Abstract

Initial investigations into the dynamics and control of the satellite attitude was conducted for the CubeSat being built by the student organization Orbit. The biggest concern was how the planed deplyable boom with a camera module would affect the dynamics. Therefore a dynamic model for a CubeSat under the influence of gravity gradient was developed. The model was then linerized for stability analysis. Two fundamental controllers were also designed and tested in simulation. One controller was based on the linerized system while the other was just a straight up PD-controller. It was shown that the positioning of the arm has a great influence on the dynamic as it is mostly governed by the inertia matrix of the satellite. It was also shown that the linerized model fits well for the pitch dynamic and is suitable for control design. For the roll and yaw dynamics on the other hand the linerized model is not that suitable.

Contents

1 Introduction **1**
- 1.1 Introduction to CubeSat . . . 1
- 1.2 Selfie-Sat . . . 1
- 1.3 Attitude Determination and Control System . . . 2
- 1.4 Selfi-Sat ADCS . . . 3

2 Background **6**
- 2.1 Rigid Body Dynamics . . . 6
- 2.2 Inertia Matrix . . . 8
- 2.3 Gravity Gradient Torque . . . 8

3 Methods **13**
- 3.1 Simulation . . . 13
- 3.2 Orbit Propagator . . . 15
- 3.3 Orbit Frame . . . 15
- 3.4 Controller . . . 16
  - 3.4.1 Pitch Linear Controller . . . 16
  - 3.4.2 Roll and Yaw Linear Controller . . . 17
  - 3.4.3 Quaternion Based Controller . . . 18

4 Results **19**
- 4.1 Open Loop Stability . . . 19
  - 4.1.1 Open Loop Pitch Stability . . . 19
  - 4.1.2 Open Loop Roll and Yaw Stability . . . 20
- 4.2 Controller . . . 22
  - 4.2.1 Pitch Control . . . 22
  - 4.2.2 Roll and Yaw Control . . . 22
  - 4.2.3 Quaternion Based Control . . . 25

5 Conclusion **28**

6 Outlook **29**

References . . . 30

Chapter 1

Introduction

Orbit is a student organization at NTNU working towards increasing its students interest in all aspects of space and space technology, with a specific focus on satellite technology. The organization mostly consists of students working as volunteers and students writing project or master thesis with the support from NTNU. The organization consists of about 30 students. Currently the organization is developing the CubeSat Selfi-Sat.

1.1 Introduction to CubeSat

A CubeSat is a satellite that follows the CubeSat standard. The CubeSat standard imposes a lot of restrictions on the satellite, most notably with regards to form factors. The CubeSat standard is built around a 'unit' called 1U. Where 1U is a $0.10\,\mathrm{m x 0.10\,m x 0.10\,m}$ cube with a maximum mass of $1.33\,\mathrm{kg}$. A CubeSat can come in different sizes which are all defined by combining several 1Us together. Normal sizes are 1U, 2U, 3U and 6U[1]. The big advantages with following this standard is that it makes the process of finding a launch a lot easier as there has also been developed standardized launch interfaces to comply with the CubeSat standard. This means that the CubeSat can be launched with any rocket having the interface and it is starting to be a very common interface to include. Even the International Space Station has the capability to deploy CubeSats[5].

Another big advantage with the standard is that a lot of companies and universities are following the standard so there is also starting to come more and more parts that are sold. All these factors make CubeSatsn an ideal starting point for small organizations without a million dollar budget.

1.2 Selfie-Sat

Selfie-Sat is the first satellite developed by the student organization Orbit. It is still in a very early stage of development and everything is not properly defined yet. The main mission of the satellite is determined. To summarize the satellite shall have a screen on it and a camera on an arm. The camera will then take a picture of the screen with the Earth in the background see, Figure 1.1. On the screen, a picture that is uploaded from

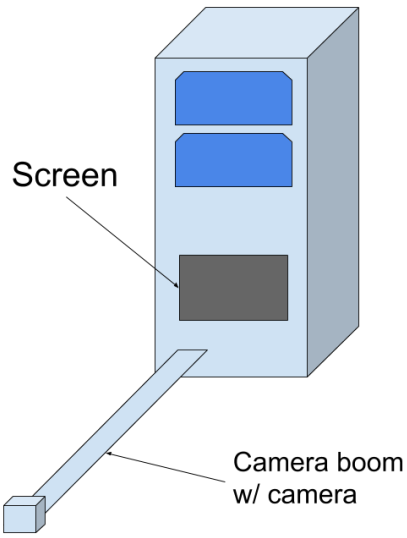


Figure 1.1: Illustration of Selfi-Sat with camera boom and screen.

Earth will be displayed. The picture or selfie will then be sent back to Earth. For the high data rate transition of the picture a S-band radio will be used. In addition to the payload the Selfie-sat will consist of the following sub subsystems to make a complete CubeSat bus: Electronic Power Supply (EPS), On-Board Computer (OBC), Ultra High Frequency (UHF) radio, the mechanical structure and the Attitude Determination and Control System (ADCS).

The EPS main purpose is to supply the satellite with the necessary electrical power. It consists of a battery and a charging circuit to charge the batteries. The power needed to charge the batteries comes from solar cells on the outside of the satellite. The OBC is in charge commands and data handling. In addition, it monitors the health of the satellite and is responsible for keeping the satellite in the correct state. The UHF radio will be used for receiving commands from the ground station and sending telemetry data from the satellite back to the ground station. The mechanical structure holds everything together and because of all the shock and vibration that the satellite experiences is especially important during launch. The ADCS is responsible for controlling and determining the attitude of the satellite. This is necessary to make sure that the Earth is in the background of when the pictures are taken.

1.3 Attitude Determination and Control System

The ADCS is in charge of determining and controlling the attitude of the satellite. This is necessary for a number of reasons. The first is to detumble the satellite after deployment from the rocket. After deployment the satellite normally has a high angular velocity which often needs to be reduced before deployables like antenna and other instruments can be released.

The second reason is that very often the payload of the satellite is an instrument that requires the satellite to be in a specific orientation. For the Selie-Sat it is necessary to making sure that the Earth is in the background when the pictures are taken. There are

also satellite designs that need a specific orientation of the satellite to maximize its solar power production.

ADCS is a complex system that can take on many forms and mean different things for different missions, but it can almost always be looked at as a classical control system consisting of a controller, actuators and sensors creating a closed loop. Very of then there is also an observer or estimator. There is large variety in both actuators and sensors. The most common actuators are magnetorquers, reaction wheels and thrusters[7]. Magnetorquers are magnetic coils used to interact with the magnetic field of the Earth to create a torque. Reaction wheels are spinning discs that use the fact that moment is preserved to change the angular velocity of the satellite. If the speed in which the discs are spinning changes, the angular velocity of the satellite will also change. Thrusters are normally only used on bigger satellites, but they are starting to see some use on CubeSat [7].

There also exist many sensors that are used as part of the ADCS and it is normal to use a combination of sensors. The three main sensors are: magnetometers, optical sensors and gyroscopes. The Magnetometers are used to measure the magnetic field of the Earth. The optical sensors are a broad category that ranges from simple light diodes to more complex camera sensors, and are used to detect anything from the horizon of the Earth, the Sun or the stars[7].

There are generally two ways of estimating the orientation of the satellite. Direct methods that mostly consist of solving Wahba's problem and filtering methods like the Kalman filter. Both methods require a reference model in addition to the sensor inputs to make an estimation of the satellite attitude[3].

1.4 Selfi-Sat ADCS

The ADCS for the Selfi-Sat has three main objectives:

- Orient the satellite so that the Earth is in the background of the selfies
- Orient the satellite so that the S-band antennas are pointing towards the ground during passes for radio transmission.
- Orient the satellite to maximize the power generated by the solar cells.

In hardware the ADCS consists of the ADCS-PCB, magetorquers-PCB and Sun sensors. The ADCS-PCB consists of a micro-controller of type TMS570 which runs all the ADCS related software, IMUs for gyroscope, magnetometer and acceleration measurements, operational amplifiers for the Sun sensor measurements and H-bridges for controlling the magnetorquers. The ADCS-PCB is connected to the rest of the satellite bus through a PCI-104 connector. The hardware architecture of the ADCS can be seen in Figure 1.2.

The software of the ADCS is broken down as shown in the Figure 1.3. For this project, the main focus will be the "Attitude Controller" while it is assumed that the rest of the system works. So in the simulations all the other parts are not considered. This means that the desired and estimated attitude of the satellite will be given directly from the simulation without any estimation. The output of the controller will be a torque that affects the system directly without a simulation of the hardware needed to create the torque.

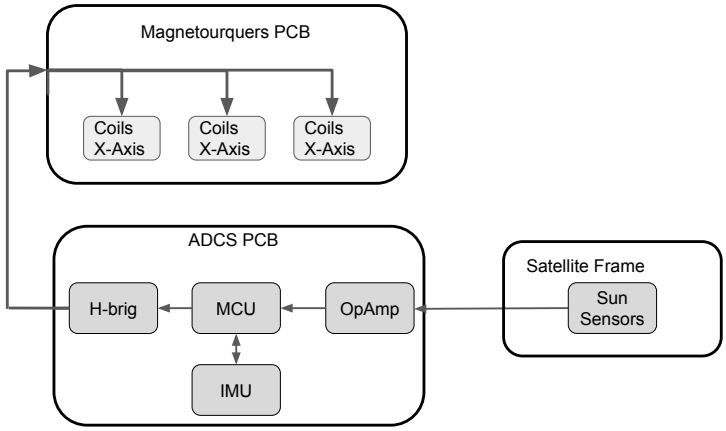


Figure 1.2: Overview of the hardware architecture for the ADCS on Selfi-Sat

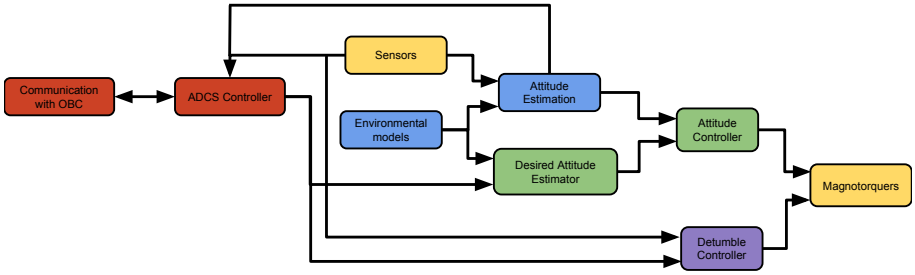


Figure 1.3: Software Architecture of the ADCS.

