired values of the position vector. Figure 9 proves that the errors do not have high values, and start to converge to a zero steady state error. This means that the PD orbital controller is able to follow trajectory of [14] with respect to the orbit of the satellite. When the simulation is ran again with smaller time steps and adjusted gains, the same outcome occurs. Moreover, the tangential velocity, as given by eq. (15) is stable, following the trajectory of [14], depicted in Figure 10.

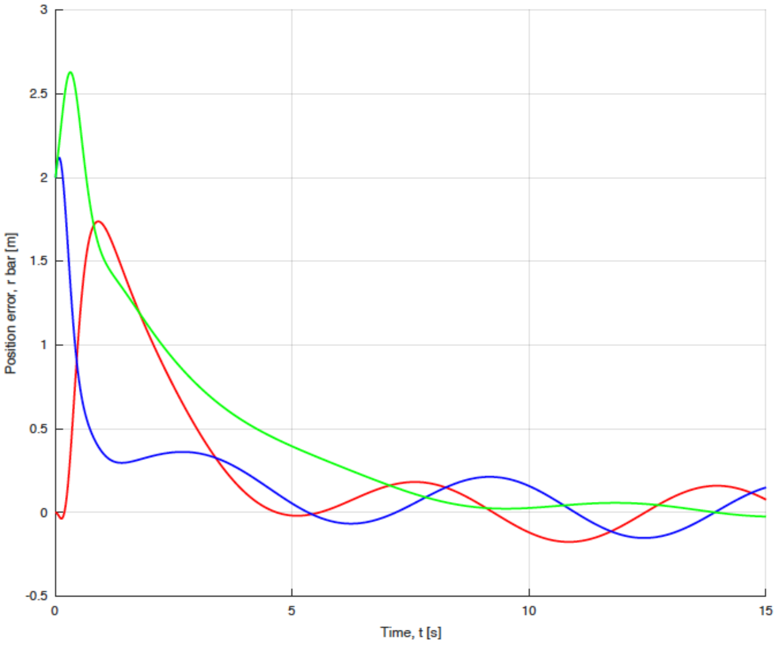

Figure 9: Error with respect to the position vector of the PD controller, controlled by eq. (35)

.

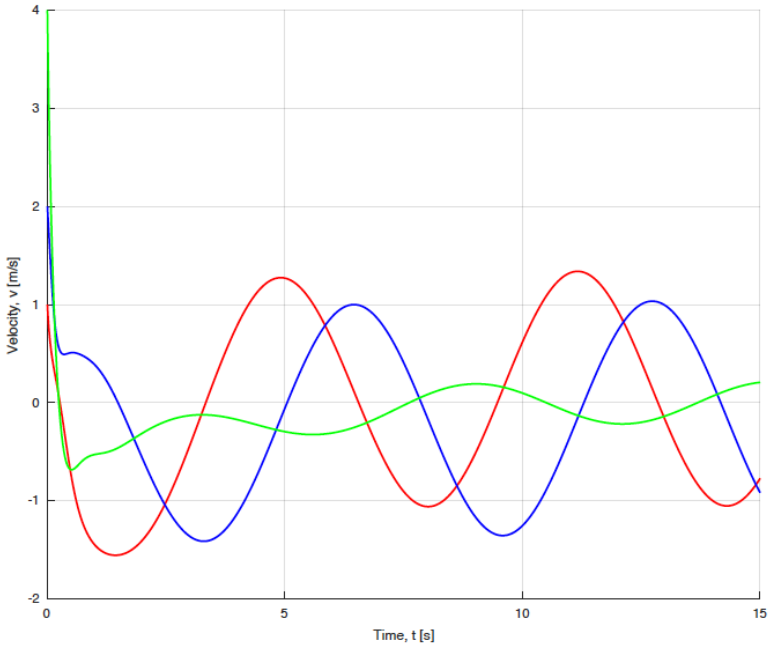

Figure 10: Tangential velocity v, as found in eq. (15) of the PD controlled trajectory of the unrealistic simulation provided by [14]. This is a part of the full trajectory defined by [14], and the velocity behaves accordingly.

.

5.4 Orbital control with the non-linear controller in the non-realistic environment

Last, the non-linear controller given by eq. (63) is able to follow accurately the desired trajectory of position vector r, which is depicted in (Figure 11), where, a fast converged steady state error of the position vector r shows that the controller is very accurate. Moreover, the dotted lines in (Figure 13) represent the desired trajectory of the orbit of the satellite, which is approximated very accurately by the non-linear controller trajectory. Summarized, the non-linear controller is able to control both the orbit and the orientation in a accurate and fast converging nature. In comparison with the PD-controller, the non-linear controller is both more accurate and converges faster. Last, (Figure 12) shows that the non-linear controller is able to successfully duplicate the results of [14].

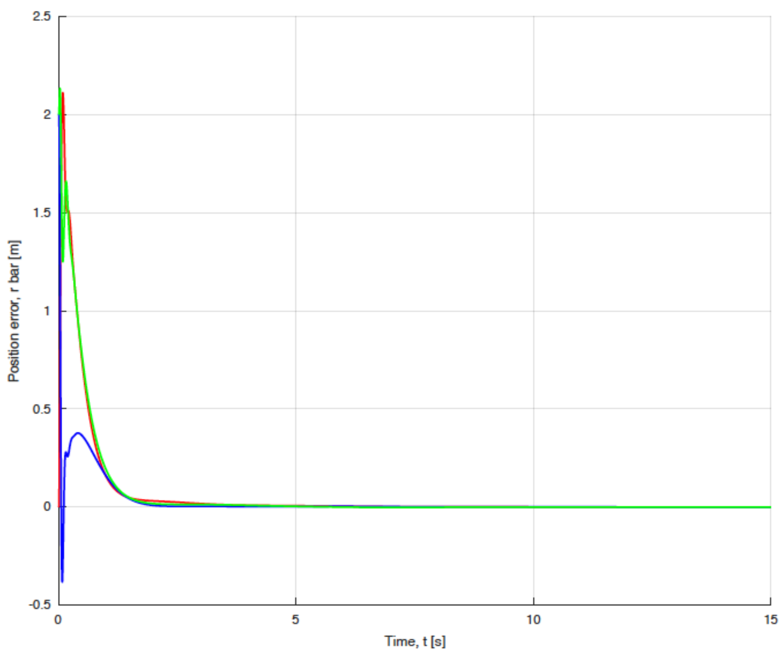

Figure 11: The error in position vector r vs Time with the non-linear controller. There can be seen clear converging behaviour, as the steady state error goes to zero

