Modeling and Simulation for Attitude Control of a Small Satellite

Bjørn Andreas Kristiansen

bjorakr@stud.ntnu.no

December 18, 2018

Project thesis

Department of Engineering Cybernetics
Norwegian University of Science and Technology

Supervisor: Jan Tommy Gravdahl
Advisor: Mariusz Eivind Grøtte

Abstract

The goal of the HYPSO satellite mission at NTNU is to monitor the ocean. To do that, the camera needs to aim in the right direction. As a part of the attitude determination and control system (ADCS) group, led by Mariusz Eivind Grøtte, my focus is on attitude control. The main purpose of this project thesis is to model the dynamics of the system containing a satellite orbiting the Earth and to create a simulation environment for further investigating nonlinear control schemes.

In this project thesis, the nonlinear orbit and attitude dynamics of the satellite is modeled when actuated by reaction wheels and magnetorquers. The nonlinear satellite model is implemented in Simulink. In this project report, the results for a 6U CubeSat are shown using Proportional-Derivative controller and B-dot tumbling controller for reaction wheels and magnetorquers, respectively.

Sammendrag

Målet for HYPSO-satellitten fra NTNU er å overvåke havområder. For å klare dette må kameraet satellitten bruker peke i riktig retning. Som en del av ADCS-gruppen på MASSIVE-prosjektet, under ledelse av Mariusz Eivind Grøtte, er mitt fokus på attitude control. Hensikten med denne prosjektoppgaven er å modellere dynamikken til systemet, før deretter å lage en simulering som kan brukes for å utvikle ulineær attitude control.

I denne prosjektoppgaven modelleres den ulineære dynamikken til satellittens bane og attitude under pådrag fra reaksjonshjul og magnetorquere. Den ulineære modellen er implementert i Simulink. I denne prosjektrapporten er resultatene vises resultatene for en 6U CubeSat ved bruk av Proposjonal-Derivasjon-regulator og B-dot tumbling regulator for henholdsvis reaksjonshjulene og magnetorquerne.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 NTNU SmallSat - HYPSO

The HYPSO [1] mission is part of the space related activities at NTNU. HYPSO, or HYPer-spectral Smallsat for ocean Observation, is a interdisciplinary CubeSat project that will be part of a multi-agent marine observation system [2].The HYPSO mission is going to use a hyper spectral imager in order to help with ocean color mapping, giving the users the ability to monitor algal blooms. The hyper spectral imager is in this text referred to as the camera.

The HYPSO project is the latest in a line of interdisciplinary projects at NTNU aiming to launch a small satellite, the previous being nCube-1, nCube-2 and NUTS [3]. The difference between the previous satellite projects and the HYPSO project from a control perspective is that the HYPSO satellite will have two different kinds of actuators, both reaction wheels and magnetorquers, whereas the previous satellites were only equipped with magnetorquers. HYPSO needs the reaction wheels is order to perform slew and point in fast maneuvers.

1.2 Contributions

The HYPSO camera will be mounted along the z-axis of the body frame. Attitude control for this system is about controlling the attitude of the satellite so that the axis with the camera points towards the area of the Earth with the interesting part of the ocean. Pointing, detumbling and slewing are three of the interesting forms of attitude control for this mission.

- For pointing, the goal is to point straight down, also called nadir, or at some other point. For the attitude control system, this translates to regulating the satellite towards a fixed attitude.
- For detumbling, the goal is to stop or reduce the tumbling of the satellite.
- For slewing, the satellite would aim at an area and then slew over it while the satellite passes, and because of this maneuver have the area in view for a longer period of time. The attitude control system would need to control both the position and the given angular velocity of the satellite. A possible control scheme for the slewing maneuver is not presented in this text.

Attitude control for the HYPSO mission is described in more detail in the concept document. [2]

This project thesis models and implements controllers for pointing, using reaction wheels, and detumbling, using magnetorquers. A solution for the slewing maneuver is not implemented, but the Simulink model developed in this project thesis makes it easy to implement and test new controllers later.

1.3 Project Thesis Outline

Chapter 2 deals with the theory used in the rest of the project thesis, from the different rotations to the physical principles that are utilised.

Chapter 3 is about the method: from the mathematical modeling, to assumptions made with respect to the controllers and perturbations.

Chapter 4 presents the simulation results and discussions.
Chapter 5 is the conclusion.
Chapter 6 is a short discussion about future work.

Chapter 2

Theory

2.1 Geometry

2.1.1 Reference Frames

A reference frame is an ordered set of three orthonormal vectors that create a right handed coordinate frame. A frame is defined by an origin and three orthonormal axes. Some frames are inertial, meaning that they have no acceleration themselves, or in other words, that no forces act upon the frames. These inertial frames, in this project the ECI and the perifocal frame, are only approximations of true inertial frames, as these frames will have some forces acting upon them from their rotation around the Sun, their rotations around the center of the galaxy and so on. If the satellite was orbiting the Sun instead of the Earth, ECI would not be a good choice for an inertial frame. ECI is assumed to be inertial here.

2.1.1.1 Earth-Centered Inertial Frame (ECI)

The Earth-centered inertial (ECI) frame is an inertial frame centered in the Earth's center. The x-axis points towards the vernal equinox, which is an axis given by the moment the sun crosses the equatorial plane on it's way up, also known as the ascending node of the sun. The z-axis goes through the North Pole, and the y-axis completes the right-handed frame. As a result, the xy-plane lie at equator.

2.1.1.2 Earth-Centered, Earth Fixed Frame (ECEF)

The Earth-Centered, Earth Fixed (ECEF) frame is a frame centered in the center of the Earth, like the ECI frame. The x-axis of the ECEF frame points to 0° longitude, 0° latitude. The z-axis is the same as the ECI z-axis, while the y-axis completes the right-handed coordinate frame. However, unlike the ECI frame, ECEF follows the Earth's rotation. The ECEF frame rotates with the angular velocity about the z-axis corresponding to a full round every day, making the angular rotation around the z-axis [4, p. 16]

$$\omega_e = 7.2921 * 10^{-5} \tag{2.1}$$

2.1.1.3 Longitude, latitude and elevation

Longitude, latitude and elevation (height) is an alternative representation to the 3-dimensional Cartesian coordinate position vector. Longitude and latitude are angles representing east-west and north-south, respectively. Longitude is 0° at the prime meridian, extends +180 ° to the east and -180 ° to the west. Latitude is 0 ° at the Equator, and 90 ° at either pole. Together with height, the distance from the center, they determine where a point is on a sphere.

2.1.1.4 Perifocal Frame (PQW)

The perifocal frame (PQW) is an inertial frame centered in the Earth's center. The first axis, p, is pointing toward the periapsis, which is the shortest radius in orbit. The second axis, q, has a true anomaly of 90° from p, making it orthogonal to the first axis. The third axis, w, is the cross product of the other two axes. Parameters in the perifocal frame are called orbital parameters. These parameters are used to uniquely describes an orbit.

The orbital parameters are

- a - Semi-major axis
  - Half of the major axis of the ellipsis created by orbit plane.
- e - eccentricity
  - A parameter that describes how different the orbit is a circle. When e is 0, the orbit is circular. Eccentricity between 0 and 1 gives an elliptical orbit.
- i, Ω, ω - inclination, longitude of the ascending node and argument of periapsis
  - three angles defining the orbit plane's attitude in space relative to the equatorial plane.
- θ - true anomaly (NB: This project uses mean anomaly instead)
  - Describes the position of the orbiting body in the orbit.
- M - Mean anomaly
  - Mean anomaly describes where the orbiting body is in the orbit. The mean anomaly is, as opposed to the true anomaly, not based on the ellipse made by the orbit, but rather an imaginary circle centered in the same center as the orbit being orbited with the same orbiting period as the real orbit. An initial value for the mean anomaly, M_0, is set to decide where in the orbit the satellite begins at time zero.

The satellite's position in the orbit can be changed by changing the mean anomaly.

2.1.1.5 Orbit Frame

The orbit frame is a non-inertial frame centered in the center of mass of the satellite. The third axis is pointing towards the Earth's center, making it point in the negative direction of the satellite position vector in the ECI frame. The first axis is the velocity of the satellite in its orbit, making the second axis the cross product of the other two. The orbit frame follows the satellite as it orbits the Earth, always having one axis, the third, pointing towards the center of the Earth.

2.1.1.6 Body Frame

The body frame is a non-inertial frame centered in the center of mass of the satellite, like the orbit frame. Unlike the orbit frame, however, its axes follow the satellite as it rotates.

Attitude control relates to controlling the attitude of the body frame of the satellite with respect to the Earth. The orbit frame constantly points towards the Earth, making attitude control a problem of controlling the attitude of the body frame relative to the orbit frame.

2.1.2 Attitude Representation and Rotation

Vectors in a three dimensional Euclidian space, like the position or velocity, can be represented in any of the previously defined frames, except the perifocal frame. To change the frame the vector is represented in, the vector needs to be rotated[1] from the frame it is currently represented in into the new frame. A rotation like this is shown in equation 2.2, where R_b^a is a rotation from frame b to frame a.

$$\mathbf{v}^a = \mathbf{R}_b^a \mathbf{v}^b \tag{2.2}$$

The rotation represented by R is a 3x3 matrix that has some properties [5, p. 219-221]. Among these are that it is part of SO(3), meaning that

$$\mathrm{SO(3)} = \{\mathbf{R}|\mathbf{R} \in R^{3x3}, \mathbf{R}^\mathrm{T}\mathbf{R} = \mathbf{I} \text{ and } \det \mathbf{R} = 1\}$$

This definition means that the rotation is orthogonal. That the matrix is orthogonal implies that the inverse of the matrix would be equivalent to the transpose, or

$$\mathbf{R}_a^b = (\mathbf{R}_b^a)^{-1} = (\mathbf{R}_b^a)^T$$

Another property is that the product of rotations creates a new rotation,

$$\mathbf{R}_a^c = \mathbf{R}_b^c \mathbf{R}_a^b$$

[1]The language in this text does not separate cases where the rotation matrix acts like a rotation matrix and a coordinate transformation matrix. If it did, equation 2.2 would have been denoted as a coordinate transformation matrix. For a more thorough discussion on the differences between the two interpretations of the rotation matrix, see Egeland and Gravdahl [5, p. 220]

Using this formula, rotations between all frames can be constructed, should the rotation between any combination of the frames be known.

An attitude is the way a frame is rotated relative to another. This rotation R_b^a can be constructed in different ways, depending on how the attitude is represented.

The different attitude representations have different advantages and drawbacks. To represent any complete rotation in three dimensional space, three parameters are required. However, any parameterisation using less than four parameters would include some singularity.

2.1.2.1 Rotation matrix

The rotation matrix is also known as a direction cosine matrix (DCM). It is overrepresented, as it uses 9 parameters to represent an attitude. The advantage of this is that itt does not contain any singularities. Using a and b as two sets of orthogonal unit vectors, the rotation matrix is constructed by [5, p. 219-220]

$$\mathbf{R}_b^a = \{\vec{a} \cdot \vec{b}\} \tag{2.3}$$

or by

$$\mathbf{R}_b^a = (\ \mathbf{b}_1^a\ \mathbf{b}_2^a\ \mathbf{b}_3^a\ ) \tag{2.4}$$

2.1.2.2 Euler angles

The rotation using Euler angles consists of the product of three rotation matrices, multiplied together. Each rotation matrix consists of a simple rotation, meaning a rotation by an angle about one of the axes. The simple rotations are shown in the matrices in the equations below, where the subscript i in $\mathbf{R}_i$ indicates which axis the rotation is about.

$$\mathbf{R}_1(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi & cos\phi \end{pmatrix} \tag{2.5}$$

$$\mathbf{R}_2(\theta) = \begin{pmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{pmatrix} \tag{2.6}$$

$$\mathbf{R}_3(\psi) = \begin{pmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.7}$$

There are twelve different ways to use Euler angles to show an orientation, based on the order of axes one rotates about. One example is

$$\mathbf{R}_b^a = \mathbf{R}_3(\psi)\mathbf{R}_2(\theta)\mathbf{R}_1(\phi) \tag{2.8}$$

Even if it uses three 3x3 matrices to represent a full rotation, the parameters needed to represent the rotation with Euler angles is just the angles used for each axis. Thus there are only three parameters used. and there is a singularity, known as gimbal lock. This singularity is always in the second of the three angles, making the order of the simple rotations an important choice.

2.1.2.3 Axis-angle

The axis-angle parameterization has four parameters: the three vector coordinates of the vector it is rotating about and the angle it is rotating with. Axis-angle is based on Euler's theorem of rotation [4, p. 21], which states that any rotation in three-dimensional space can be represented by a single axis and an angle. With this representation one only needs the axis and the angle, making it four parameter, to represent a rotation.

2.1.2.4 Unit quaternions

The quaternion has one scalar part and one vector part, defined by the angle θ and vector k as [5, p. 233]

$$q = \begin{bmatrix} \eta \\ \epsilon \end{bmatrix} = \begin{bmatrix} cos(\theta/2) \\ ksin(\theta/2) \end{bmatrix} \tag{2.9}$$

The unit quaternions have length 1, meaning that

$$q^{\intercal} q = 1 \Rightarrow \eta^2 + \epsilon^{\intercal}\epsilon = 1 \tag{2.10}$$

The rotation matrix defined by the unit quaternion is given by [4, p. 28]

$$\mathbf{R}_{\eta,\epsilon} = \mathbf{I}_{3x3} + 2\eta\mathrm{S}(\epsilon) + 2\mathrm{S}^2(\epsilon) \tag{2.11}$$

where S() is the skew symmetric matrix.

