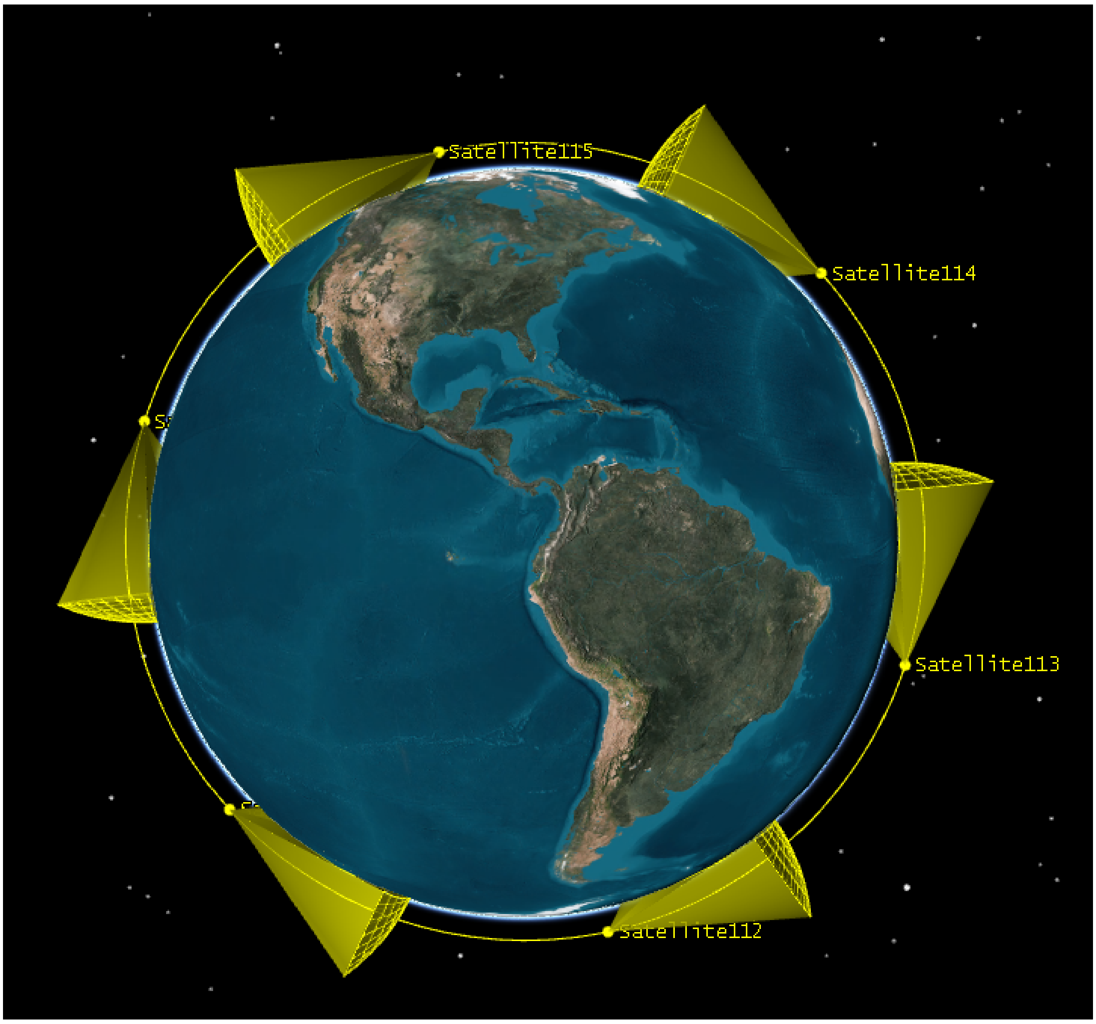
Phasing of Satellites in a Single Orbital Plane