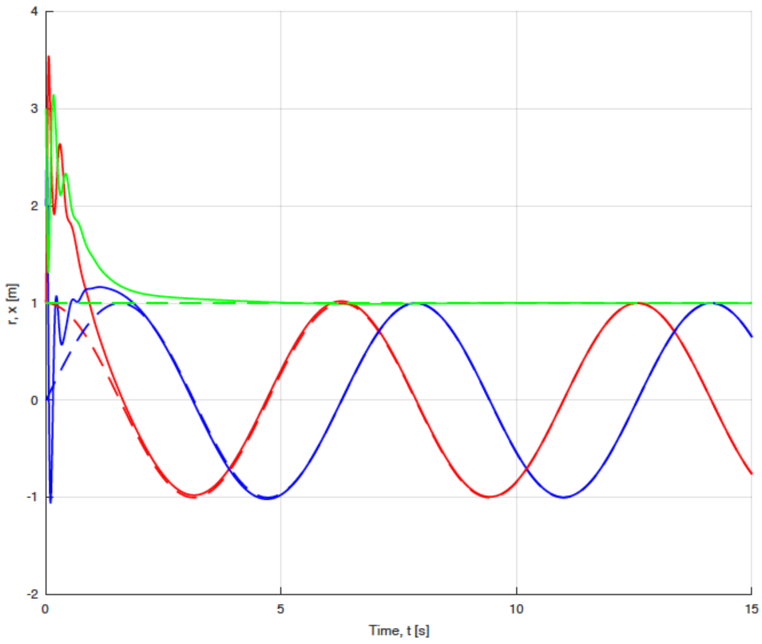

Figure 12: Position Vector r vs Time t. The dotted lines represent the desired trajectory of the position vector r, and the solid lines represent the controlled trajectory of the position vector r. It can be seen that the lines coincide

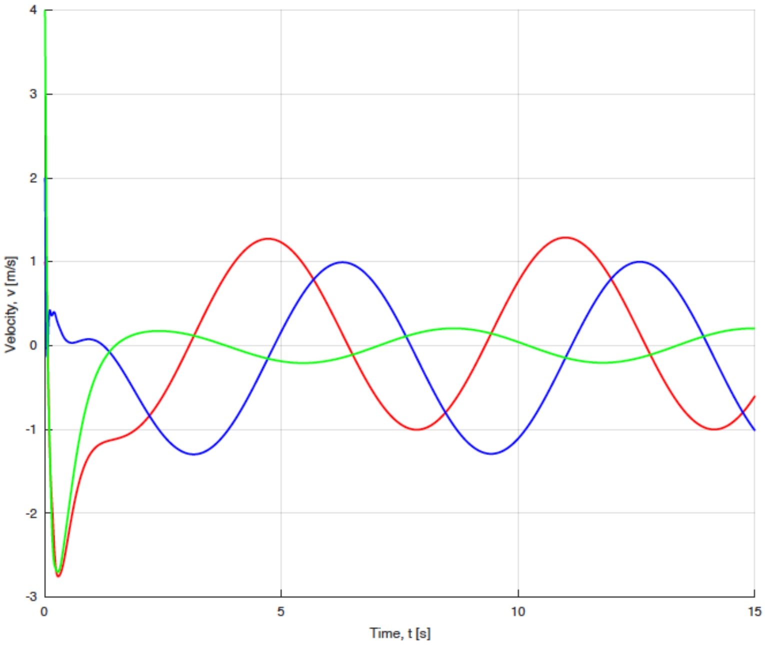

Figure 13: The velocity v as found in (15) of the non-linear controlled unrealistic simulation, following the trajectory of [14]

5.5 Simulation in a realistic trajectory

After the transformation of r_d, as shown in section [4.2], the development of a realistic scenario requires the radius r to be given by $R_0 = R_{earth} + R_{orbit} = 6.5783e+06m$. Here lies the second complexity of the implementation of the realistic scenario. The errors in the orbital control become very large, because when the position vector r is scaled up with a magnitude of 10^e6, so do the velocity and the acceleration. The propagated time, however, stays the same. This causes a highly unrealistic trajectory and velocity change very short time span, analogous of traveling multiple times around the world in no less then 5 sec. Therefore, as the trajectory is scaled up realistically with a factor of 10^6, the time elapsed is required to be scaled down with the same magnitude, to keep a realistic velocity and acceleration. To show this, first the results of the new realistic trajectories and the according transformations are considered in section [4.2]. Thereafter, three simulations with both the PD-controller and the non-linear controller provide clear inside into the physical interpretation of the control of the developed realistic scenario.

Now that the new desired realistic trajectories are transformed, three simulations are performed, summarized by Table 1.

Simulation	Controller	Time Span	Phase	Gains
1	PD	20 s	R_0/pi	$K_p = 3$, $K_d = 0.5$, $D_t = 40$, $K_r = 16$
1	Non-Linear	20 s	R_0/pi	C = [90 90 90 10 10 10], k=(1e-12)*12, l=40
2	Non-Linear	10 s	pi	C = [90 90 90 10 10 10], k=(1e-12)*12, l=40
3	PD	10 s	180000/pi	$K_p = 3$, $K_d = 0.5$, $D_t = 40$, $K_r = 16$
3	Non-Linear	10 s	180000/pi	C = [90 90 90 10 10 10], k=(1e-12)*12, l=40

Table 1: Properties of the three different simulations conducted in the realistic simulation

To make clear that complications arise when the realistic trajectory is scaled up according to R_0, simulation 2 of Table 1 is conducted. What is apparent is that the trajectory is followed, which is shown in figure 14. The dotted lines, resembling the desired trajectory in every figure, and the full lines coincide, thus it seems that in this simulation the non-linear controller is following trajectory. However, in figure 15, we still see a fairly large error that is not converging.