The quaternion representation does not have a singularity like the Euler angles does, as it has four parameters.

The main advantage of Euler angles has over quaternions is that they are easier to understand. Roll-pitch-yaw is used for aircrafts and ships, and if the vessel is stable enough to stay away from the singularity most of the time, using them will not be a problem. However, in the satellite application, in particular for the HYPSO mission, the satellite could be tumbling. During this tumble, it might have any attitude, including the attitude specific attitude that would make a parameterisation with three parameters singular. For this reason, a parameterisation that does not contain any singularities is preferable for the satellite.

2.1.2.5 Rotating Reference Frame

A reference frame can rotate relative to another. For example, the orbit frame will rotate relative to the ECI frame as the satellite orbits the Earth, and the body frame will rotate relative to the orbit frame when the attitude changes.

Differentiating a vector u, where b is a rotating reference frame with respect to a

gives the transport theorem [5, p. 243]

$$\frac{d}{dt}^a u = \frac{d}{dt}^b u + \omega_{ab} \times u \tag{2.12}$$

or, on coordinate vector form [5, p. 242]

$$\dot{\mathbf{u}}^a = \mathbf{R}_b^a[\dot{\mathbf{u}}^b + (\omega_{\mathbf{ab}}^{\mathbf{b}})^\times \mathbf{u}^b] \tag{2.13}$$

2.1.2.6 Kinematic Differential Equation

The kinematic differential equation shows the change in attitude in time. The kinematic equation for quaternions is given by [4, p. 30-31]

$$\dot{q} = \begin{bmatrix} \dot{\eta} \\ \dot{\epsilon} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\epsilon^\top \\ \eta \mathbf{I}_{3x3} + \mathrm{S}(\epsilon) \end{bmatrix} \omega_{b/a}^b = \frac{1}{2} \begin{bmatrix} -\epsilon^\top \\ \eta \mathbf{I}_{3x3} + \mathrm{S}(\epsilon) \end{bmatrix} \omega_{b/a}^b = \mathbf{T}(\mathbf{q}) \omega_{b/a}^b \tag{2.14}$$

2.2 Physics

2.2.1 Newton's Second Law

Newton's second law is defined as [4, p. 46]

$$\sum F = m \frac{d}{dt}^i v \tag{2.15}$$

when the mass of the body is constant. F is force, m is the mass of the body, $\frac{d}{dt}^i$ is the time derivative in an inertial frame and v is the velocity.

2.2.2 Newton's Law of Universal Gravitation

Newton's law of universal gravitation states that every particle attracts every other particle with a force equivalent to [6, p. 423]

$$F = \frac{G m_1 m_2 r}{||r||^3} \tag{2.16}$$

where G is the gravitational constant, m_1 and m_2 are the masses of the objects, and r is the vectorial distance between the two centers of masses.

The Earth has substantially larger mass than the satellite, so the force acting on the Earth because of the satellite orbiting it is ignored in this project.

Dividing equation 2.16 by the satellite's mass on both sides, gives an equation for specific force.

$$a = \frac{F}{m} = \frac{\mu r}{||r||^3} \tag{2.17}$$

where μ is the standard gravitational parameter for the Earth, given as the product of G and the Earth's mass.

2.2.3 Newton's Second Law for Rotational Motion

Also called Euler's second law of motion, or Euler's second axiom. It states that [4, p. 46]

$$\frac{d}{dt}{}^{i}H = \tau^{b} \tag{2.18}$$

where H is the angular momentum and τ is the torque, both about a point. Angular momentum is defined as

$$H = \mathbf{I}\omega \tag{2.19}$$

where $\mathbf{I}$ is the inertia matrix and ω is the angular velocity.

The torque from a force is defined as the arm crossed with the force,

$$\tau = \mathbf{r} \times \mathbf{F} \tag{2.20}$$

2.2.4 Rotational Vector Properties

Let a, b, c and d be different frames. Then the angular velocity omega ad is given as [5, p. 242]

$$\omega_{ad} = \omega_{ab} + \omega_{bc} + \omega_{cd} \tag{2.21}$$

The total angular momentum of a system is the vector sum of angular momentum from all parts of the system,

$$H_{total} = \sum H \tag{2.22}$$

2.2.5 Rotational Kinetic Energy

Rotational kinetic energy is given by [6, p. 308]

$$E_k = \frac{1}{2}\omega^\intercal \mathbf{I}\omega \tag{2.23}$$

Chapter 3

Method

3.1 NTNU SmallSat - HYPSO

The satellite used in the HYPSO mission will be a 6U CubeSat, delivered by NanoAvionics [7]. The 6U dimension means that the satellite will be 20x10x30, all in centimeters. It will contain solar panels, a case, wires, and more. In order to use this information as the basis for simulation in this project, only the parameters of the satellite that are involved in the mechanics of the system, and the control, are required. Those parameters are:

- The size of the satellite
- The inertia matrix of the satellite
- The satellite's mass
- The actuators
  - Reaction wheels
  - Magnetorquers

The reaction wheels and the magnetorquers are the satellite's actuators. Information about these were made available by NanoAvionics. The outer dimension, 20x10x30 in cm, is fixed. The inertia matrix and the satellite's mass used in this project, however, comes from a computer simulation done on a preliminary model of the satellite. The values are subject to change before launch.

3.1.1 Transformation between Orbital Parameters (PQW) and ECI Cartesian Coordinates

The mission has a desired orbit, with defined orbital parameters. The perifocal frame is defined in orbital parameters. For more about the orbital parameters or the perifocal frame, see section 2.1.1.4.

The simulation of the orbit is in the ECI frame. To get the initial ECI position and velocity vectors from the orbital parameters, there needs to be a way to transform the orbital parameters to the ECI frame. A conversion between the frames released by the Colorado Center for Astrodynamics Research at University of Colorado Boulder [8] (Warning: the link leads to a direct download of a .doc file) has been implemented in a Matlab function called kep2ECI. A similar file [9] from the same source describes how to go from ECI to orbital parameter, but this functionality has not been implemented here.

3.2 Mathematical Model

For the models developed, as well as for the simulation, the Earth will be modeled as a sphere, as it is easier to use in models. This is a simplification, as the Earth is not completely spherical.

The model of the Earth is defined by it's standard gravitational parameter, see section 2.2.2, and its radius, here taken to be 6 371 km. If a more accurate model of the Earth was used rather than a sphere, like an ellipsoid, the radius would change depending on the location.

3.2.1 Orbital Mechanics

Using Newton's second law, equation 2.15, and the fact that

$$\dot{r} = v \tag{3.1}$$

the differential equation describing the motion of the satellite orbiting the Earth is given as

$$\frac{d}{dt}^{i} v = \mu \frac{r}{\|r\|^3} + \frac{f}{m_{sat}} \tag{3.2}$$

given that the satellite's mass is constant, where all values are given in the ECI frame, and r is the position vector of the satellite center of mass. f denotes the other forces acting on the satellite. For these other forces, see section 3.3.

3.2.2 Attitude Kinetics

Attitude kinetics is how torques change the attitude of the satellite.

The satellite's total angular momentum is given by the sum of the angular momentum from the satellite and the angular momentum of the reaction wheels (equation 2.22),

$$H = H_{total} = H_{satellite} + H_{RW} = \mathbf{I}_{sat}\omega_{sat} + \mathbf{I}_{RW}\omega_{RW} \tag{3.3}$$

Using equation 2.18, the transport theorem given in equation 2.12 and, writing ω_{sat} as $\omega^b_{b/i}$ and ω_{RW} as ω^b_w, gives

$$\frac{d}{dt}^{i} H = \frac{d}{dt}^{i} (\mathbf{I}_{sat}\omega^b_{b/i} + \mathbf{I}_w\omega_w) = \mathbf{I}_{sat}\frac{d}{dt}^{b} \omega^b_{b/i} + (\omega^b_{b/i})^{\times}(\mathbf{I}_{sat}\omega^b_{b/i} + A_w H_w) + \frac{d}{dt}^{b} H_w = \tau \tag{3.4}$$

where it has been used that the time derivative of the inertial matrices $\mathbf{I}_{sat}$ and I_w are constant in the body frame. A_w is the matrix mapping the vectors from each of the reaction wheels to the axes of the body frame. Using the negative of the body frame derivative of H_w as the command torque for the reaction wheels [10], the final differential equation is

$$\mathbf{I}_{sat} \frac{d}{dt}^{b} \omega^b_{b/i} = -(\omega^b_{b/i})^{\times}(\mathbf{I}_{sat}\omega^b_{b/i} + A_w H_w) + \tau_{RWc} + \tau_{MTQ} + \tau_d \tag{3.5}$$

$$\tau_{RWc} = -\frac{d}{dt}^{b} H_w = -\mathbf{I}_w \frac{d}{dt}^{b} \omega_w \tag{3.6}$$

where τ_{RWc} is the torque from the reaction wheels, τ_{MTQ} is the torque from the magnetorquers and τ_d is the torque caused by perturbations.

3.2.3 Attitude Kinematics

Kinematics is the description of the motion of objects, meaning the geometrical aspect of motion. In this context, attitude kinematics describes how the attitude, described in quaternions, changes over time. The assumption is perfect knowledge of the system for this project, so there is no filtering involved. Therefore, the differential equation is given by equation 2.14:

$$\dot{\mathbf{q}} = \mathbf{T}(q)\omega^b_{b/o} \tag{3.7}$$

A comment about how this could have been improved can be found in section 6.

3.2.4 Angular Velocities

$\omega^b_{b/o}$ is the angular velocity of the satellite body frame relative to the orbit frame. Using the property of angular velocity described in equation 2.21, $\omega^b_{b/o}$ is found by subtracting the angular velocity of satellite in orbit from the angular velocity of the body frame relative to the inertial frame,

$$\omega^b_{b/o} = \omega^b_{b/i} - \mathbf{R}^b_o \omega^o_{o/i} \tag{3.8}$$

where $\omega^o_{o/i}$ is given by [11]

$$\omega^o_{o/i} = \mathbf{R}^o_i \frac{\mathbf{S}(\mathbf{r})\mathbf{v}}{\mathbf{r}^\mathsf{T}\mathbf{r}} \tag{3.9}$$

3.2.5 Magnetic Field Model

The International Geomagnetic Reference Field [12] is a mathematical model of the Earth's magnetic field. Coefficients for the formula are set released for a particular year. In this case the most recent model, 2015, was chosen.

In this project, the code from [13] is used, with some changes to fit the input parameters and to make the reading of the coefficient table faster. The functions are implemented in Simulink, see section 3.5.2.

3.3 Perturbations

Perturbations are forces and torques that act on the satellite as it propagates its orbit around the Earth. This does not include gravity, which is considered the main force acting on the orbiting body. The perturbations that are discussed here are

- Gravitational gradient
- Magnetic disturbance
- Atmospheric drag
- Solar radiation pressure

While some of the main perturbations are discussed, the discussion is not exhaustive. Some of the perturbations not discussed here are the oblateness of the Earth and the effect of tides. These perturbations could be part of future work.

3.3.1 Gravitational Gradient

The gravitational gradient comes from that the object orbiting the Earth, the satellite, is not a point mass. The gravitational gradient torque is given by [14, p. 237]

$$g_c = 3(\frac{\mu}{R_c^3})\mathbf{c}_3^{\times}\mathbf{I}\mathbf{c}_3 \tag{3.10}$$

where $\mathbf{c}_3$ is the third column of the rotation matrix from the orbit frame to the body frame.

3.3.2 Magnetic Disturbance

The satellite sets up a dipole, both from the the electronics and from the magnetorquers when they are in use. The dipole from the electronics is called the residual magnetic dipole [15], and can be represented as

$$\tau = D\mathbf{B}^b \tag{3.11}$$

In this project, D is the product of a constant and area.

3.3.3 Atmospheric drag

Atmospheric drag is a force that works in the opposite direction of the motion of the body has through the fluid it is moving through. If the fluid is air, this drag force is often referred to as air resistance.

3.3.3.1 Force

The force is given by

$$F = \frac{1}{2}\rho A V^2 C_D \tag{3.12}$$

where ρ is the air density and A is the area of the largest surface of the satellite, which for the HYPSO satellite is the y-face at 0.06 m^2. V is the velocity of the satellite relative to the fluid and C_D is the drag coefficient, which is set to 2.1. In the simulations, $0.5 * 3.5481 * 10^{-12}$ is used for ρ.

3.3.3.2 Torque

The formula for a torque made by a force is given by equation 2.20. The arm is taken as the largest possible value from the center of gravity to the center of pressure, which span the length of the largest surface, one on each side. The problem with this is that the points are set at these points no matter how the satellite rotates. This, in addition to that the center of gravity and center of pressure are not supposed to be at these fixed points, is not correct, and fixing this is part of future work. What this simplification of the model means to the results is that there will be a perturbing torque that is as large as the largest value the torque can have, but the direction of the torque does not always physically make sense.

3.3.4 Solar radiation pressure

Solar radiation pressure is pressure on the satellite due to radiation from the sun.

3.3.4.1 Force

The force is given by

$$F = \frac{F_{SRP}}{c}\frac{A}{2}(1+\eta)cos(\alpha)\begin{bmatrix}0\\1\\0\end{bmatrix} \tag{3.13}$$

where

- F_{SRP} is the solar constant, with value 1367 W/m^2.
- c is the speed of light, $3 * 10^8$ m/s.
- A is the area of the largest face of the satellite, here 0.06 m^2.
- η is the emittance, set to 0.2.

- α is the angle defining the angle of the sunlight, set to 0° for maximum force. With the force hitting the largest face of the satellite, the vector at the end becomes [0;1;0].