As the satellite is still in development there are a few assumptions that are made about the satellite. These are made in a way that the dynamics of the satellite can be well defined and can therefore be used for stability analysis, simulations and control design. The assumptions are:

- The Selfi-Sat follows the CubeSat standard for a 2U.
- The Selfi-Sat has an arm of length $0.30\,\mathrm{m}$.
- The camera module at the end of the arm has a mass of $50\,\mathrm{g}$.
- The mass distribution within the main frame of the satellite is uniform.
- The mass of the camera can be seen as a point mass.
- The center of mass is at the geometric center of the satellite.
- Before the arm is released the principal axes of the satellite are aligned with the axis of symmetry. As shown in Figure 1.4.

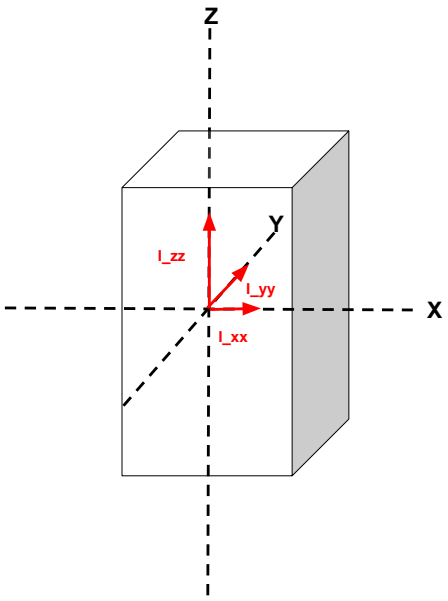


Figure 1.4: The dotted lines are the axis of symmetry for the satellite. The read arrows show the inertia matrix.

- The Selfi-Sat has a estimator capable of determining the quaternion and angular velocity.
- The Selfi-Sat has magnetorquers capable of producing $550\,\mathrm{mA}^2$ on each axis resulting in a max torqu of about $1\,\mu\mathrm{N\,m}$.

Chapter 2

Background

2.1 Rigid Body Dynamics

Satellite attitude dynamics are normally modeled with rigid body dynamics. In this work we only look at rotational motion about the center of gravity, as this is what affects the attitude of the satellite. The rotational motions are independent of the translation motions as can be seen in Equation 2.1 [2].

$$\begin{bmatrix} m\mathbf{I}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{I}_g \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}^b_{g/n} \\ \dot{\boldsymbol{\omega}}^b_{b/n} \end{bmatrix} + \begin{bmatrix} m\mathbf{S}(\boldsymbol{\omega}^b_{b/n}) = & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & -\mathbf{S}(\mathbf{I}_g\boldsymbol{\omega}^b_{b/n}) \end{bmatrix} \begin{bmatrix} \mathbf{v}^b_{g/n} \\ \boldsymbol{\omega}^b_{b/n} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^b_g \\ \mathbf{g}^b_g \end{bmatrix} \tag{2.1}$$

Where $\mathbf{I}_g$ is the inertia matrix of the satellite in the center of gravity and will for the reminder of the text simply be denoted $\mathbf{I}$ for simplicity. $\mathbf{v}^b_{g/n}$ is the transnational velocity between center of gravity and an inertial frame resolved in the body reference frame. $\boldsymbol{\omega}^b_{b/n}$ is the angular velocities between the body frame and an internal frame resolved in body frame. $\mathbf{S}(x)$ is the skew symmetric matrix of x. As we are only interested in the rotation we can reformulate Equation 2.1 into Equation 2.2.

$$\dot{\boldsymbol{\omega}}^b_{b/n} = \mathbf{I}^{-1}(\mathbf{S}(\mathbf{I}\boldsymbol{\omega}^b_{b/n})\boldsymbol{\omega}^b_{b/n} + \mathbf{g}^b_g) \tag{2.2}$$

The attitude of the satellite is represented through a rotation between the body frame of the satellite and an inertial frame. For satellites, a common inertial frame is the Earth Centered Inertial frame (ECI). It is defined as having its origin at the center of mass of the Earth, the z-axis is aligned with the Earth's north pole, the x-axis is aligned with the vernal equinox and the y-axis is defined by completing the right hand rule. The vernal equinox is the intersection between Earth's orbital plane and Earth's equatorial plane[3].

Unit quaternions are used to represent the rotation between the frames. The unit quaternion is defined as $q = [\eta \; i\epsilon_1 \; j\epsilon_2 \; k\epsilon_3]^T$ and $|q| = 1$. Quaternions are used as a representation when a singularities free representation is needed are needed. This is need for the satellite dynamics because the satellite unlike many other rigid bodies on Earth can rotate freely around any of its axes. It can therefore have any attitude. The relationship between quaternions and angular velocity can be seen in Equation 2.3a. Where $\mathbf{T}(\mathbf{q})$ is defined

as in Equation 2.3b and represents the relation between quaternions and angular velocity [2].

$$\dot{\mathbf{q}}_b^i = \mathbf{T}(\mathbf{q}_b^i)\boldsymbol{\omega}_{b/n}^b \tag{2.3a}$$

$$\mathbf{T}(\mathbf{q}) = \begin{bmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \eta & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \eta & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \eta \end{bmatrix} \tag{2.3b}$$

Combining the kinematic and dynamic equations gives the full state space model as shown in Equation 2.4. The state space model is highly nonlinear with multiplication between states for both the quaternion and the angular velocity. It is also worth noting that the system is only in equilibrium when the angular velocity is zero as any angular velocity will make $\dot{\mathbf{q}} \neq 0$. This comes from the requirement that the quaternion should be an unit quaternion that means that one of the elements of the quaternion must be non zero, and as can be seen by Equation 2.3b all elements of the quaternion are multiplied with all elements of the angular velocity, making $\boldsymbol{\omega} = 0$ the only equilibrium.

$$\dot{\mathbf{q}}_b^i = \mathbf{T}(\mathbf{q}_b^i)\boldsymbol{\omega}_{b/n}^b \tag{2.4a}$$

$$\dot{\boldsymbol{\omega}}_{b/n}^b = \mathbf{I}^{-1}(\mathbf{S}(\mathbf{I}\boldsymbol{\omega}_{b/n}^b)\boldsymbol{\omega}_{b/n}^b + \mathbf{g}_g^b) \tag{2.4b}$$

If we are only looking at the angular velocity, there will be one more equilibrium points for a pure rotation around each axes. This can easily be seen as each element of $\dot{\boldsymbol{\omega}}$ is dependent on the multiplication of two different elements of $\boldsymbol{\omega}$ which will always be zero as long as $\boldsymbol{\omega}$ only has one non zero element. Only the rotation around the minor and major principle axis are locally stable but not asymptotically stable[4]. This can be illustrated by looking at the linerized model of the angular velocity. The Jacobin for Equation 2.2 is given by Equation 2.5 where $I_x = \frac{I_{yy}-I_{zz}}{I_{xx}}$, $I_y = \frac{I_{zz}-I_{xx}}{I_{yy}}$ and $I_z = \frac{I_{xx}-I_{yy}}{I_{zz}}$. I_{xx}, I_{yy}, I_{zz} are the diagonal elements of the principal inertia matrix.

$$\mathbf{A} = \begin{bmatrix} 0 & I_x\omega_z & I_x\omega_y \\ I_y\omega_z & 0 & I_y\omega_x \\ I_z\omega_y & I_z\omega_x & 0 \end{bmatrix} \tag{2.5}$$

If we look at rotation only around the z-axis, the expression for the eigenvalues is given by $-\lambda(\lambda^2 - I_x I_y \omega^2) = 0$. This leads to the eigenvalues being $\lambda_1 = 0$ and $\lambda_{2,3} = \sqrt{I_x I_y \omega^2}$. So if $I_x I_y \omega^2 > 0$ then $\lambda_{2,3}$ will become a positive and a negative real number and the system is unstable. If $I_x I_y \omega^2 < 0$ then $\lambda_{2,3}$ becomes imaginary conjugates with no real part. In the later case the system will be a harmonic oscillator. The sign of $I_x I_y \omega^2$ is dependent on the relation between I_{xx}, I_{yy} and I_{zz} as shown if Table 2.1. It can be seen that the rotation around the z-axis is stable, if the z-axis is a minor or major axis and unstable, if the the z-axis is an intermediate axis.

$I_{xx} < I_{zz}$	$I_{yy} < I_{zz}$	$I_x I_y \omega^2 < 0$	stable
$I_{xx} < I_{zz}$	$I_{yy} > I_{zz}$	$I_x I_y \omega^2 > 0$	unstable
$I_{xx} > I_{zz}$	$I_{yy} < I_{zz}$	$I_x I_y \omega^2 > 0$	unstable
$I_{xx} > I_{zz}$	$I_{yy} > I_{zz}$	$I_x I_y \omega^2 < 0$	stable

Table 2.1: Shows the sign of the argument for the square root of the eigenvalues of the linearized angular rotation dynamics.