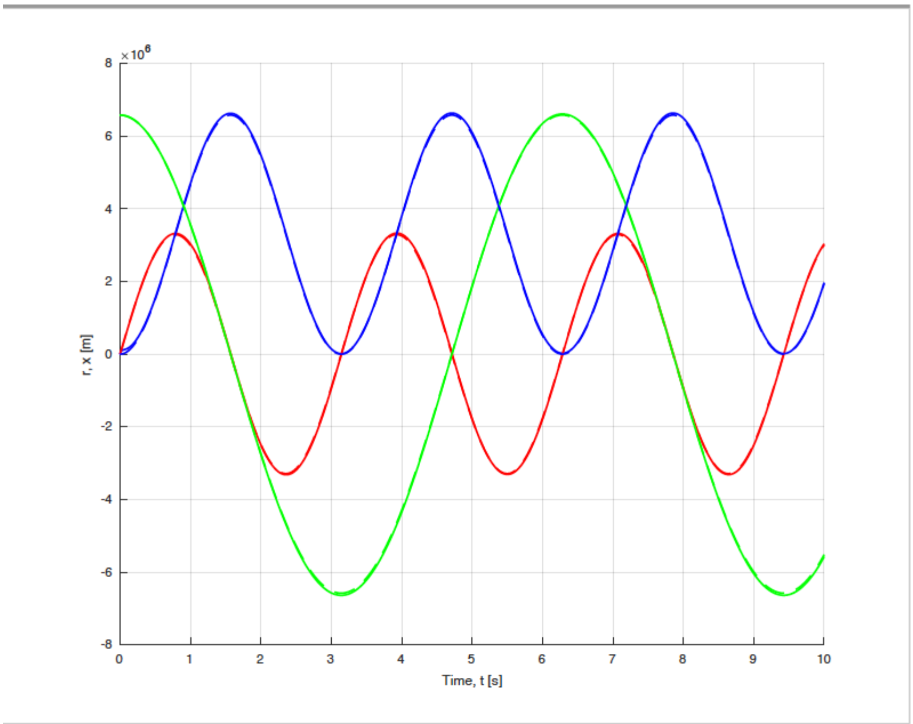

Figure 14: Plot of the r-vector in simulation 2 of Table 1 of the non-linear controller. The trajectory itself is realistic, however the time span is highly unrealistic. Here the blue, red and green line represent the x, y and z axis respectively. It must be noted that the dotted and full lines coincide in this plot, showing that the trajectory is properly approximated, even with the unrealistic time span

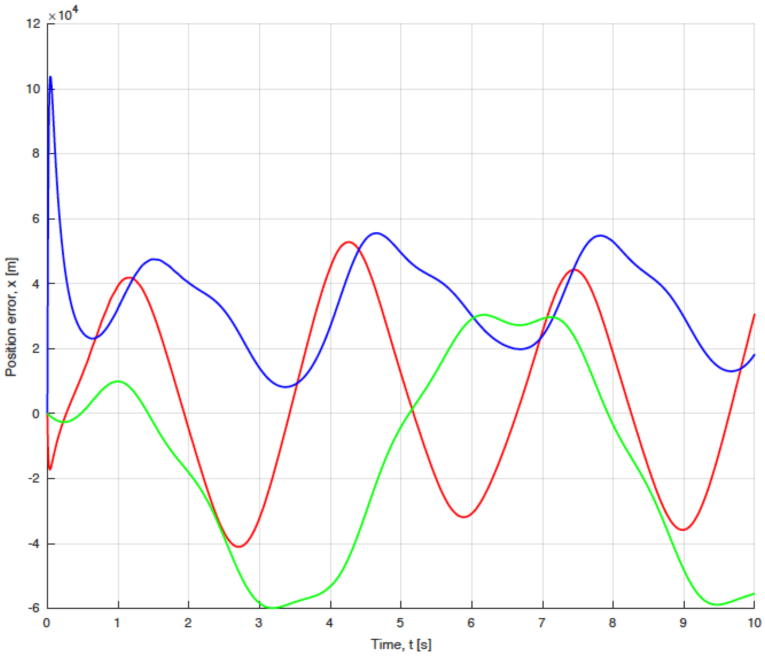

Figure 15: The error in r-vector in simulation 2 of Table 2 of the non-linear controller. Here the blue, red and green line represent the error in the x, y and z axis respectively. The errors seem large, however when comparing them to the magnitude of the trajectory, are relatively small. These large errors arise due to very large changes in r, v and $\dot{v}$ that are unrealistic.

The reason for the large errors depicted in 15, is the fact that the speed and the acceleration are now highly unrealistic. For instance, the z-direction, depicted by the green line, travels a distance of $14e10m$ in approximately 7 seconds. The speed and acceleration required for this trajectory change, are highly unrealistic. Therefore, it is clear that if r, v and $\dot{v}$ are up scaled for a realistic simulation, the phase should be down scaled accordingly, so that the time required for the trajectory also remains realistic. The next step is thus shown in simulation 3, found in Table 1. Here, the time t is scaled down by a phase of $\pi/180000$, as shown in Table 1. As can be seen in the Appendix, the trajectory has changed drastically. This is because now only a very small part of the trajectory has taken place in the elapsed time, compared to simulation 2. What must be noted, is that the non-linear controller converges faster and more accurately then the PD controller. When comparing Figure 16 and Figure 17, it can also be concluded that the position error is much smaller in the non-linear controlled simulation, then in the PD controlled situation.

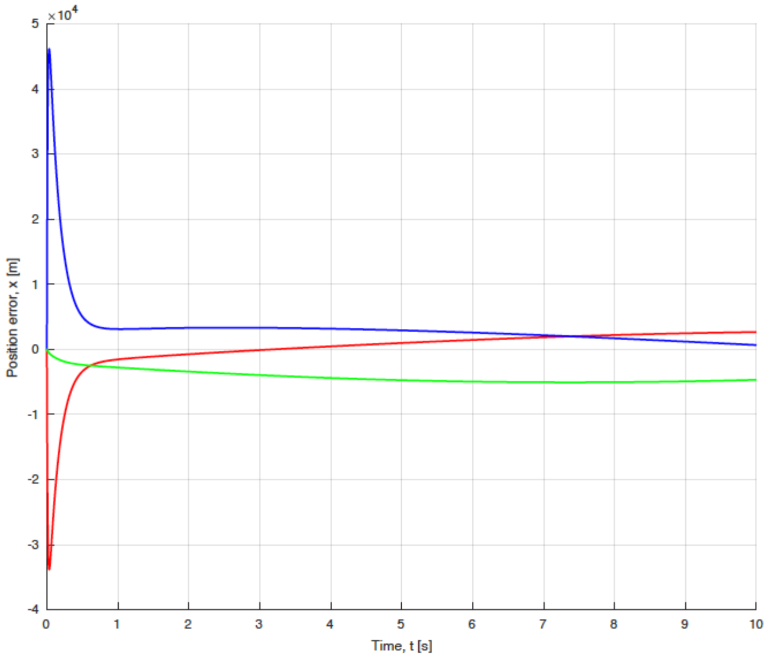

Figure 16: The error in r-vector in simulation 3 of Table 1 of the non-linear controller. Here the blue, red and green line represent the error in the x, y and z axis respectively.