3.3.4.2 Torque

Like the torque from the atmospheric drag, section 3.3.3.2, the torque from the solar radiation pressure is based on the maximum arm possible. This results in a torque that, while it is the maximum possible, only makes sense if the sun follows the satellite around as it rotates about its own axes.

Making a model that takes the sun's position relative to the satellite into account and uses this to calculate the solar radiation pressure force and torque is part of future work, see section 6.

3.4 Control

In order to perform the attitude control objectives, the satellite has two kinds of actuators. These actuators are called reaction wheels and magnetorquers.

3.4.1 Reaction Wheels

A reaction wheel is a flywheel, meaning a device that stores rotational energy. On the HYPSO satellite, four reaction wheels will be used. A matrix, A_w, is used to map the response of the wheels to the axis they correspond to. In the simulations in this project thesis, only one wheel on each axis has been used, making it three in total. The model can be expanded to include more wheels by expanding the A_w matrix by one or more columns.

Reaction wheels provides a more accurate attitude control option for pointing than magnetorquers would. The aim is to use the reaction wheels for both pointing and slewing maneuver. Only the pointing task is explored here.

One controller will in this project thesis be used with the reaction wheels: The Proportional-Derivative controller. The PD controller is given by

$$\tau = -\mathbf{K_p}\epsilon_{error} + \mathbf{K_d}\omega^b_{b/o} \tag{3.14}$$

where ϵ_{error} is the error in the last three elements of the quaternion, and $\omega^b_{b/o}$ is the angular velocity of the body frame relative to the orbit frame. The gains $\mathbf{K_p}$ and $\mathbf{K_d}$ are based on the satellite's inertia matrix, since the amount of torque needed to control the satellite is dependant by the satellite's inertia.

3.4.2 Magnetorquers

A magnetorquer is made up of magnetic coils. They can be used to control the attitude of a satellite works by creating a magnetic dipole by running a current through its coils, which in turn interacts with the Earth's magnetic field. This resulting torque is given by

$$\tau^b = m^b \times \mathbf{B}^b \tag{3.15}$$

where the components of m^b in all three axes are given by the product of the current that flows through the coil, the number of windings in the coil and the surface area of the coil, or

$$m^b = \begin{bmatrix} i_x N_x A_x \\ i_y N_y A_y \\ i_z N_z A_z \end{bmatrix} \tag{3.16}$$

In this project thesis, magnetorquers are used for detumbling.

To control the amount of torque applied from the magnetoruqers for detumbling, the B-dot algorithm is used. In the B-dot algorithm, m^b is set to

$$m^b = -k\dot{\mathbf{B}}^b \tag{3.17}$$

where k is a positive constant.

Where $\mathbf{B}^b$ can be read from the IGRF model, the derivative, $\dot{\mathbf{B}}^b$, cannot be read from a table. A simplification is made in assuming that the time derivative of the magnetic field in the inertial frame is zero [16], making the derivative of the magnetic field

$$\dot{\mathbf{B}}^b = -\omega^b_{b/i} \times \mathbf{B}^b \tag{3.18}$$

3.5 Simulation setup

The tool used for simulating the system is Simulink. To set up the variables needed in simulink, and to call the simulation, a script is made in MATLAB. Calling this script, main.m, starts the simulation.

There are additional functions that aid in the initialisation of the simulation, as well as plotting when the simulation is done.

3.5.1 Top Level

The system consists of three blocks: The control block, the IGRF magnetic field block and the satellite dynamics block. The IGRF block produces the magnetic field in the body frame, which is needed for both the magnetorquer and the magnetic disturbance. The control block produces the torque input to the spacecraft, which is modelled in the spacecraft dynamics block. There are two blocks that sends variables to workspace, namely the position vector describing the position of the satellite in the ECI (r, with the variable name "position"), and the attitude of the spacecraft (q, with the variable name "attitude_quaternions"). These variables are used to visualise the results after the simulation ends.

In this project, perfect knowledge about the attitude is assumed. For that reason, there is no attitude determination system, ADS, before, meaning to the left, of the (attitude) control system, ACS.

3.5.2 IGRF Block

The job of the IGRF block is to produce the magnetic field in the body frame. The block uses functions in "Field in ECEF" and "Transform to ECI" that were taken from [13].

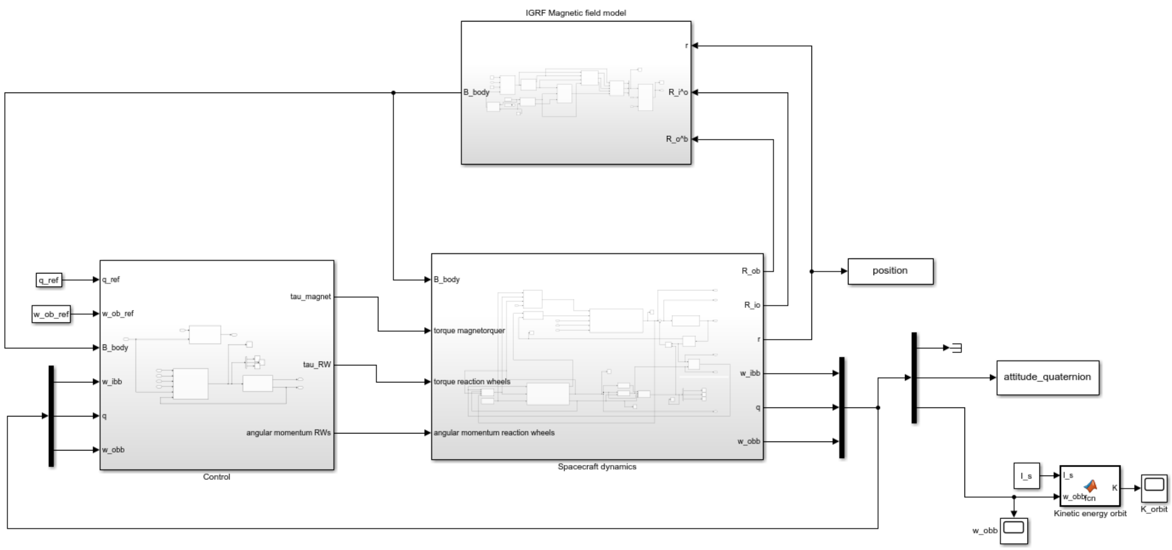

Figure 3.1: The top level of the Simulink model

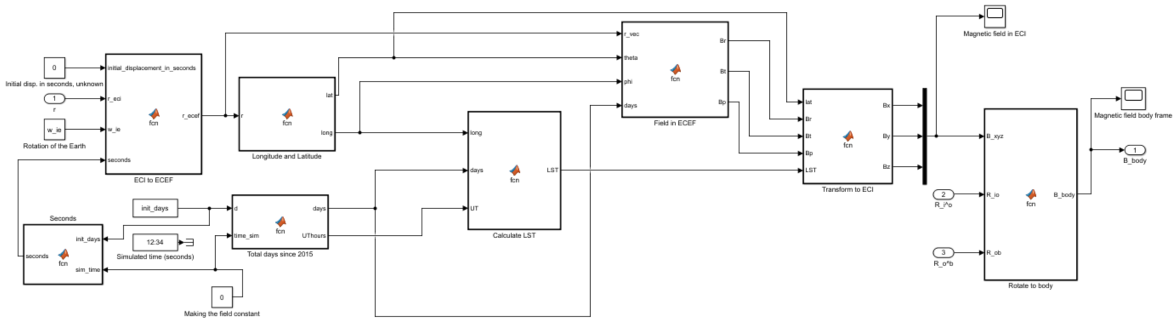

Figure 3.2: Block diagram of the IGRF model.

The only changes made in the functions are that the "Field in ECEF" function now has the IGRF values for 2015 directly in a matrix, instead of reading a file each time.

The block uses the number of days since the data is from, here Jan 1, 2015, to determine the current orientation of the field. An initial number of days from that point can be set in main.m, which is the number of days from this date the simulation starts. This info, combined with how much time has been spent simulating, is used to calculate local sidereal time, LST.

The inputs to the IGRF subsystem are r, which is the position of the satellite's mass center in the ECI frame, $\mathbf{R}_i^o$, which is the matrix describing the rotation from ECI to the orbit frame, and $\mathbf{R}_o^b$, which is the rotation from the orbit frame to the body frame. These matrices are used in the end to rotate the field from ECI to the body frame, which is where it will be used.

The ECI position vector is rotated to the ECEF frame in order for it to be used to calculate longitude and latitude. The formulas for longitude and latitude assume that the Earth is a sphere.

The initial displacement in seconds block is a block that says how far away from being aligned the ECI and ECEF frames were on the day the IGRF data are from. This bias is not known, and is part of future work. The value is set to zero.

The simulated time block keeps track of the time that has passed since the simulation started, so the rotation from ECI to ECEF does not have to be constant for the duration of the simulation. In the simulations used in the results, however, they are set as constant by replacing the simulated time by a zero, even if the simulated time block works as intended.

3.5.3 Control Block

The control block, or the attitude control system (ACS), consists of three blocks: The reaction wheel controller, the reaction wheel dynamics and the magnetorquers. The reaction wheel controller is set up to easily accommodate other control types than the implemented PD controller. The reaction wheel dynamics are idealised, meaning that their response follows the input exactly [10, p. 16]. Because of the matrix inside the

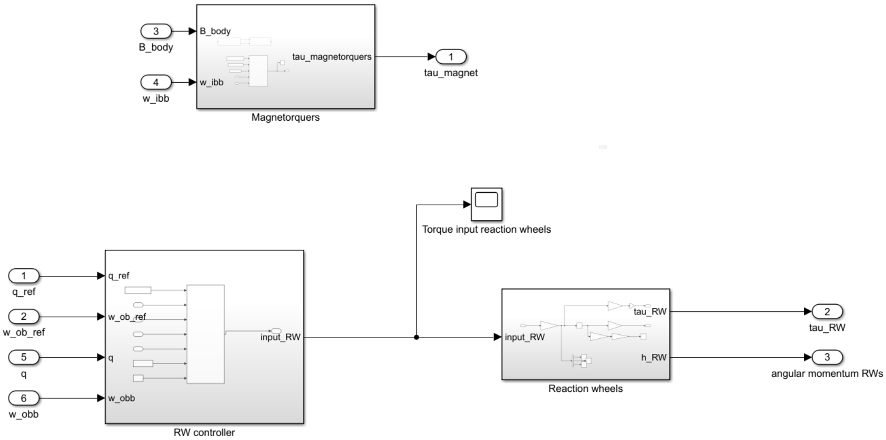

Figure 3.3: The control block in Simulink, with blocks for control for the magnetorquers, the reaction wheels, as well as one for the model of the reaction wheels.

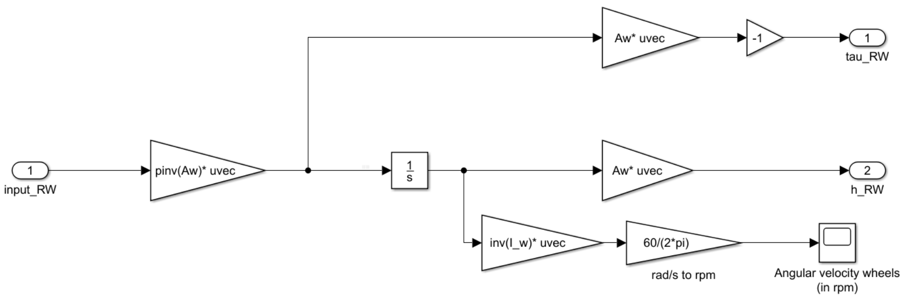

Figure 3.4: Inside the reaction wheels block, showing the reaction wheel dynamics. Idealised.

reaction wheel dynamics block, however, no change needs to be made in the Simulink block diagram in order for the system to work if another reaction wheel is added. With the reaction wheels, both the torque produced and the angular momentum are used in calculations inside the spacecraft dynamics block (see section 3.2.2), while the magnetorquers' impact on the spacecraft dynamics are only represented by the torque they produce.

3.5.4 Satellite Dynamics

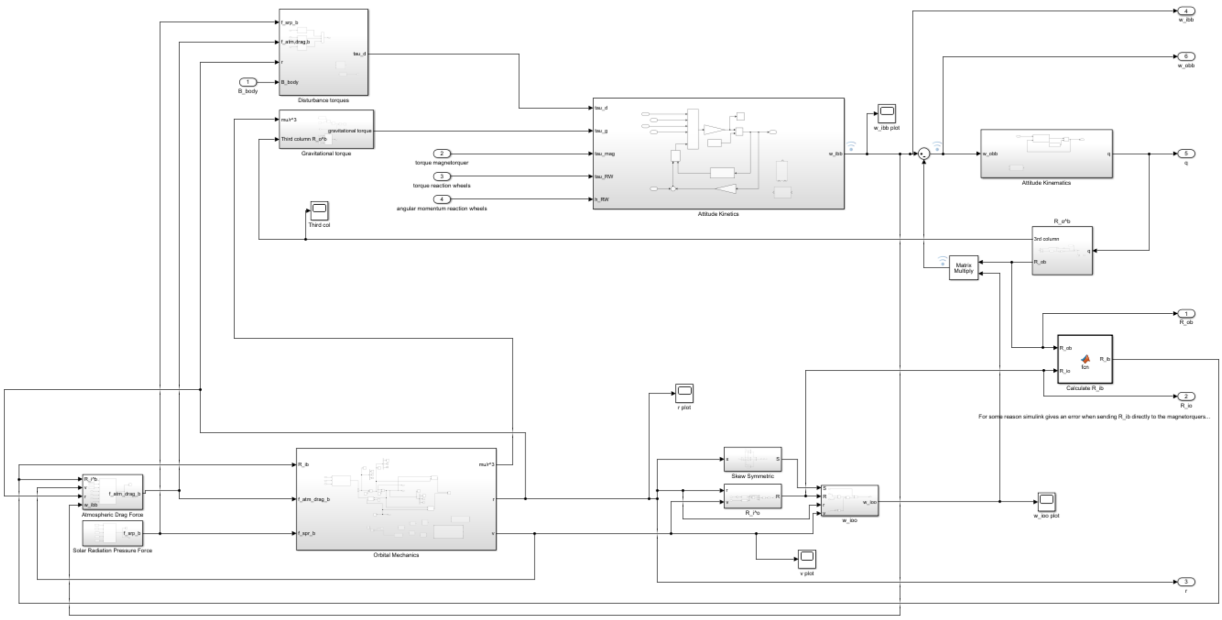

Figure 3.5: The satellite dynamics block.