2.2 Inertia Matrix

From Equation 2.1 it can also be seen that the angular velocity is only dependent on external moments $\mathbf{g}_g^b$ and the inertia matrix. The external forces will be discussed in more detail later in the chapter. The inertia matrix is defined as in Equation 2.6 [2].

$$\mathbf{I} = \begin{bmatrix} \mathbf{I}_{xx} & \mathbf{I}_{xy} & \mathbf{I}_{xz} \\ \mathbf{I}_{yx} & \mathbf{I}_{yy} & \mathbf{I}_{yz} \\ \mathbf{I}_{zx} & \mathbf{I}_{zy} & \mathbf{I}_{zz} \end{bmatrix} \tag{2.6}$$

Where the elements are defined as in Equation 2.7

$$\mathbf{I}_x = \int_V \rho(y^2 + z^2) dV \qquad \mathbf{I}_{xy} = \mathbf{I}_{yx} = \int_V \rho xy dV \tag{2.7a}$$

$$\mathbf{I}_y = \int_V \rho(x^2 + z^2) dV \qquad \mathbf{I}_{xz} = \mathbf{I}_{zx} = \int_V \rho xz dV \tag{2.7b}$$

$$\mathbf{I}_z = \int_V \rho(x^2 + y^2) dV \qquad \mathbf{I}_{yz} = \mathbf{I}_{zy} = \int_V \rho yz dV \tag{2.7c}$$

There always exists a principal inertia matrix $\mathbf{\Lambda}$ which has only diagonal terms where the diagonal terms, then become the inertia of the principal axis. Any inertia matrix can be transformed into a principal inertia matrix by rotation. This also means that any angular velocity defined in a frame that does not creating a principal inertia matrix can be transformed into this frame by a rotation. So there is no loss of generality by only considering the principal inertial in the stability analysis in section 2.1. The rotation matrix, to go from any inertia matrix to the principal inertia matrix, can be found by eigendecomposition of the inertia matrix into the rotation matrix and the principal inertia matrix as in Equation 2.8.

$$\mathbf{I} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^T \tag{2.8}$$

2.3 Gravity Gradient Torque

Gravity gradient is one of three main torques acting upon a satellite orbiting the earth. The parasitic magnetic dipole moment and drag from air resistance are the other two torques. Gravity gradient is a result of an uneven mass distribution in the satellite. Which

leads to uneven gravitational forces on the satellite creating a torque. Satellites with long deployables like an arm will therefore be affected a lot by gravity gradient torques as they have a very uneven mass distribution due to the deployables.

For the derivation of the equations for the gravity gradient we will follow [6] and [8]. The gravity gradient torque can be calculated by considering a small mass element dm on the satellite. The gravitational force acting on this from Earth is given by Equation 2.9. $\mu = GM$ and G is the gravitational constant and M is the mass of Earth. $\vec{R}$ is a vector going from the center of the Earth to the center of the satellite. $\vec{r}$ is a vector going from the center of the satellite to the mass element dm.

$$d\vec{f} = -\mu \frac{\vec{R} + \vec{r}}{|\vec{R} + \vec{r}|^3} dm \tag{2.9}$$

The gravity gradient torque is expressed by the integral in Equation 2.10a. As this integral has no closed form solution [6], an approximation using a Taylor series expression is done as it is shown in Equation 2.10b. This leads to the solution shown in Equation 2.10c.

$$\vec{g} = \int_v \vec{r} \times d\vec{f} dm = -\int_V \vec{r} \times \mu \frac{(\vec{R} + \vec{r})}{|\vec{R} + \vec{r}|^3} dm \tag{2.10a}$$

$$\frac{(\vec{R} + \vec{r})}{|\vec{R} + \vec{r}|^3} = \frac{(\vec{R} + \vec{r})}{((\vec{R} + \vec{r})(\vec{R} + \vec{r}))^{3/2}} = \frac{(\vec{R} + \vec{r})}{R^3(1 + 2\vec{R}\vec{r}/R^2 + r^2/R^2)^{3/2}} \approx \frac{(\vec{R} + \vec{r})}{R^3}(1 - 3\frac{\vec{R} \cdot \vec{r}}{R^2}) \tag{2.10b}$$

$$\vec{g} \approx -\frac{\mu}{R^3} \int_V (1 - 3\frac{\vec{R} \cdot \vec{r}}{R^2}) \cdot \vec{r} \times (\vec{R} + \vec{r}) dm = -\frac{\mu}{R^3} \int_V (1 - 3\frac{\vec{R} \cdot \vec{r}}{R^2}) \cdot (\vec{r} \times \vec{R}) dm \tag{2.10c}$$

The integral can be solved by using the fact that dm is only dependent on $\vec{r}$ so that $-\frac{3\mu}{R^3} \int_v \vec{r} dm \times \vec{R} = 0$. The remaining integral can be expressed using the inertia tensor $\tilde{\mathbf{I}}$ as shown in Equation 2.11.

$$\vec{g} = \frac{3\mu}{R^5} \int_v (\vec{R} \cdot \vec{r})(\vec{r} \times \vec{R}) = \frac{3\mu}{R^5} \vec{R} \times \tilde{I} \cdot \vec{R}) \tag{2.11}$$

Resolving all the vectors in the body frame gives rise to a new state-space model as shown in Equation 2.12. The individual components of the gravity gradient torque can be seen in Equation 2.13. The inertia matrix is a principal inertia matrix. From this it can be seen that the difference in the principal components is what creates gravity gradient torques. It can also be seen that there is a new requirement to have an equilibrium. Only one of the components of $\mathbf{R}^b$ can be none zero. This is achieved if any of the axes of the body frame is aligned with z-axis of the orbit frame. Where the orbit frame is defined so that the z-axis points directly toward the center of Earth, the x-axis is in the same direction as the velocity of the satellite and the y-axis is found by completing the right hand rule.

$$\dot{\boldsymbol{\omega}}^b_{b/n} = \mathbf{I}^{-1}(\mathbf{S}(\mathbf{I}\boldsymbol{\omega}^b_{b/n})\boldsymbol{\omega}^b_{b/n} + \frac{3\mu}{R^5}\mathbf{S}(\mathbf{R}^b)\mathbf{I}\mathbf{R}^b) \tag{2.12}$$

$$g_x = \frac{3\mu}{R^5} R_y R_z (I_{zz} - I_{yy}) \tag{2.13a}$$

$$g_y = \frac{3\mu}{R^5} R_x R_z (I_{xx} - I_{zz}) \tag{2.13b}$$

$$g_z = \frac{3\mu}{R^5} R_x R_y (I_{yy} - I_{xx}) \tag{2.13c}$$

For the stability analysis of the system we will replace the non linear kinematic equation. The quaternion parameterisation will also be replaced by Euler angels. To derive this new equations a similar approach as in [6] and [8] will be used. We will also assume to have circular orbit which means that the angular velocity between the orbit frame and the inertia frame can be expressed as $\boldsymbol{\omega}^o_{o/n} = [0 \ -\omega_c \ 0]^T$. ω_c is the angular velocity of the orbit and for a circular motion it is given by $\omega_c = \sqrt{GM/R^3}$. The angular velocity between the body and orbit frame is $\boldsymbol{\omega}^b_{b/o} = \dot{\boldsymbol{\theta}}$ and $\boldsymbol{\omega}^b_{b/n} = \boldsymbol{\omega}^b_{b/o} + \boldsymbol{\omega}^b_{o/n}$. This gives rise to the new kinematic equations given in Equation 2.14.

$$\dot{\boldsymbol{\theta}} = \boldsymbol{\omega}^b_{b/i} - \mathbf{R}^0_b(\boldsymbol{\theta})^T \boldsymbol{\omega}^o_{o/n} \tag{2.14}$$

To linerize the kinematic equation we use the linerized rotation matrix $\mathbf{R}^o_b(\boldsymbol{\theta})$ which assumes that the angles are so small that $sin\theta_i = \theta_i$, $cos\theta_i = 1$ and $\theta_i\theta_j = 0$. This leads to the rotation matrix shown in Equation 2.15. Leading to a new linear kinematic equation Equation 2.16.

$$\mathbf{R}^b_o(\boldsymbol{\theta}) = \mathbf{1} - \mathbf{S}(\boldsymbol{\theta}) = \begin{bmatrix} 1 & \theta_3 & -\theta_2 \\ -\theta_3 & 1 & \theta_1 \\ \theta_2 & -\theta_1 & 1 \end{bmatrix} \tag{2.15}$$

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} \omega_x + \omega_c\theta_3 \\ \omega_y + \omega_c \\ \omega_z - \omega_c\theta_1 \end{bmatrix} \tag{2.16}$$

The gravity gradient torque can be expressed easily in the orbit frame as z-axis $\vec{o}_z = -\vec{R}/R$ and using the relationship between R and ω_c leads to Equation 2.17. To exprec the gravity gradient torque in the body frame the rotation matrix $\mathbf{R}^o_b(\boldsymbol{\theta})^T$ is used so that $\vec{o}_z = R(1,3)\vec{b}_x + R(2,3)\vec{b}_y + R(3,3)\vec{b}_z$. Combining this with Equation 2.12 gives the new dynamic Equation 2.18, where $\mathbf{R}(3)_{b/o}$ is the third column of $\mathbf{R}^o_b(\boldsymbol{\theta})$.

$$\vec{g}_c = 3\omega_c^2 \vec{o}_z \times \tilde{I} \cdot \vec{o}_z \tag{2.17}$$

$$\dot{\boldsymbol{\omega}}^b_{b/n} = \mathbf{I}^{-1}(\mathbf{S}(\mathbf{I}\boldsymbol{\omega}^b_{b/n})\boldsymbol{\omega}^b_{b/n} + 3\omega_c^2 \mathbf{S}(\mathbf{R}(3)_{b/o})\mathbf{I}\mathbf{R}(3)_{b/o}) \tag{2.18}$$

Equation 2.16 is used to get an expression for $\dot{\boldsymbol{\omega}}^b_{b/n}$. Substituting this and Equation 2.16 into Equation 2.18 gives Equation 2.19. The assumption that $\theta_i\theta_j = 0$ and $\dot{\theta}_i\dot{\theta}_j = 0$ is also used to keep the system linear.