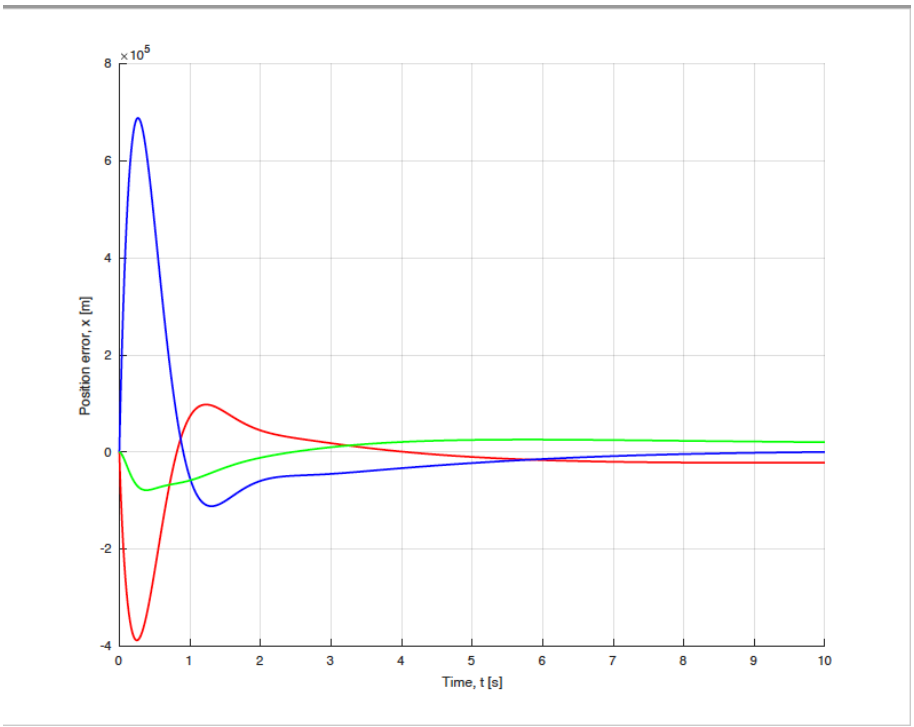

Figure 17: The error in r-vector in simulation 3 of Table 1 of the PD controller. Here the blue, red and green line represent the error in the x, y and z axis respectively.

Moreover, Figure 18 and figure 19 show that the velocity is now behaving in a realistic way both in the PD-controlled and non-linearly controlled simulation.

Apart from its initial change staring from the initial conditions, the velocity has a normal trajectory without unrealistic accelerations and behaves stable.

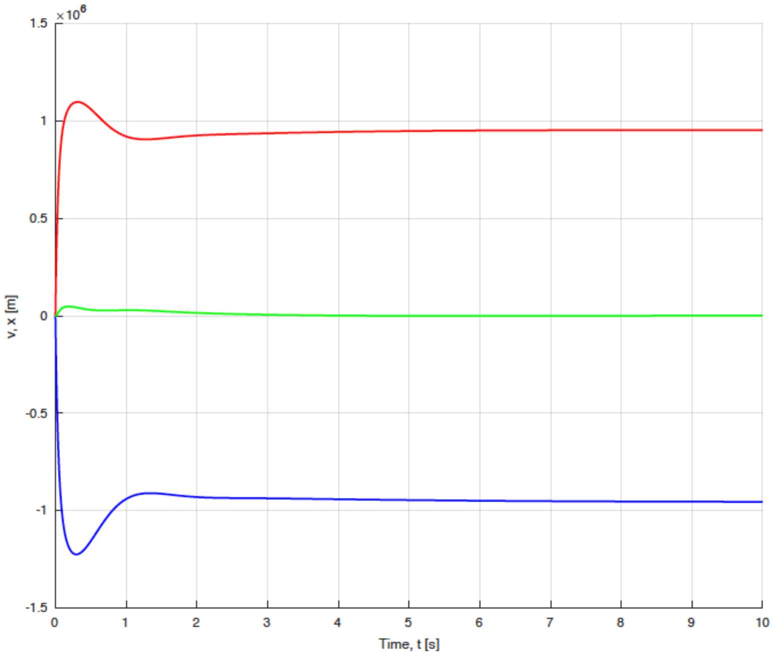

Figure 18: The velocity plot of simulation 3 of Table 1 of the PD controller. Here the blue, red and green line represent the error in the x, y and z axis respectively.

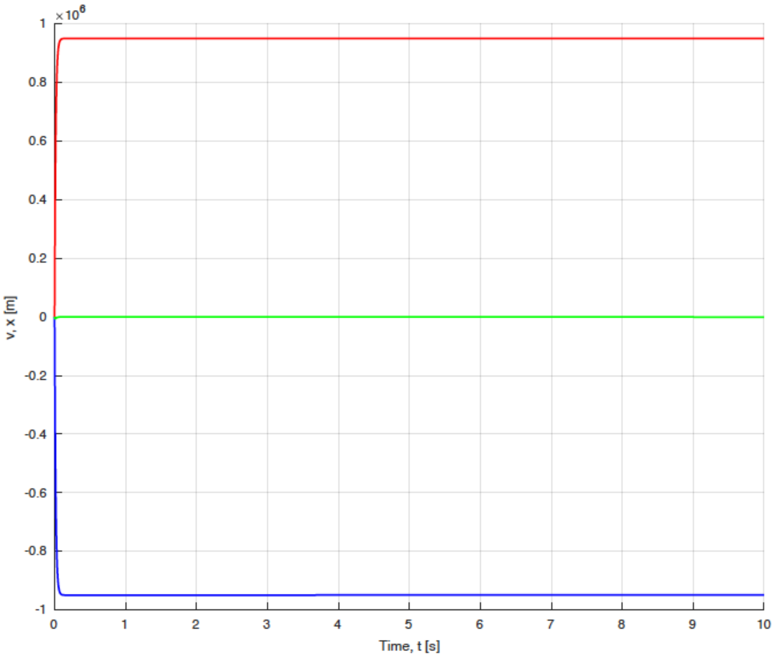

Figure 19: The velocity plot of simulation 3 of Table 1 of the non-linear controller. Here the blue, red and green line represent the error in the x, y and z axis respectively.