- A Lyapunov Control Approach -

Master's thesis

CA4 - Group 1037



Aalborg University
Control and Automation

AALBORG UNIVERSITY
STUDENT REPORT

Department of Electronic Systems
Aalborg University
https://www.es.aau.dk/

Title:
Phasing of Satellites in a Single Orbital Plane

Theme:
Control Theory

Project Period:
Spring Semester 2020 03/02/2020 - 04/06/2020

Project Group:
CA4 - Group 1037

Participant(s):
Jacob Wesenberg - 20157653
Michael Juul Nielsen - 20155267

Supervisor(s):
Jesper Abildgaard Larsen
Aitor Ramirez Gomez

Copies: 1

Page Numbers: 89

Date of Completion:
June 4, 2020

Abstract:

The aim of this project was to phase grouped satellites in a single orbital plane. The project was done as a part of the research project MARIOT, which seeks to develop a satellite based maritime IoT network using VDES communication. First, a problem analysis investigating the different aspects of a satellite in orbit are presented. This included orbit parameters, different actuation methods, specifications of antennas, and some satellite constellations currently providing communication coverage around the world. Kinematics and dynamics describing a satellite with respect to another satellite in orbit were defined. The dynamics included modelling of gravitational and atmospheric drag perturbations, which was verified using the program AGI STK. A PD-, LQR-, and Lyapunov controller was implemented and tested in MATLAB/Simulink. The PD controller was stable distant to the reference, the LQR controller was stable near the reference, and the Lyapunov controller was stable all the way. The Lyapunov control was introduced to drag and remained stable. The implementation of the system was deemed successful.

Contents

Preface

This report was written by group 1037 on the 4th semester of Control & Automation at Aalborg University as a master's thesis.

Reading Guide

Throughout this report when referencing to figures this will be shown as Figure 1.3 for the third figure in the first chapter and citations will be shown as [1]. Furthermore, references to equations will be shown as Equation (1.3) for the third equation in section 1.
In equations vectors are denoted in bold as $\mathbf{V}$ while matrices are denoted in bold while also being underlined as $\underline{\mathbf{M}}$

Aalborg University, June 4, 2020

Jacob Wesenberg
<jwesen15@student.aau.dk>

Michael Juul Nielsen
<micnie15@student.aau.dk>

Chapter 1

Introduction

Sea based industries such as the seafood industry and the maritime transport sector, are massive businesses in growth. The maritime transport sector shipped 11 billion tons of cargo in 2018 alone and expecting an annual growth of 3.4% over the coming years [28].
Just as in other industries, the concept of Internet of Things (IoT) is starting to show promising application areas in the sea based industries. Start-up companies are developing products to collect and utilize data from ships throughout the industry. For example, the UK company Green Sea Guard, is developing equipment to monitor the emissions of the engines of ships. This data is to be used to minimize pollution and provide an early warning, should the ships engine begin to malfunction. In much the same way, the US company Augury is developing a sensor to be fitted on the ships machinery. This sensor will diagnose any problems with the ships equipment by analyzing the vibrations, sound, and temperature of the equipment.
Both of these products will be able to warn shore-based mechanics of problems with the ships, allowing them to be ready to deal with these as soon as the ship docks, minimizing the down time of the ships.

Similar products are being developed focusing on everything from the health of crew members, to the state of individual packages. The one thing all of these products have in common, is the need for a communication links between the ships at sea and the shore, where the data is to be utilized by companies or costumers.

One communication system currently used by ships is the Automatic Identification System (AIS). Using this system, crucial information about the ship such as identification, position, course, and speed is transmitted to other ships and shore based facilities. The information is transmitted through the Very High Frequency (VHF) maritime radio band (156.025–162.025 MHz), and is mainly used for collision avoidance.
However, since the introduction of AIS, the system has been used to transmit more

and more different types of data. The existence of a data transfer system from ship to ship, and from shore to ship, proved to be very useful in areas such as canals where information about water levels and weather could be broadcasted to ships seeking to pass through the canal.
This lead to the AIS network being overloaded, and in response the AIS system was expanded with the AIS Specific Message (ASM) channels. These have become widely used, however since only four channels exist, each channel has several uses in different parts of the world. A register of the use of each channel is maintained by the International Association of Marine Aids to Navigation and Lighthouse Authorities (IALA). However, this cannot be assumed to be up to date at all times, since the reporting of the use of ASM channels is only a recommendation. This means that if the ASM channels were to be used globally for data transfer, the channels would quickly get overloaded.