$$I_{xx}\ddot{\theta}_1 + (I_{xx} + I_{zz} - I_{yy})\omega_c\dot{\theta}_3 - 4(I_{zz} - I_{yy})\omega_c^2\theta_1 = 0 \tag{2.19a}$$
$$I_{yy}\ddot{\theta}_2 + 3\omega_c^2(I_{xx} - I_{zz})\theta_2 = 0 \tag{2.19b}$$
$$I_{zz}\ddot{\theta}_3 + (I_{xx} + I_{zz} - I_{yy})\omega_c\dot{\theta}_1 + (I_{yy} - I_{xx})\omega_c^2\theta_3) = 0 \tag{2.19c}$$

It can be seen from Equation 2.19 that θ_2 is decoupled from the other states and can therefore be analyzed independently of the rest of the system. It can also be seen that the equation is on the form of an harmonic oscillator. The state is therefore stable if $I_x x > I_z z$. For the two remaining states the equations can be reformulated as the form in Equation 2.20, where the matrices are defined as in Equation 2.21 and $\tilde{\boldsymbol{\theta}} = [\theta_1\ \theta_3]^T$.

$$\mathbf{M}\ddot{\tilde{\boldsymbol{\theta}}} + \mathrm{G}\dot{\tilde{\boldsymbol{\theta}}} + \mathbf{K}\tilde{\boldsymbol{\theta}} = 0 \tag{2.20}$$

$$\mathbf{M} = \begin{bmatrix} I_{xx} & 0 \\ 0 & I_{zz} \end{bmatrix} \tag{2.21a}$$
$$\mathbf{G} = \omega_c(I_{xx} + I_{zz} - I_{yy}) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{2.21b}$$
$$\mathbf{K} = \omega_c^2 \begin{bmatrix} 4(I_{yy} - I_{zz}) & 0 \\ 0 & (I_{yy} - I_{xx}) \end{bmatrix} \tag{2.21c}$$

To investigate the stability of the system the, poles of the transfer function are calculated. The transfer function is given in Equation 2.22 and the poles can be calculated by finding the s for which the determinant of $\mathbf{M}s^2 + \mathbf{G}s + \mathrm{K}$ is zero.

$$\tilde{\boldsymbol{\theta}}(s) = \frac{(\mathbf{M}s + G)\tilde{\boldsymbol{\theta}}(0) + \mathbf{M}\dot{\tilde{\boldsymbol{\theta}}}(0)}{(\mathbf{M}s^2 + \mathbf{G}s + \mathbf{K})^{-}1} \tag{2.22}$$

The determinant can be expressed as in Equation 2.23a and reduced to Equation 2.23b by dividing by $I_{xx}I_{zz}$ and inserting the new variables $k_1 = (I_{yy} - I_{zz})/I_{xx}$ and $k_2 = (I_{yy} - I_{xx})/I_{zz}$. Looking at Equation 2.23b, it becomes clear that it is a biquadratic function and can be solved by $\lambda = s^2$. This means that the only time s has none positive real parts of its roots is when $\lambda < 0$. The resulting quadratic function $\lambda^2 + b\lambda + c = 0$ is only negative following the criteria in the Table 2.2. The criteria is also translated into dependencies on k_1 and k_2 and the the criteria for stability for θ_2 is added.

$$s^4 I_{xx}I_{zz} + s^2\omega_c^2[I_{xx}(I_{yy} - I_{xx}) + 4I_{zz}(I_{yy} - I_{zz}) + (I_{xx} + I_{zz} - I_{yy})^2] + 4\omega_c^4(I_{yy} - I_{xx})(I_{yy} - I_{zz}) = 0 \tag{2.23a}$$
$$s^4 + s^2\omega_c^2(1 + 3k_1 + k_1k_3) + 4\omega_c^4 k_1 k_3 = 0 \tag{2.23b}$$

This also leads the criteria on the inertia on the principal axis as shown in Equation 2.24.

$$I_{yy} > I_{xx} > I_{zz} \text{ and } I_{yy} < I_{xx} + I_{zz} \tag{2.24a}$$
$$I_{xx} > I_{zz} > I_{yy} \text{ and } I_{xx} < I_{yy} + I_{zz} \tag{2.24b}$$

	$k_1 > k_2$
$c > 0$	$k_1 k_3 > 0$
$b > 0$	$1 + 3k_1 + k_1 k_3 > 0$
$b^2 - 4c > 0$	$(1 + 3k_1 + k_1 k_2)^2 - 16 k_1 k_3 > 0$

Table 2.2: Stability criteria for satellite when considering gravity gradient.

Chapter 3

Methods

3.1 Simulation

To test the controller and verify the stability analysis a simulation was created. The simulation consists of the rigid body kinematics and dynamics as described in section 2.1, and simulation of the gravity gradient torque as described in section 2.3. The gravity gradient torque also requires the simulation of the orbit, so an orbit propagator was created. The stability analysis was also done in relation to the orbit frame and not the inertia frame, so it was also necessary to create the kinematics for the body orbit frame relation. This was done by calculation the rotation between the inertia frame and the orbit frame and using the relation sheep $q_b^o = q_i^o \otimes q_b^i$ where $\otimes$ is the quaternion product. Both the orbit propagator and the Orbit frame calculations are discussed in more detail in later sections. An overview of the simulation can be seen in Figure 3.1.

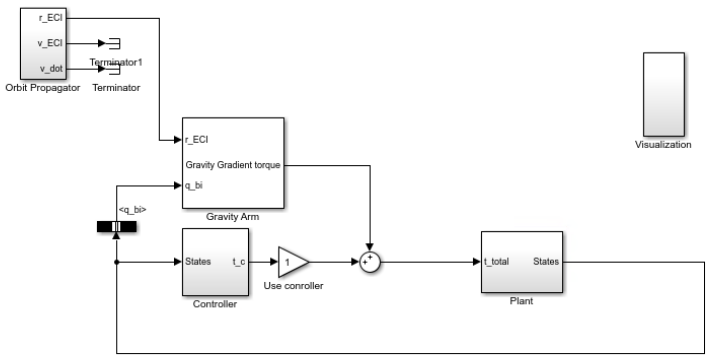


Figure 3.1: Overview of the simulation's main components and how they are connected to each other.

For the dynamic equation and the gravity gradient torque, the inertia of the satellite is needed. As the satellite has a deployable arm that changes the inertia matrix, two inertia matrices where needed. The inertia matrices where calculated based on the assumptions stated in section 1.4. For an uniform prism only four parameters are needed: height h,

width w, depth d and mass m, where the parameters are related to the coordinate frame as shown in Figure 3.2. Then the inertia matrix $\mathbf{I}$ becomes as shown in Equation 3.1, where the parameter values follows the CubeSat standard so $h = 0.20\,\mathrm{m}$, $w = 0.10\,\mathrm{m}$, $d = 0.30\,\mathrm{m}$ and $m = 2.66\,\mathrm{kg}$. It is worth noting that $d = w$, which means that $I_{xx} = I_{yy}$ and only spin around the z-axis is stable.

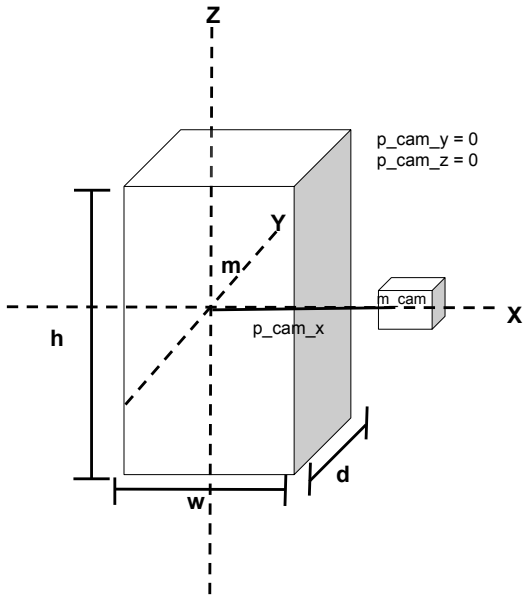


Figure 3.2: Illustration showing what the different parameters used in the calculation of the inertia matrix represent.

$$\mathbf{I} = \begin{bmatrix} \frac{1}{12}m(h^2+d^2) & 0 & 0 \\ 0 & \frac{1}{12}m(h^2+w^2) & 0 \\ 0 & 0 & \frac{1}{12}m(w^2+d^2) \end{bmatrix} \tag{3.1}$$

The inertia matrix $\mathbf{I}_{arm}$ may also contain non diagonal elements. As the camera is modeled as a single point mass, the integrals in Equation 2.6 may be replaced by sums and the inertia matrix $\mathbf{I}_{arm}$ is simply found by adding the contribution from the arm to $\mathbf{I}$ as shown in Equation 3.2. The contribution of the camera to the inertial matrix is only dependent on the position of the camera in the body frame given by $p^b_{cam} = [p_{cam_x}\ p_{cam_y}\ p_{cam_z}]^T$ and the mass of the camera m_{cam}. As the new inertia matrix may have none diagonal elements it is not given that any of the axes are stable. The frame where the axes are stable can be foun using Equation 2.8.

$$\mathbf{I}_{arm} = \begin{bmatrix} I_{xx} + m_{cam}(p^2_{cam_y} + p^2_{cam_z}) & m_{cam}p_{cam_x}p_{cam_y} & m_{cam}p_{cam_x}p_{cam_z} \\ m_{cam}p_{cam_x}p_{cam_y} & I_{yy} + m_{cam}(p^2_{cam_x} + p^2_{cam_z}) & m_{cam}p_{cam_y}p_{cam_z} \\ m_{cam}p_{cam_x}p_{cam_z} & m_{cam}p_{cam_y}p_{cam_z} & I_{zz} + m_{cam}(p^2_{cam_x} + p^2_{cam_y}) \end{bmatrix} \tag{3.2}$$

3.2 Orbit Propagator

The orbit propagator is based on a state space model for the position and translational velocity of the satellite, r^i and v^i respectively. It is assumed that the only external force acting action on the satellite is the gravitational pull from earth. No other forces are considered. The goal is to get a perfectly circular orbit because this is what is assumed in the stability proof. The state space equations can then be derived from Newton's universal gravitation law and the fact that $\dot{r^i} = v^i$. This gives rise to the state space model shown in Equation 3.3, where G and M are the gravitational constant and mass of the Earth respectively.

$$\dot{r^i} = v^i \tag{3.3a}$$

$$\dot{v^i} = -\frac{GMr^i}{||r^i||^3} \tag{3.3b}$$

There are some numerical issues with the numerical solution to the set of differential equations. The simple solution to this problem is to force the length of both vectors to stay the same as they should not change in a perfectly circular orbit. This has the down side of only allowing for perfectly circular. This makes the simulation less ideal for future testing as it is not expected that the Selfi-Sat will have a perfectly circular orbit. Another possible solution to the numerical issue might be to use a solver of higher order as the current implementation only uses "Euler's method". A different and more physical approach is to force the answers to maintain the systems energy as the system has no loss. Therefore the sum of potential and kinetic energy should stay the same at all times.