Figure 20 shows that the attitude of the satellite is still controlled properly by the PD controller, as is also the case with the non-linear controller.

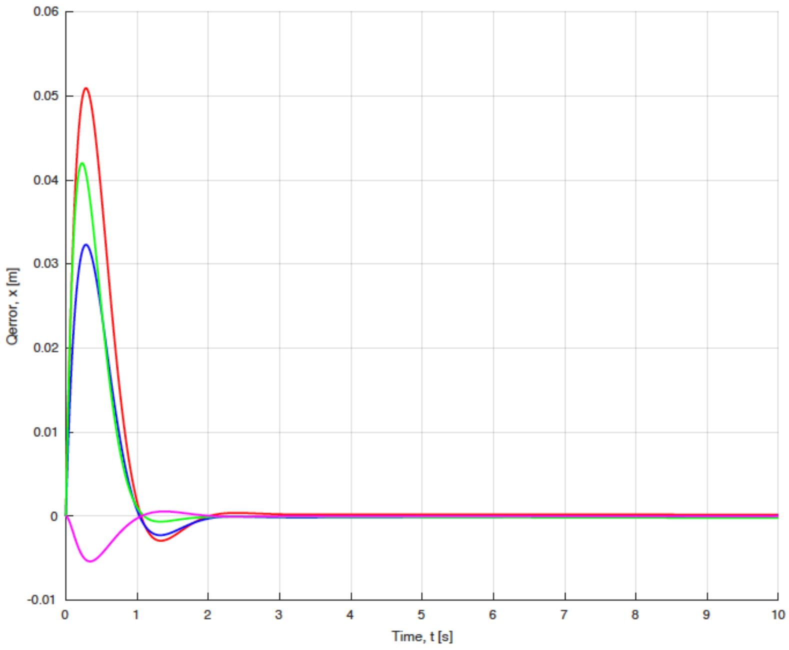

Figure 20: The error in q-vector in simulation 3 of Table 1 of the PD controller. Here, the blue, green, red and pink lines represent q_1, q_2, q_3 and q_4 respectively.

Lastly, simulation 1, as can be found in Table 1, is the most realistic simulation. The phase is scaled down with the same magnitude as orbital variables are scaled up. This means that not only does this simulation follows a realistic trajectory, it does so in the most realistic time, as can be seen in the Appendix, which shows that the trajectory hardly changes in the elapsed time. The behaviour is fairly similar to simulation 3, with Figure 21 and Figure 22 showing the same behaviour as Figure 17 and figure 16. However, this trajectory is more realistic. Since in 10 sec, only a very small fraction of the total trajectory is completed.

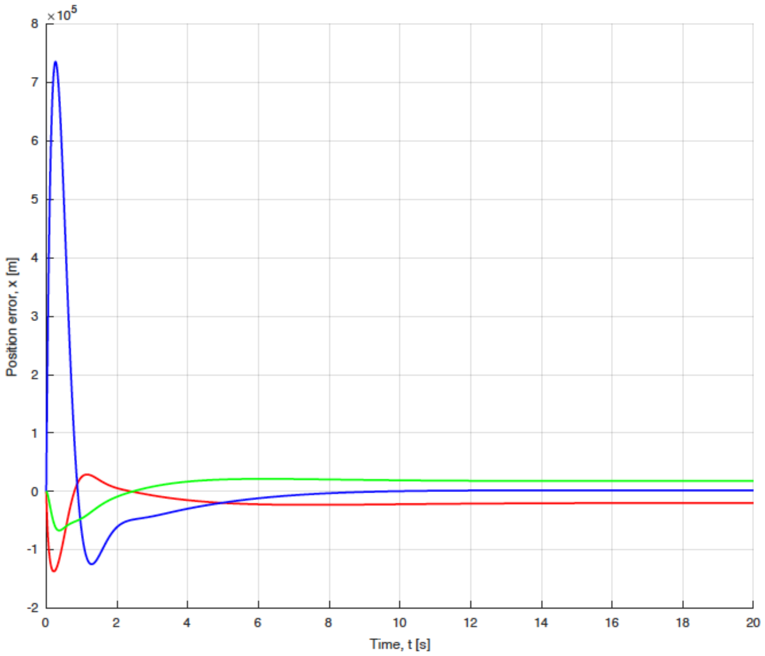

Figure 21: The error in r-vector in simulation 1 of Table 1 of the PD controller. Here the blue, red and green line represent the error in the x, y and z axis respectively.

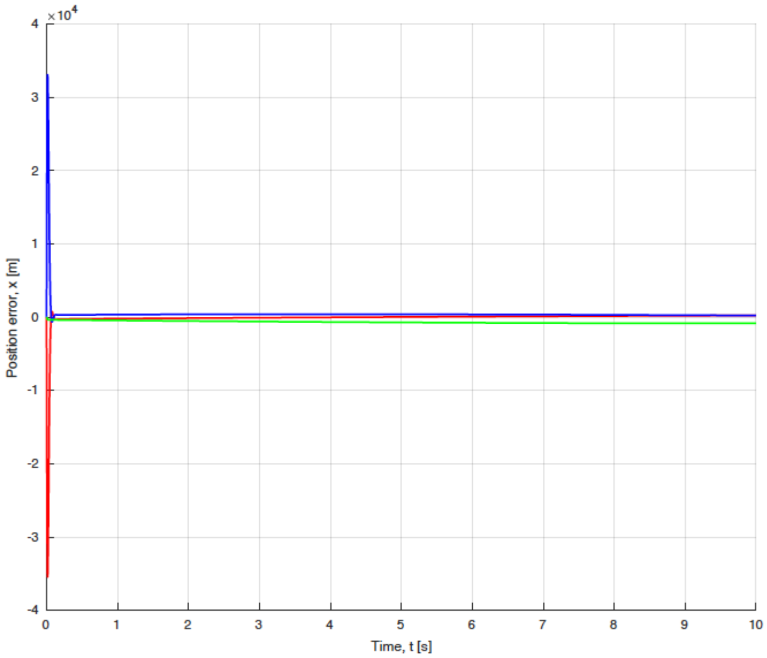

Figure 22: The error in r-vector in simulation 1 of Table 1 of the non-linear controller. Here the blue, red and green line represent the error in the x, y and z axis respectively.