As a consequence of the already widespread use of the ASM channels, and the expected need for data exchange in the future, a new radio communication system has been developed specifically with data transfer in mind. The VHF Data Exchange System (VDES) consists of the AIS channels for collision avoidance, the ASM channels for certain data transfers, and a new set of channels for VHF Data Exchange (VDE). The VDE channels allows for the transfer of any kind of data, as opposed to the ASM channels which are only for specific predefined messages [22].

VDES is supported by IALA in their plans for implementation of the concept of E-navigation. E-navigation is a concept developed by the International Maritime Organization (IMO), who seeks to digitize information and navigational tools used when operating at sea [15].
With E-navigation, IMO seeks to standardize the design of the bridge[1] of ships around the world, utilizing VDES in the navigational tools, hence global VDES coverage is needed. VDES coverage provided by shore-based ground stations over the VHF band, covers a range of about 15-60 nautical miles or about $28 - 111km$. Furthermore, line of sight is needed for a strong reception. Hence for global coverage using the VHF band, satellites would be required [14].

The rest of this chapter will explore the problem of establishing VDES coverage using satellites, beginning with a general definition of a satellite, the variables defining its orbit, and the essential components of such a satellite.
An analysis of the state of the art in satellites will also be presented, along with the current state of the development of VDES satellite systems.
Furthermore, an analysis will be conducted to investigate the amount of satellites and orbits required to provide different levels of VDES satellite coverage under a series of given assumptions.

[1]The bridge being the room or platform from which the ship is commanded.

A common control problem to be solved in the establishment of the orbits analysed, is found to be the phasing of satellites occupying the same orbit. A problem statement describing this problem is presented and a series of requirements to a solution solving this problem are listed.

Chapter 2

Problem Analysis

A satellite is defined as a celestial body orbiting another of larger size. This definition includes natural satellites, such as the moon orbiting Earth, as well as the ones dealt with in this report; artificial satellites.
These are objects moving around Earth, performing some kind of task such as surveillance, providing communication services, or the acquisition of scientific data. How these satellites are moving around Earth is defined by the orbit in which they are placed.

2.1 Orbit

An orbit is the gravitationally curved trajectory of a celestial body. This can be seen as the trajectory of a planet around a star, or a natural satellite around a planet. Planets and satellites follow elliptic orbits, where the center of mass is orbited at a focal point of the ellipse [35].
One way of defining the orbit of a satellite, is by defining it as a Kepler orbit, which contains the 6 Keplerian elements. These elements are listed and defined in Table 2.1, and visualized in Figure 2.1. With the use of the 6 Keplerian elements, any orbit can be defined and analysed.

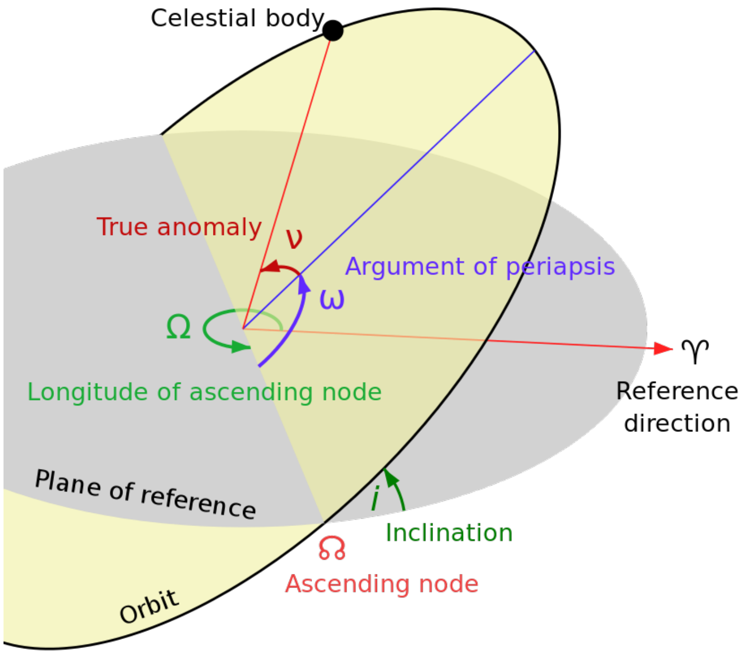


Figure 2.1: Keplerian elements of an orbit [36]