The factor that is deciding if a orbit becomes circular or not, is its initial conditions. For a circular motion the translational velocity is a tangent to the circle, meaning that the initial state of v^i should be perpendicular to the initial state of r^i. It is also a well known fact that the centripetal acceleration is given by $a = v^2/r$, and always in the direction of the circle's center. As gravity is the only source of a force in the system, the equation for acceleration in Equation 3.3 can substitute a solving for v gives the expression as shown in Equation 3.4 for the magnitude of the initial velocity for v^i.

$$||v_0^i|| = \sqrt{\frac{GM}{||r_0^i||}} \tag{3.4}$$

3.3 Orbit Frame

As discussed earlier calculating, q_b^o has great value for visualization and is needed for the controller. The rotations needed to go from the inertia frame to the orbit frame can be done in two steps. The first step is to rotate the inertia frame so that the z-axis of the inertia frame aligns with the $-r^i$. The second step is to rotate around the new z-axis so that the x-axes align.

A rotation can be described using an axis angle representation. An axis angle representation consist of a vector describing an axis that will be rotated around, and an angle

describing how much to rotate around the axis. This can be used to describe the rotation in the first step. To find the axis, the cross product can be used. As the result of a cross product is a new vector that is normal to both vectors. This vector can be used as our rotation vector in the axis angle representation. The definition of the dot product can be used to find the angle. It can be seen that $acos(a \cdot b)$ gives the angel θ. The axis angle representation can be written as in Equation 3.5, where w_1^i and θ represent the rotation axis and angle respectively and z^i and r^i are the z-axis of the inertial frame and the position of the satellite.

$$\mathbf{w}_1^i = \begin{bmatrix} w_1^i \\ \theta \end{bmatrix} = \begin{bmatrix} z^i \times -r^i \\ -\frac{r^i}{||r^i||} \cdot z^i \end{bmatrix} \tag{3.5}$$

For the rotation in step two, the same idea can be used. Here it is not necessary to find the rotation axis as this will always be the z-axis in the orbit frame z^o. z^o in the inertia frame can be expressed as $q_i^{z_o} z^i (q_i^{z_o})^{-1}$, and $q_i^{z_o}$ is the quaternion representing the rotation in step one. This gives rise to the angle axis representation as shown in Equation 3.6, where once again the dot product is used to find the angle. Where v^i and $\hat{x}$ are the velocity of the satellite, and the inertia x-axis undergone the rotation $\mathbf{w}_1^i$.

$$\mathbf{w}_2^i = \begin{bmatrix} z^o \\ \frac{v^i}{||v^i||} \cdot \hat{x} \end{bmatrix} \tag{3.6}$$

To go from an angle axis representation to quaternion representation, see the formula in Equation 3.7. The final rotation to go from the inertia frame to the body frame can then be expressed as the quaternion product of the two rotations.

$$q = \begin{bmatrix} cos(\frac{\theta}{2}) & sin(\frac{\theta}{2})w_x & sin(\frac{\theta}{2})w_y & sin(\frac{\theta}{2})w_z \end{bmatrix}^T \tag{3.7}$$

3.4 Controller

As it was shown in section 2.3, the linearized pitch dynamic is decoupled from the linearized roll and yaw dynamics. Therefore the control design is split into two controllers, one for pitch and one for roll and yaw. Both controller are PD controllers based on the linearized dynamics about the orbit frame.

3.4.1 Pitch Linear Controller

The linearized pitch dynamic is a mass-damper-spring system without any dampening. This gives rise to the harmonic oscillations seen in the open loop system. A PD controller is suitable for this type of system as it allows to the control both the dampening and the natural frequency. The linearized dynamics can be written as in Equation 3.8, where $K = 3 * \omega_c^2 (I_{xx} - I_{zz})/I_{yy}$ and τ is the control input to the system. By choosing a controller of the form in Equation 3.9 with θ_{2d} as the desired pitch angle, the result becomes a system as shown in Equation 3.10a and transfer function Equation 3.10b.

$$\ddot{\theta}_2 + K\theta = \frac{\tau}{I_{yy}} \tag{3.8}$$

$$\tau = I_{yy}(-K_d\dot{\theta}_2 + K_p(\theta_{2d} - \theta_2) \tag{3.9}$$

$$\ddot{\theta}_2 + K_d\dot{\theta}_2 + (K + K_p)\theta_2 = K_p\theta_{2d} \tag{3.10a}$$

$$\frac{\theta_2(s)}{\theta_{2d}(s)} = \frac{K_p}{s^2 + K_d s + (K + K_p)} \tag{3.10b}$$

The transfer function in Equation 3.10b can be compared to the characteristic function of a mass damper spring system $s^2 + 2\zeta\omega_0 s + \omega_0$ to find values for K_p and K_d, that depends on the natural frequency ω_0 and dampening ζ. This gives rise to the expression for K_d and K_p shown in Equation 3.11, including the divided by I_{yy}. As this controller is dependent on I_{yy} different values for K_d and K_p should be calculated depending on if the arm is out or not.

$$K_d = 2\zeta\omega_0 I_{yy} \tag{3.11a}$$

$$K_p = (\omega_0^2 - K)I_{yy} \tag{3.11b}$$

3.4.2 Roll and Yaw Linear Controller

For the roll and yaw controller it is the same principal as for the pitch controller as this is also a mass damper spring system. The only difference is that this time the states are coupled. From section 2.3 we know that the linearized equations can be written as in Equation 3.12. Choosing an controller as in Equation 3.13 gives rise to Equation 3.14a and the transfer function Equation 3.14b. As we have two states now all the constants are matrices.

$$\mathbf{M}\ddot{\tilde{\boldsymbol{\theta}}} + \mathrm{G}\dot{\tilde{\boldsymbol{\theta}}} + \mathbf{K}\tilde{\boldsymbol{\theta}} = \boldsymbol{\tau} \tag{3.12}$$

$$\boldsymbol{\tau} = \mathbf{M}(-\mathbf{K}_d\dot{\tilde{\boldsymbol{\theta}}} + \mathbf{K}_p(\tilde{\boldsymbol{\theta}}_d - \tilde{\boldsymbol{\theta}})) \tag{3.13}$$

$$\ddot{\tilde{\boldsymbol{\theta}}} + (\mathbf{G} + \mathbf{K_d})\dot{\tilde{\boldsymbol{\theta}}} + (\mathbf{K} + \mathbf{K_p})\tilde{\boldsymbol{\theta}} = \mathbf{K_p}\tilde{\boldsymbol{\theta}}_d \tag{3.14a}$$

$$\frac{\tilde{\boldsymbol{\theta}}(s)}{\tilde{\boldsymbol{\theta}}_d(s)} = \frac{\mathbf{K_p}}{\mathbf{I}_{2x2}s^2 + (\mathbf{G} + \mathbf{K_d})s + (\mathbf{K} + \mathbf{K_p})} \tag{3.14b}$$

To calculate values for $\mathbf{K}_d$ and $\mathbf{K}_d$ it can once again be done by comparing the transfer function Equation 3.14b to the characteristic equation of the mass damper spring system. If ω_0 and ζ are 2x2 matrices where element (1,1) and (2,2) representing the natural frequency and dampening for roll and yaw respectively.

3.4.3 Quaternion Based Controller

A much simpler approach is also designed where the controller is simply given by Equation 3.15. $\tilde{\mathbf{q}}_b^o$ is the imaginary parts of the quaternion product between the conjugate of the desired quaternion and the current quaternion $\tilde{\mathbf{q}}_b^o = \bar{\mathbf{q}}_d \otimes \mathbf{q}_b^o$. This is a better representation of the difference in rotation between the two quaternions[3]. A quaternion that represents rotation has to be a unit quaternion. Therefore it is enough to specify only the three imaginary parts to fully describe any orientation. $\tilde{\boldsymbol{\omega}}_{b/n}^b = \boldsymbol{\omega}_d^b - \boldsymbol{\omega}_{b/n}^b$ is the difference between the desired angular velocity and the true angular velocity.

$$\boldsymbol{\tau}_c = -\mathbf{I}(\mathbf{K}_p \tilde{\mathbf{q}}_b^o + \mathbf{K}_d \tilde{\boldsymbol{\omega}}_{b/n}^b) \tag{3.15}$$

This controller should work as long as $\mathbf{K_p}$ and $\mathbf{K_d}$ are sufficiently large compared to $\mathbf{I}$, as this is what drives the rest of the system's dynamic.