6 Conclusion

To conclude, the PD controlled system shows strong controlled and converging behaviour with respect to the attitude, described by angular velocity ω and quaternions q. The orbital control is fairly accurately controlled, and it seems that the trajectory is only followed, and the error is converging to a zero steady state error. The non-linear controller is also able to control the attitude and the orbit of both the unrealistic and the realistic simulation. It shows better performance in orbital control for every scenario and better performance in the attitude control in the realistic scenario. However, in the attitude control, the performance is not necessarily better. This could be due to the fact that the non-linear controller with its higher gains, is better suitable for larger, more realistic values. Overall, both controllers showed proper results. The realistic transformed trajectory provided a logical and stable trajectory and was thus transformed properly. To run the entire simulation takes great computational power, as every time step is divided by a magnitude of $10e6$. For this reason, the total trajectory could not be plotted and interpreted. However, this small fraction is already enough to conclude that both the non-linear and the PD controller were able to control in a realistic simulated environment, as was a main goal of this integration project. Both the PD- and the non-linear controller were able to simulate the followed trajectory of [14] and perform properly in a developed realistic environment.

6.1 Future Research

In further research, the simulation, in combination with the modelled dynamics, should be run by a computationally powerful computer, able to plot the full trajectory, to show and prove that indeed the complete orbital trajectory can be followed, for both the attitude- and the orbital dynamics. To show that the computer must have large computational force, Figure 22 must be noted in the Appendix. Last, the project has not reached the stage of implementing the dynamical extension of [4] into the model. This way, the non-linear controller would be able to control trajectory with only input measurement as input. When implementing [4] into the model, and computing the entire trajectory, very interesting results could be accomplished in future research.

7 Discussion

By looking at the research goal, there are certainly some factors that the research project was not able to fulfill. Nonetheless, the results that are obtained tell us a lot about attitude control. For starters, using a script for the modelling of the dynamics of a typical satellite and the controllers has hopefully contributed to a solid foundation for future research projects about attitude control. Moreover, the collection and thorough explanation of all theoretical parts connected to attitude and orbital control could provide a lay with a solid foundation to start working on similar research projects. The controller of [14] is proven to be a powerful controller, not only in theory, but also in practice. With proper results in a realistic scenario, and a very thorough explanation of the involved mathematics behind it, this project could possibly also enhance ones appreciation and understanding of [14]. On the other hand, there are some important lacks in the research project. First of all, as discussed before, the dynamical extension of [4] is not yet implemented in the model. This could however have shown promising and interesting results and complete the research goal. More detailed, in the attitude control for the realistic simulation, the trajectory of the quaternions and the angular velocity is kept the same. However, when we look at eq. (14), one can see that the angular velocity is dependent on the tangential velocity. In this research, there is chosen to keep the angular velocity ω constant. However, when we upscale r, $\dot{r}$ and the acceleration $\dot{v}$, a change in the trajectory of the angular velocity ω and thus quaternion vector q might also occur. When this would be investigated, the trajectory could possible become even more realistic, and thus for future research this would be something that needs to be further investigated. Also, in the inequality that hold for the gains of (63) of [4] should be tested. If they hold, it could give good insight about the boundaries that the gains for the developed controller should adhere.

References

[1] Sumit Agrawal, Sanket Chirame, Anant Joshi, Ravit Anand, Ram Milan Verma, Sukanya Kudva, Keshav Janyani, Eashan Gupta, and Charit Verma. Attitude determination and control subsystem, satellite 101 wiki.

[2] NASA. State of the art of small spacecraft technology. 2008.

[3] Marcel J. Sidi. Spacecraft Dynamics and Control: A Practical Engineering Approach. Cambridge Aerospace Series. Cambridge University Press, 1997.

[4] D. A. Dirksz and J. M. A. Scherpen. On tracking control of rigid-joint robots with only position measurements. IEEE Transactions on Control Systems Technology, 21(4):1510–1513, 2013.

[5] Adam Neil Philip. Attitude sensing, actuation, and control of the brite and canx-4&5 satellites. 2009.

[6] A.M. Mohammed, Benyettou Mohammed, Y. Bentoutou, Abdelmadjid Boudjemai, Y. Hashida, and Martin Sweeting. Three-axis active control system for gravity gradient stabilised microsatellite. Acta Astronautica, 64:796–809, 04 2009.

[7] Kristian Svartveit. Attitude determination of the ncube satellite. 2003.

[8] Karl J. Astrom. Introduction to Stochastic Control Theory. 1970.

[9] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch. Rigid-body attitude control. IEEE Control Systems Magazine, 31(3):30–51, 2011.

[10] Kenji Fujimoto, Tomoya Takeuchi, and Yuki Matsumoto. On port-hamiltonian modeling and control of quaternion systems. IFAC-PapersOnLine, 48:39–44, 12 2015.

[11] R. Wisniewski and P. Kulczycki. Rotational motion control of a spacecraft. IEEE Transactions on Automatic Control, 48(4):643–646, 2003.

[12] P. Pounds, T. Hamel, and R. Mahony. Attitude control of rigid body dynamics from biased imu measurements. In 2007 46th IEEE Conference on Decision and Control, pages 4620–4625, 2007.

[13] O. Egeland and J. . Godhavn. Passivity-based adaptive attitude control of a rigid spacecraft. IEEE Transactions on Automatic Control, 39(4):842–846, 1994.

[14] K. Fujimoto and T. Nishiyama. On trajectory tracking control of port-hamiltonian systems with quaternions. In 53rd IEEE Conference on Decision and Control, pages 4820–4825, 2014.

[15] Thomas Witelski and Mark Bowen. Fast/slow Dynamical Systems, pages 201–213. Springer International Publishing, Cham, 2015.

[16] D. A. Dirksz, J. M. A. Scherpen, and R. Ortega. Interconnection and damping assignment passivity-based control for port-hamiltonian mechanical systems with only position measurements. In 2008 47th IEEE Conference on Decision and Control, pages 4957–4962, 2008.

[17] Salim Farissi, Stefano Carletta, Augusto Nascetti, and Paolo Teofilatto. Implementation and hardware-in-the-loop simulation of a magnetic detumbling and pointing control based on three-axis magnetometer data. 12 2019.

[18] Carlos Andrés Fernández Cerdas. Attitude control design for station keeping mode in the gw-sat nanosatellite. 2009.

[19] E. LEVIN. Solar radiation pressure perturbations of earth satellite orbits. AIAA Journal, 6(1):120–126, 1968.