The Satellite Dynamics block is by far the most complex submodule. It models all the movement of the satellite about the Earth and about its own center. The main blocks, namely the orbital mechanics, the attitude kinetics and the attitude kinematics block, are described in sections 3.2.1, 3.2.2 and 3.2.3. The remaining blocks are there to supply these main blocks with the information they need. An example is how the angular velocity of the orbit is calculated from the output of the Orbital Mechanics block, before the result is rotated and subtracted from the output of the Attitude Kinetics block and used as input to the Attitude Kinematics block.

The last points of interest in the Satellite Dynamics block are the perturbation blocks. The forces from the solar radiation pressure and the atmospheric drag pressure (section 3.3.4.1 and 3.3.3.1) are calculated to the left of and are fed into the Orbital Mechanics subsystem as well as the perturbation torque subsystem. While also a perturbation, gravitational gradient torque has its own block.

3.5.5 Orbital Mechanics

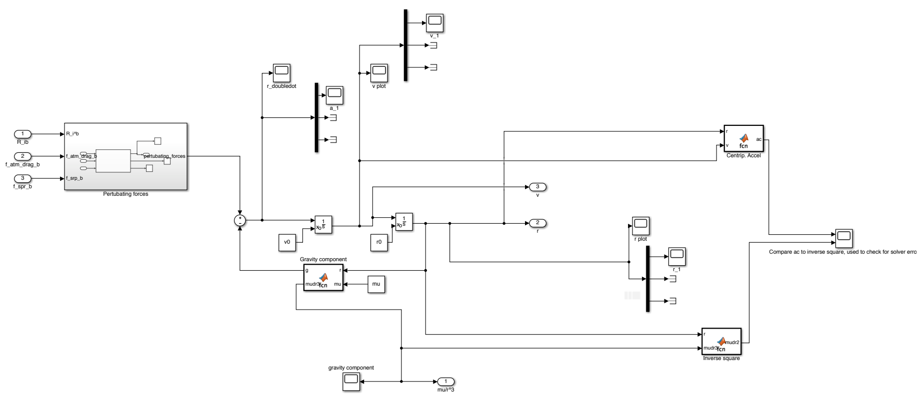

Figure 3.6: Inside the orbital mechanics block. Its main function is a double integration. The comparison on the right of the figure was used to detect an error created by having too high relative tolerance when a variable step solver was used.

The Orbital Mechanics block implements the functions in section 3.2.1. It consists of a block rotating and summing the perturbing forces, a function block that calculates the gravity component given by Newton's gravitational law, section 2.2.2, and two integrators. The initial conditions on the integrators are set in the Matlab script main.m, on the basis of the mission's orbital parameters. There is some extra functionality that compares the centripetal acceleration to the inverse square law in order to determine if the system suffers numerical errors. This functionality is not in use in any other part of the system.

3.5.6 Attitude Kinetics

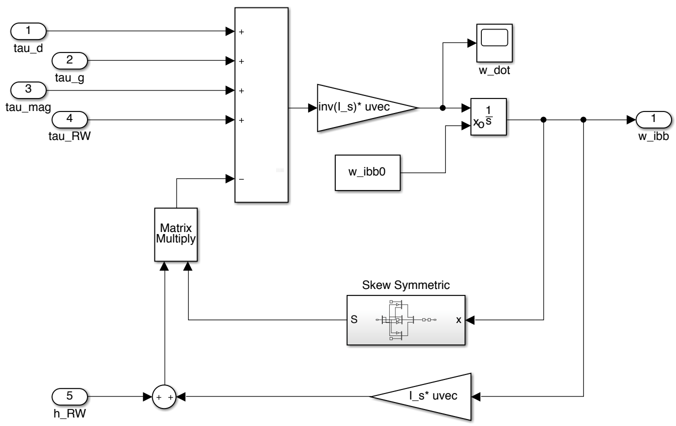

Figure 3.7: The attitude kintetics block.

The Attitude Kinetics block contains the implementation of the functions in section 3.2.2. "tau_g" and "tau_d" are the perturbing torques from gravitational gradient and other perturbations, respectively.

3.5.7 Attitude Kinematics

The Attitude Kinematics block contains the implementation of the functions in section 3.2.3. The function is inside a MATLAB function.

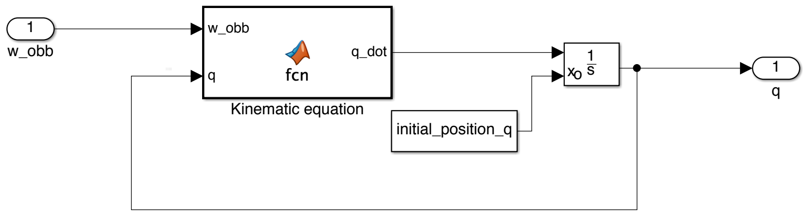

Figure 3.8: The attitude kinematics block.

3.5.8 Perturbation Torques

Perturbation Torques contains all the torques from section 3.3, apart from the gravity gradient torque, which has its own subsystem. The functions are for each of the disturbances implemented in MATLAB function blocks, and the results of the individual disturbances are summed at the output of the Perturbation Torques block.

3.5.9 Visualisation

There are five different ways that will be used to visualise the results from the simulations. The first option is inside Simulink itself, while the other four requires a specific timeseries to be exported from Simulink to workspace in MATLAB after the simulation is done for visualisation. There are boolean variables in main.m available to enable or disable either of the last four options.

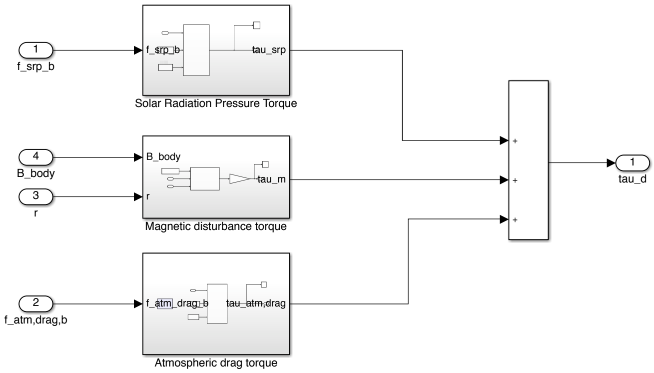

Figure 3.9: Perturbing torques block, input to Attitude kintetics.

3.5.9.1 Simulink Scopes

Scopes in Simulink will be used for most of the results. Scopes give the user the possibility to check the simulation in real time, as well as easy access to stop the simulation when enough time has passed. This makes scopes far more flexible than the other four options, which will run through a full timeserie once they have started.

3.5.9.2 Plot Orbit - Short Tail

This plot shows the satellite as it moves around the Earth. The Earth is fixed in the plot. When the satellite moves, it leaves a short tail marking where it has recently been.

The plot can be enabled by setting the variable "bool_plot_orbit" to "true" in main.m.

3.5.9.3 Plot Orbit - Complete

This plot is almost identical to the one in section 3.5.9.2, but instead of being custom made it uses comet3, a MATLAB function. This causes the tail to show the complete orbit the satellite has taken from start to finish of the simulation.

The plot can be enabled by setting the variable "bool_plot_complete_orbit" to "true" in main.m.

3.5.9.4 Plot Attitude as Euler Angles

This plot shows the attitude from the simulation plotted as Euler angles, given in angular degrees. The purpose of this is to show plots that are easier to understand than simply showing the attitude given in quaternions from the scopes in Simulink.

The plot can be enabled by setting the variable "bool_plot_attitude" to "true" in main.m.

3.5.9.5 Plot Satellite Attitude

This plot shows a model, a box, of the satellite, the initial attitude, the current attitude, and the reference attitude. This plot exists to show how the attitude control of the satellite would affect the system.

The plot can be enabled by setting the variable "bool_plot_attitude_w_satellite" to "true" in main.m.

Chapter 4

Results and Discussion

The results are from simulations in Simulink. A fixed-step size solver has been used with 4 Hz sampling frequency, making the step size 0.25.

Kinetic energy will in this context refer to rotational kinetic energy, or more specific, the rotational kinetic energy of the satellite's body frame about the orbit frame. When the satellite is at not rotating relative to the orbit frame, this kinetic energy will be zero. The formula for rotational kinetic energy is given in equation 2.23. The values presented in these plots are twice as large as the true value, as the $\frac{1}{2}$ factor was omitted by a mistake during simulations. As this energy plot is used to observe the curve and that it reaches zero, the plots presented will serve the same purpose.

The scopes, meaning the majority of the results, have the time in seconds on the horizontal axis. On the vertical axis, the plots have Joule, radians per second, Newton meter and RPM for the plots with kinetic energy, angular velocity, torque and reaction wheel velocity, respectively.

The perturbing forces are in use for all simulations. When a subsection states that there were no perturbations in use, it means that the perturbing torques were deactivated.

4.1 Simulation Without Control

These plots are showing the simulation without any actuators controlling the attitude. They are included to show the uncontrolled response of the system with and without perturbations.

4.1.1 Without Perturbations

Figure 4.1 and 4.2 show the kinetic energy and the angular velocity of the system.

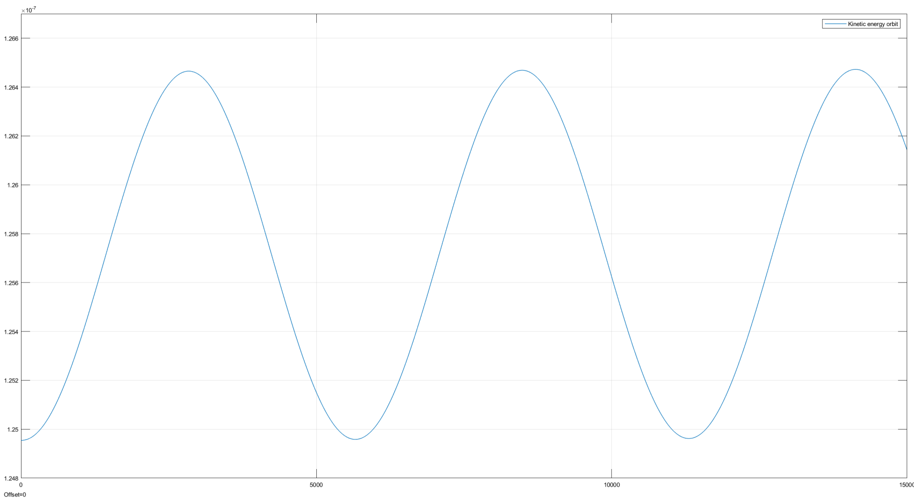

Figure 4.1: The kinetic energy with no control and no perturbations.

4.1.2 With Perturbations

Figures 4.3 and 4.4 show the kinetic energy and the angular velocity of the system.

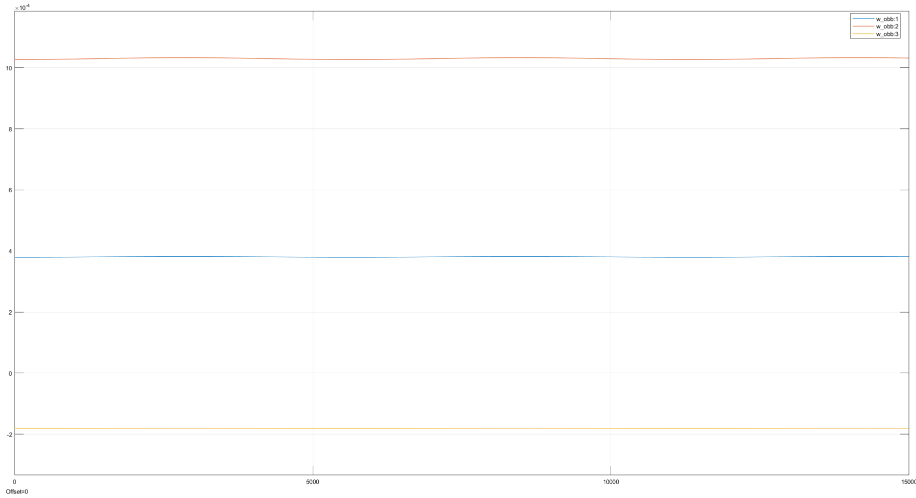

Figure 4.2: The angular velocities with no control and no perturbations.