Chapter 4

Results

The results will mainly focus on verifying the linearized analysis as it was not not conclusive, looking at the effects of releasing the arm and the effectiveness of the proposed controller.

4.1 Open Loop Stability

To invest gate the open loop stability two scenarios are investigated. One for when the camera is not yet released and the second for when the camera is released and positioned along the x-axis of the body frame. This results in the two different inertia matrices I and I_{arm} shown in Equation 4.1.

$$\mathbf{I}_{arm} = \begin{bmatrix} 0.0111 & 0 & 0 \\ 0 & 0.0172 & 0 \\ 0 & 0 & 0.0106 \end{bmatrix} \mathbf{I} = \begin{bmatrix} 0.0111 & 0 & 0 \\ 0 & 0.0111 & 0 \\ 0 & 0 & 0.0044 \end{bmatrix} \tag{4.1}$$

4.1.1 Open Loop Pitch Stability

According to the linearized pitch equations from section 2.3, the pitch dynamics should form a harmonic oscillation around the equilibrium. The oscillation should have a period according to Equation 4.2. The stability of the dynamic is only dependent on $I_{xx} > I_{zz}$. As this is the case for both I and I_{arm} only the former will be investigated for the pitch.

$$T_p = \frac{2\pi}{\omega_n} = \frac{2\pi}{\sqrt{3\omega_c^2(I_{xx} - I_{zz})/I_{yy}}} \approx 1.83 \cdot 10^4 s \tag{4.2}$$

Looking at the pitch dynamics from different initial conditions as seen in Figure 4.1, it is clear that the linearized model fits very well with the nonlinear model. The dynamic behaves perfectly like a harmonic oscillator and the period is almost exactly the same as the theoretical period calculated in Equation 4.2.

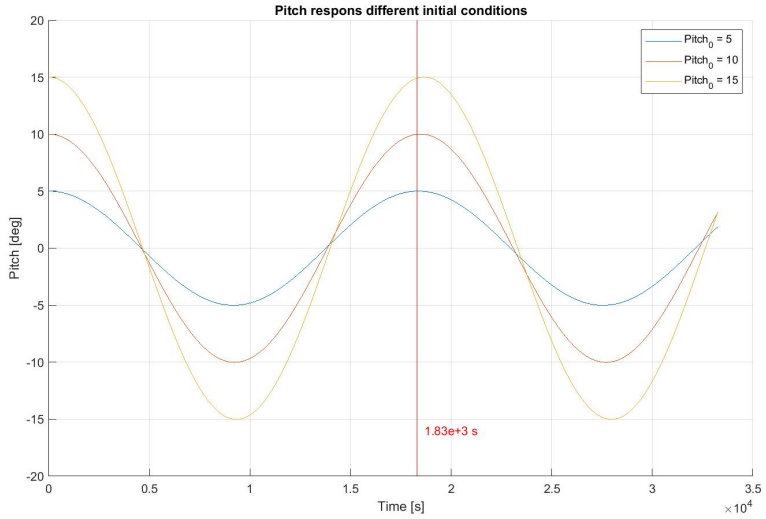


Figure 4.1: Pitch dynamics for different initial conditions 5, 10 and 15 deg. The read line marks the expected period.

4.1.2 Open Loop Roll and Yaw Stability

For the roll and yaw dynamics there are two cases, as the dynamic of the linearized system changes drastically if $I_x x = I_y y$. If $I_x x = I_y y$, it can be seen from Equation 2.19 that the dynamic of yaw θ_3 only depends on the roll θ_1. This can be used to get a second order equation for the roll as shown in Equation 4.3 where the equation is transformed to the laplace domain. This is again a harmonic oscillator as long as $I_{zz}\omega_c^4 + \omega_c^2 4(I_{yy} - I_{zz} > 0$. For the situation where the arm is not released yet, this is the case as the inertia matrix I is given by Equation 4.1.

$$\theta_3(s) = \frac{-\omega_c \theta_1(s)}{s} \tag{4.3a}$$

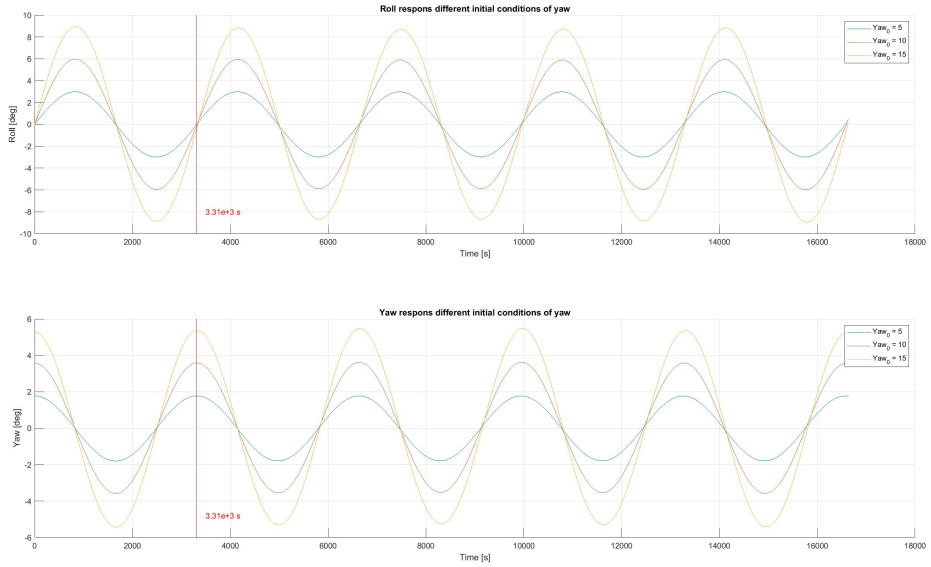
$$I_{xx}\theta_1(s)s^2 + (I_{zz}\omega_c^2 + \omega_c^2 4(I_{yy} - I_{zz})\theta_1(s) = 0 \tag{4.3b}$$

As time period can be calculated for roll using Equation 4.4. The period for the yaw will be the same as the period for the roll.

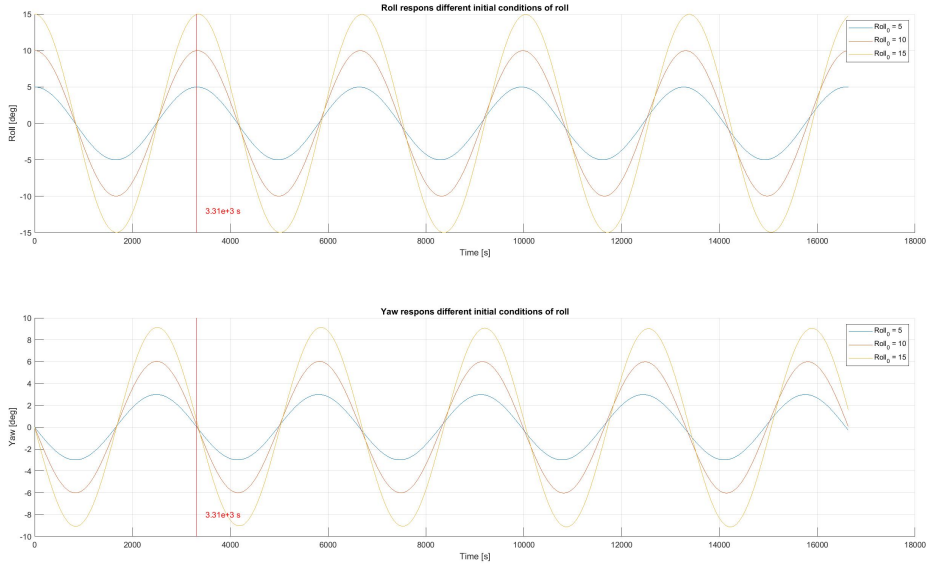
$$T_p = \frac{2\pi}{\omega_n} = \frac{2\pi}{\sqrt{\omega_c^2 I_{zz} + \omega_c^2 (I_{yy} - I_{zz})/I_{xx}}} \approx 3.31 \cdot 10^3 s \tag{4.4}$$

In Figure 4.2a and Figure 4.2b it can be seen that both roll and yaw have the expected behaviour as harmonic oscillations that are phase shifted.

When the camera arm is released the roll and yaw dynamics can no longer be understood in terms of harmonic oscillations, because of the compelling between them. They quickly become very nonlinear as shown in Figure 4.3. It also shows the the pitch dynamics are actually not decoupled from the rest of the dynamics as it can be be seen clearly in Figure 4.3.



(a) Roll and yaw dynamics for different initial conditions for yaw. The read line marks the expected period. The plots for yaw are centered around 0 deg make it more readable.



(b) Roll and yaw dynamics for different initial conditions for roll. The read line marks the expected period.

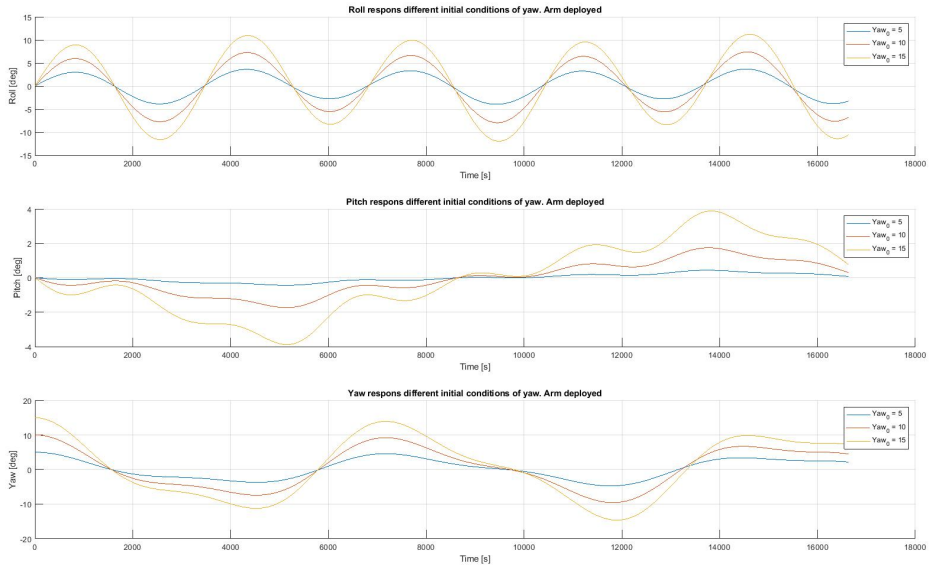


Figure 4.3: Roll, pitch and yaw response to different initial conditions when the arm is deployed.