Parameter	Definition	Symbol
Eccentricity	Shape of the ellipse, elongation compared to a cicle	e
Semi-major axis	Mean of periapsis and apoapsis distance, apoapsis being the	a
Inclination	Vertical tilt of the ellipse	i
Longitude	Horizontal tilt of the ellipse	Ω
Argument of periapsis	Orientation of the ellipse	ω
True anomaly	Position of the orbiting body along the ellipse at a specific time, described at epoch (t_0)	v

Table 2.1: Keplerian elements of an orbit

2.1.1 Altitude

There are different altitude classifications for an orbit. Some of these are Low Earth Orbit (LEO), Medium Earth Orbit (MEO), and Geostationary Orbit (GEO), where the orbit have an altitude below $2000km$, from $2000km$ to $35786km$, and at $35786km$ to Earth respectively, shown in Figure 2.2. The orbit period is around 100 minutes in LEO, 12 hours in MEO, and 24 hours in GEO. This means, that a satellite in GEO has the same angular velocity as Earth, whereas a satellite in LEO has an angular velocity around 14 times greater. Due to the gravitational pull from Earth, the closer a satellite is to Earth, the higher speed is required to stay in orbit. Combining this with aerial drag and solar pressure, a satellite in orbit will lose altitude over time. This is accounted for with a propulsion system, which is described in section 2.2.

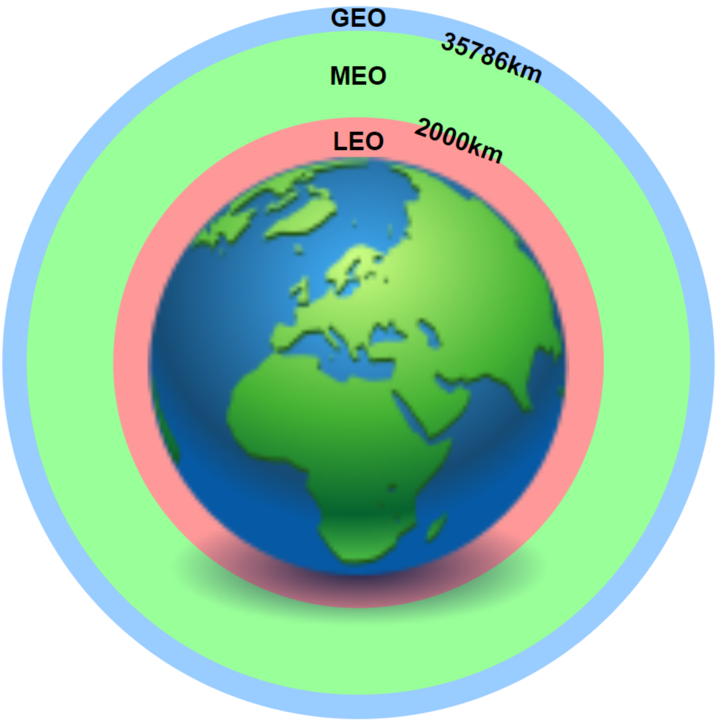


Figure 2.2: Visual representation of LEO, MEO, and GEO. Scaling is not accurate

As shown in Table 2.1, the semi-major axis of an orbit describes the altitude with respect to the center of Earth, instead of the distance to the surface of Earth. For instance, if a satellite has a semi-major axis of $6928km$ and an eccentricity of 0, it is placed in a circular LEO at around $550km$ altitude.