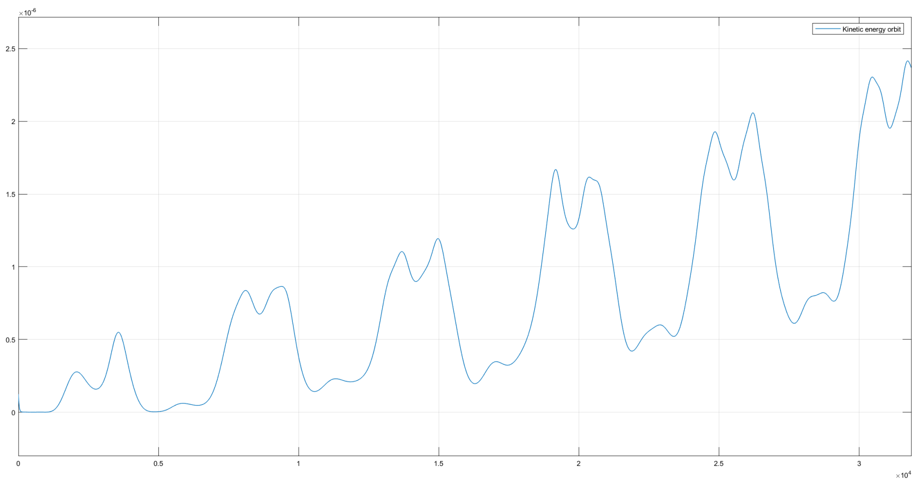

Figure 4.3: The kinetic energy with no control, with perturbations.

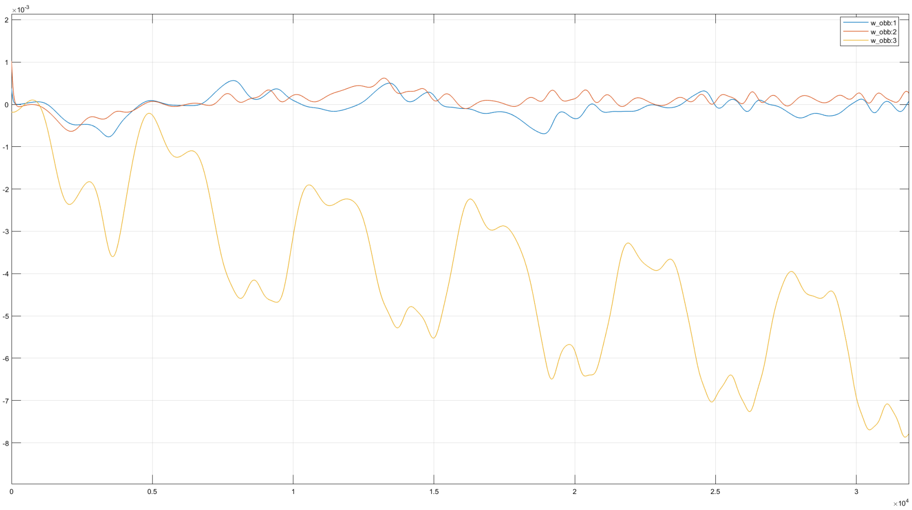

Figure 4.4: The angular velocities with no control, with perturbations.

4.1.3 Discussion

The angular velocities without perturbations are near-constant because the angular velocity of the body frame relative to the inertial frame, ECI, is set to zero at the beginning of the simulation, and no torque is applied. The near-constant values are not zero because they represent the negative of the rotation from the orbit frame about the inertial frame. See equation 3.8. This would also explain the small fluctuations in kinetic energy, when taking the perturbing forces into account.

The plots with perturbations shows an unstable system. This is most likely because of the nonphysical modeling of some of the perturbations, see section 3.3.

4.2 Control with Reaction Wheels, No Perturbations

The goal of the controller implemented for the reaction wheels is to control the satellite from one attitude to another. Initialising the attitude in Euler angles, and using the eul2quat[17] built-in transformation in Matlab to change the attitude representation to quaternions for simulation. The attitudes used were

- Initial attitude: [20°, 0°, 10°]
- Reference attitude: [-20°, 0°, 0°]

The PD controller used the controller gains

- $K_p = I_s/1000$
- $K_d = I_s * 45/1000$

where I_s is the inertia matrix of the satellite.

4.2.1 Simulation Results

Figures 4.5, 4.6, **??** and **??** show the simulation result for the PD controller with no perturbations.

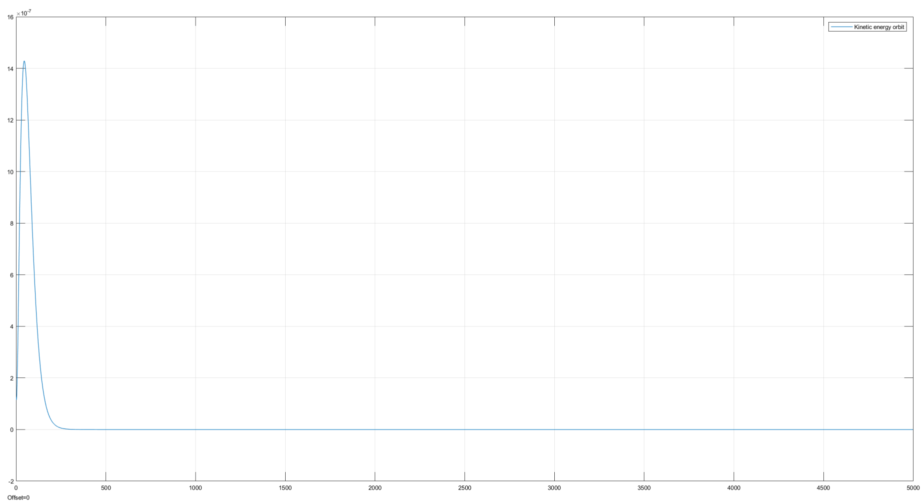

Figure 4.5: The kinetic energy with PD control on the reaction wheels, without perturbations.

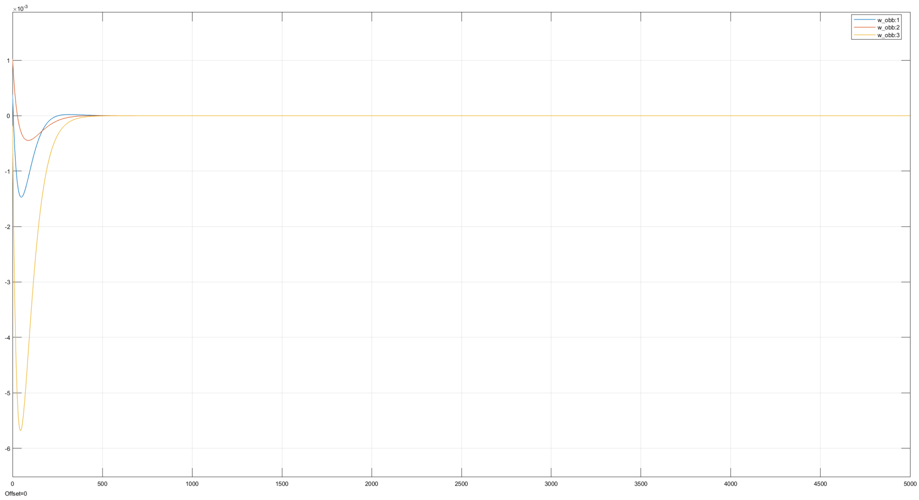

Figure 4.6: The angular velocities with PD control on the reaction wheels, without perturbations.

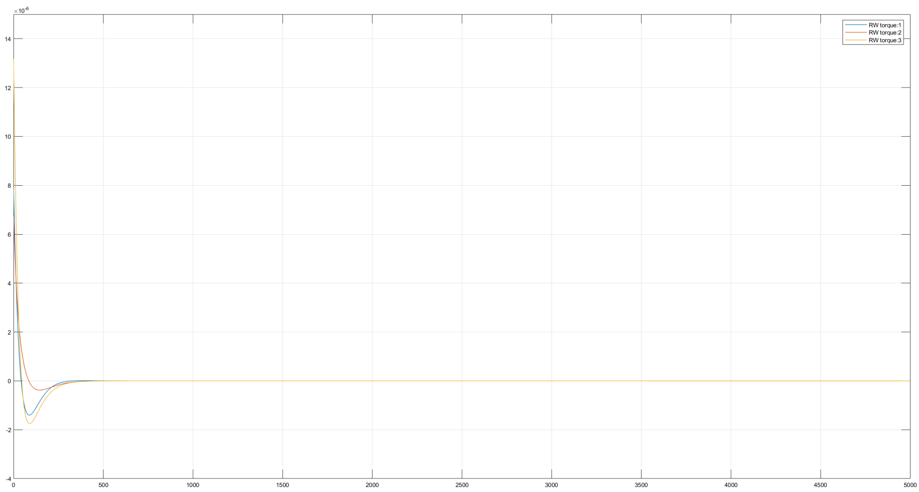

Figure 4.7: The torque input to the reaction wheels, using PD control on the reaction wheels, without perturbations.

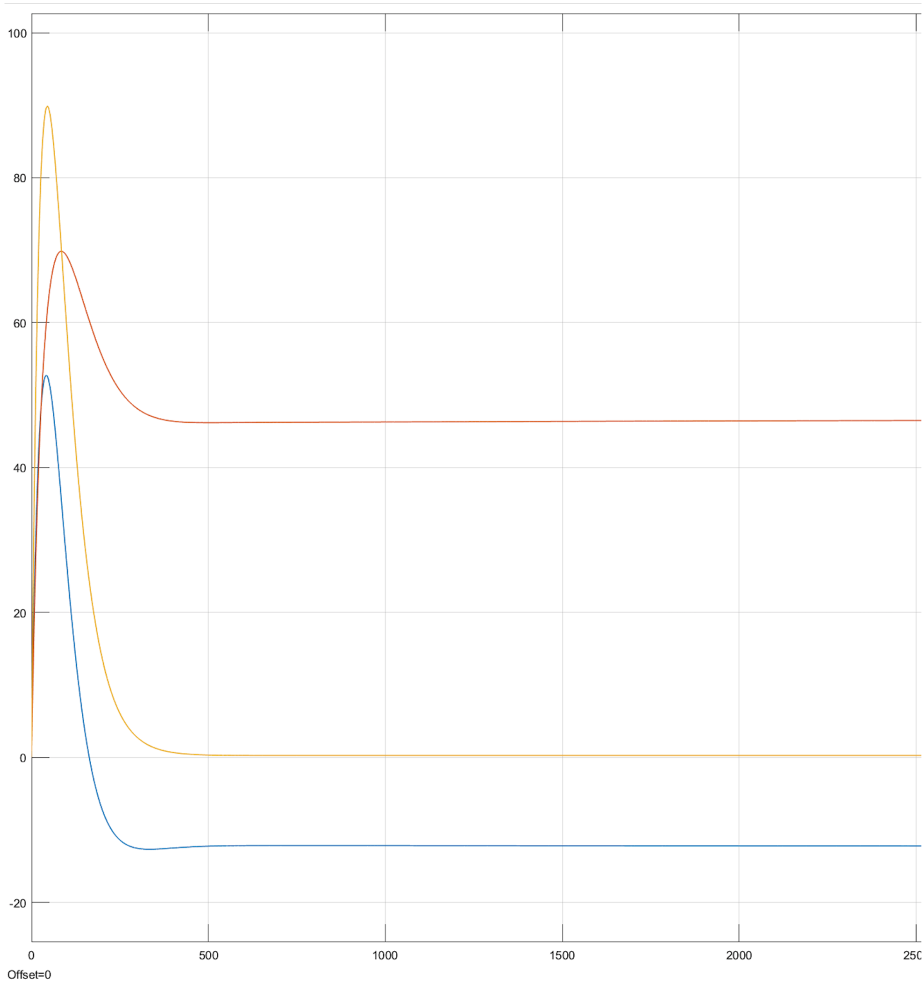

Figure 4.8: The angular velocity of the reaction wheels in RPM, using PD control on the reaction wheels, without perturbations.

4.2.2 Visualisation

The visualisations in this subsection are all post-simulations, described in section 3.5.9.

4.2.2.1 Plot Orbit - Short Tail

As described in section 3.5.9.2. The advantage of this plot is that, unlike the complete plot, it is easy to see the direction the satellite is moving. Figure 4.9 and 4.10 show the plot at two different times.

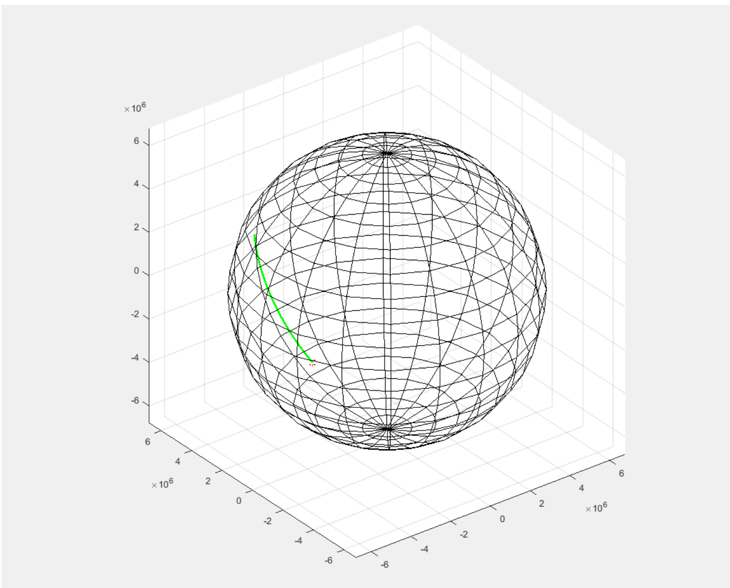

Figure 4.9: First part of orbit, plot with short tail.

4.2.2.2 Plot Orbit - Complete

See section 3.5.9.3. The plot is shown in figure 4.11.