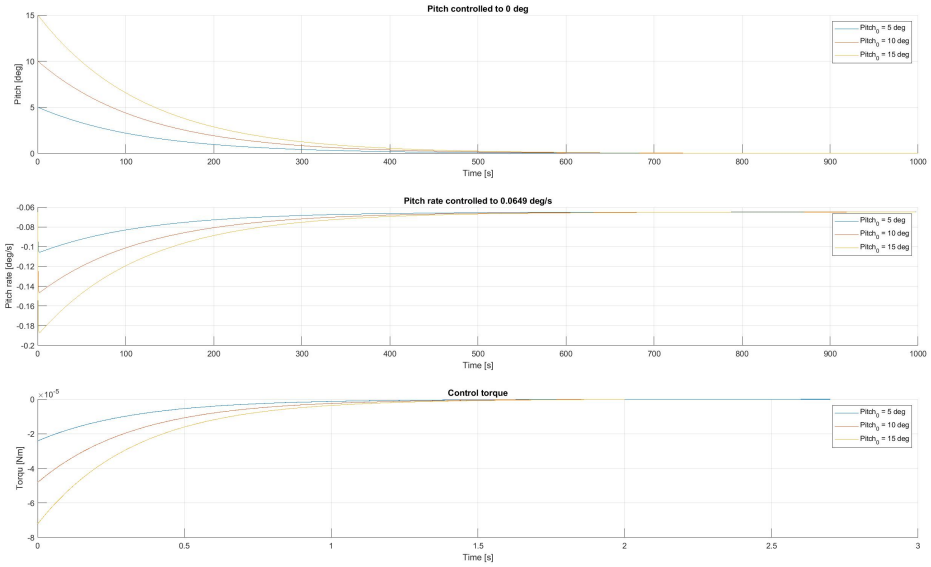
4.2 Controller

4.2.1 Pitch Control

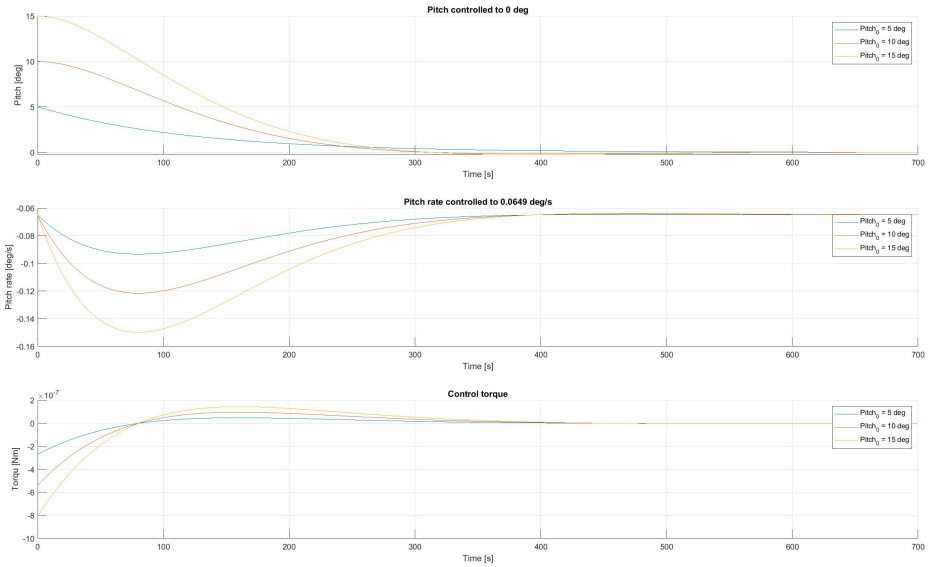
It can be seen in Figure 4.4a and Figure 4.4b the linear pitch controller works well for both cases with and without the arm out. This is not very surprising as the results shown in subsection 4.1.1 show that the linearized model is a good approximation of the nonlinear system. It can also be seen that the controller struggles a bit more when the arm is deployed and from 10° and up there is a visible overshoot. It is the easiest to see that in the control torque plot. Additionally the controller is rather slow. Asit needs around $500\,\mathrm{s}$ to reach the desired value. It can also be seen that less torque is needed from the controller when the arm is deployed. This is as expected as the gravity gradient is pulling the satellite towards the equilibrium and all the controller need to do is to damp the system.

4.2.2 Roll and Yaw Control

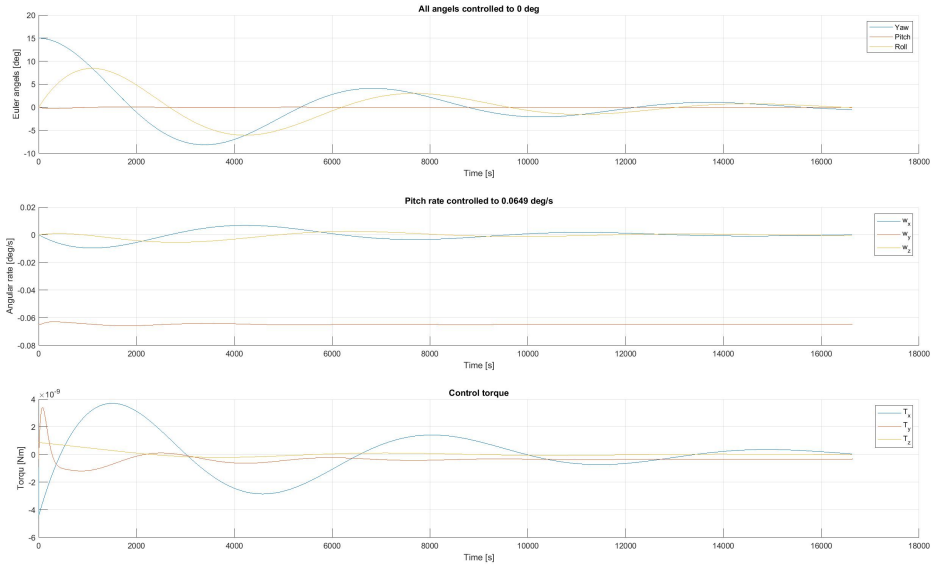
In Figure 4.4a and Figure 4.5b the controller struggles a lot more than for the pitch. The system ends up being under-damped and slowly settling towards the desired values. In both cases it has not been able to reach the desired value within 3 orbits. This is mainly because the controller had to be tuned down to be stable where $\zeta_{rool} = 3$ and $\zeta_{yaw} = 0.001$. So the roll is extremely over-damped and the yaw is extremely under-damped. There might be better values for the ζ, but it would require more investigation. Anyways, the linearization is not a good way of finding values for K_p and K_d as a lot of tuning is required for tuning.



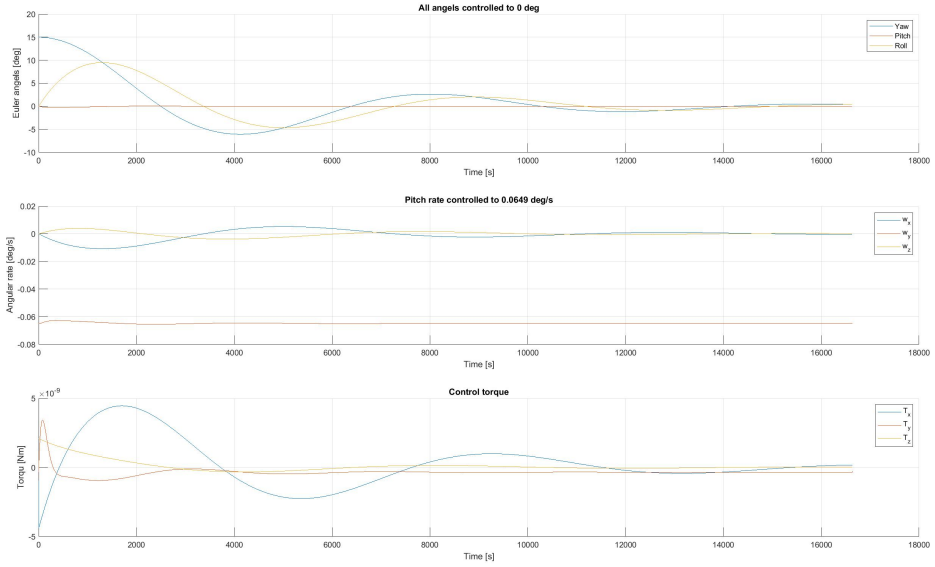
(a) Pitch controlled towards zero by linear controller. The arm is not deployed.



(b) Pitch controlled towards zero by linear controller. The arm is deployed.



(a) Roll and yaw controlled towards zero by linear controller. The arm is not deployed.