[20] Mauricio Muñoz Arias. Force control of a class of standard mechanical systems in the port-hamiltonian framework. pages 377–382, 09 2013.

[21] Paolo Forni, Dimitri Jeltsema, and Gabriel Lopes. Port-hamiltonian formulation of rigid-body attitude control. IFAC-PapersOnLine, 48:164–169, 12 2015.

[22] Arjan van der Schaft. Port-hamiltonian systems: an introductory survey. In M. Sanz-Sole, J. Soria, J.L. Varona, and J. Verdera, editors, Proceedings of the International Congress of Mathematicians Vol. III, number suppl 2, pages 1339–1365. European Mathematical Society Publishing House (EMS Ph), 2006. null ; Conference date: 22-08-2006 Through 30-08-2006.

[23] Arjan van der Schaft and Hans Zwart. The port-hamiltonian approach to physical system modeling and control. http://www.math.rug.nl/ arjan/DownloadLectures/Benelux08.pdf, 2008.

[24] Energy-based modelling of electromechanical systems.

[25] Korn TM Korn GA. Mathematical handbook for scientists and engineers.

[26] Fundamentals of spacecraft attitude determination and control.

[27] Ewoud Vos, Jacquelien Scherpen, and Arjan Schaft. Equal distribution of satellite constellations on circular target orbits. Automatica, 50, 09 2014.

8 Appendix

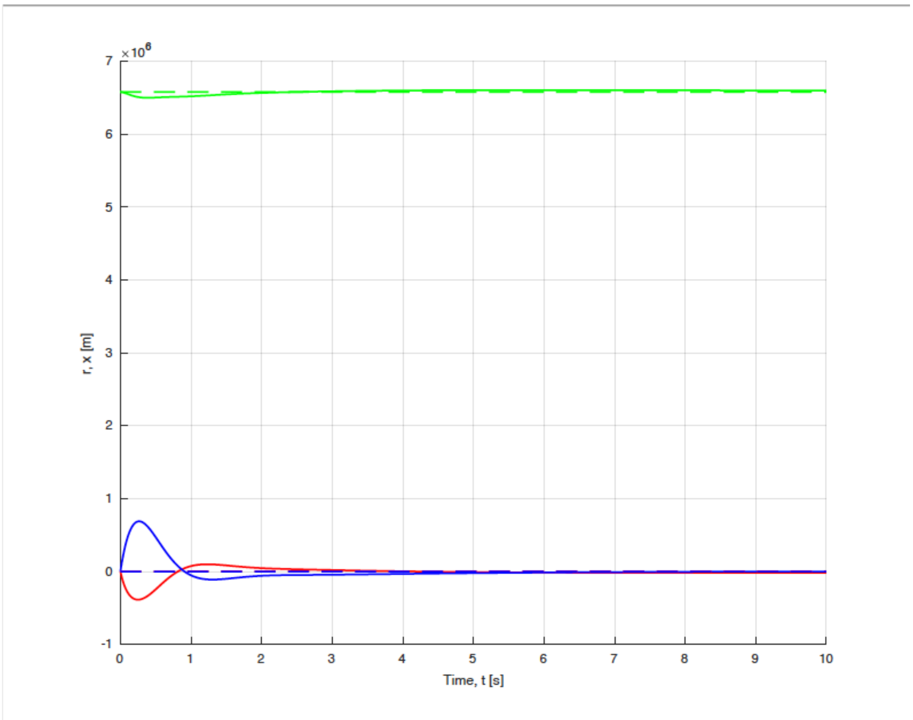

Figure 23: Plot of the r-vector in simulation 3 of Table 1 of the PD controller. Here, the blue, red and green line represent the x, y and z axis respectively. The dotted lines represent the desired trajectory, and as can be seen the controlled measured trajectory approximates the desired trajectory. It must be noted that the straight line do change in this plot, however only a very small amount, as it is only a fraction of the full desired trajectory due to the scaled down phase.

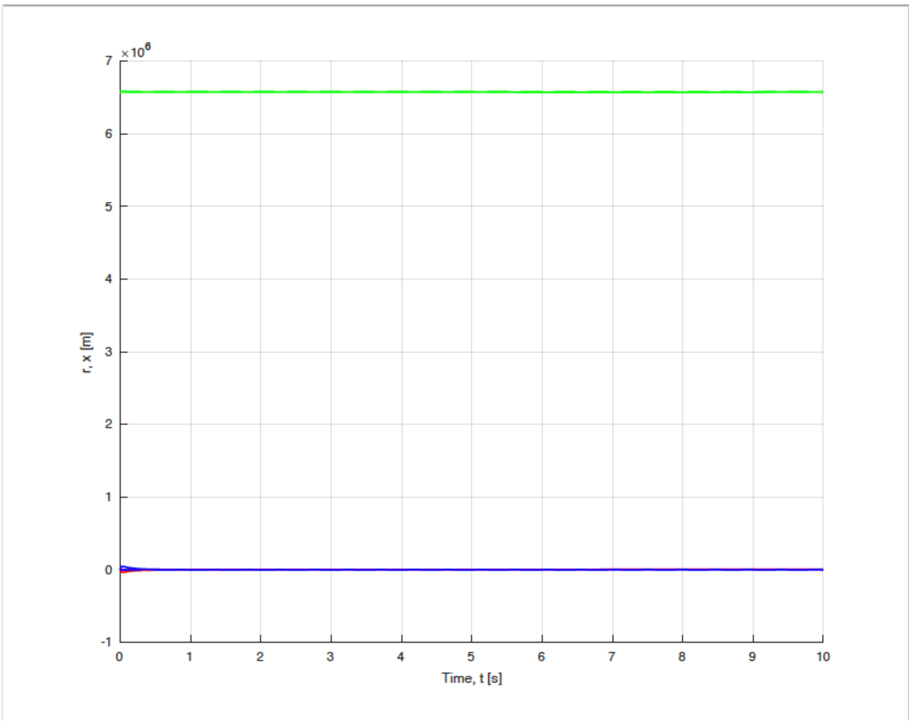

Figure 24: Plot of the r-vector in simulation 3 of Table 1 of the non-linear controller. Here, the blue, red and green line represent the x, y and z axis respectively. The dotted lines represent the desired trajectory, and as can be seen the controlled measured trajectory approximates the desired trajectory. It must be noted that the straight line do change in this plot, however only a very small amount, as it is only a fraction of the full desired trajectory due to the scaled down phase.