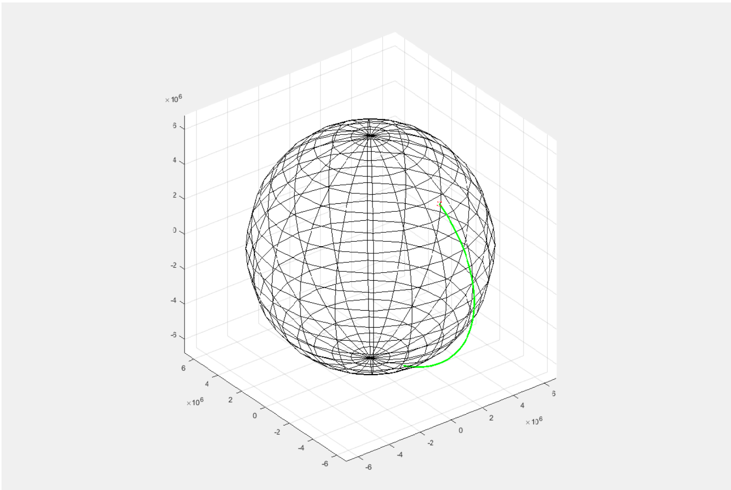

Figure 4.10: Second part of orbit, plot with short tail.

4.2.2.3 Plot Attitude

See section 3.5.9.4. Figure 4.12 shows the attitude over time for the current section.

4.2.2.4 Plot Satellite Attitude

See section 3.5.9.5. Figure 4.13 shows the satellite, represented by a blue box, at its initial attitude. Figure 4.14 is taken after the reference attitude has been reached.

4.2.3 Discussion

The PD controller works well for attitude control with no torque perturbations.

4.3 Detumbling with Magnetorquers

For detumbling, only the magnetorquers are used. Tumbling is simulated by setting the w_ibb0 variable to something other than the zero vector.

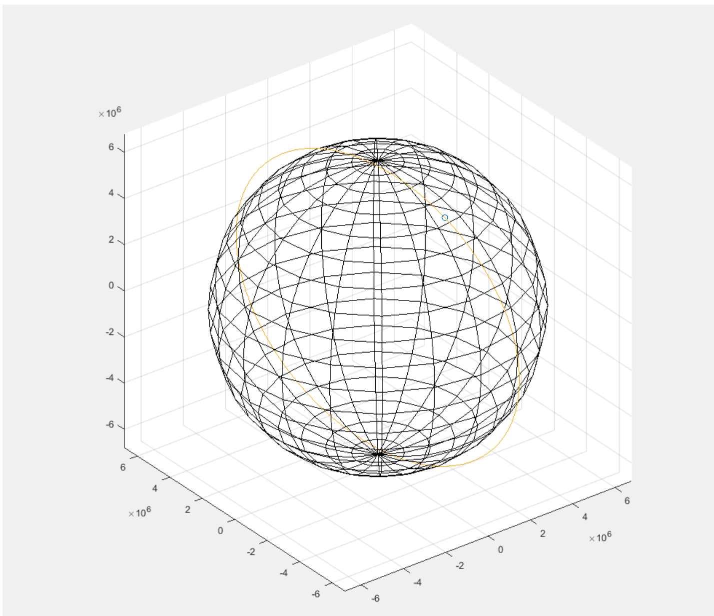

Figure 4.11: Full path of the orbit follows the satellite as a tail.

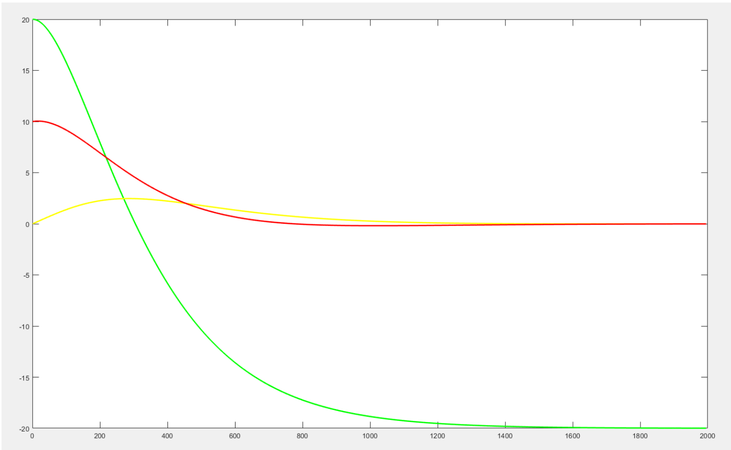

Figure 4.12: Plot of how the Euler angles representing the attitude of the satellite changes over time.

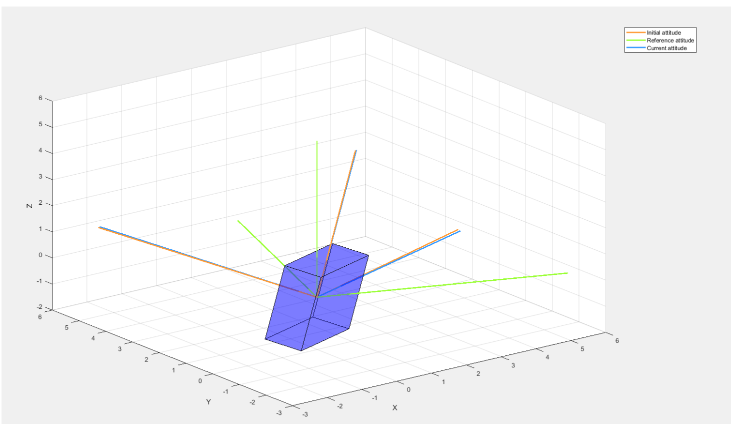

Figure 4.13: Plot showing the satellite's attitude when the control starts.

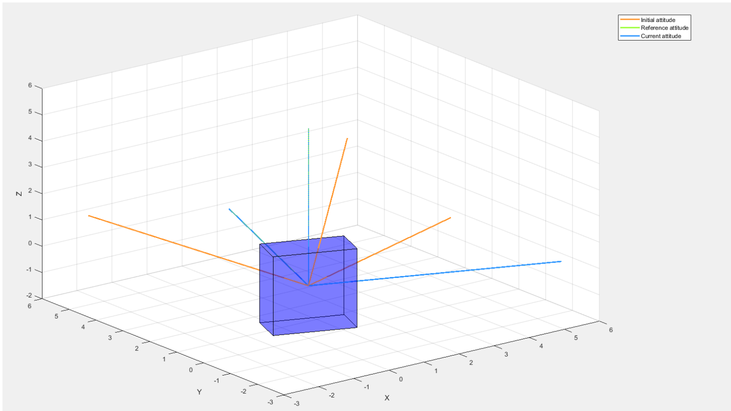

Figure 4.14: Plot showing the satellite's attitude when it has reached the reference attitude.

4.3.1 Detumbling

Setting $w_ibb0 = [0.1; 0.4; 0.9]$, all in radians per second, gives figure 4.15, 4.16 and 4.17.

4.3.1.1 Discussion

The torque and angular velocity fluctuates, and goes to zero, as it is supposed to.

4.3.2 Detumbling of 10x the Angular Velocity

Now, w_ibb0 is set ten times larger than what it was in the previous subsection. This gives the figures 4.18, 4.19 and 4.20. Figure 4.21 and 4.22 shows plots of the angular velocity and the magnetorquer torque, respectively, zoomed in to the beginning of the simulation.

4.3.2.1 Discussion

It is clear that the detumbling controller works even with a larger initial tumbling.

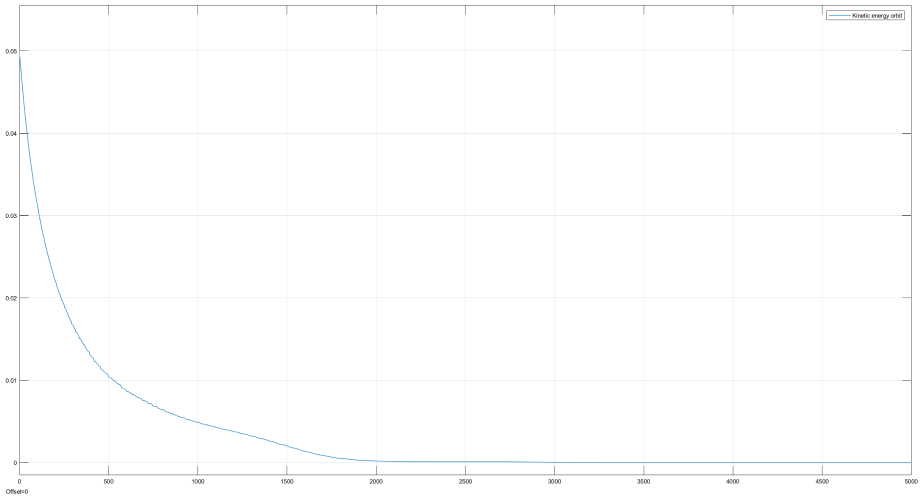

Figure 4.15: Detumbling with magnetorquers, kinetic energy.

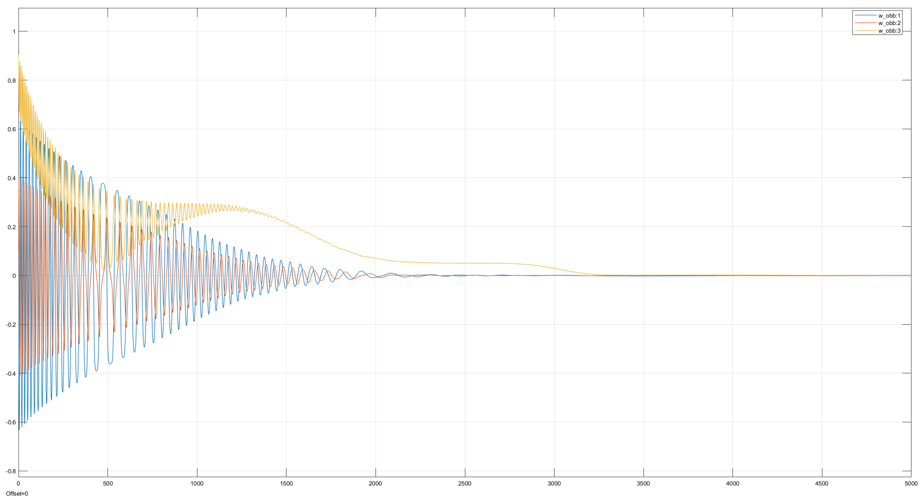

Figure 4.16: Detumbling with magnetorquers, angular velocity.

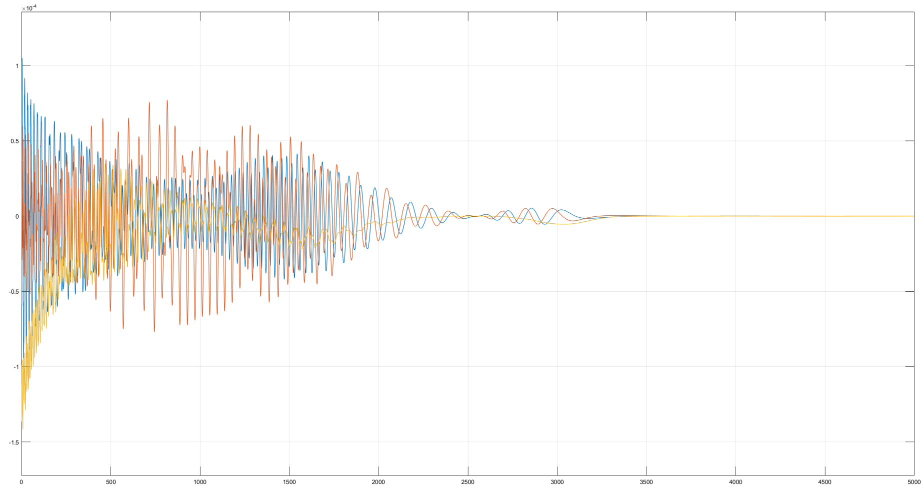

Figure 4.17: Detumbling with magnetorquers, torque.

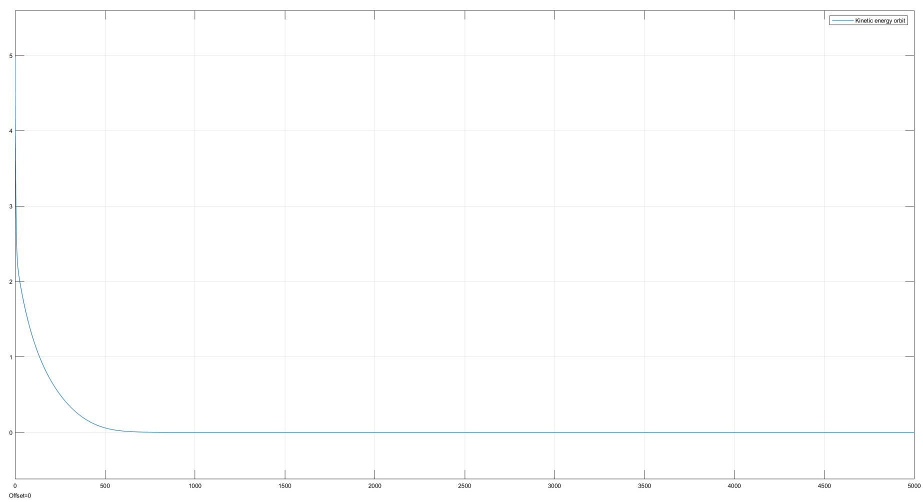

Figure 4.18: Detumbling with magnetorquers using 10x angular velocity, kinetic energy.