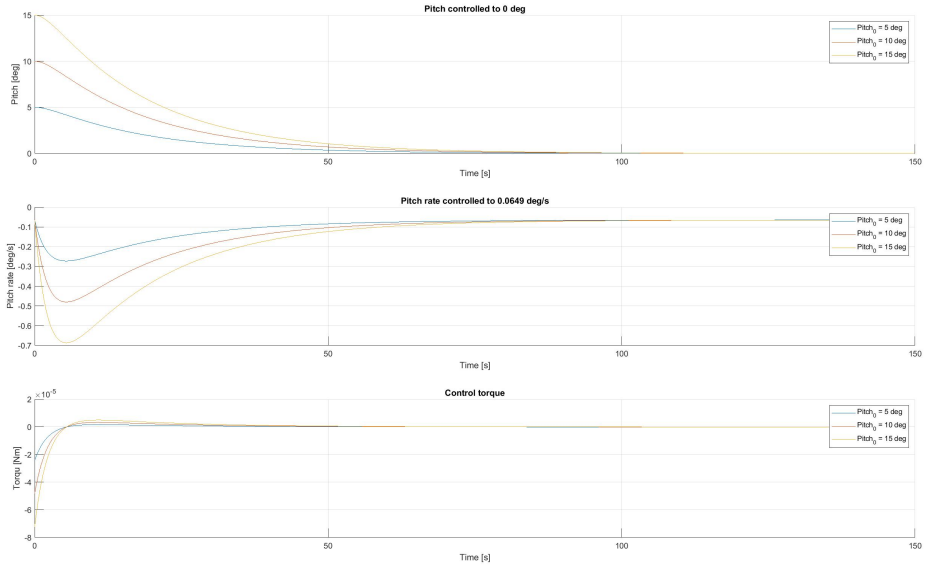


(b) Roll and yaw controlled towards zero by linear controller. The arm is deployed.

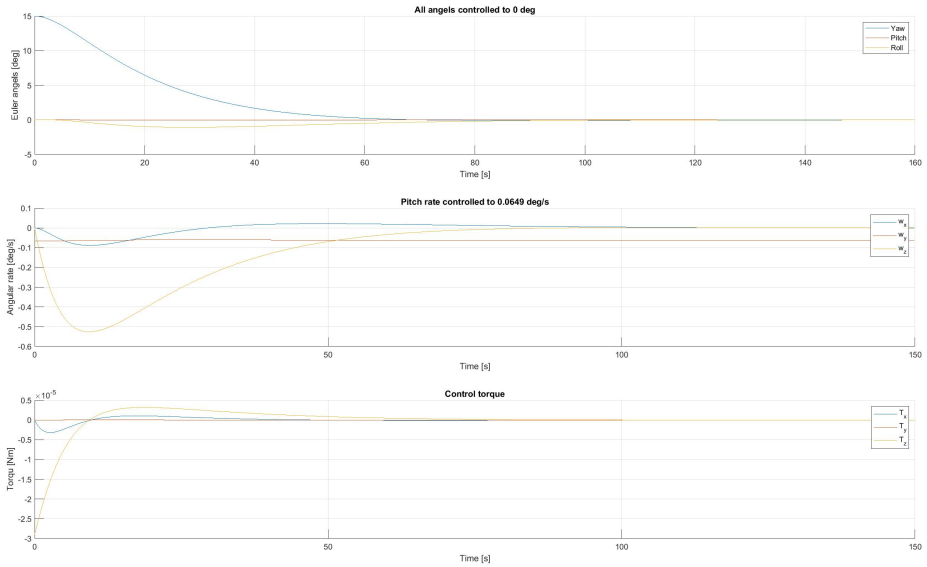
4.2.3 Quaternion Based Control

The quaternion based control shows a lot of promises for all the angles. It can be seen in Figure 4.6a and Figure 4.6b. In both cases the controller gives a good response that is much faster than the other controllers. The biggest concern with this controller is that it uses a lot of torque and it is not necessarily given that the satellite can produce such a high torque. This is not that surprising as it is designed around the controller dominating the dynamics of the system. It is therefore required that the controller produces a large torque.

As this controller is quaternion based and therefore has no singularities it can hold any desired orientation. It is shown in Figure 4.7 where the desired orientation is roll = 170°, pitch = -10° and yaw = 80°. It can be seen from the plot that there is some steady state error in the controller, but that it is reasonably fast even for such large jumps in desired orientation.



(a) Pitch controlled towards zero by nonlinear controller. The arm is deployed.



(b) Roll and yaw controlled towards zero by nonlinear controller. The arm is deployed.

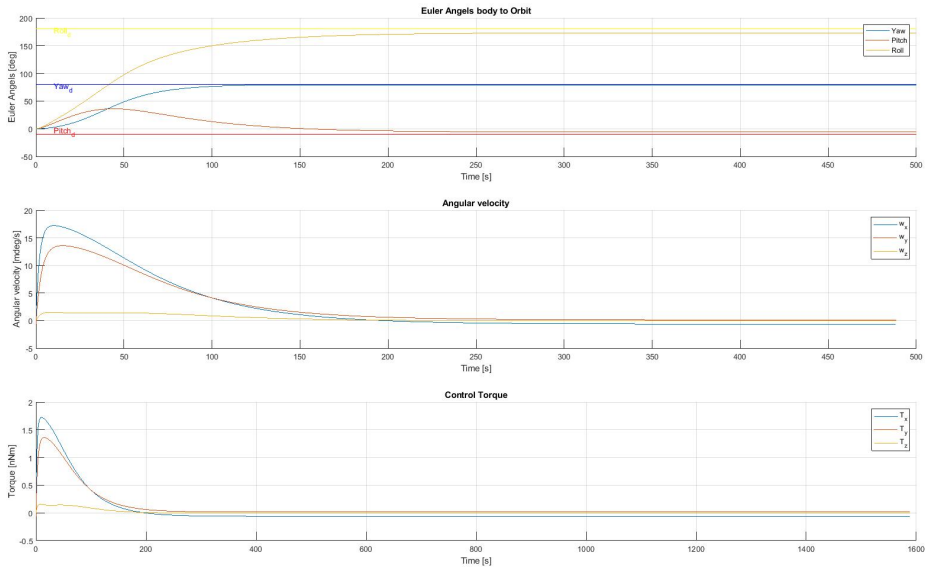


Figure 4.7: Quaternion based controller used to reach desired values. The Desiree values are shown as horizontal lines.

Chapter 5

Conclusion

As the Selfi-Sat project still is in an early phase of development, it is important that some early investigation into some of the design desertion are tested. Therefore an investigation into how the deployable camera boom would affect the dynamics of the satellite. A model of the satellite dynamic under the influence of a gravity gradient torque was created for this reason. The model was linearized for stability investigation and it was reviled that there were two stable configurations as described in Equation 2.24. This was only dependent on the inertia matrix which is in return where dependent on the position of the camera. It was also shown that the magnitude of the gravity gradient torque was dependent on the difference between the diagonal elements of the inertia matrix as shown in Equation 2.13. So it is clear that great care should be taken when placing the camera and the internal components of the satellite so that the inertia matrix is the most suitable for the mission. If the mission requires a stationary pointing towards the Earth for the most part, the design should aim to maximize the difference between the elements of the inertia matrix. This will create a large gravity gradient that will have a stabilizing effect. It was shown with the pitch controller described in subsection 3.4.1 that it was easier to stabilize the satellite around the pitch when the boom is deployed.

On the other hand, if the mission require a lot of dynamic movement, the design should aim to make the difference between all the elements of the inertia matrix as small as possible. If the goal is to have any other position than the equilibrium, the gravity gradient torque will work against the controller and drain a lot of power from the satellite.

There were also some simple attempts at controller design. They showed that though the linearized model is suitable for stability analysis, it is not very suitable for control design for roll and yaw, but for pitch it worked well. This was also reflected in how well the linearized model fitted compared to the nonlinear model. For pitch the linearized model resembled the nonlinear one much more than for roll and yaw.

The controllers also indicated what is required from the magnetorquers with maximum values in the range of $10\,\mu\mathrm{N\,m}$ while the maximum values that can be expected from the magnetorquers are in the range of $1\,\mu\mathrm{N\,m}$. This should not pose a too big problem for the project as the controllers can always be tuned down or a more optimal controller can be choosen.

Chapter 6

Outlook

Going forward there are many things that can be investigated in the future as the project is still in an early phase. To make sure that the mission is feasible from an ADCS point of view, the model should be expanded so that it can take more of the expected disturbance torque like the parasitic diplomoment and drag torques into account.

The orbit propagator could be expanded to become a more general one and the numerical efforts could be investigated further.

Some effort should also be invested into improving upon the control design in order to find something that is more energy efficient. It should also be noted that this simulation and control design does not consider the actuation that is needed to generate the torque. So a more accurate simulation of the actuation would be needed to make sure that the controller also works in real life conditions.

The current control design also assumes that it has perfect information about both attitude and angular velocity. This is not very realistic, so appropriate sensor models and estimator models should be developed to make sure that the controller can work with real inputs.

Bibliography

[1] *CubeSat Design Specification Rev. 13*. The CubeSat Program, Cal Poly SLO.

[2] Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd, Apr. 2011. DOI: 10.1002/9781119994138.

[3] F. Landis Markley and John L. Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control*. Springer New York, 2014. DOI: 10.1007/978-1-4939-0802-8.

[4] Jerrold E. Marsden and Tudor S. Ratiu. *Introduction to Mechanics and Symmetry*. Springer New York, 1994. DOI: 10.1007/978-1-4612-2682-6.

[5] NASA CubeSat Launch Initiative. *CubeSat101 Basic Concepts and Processes for First-Time CubeSat Developers*. Oct. 2017. URL: https://www.nasa.gov/sites/default/files/atoms/files/nasa_csli_cubesat_101_508.pdf.

[6] Rudrapatna V. Ramnath. *Computation and Asymptotics*. Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-25749-0.

[7] Xiwang Xia et al. "NanoSats/CubeSats ADCS survey". In: *2017 29th Chinese Control And Decision Conference (CCDC)*. IEEE, May 2017. DOI: 10.1109/ccdc.2017.7979410.

[8] Julio Zorita. "Dynamics of small satellites with gravity gradient attitude control". MA thesis. KTH, Space and Plasma Physics, 2011, p. 118.