2.1.2 Attitude

The attitude of a satellite is the orientation with respect to an inertial frame of reference, for instance placed at the center of Earth or nearby objects. The orientation of a satellite is computed by an orbit frame and a body frame. Controlling the attitude is an essential aspect of an satellite in orbit, due to actions requiring a certain attitude to work, but can also be used to optimise the effectiveness of a task [29]. An example of this, is a satellite which antennas have to be aligned parallel to the surface of Earth in order to communicate with ground stations, or even tilting the antenna to increase the area of coverage.
Attitude control is not in the scope of this project, and will therefore not be investigated further.

2.1.3 Constellation

When multiple satellites are working together as a system, for example to provide global or near-global signal coverage, the satellites can be defined as a satellite constellation. This is achieved by having a single or more orbital planes and ground stations distributed globally. For a constellation in a single orbital plane, each satellite will have connection to different ground stations at different times, and the satellites will have the same true anomaly only with a constant time lag. From a user's point of view, when considering global coverage, at least 1 satellite can be communicated with at all times, no matter where the user is located.
By having a constellation with a single orbital plane, specific regions can have permanent coverage, and by having multiple orbital planes, permanent global coverage can be achieved. Depending on the design of the constellation and the satellites, multiple orbital planes can also be utilized to cover only specific regions.

A method of designing a constellation, is the Walker Delta Pattern, which designs the satellites to have similar orbits, eccentricity, and inclination. This is done, so that any perturbations will affect the satellites similarly, and by having circular orbits, the altitude is constant.
The Walker Constellation has the notation:

$$i : t/p/f \tag{2.1}$$

where i is the inclination, t is the total number of satellites, p is the number of equally spaced planes, and f is the relative spacing between the satellites in adjacent planes. An example of this, is the Galileo Navigation System, which is a 56°:24/3/1 constellation [40]. This means that the system consists of 24 satellites in 3 planes, inclined at 56 degrees, as shown in Figure 2.3.

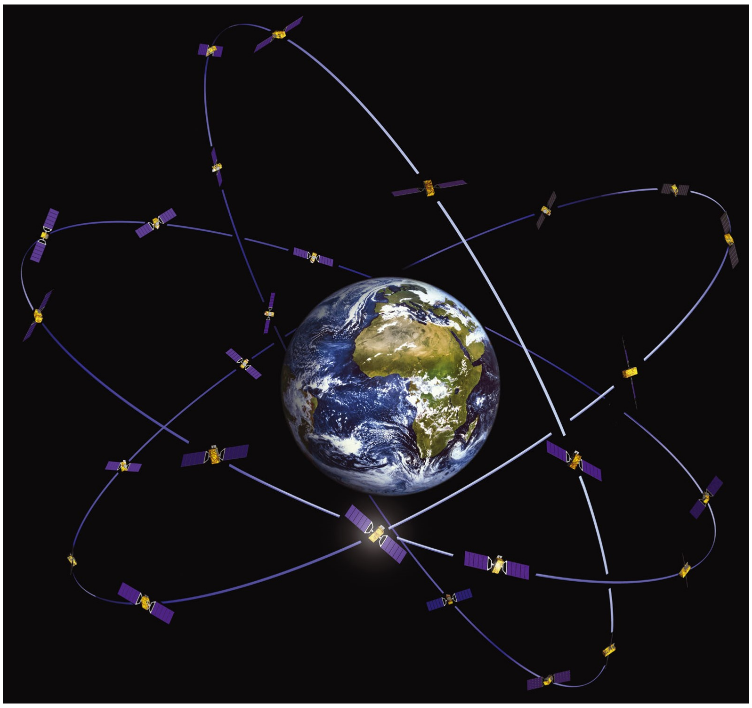


Figure 2.3: Visual representation of the Galileo Navigation System, a 56°:24/3/1 Walker Delta constellation [40]

Satellites are placed in their desired orbits by launch vehicles, which takes them from the surface of Earth, to the desired altitude, where they are placed at the required velocity to maintain this altitude as they orbit Earth.
However, the placement of the satellite by the launch vehicle is not always exact. Therefore, onboard actuation of the satellite is often necessary to reach and maintain the desired orbit and attitude. Different kinds of actuation of satellites are presented in Section 2.2.