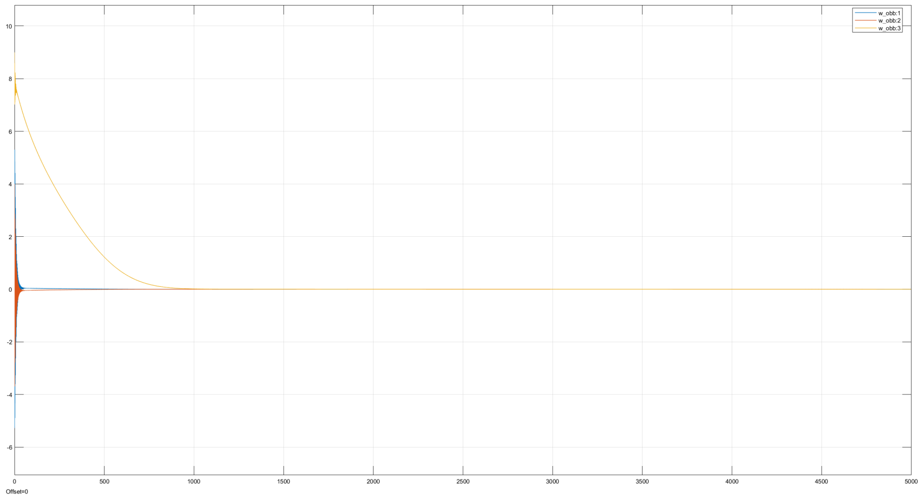

Figure 4.19: Detumbling with magnetorquers using 10x angular velocity, angular velocity.

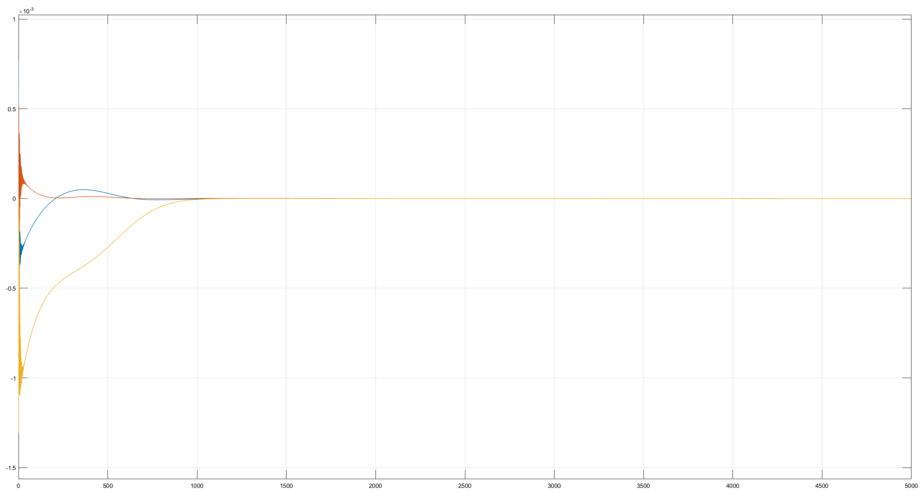

Figure 4.20: Detumbling with magnetorquers using 10x angular velocity, torque.

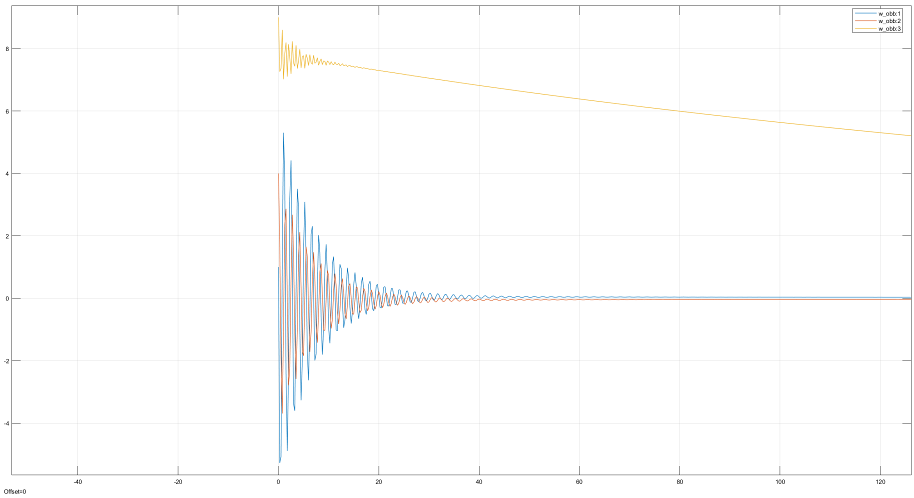

Figure 4.21: Detumbling with magnetorquers using 10x angular velocity, angular velocity at the beginning of the simulation.

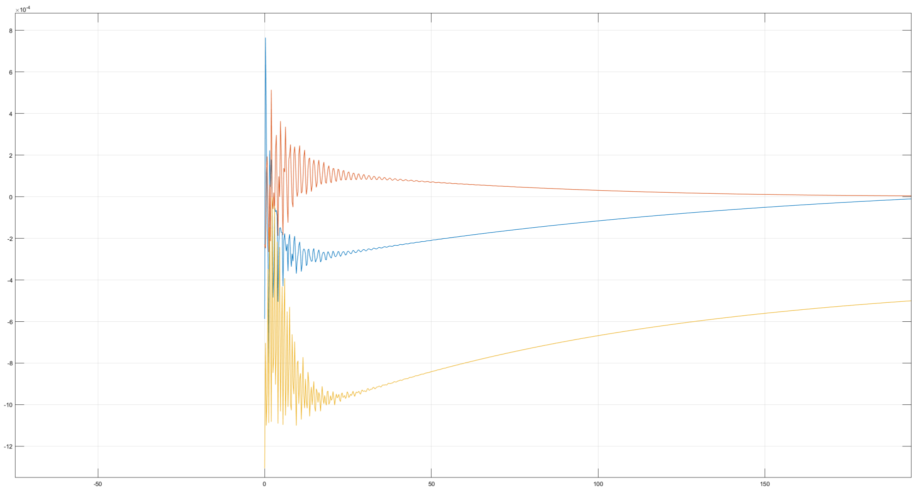

Figure 4.22: Detumbling with magnetorquers using 10x angular velocity, torque at the beginning of the simulation.

4.4 Control with Reaction Wheels, With Perturbations

4.4.1 With Only Gravitational Gradient

While adding the gravitational gradient, the controller gains for the reaction wheels PD controller were kept the same. The results are in figures 4.23, 4.24, 4.25 and 4.26.

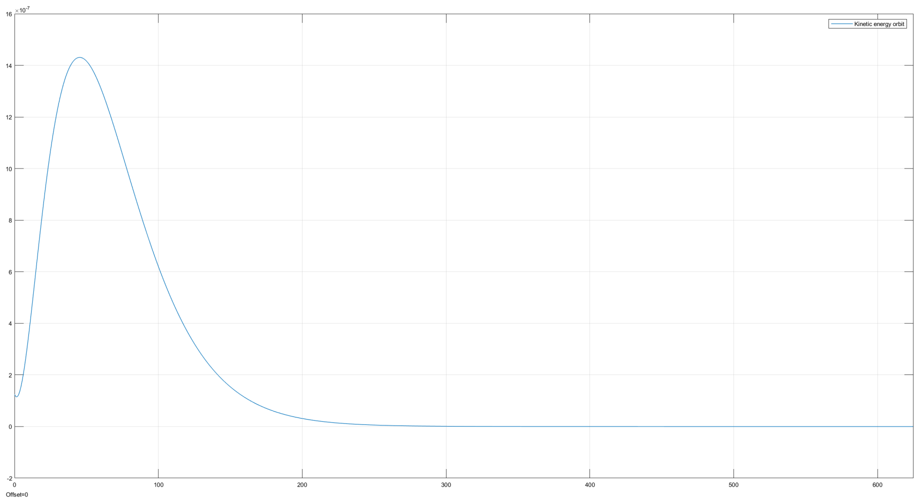

Figure 4.23: The kinetic energy with PD control on the reaction wheels, with gravitational gradient.

4.4.1.1 Discussion

There seems no major downside to have the gravitational torque applied to the simulated system. The same controller manages to regulate the attitude of the satellite. The only difference is a small wave in the RPM plot.

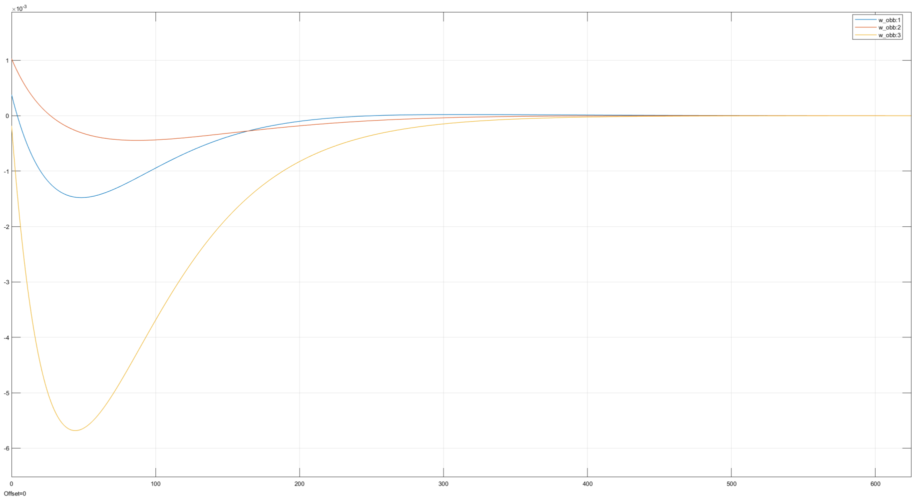

Figure 4.24: The angular velocities with PD control on the reaction wheels, with gravitational gradient.

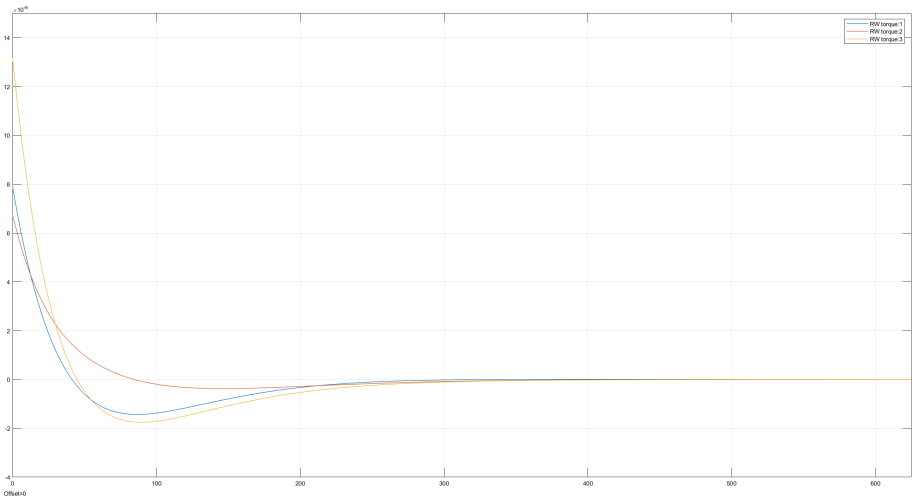

Figure 4.25: The torque input to the reaction wheels, using PD control on the reaction wheels, with gravitational gradient.

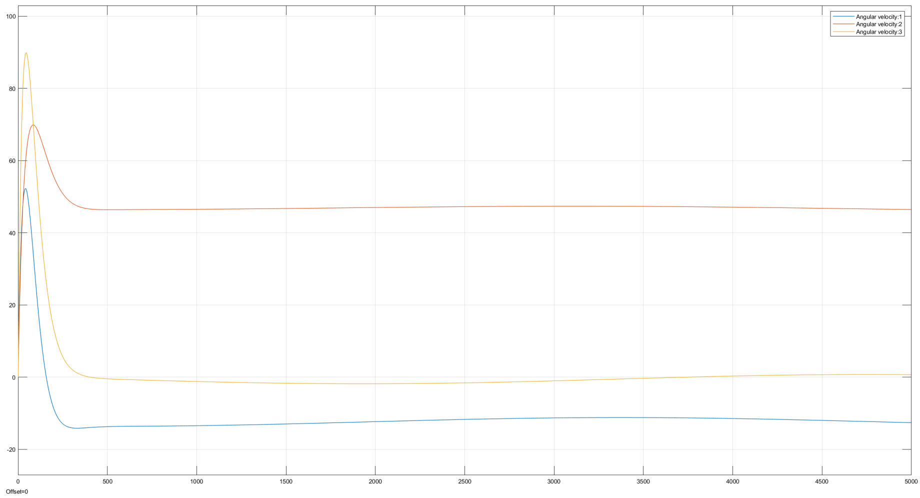

Figure 4.26: The angular velocity of the reaction wheels in RPM, using PD control on the reaction wheels, with gravitational gradient.

4.4.2 With All Perturbations

With all the perturbations applied, and a PD controller with gains set to

- $K_p = I_s/100$
- $K_d = I_s * 45/100$

gives the results shown in figure 4.27, 4.28, 4.29 and 4.30. Figure 4.31 and 4.32 show the torque and RPM plots over a longer simulation time.

4.4.2.1 Discussion

In the shorter horizon, it is clear that the controller manages to find the reference attitude and point in the right direction. However, the reaction wheels struggle to keep the reference attitude because of the perturbations. An integral term or a noise cancellation term can be added to the PD controller for attitude control in order to cancel disturbances[16]. Some different feed-forward techniques could be also be tried.