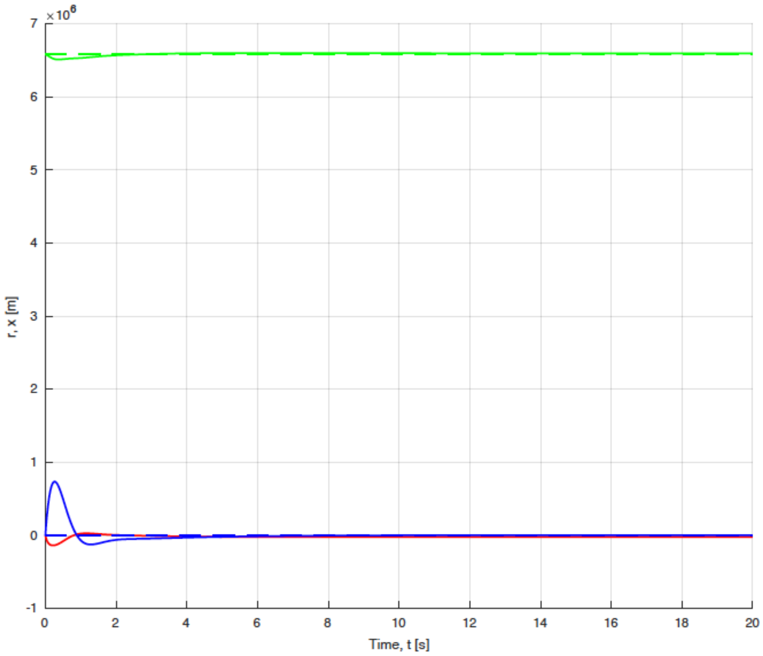

Figure 25: Plot of the r-vector in simulation 1 of Table 1 of the PD controller. Here, the blue, red and green line represent the x, y and z axis respectively. The dotted lines represent the desired trajectory, and as can be seen the controlled measured trajectory approximates the desired trajectory. It must be noted that the straight line do change in this plot, however only a very small amount, as it is only a fraction of the full desired trajectory due to the scaled down phase. The phase is scaled down with the same magnitude of R_0, making this the most realistic trajectory.

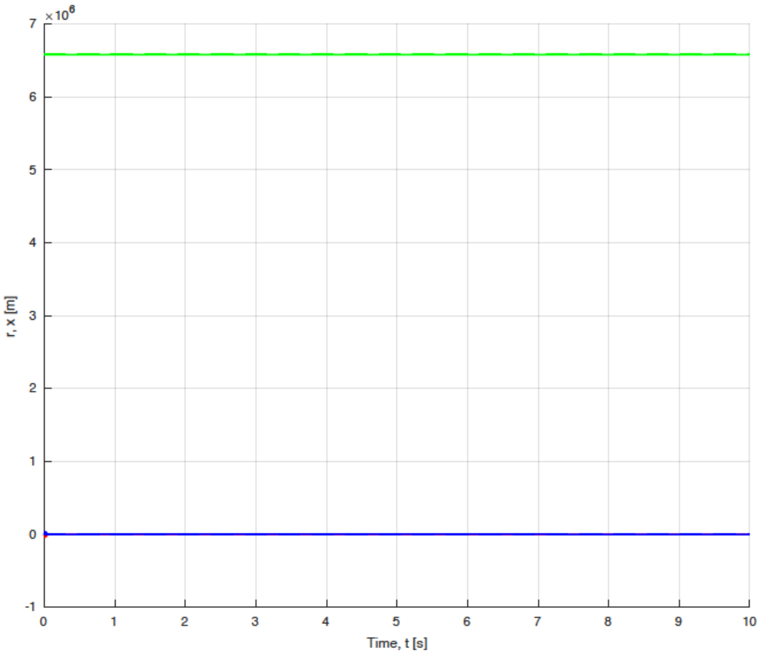

Figure 26: Plot of the r-vector in simulation 1 of Table 1 of the non-linear controller. Here, the blue, red and green line represent the x, y and z axis respectively. The dotted lines represent the desired trajectory, and as can be seen the controlled measured trajectory approximates the desired trajectory. It must be noted that the straight line do change in this plot, however only a very small amount, as it is only a fraction of the full desired trajectory due to the scaled down phase. The phase is scaled down with the same magnitude of R_0, making this the most realistic trajectory.

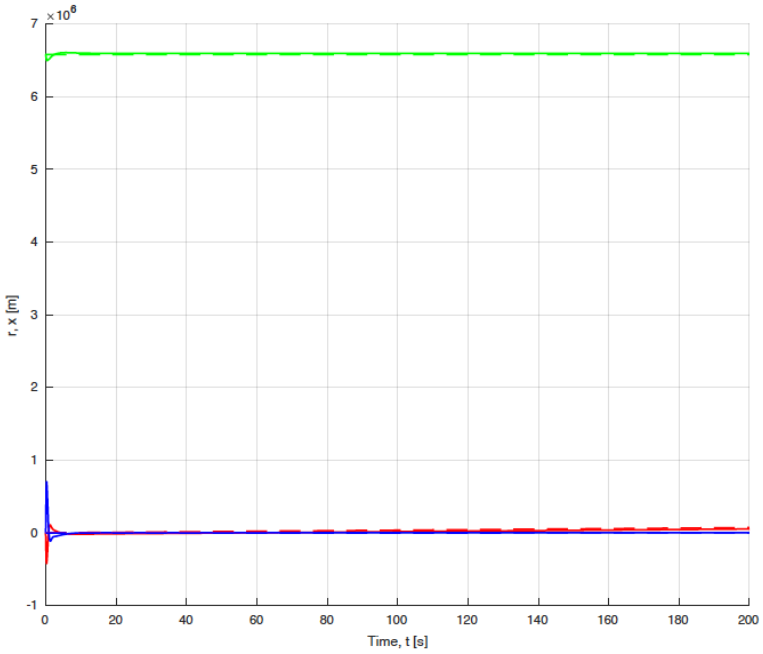

Figure 27: The plot of the r-vector in a simulation of a time span of 200 sec of the non-linear controller. Here the blue, red and green line represent the error in the x, y and z axis respectively. This shows the simulation of a time span of ten times more then simulation 1 of Table 1. It shows that even in 200 sec, the trajectory of r makes no noticeable fluctuations and so to see significant change, it has to run for a time span in the same order of R_0.