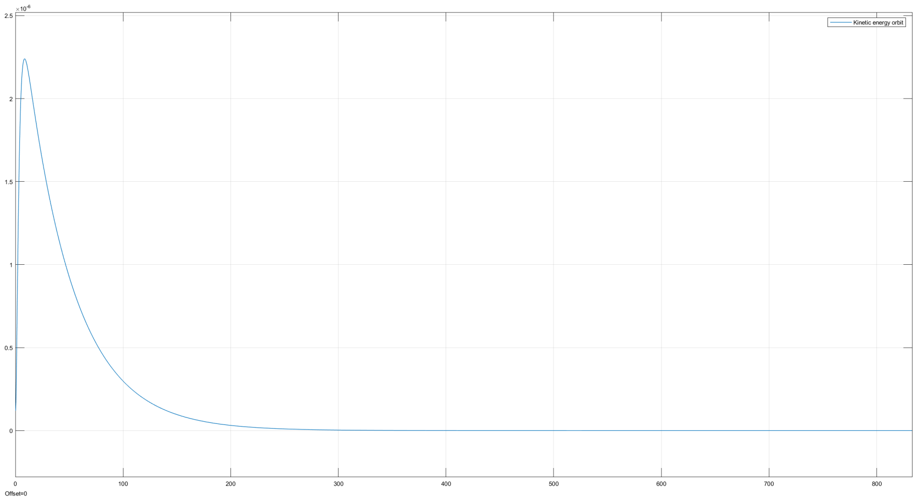

Figure 4.27: The kinetic energy with PD control on the reaction wheels, with all perturbations.

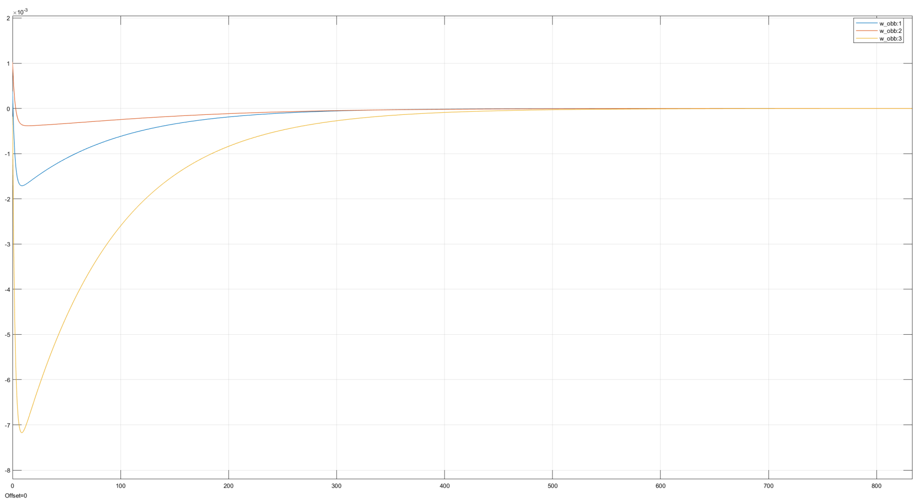

Figure 4.28: The angular velocities with PD control on the reaction wheels, with all perturbations.

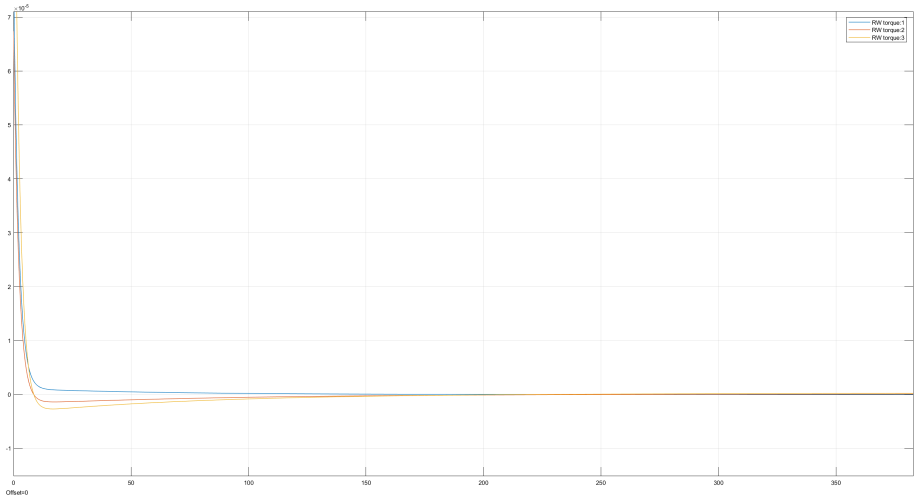

Figure 4.29: The torque input to the reaction wheels, using PD control on the reaction wheels, with all perturbations.

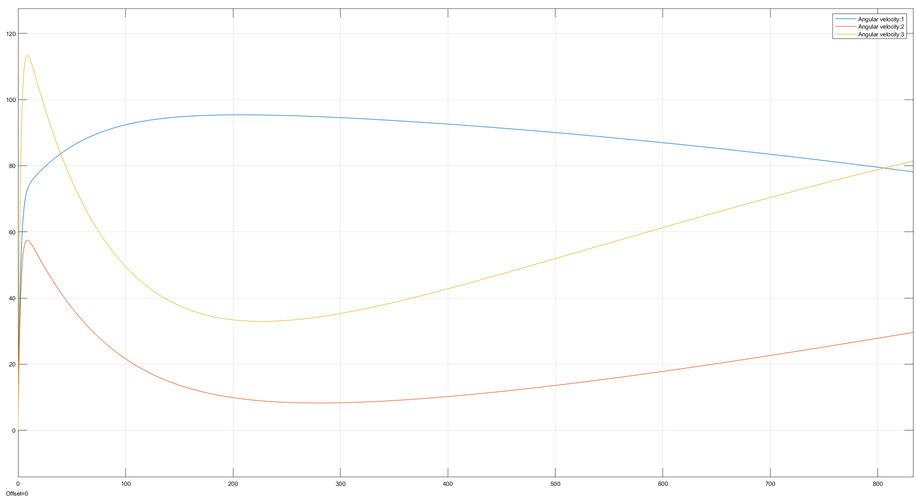

Figure 4.30: The angular velocity of the reaction wheels in RPM, using PD control on the reaction wheels, with all perturbations.

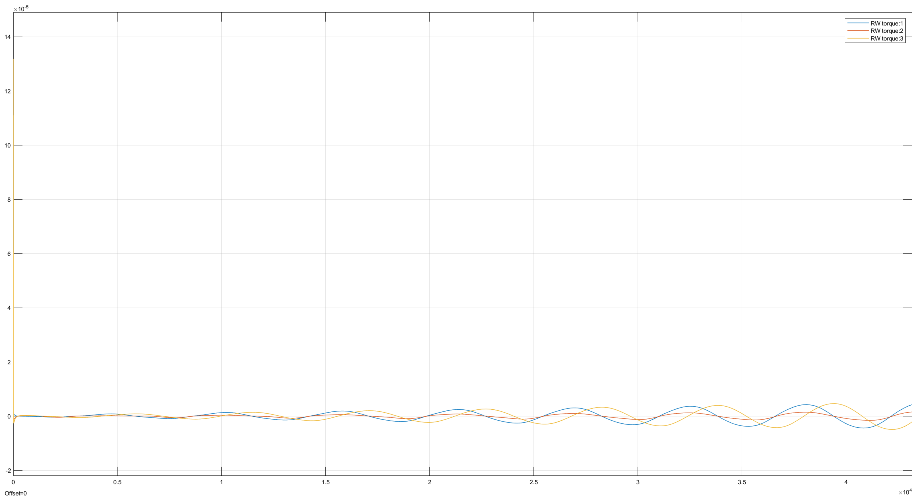

Figure 4.31: The torque input to the reaction wheels, using PD control on the reaction wheels, with all perturbations, for a longer simulation time.

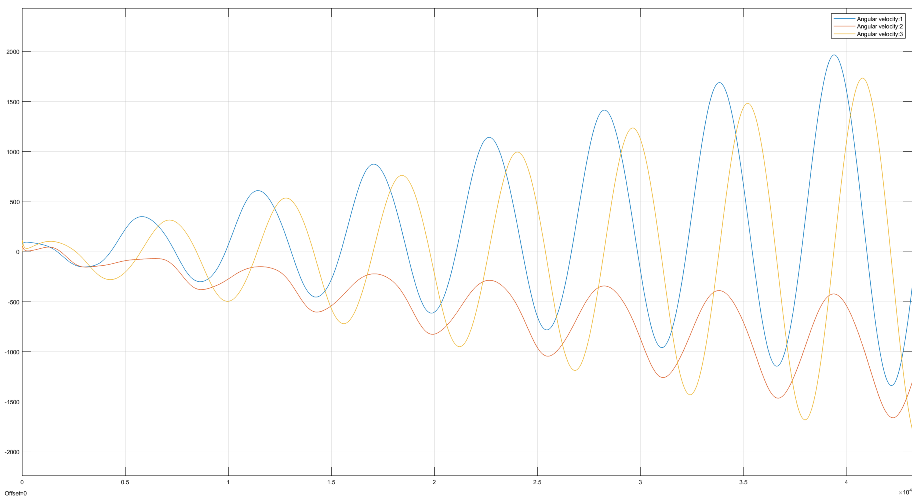

Figure 4.32: The angular velocity of the reaction wheels in RPM, using PD control on the reaction wheels, with all perturbations, for a longer simulation time.

Chapter 5

Conclusion

The magnetorquers using the B-dot algorithm worked well for detumbling.

Without perturbations, the PD controller worked fine. However, the control law used for the reaction wheels did not handle the perturbations well enough over time when they were introduced. This calls for a change in which control law to use.

In total, the work in this project thesis, the modeling and the Simulink model, creates a good foundation for further investigation of different control laws to use in the HYPSO satellite.

Chapter 6

Future Work

This chapter describes further work that could be done on the basis of this project thesis. Suggested improvements are summarised in the list below.

- Fix simplifications in the model
  - Change the assumption of a spherical Earth
  - Model the Sun's position for the perturbing force and torque in section 3.3.4.1 and 3.3.4.2.
  - Use a more accurate model for the atmospheric drag arm in section 3.3.3.2.
  - Model the dynamics of the reaction wheels. Currently, the reaction wheels are modelled using an ideal model. In the real system, the reaction wheels will need an inner controller to meet the desired response. Modeling the real reaction wheels includes adding saturation limits for when the angular velocity goes too high.
  - Assume that the model is not perfect, include noise and an attitude determination system (ADS) in the loop. An assumption that has been made in this project thesis is that Simulink manages to keep the length of the quaternion perfectly at unity, avoiding numerical errors, which is not always true. A

possible implementation for unit quaternion normalisation for Simulink is presented in [4, p. 31].

- Investigate other control schemes
  - Develop and implement other controllers than the PD controller for pointing for the reaction wheels. The current block diagram setup is ideal for switching controllers.
  - The aim of the satellite mission is to perform a slewing maneuver. No controller like this has been implemented here yet.
  - Investigate the interactions between the magnetorquers and the reaction wheels. Moment dumping and desaturation of the reaction wheels in particular.

Bibliography

[1] NUTS 1 (satellite), “Hyper spectral imager for ocenaograpic applications,” [Online; accessed 18-December-2018].

[2] M. E. G. et al., “Hyperspectral imaging small satellite in multi-agent marine observation system,”

[3] N. S. Lab, “Nuts 1 (satellite) — Wikipedia, the free encyclopedia,” [Online; accessed 18-December-2018].

[4] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control.* Wiley, 2014.

[5] O. Egeland and T. Gravdahl, *Modeling and Simulation for Automatic Control.* Tapir Trykkeri, 2003.

[6] H. D. Young and R. A. Freedman, *University Physics with Modern Physics. Scandinavian Edition, Volume One.* Dover Publications, Inc., 2016.

[7] N. A. .-.-. P. Release, *Innovative ocean research from ntnu to ride on the nanoavionics m6p nano-satellite bus*, [Online; accessed 18-December-2018].

[8] U. o. C. B. Colorado Center for Astrodynamics Research, *Kepler orbit elements to eci cartesian coordinates conversion*, [Online; accessed 17-December-2018; Handout to ASEN 5070].

[9] ——, *Eci cartesian coordinates to kepler orbit elements conversion*, [Online; accessed 17-December-2018; Handout to ASEN 5070].

[10] M. Blanke and M. B. Larsen, *Satellite Dynamics and Control in a Quaternion Formulation. [Lecture note for course 31365 Spacecraft Dynamics and Control at DTU]*. 2010, Version 2f.

[11] E. Oland and R. Schlanbusch, "Reaction wheel design for cubesats," 2009.

[12] N. Oceanic and A. Administration, *International geomagnetic reference field*, [Online; accessed 18-December-2018].

[13] J. Davis, *Mathematical Modeling of Earth's Magnetic Field, Technical Note*, [Online; accessed 18-December-2018].

[14] P. C. Hughes, *Spacecraft Attitude Dynamics*. Dover Publications, Inc., 1993.

[15] J. T. Gravdahl, "Magnetic attitude control for satellites," 2004, [Online; accessed 18-December-2018].

[16] C. M. Pong and D. W. Miller, "High-precision pointing and attitude estimation and control algorithms for hardware-constrained spacecraft," 2014.

[17] MathWorks, *eul2quat, Convert Euler angles to quaternion*, [Online; accessed 18-December-2018].