2.2 Orbit Actuation

This section will describe different actuators for linear control of a satellite. This is used for orbital control, namely for raising or lowering the orbit by adjusting the velocity of the satellite.
When launching a satellite into space, a predefined orbit is desired. In order to achieve this orbit, different maneuvers are done. Initially, a satellite is typically spinning fast when entering space, and needs to be slowed down to a stable orientation. This is accomplished by using attitude actuators, described in Appendix A. When the satellite is oriented correctly, the next step is to change the altitude to achieve the correct orbit. This is accomplished by using a propulsion system, which will be described in this section.
The predefined orbit is now achieved, and the attitude actuators and propulsion system is used throughout the life cycle of the satellite, in order to maintain correct attitude and altitude.

To increase and decrease the altitude of a spacecraft, the speed of the spacecraft has to be increased or decreased respectively. This is done with a propulsion system, which in most cases, consumes fuel to produce a force, causing an acceleration, and thereby adjusting the altitude. Combining this with attitude control, the spacecraft can be oriented accordingly before performing an adjustment in speed. The orbital maneuvers done by actuating the spacecraft which are needed to keep a spacecraft in orbit, are called orbital station-keeping.
Propulsion systems each has their own advantages and disadvantages. An equation calculating the term delta-v, Δv, is a measure of the impulse per unit of mass. It is a scalar, which is calculated by exhausting the entire usable propellant of a spacecraft in a straight line in free space, where the velocity change to the vehicle is Δv [41].
This section will cover some of the typical propulsion systems for orbital station-keeping, which is the action of maintaining the current orbit.

2.2.1 Combustion Engine

A rocket engine is a chemical thruster, which uses stored rocket propellant as reaction mass, for producing a high speed propulsive jet of fluid to produce torque. A combustion of reactive chemicals, such as jet fuel, is used for producing torque. Currently, rocket engines are used for most launches, since they are very light and provide a high thrust, however, they have a low specific impulse, which is how effectively a rocket uses propellant. Rocket engines are also frequently used in space to control both altitude and attitude of a spacecraft.

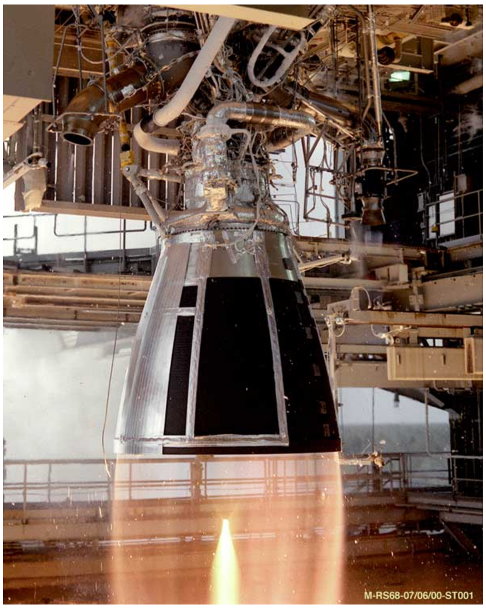


Figure 2.4: Rocket engine RS-68 being tested at NASA's Stennis Space Center [39]

2.2.2 Gridded Ion Thruster

An ion thruster is an electric propulsion system, which uses electricity to accelerate ions to generate thrust. It extracts electrons out of atoms to ionize a neutral gas, which creates a cloud of positive ions. Electrons are stored and re injected in the cloud of positive ions by a neutralizer, so the gas becomes neutral again. This is the main difference between ion thrusters and plasma thrusters, since plasma thrusters accelerate all species, free electrons as well as positive and negative ions, in the same direction whatever their electric charge is. Ion thrusters can only be used in the vacuum of space.

The Deep Space 1 spacecraft which used ion thrusters, changed its velocity by $4.3 km/s$ by consuming less than $74 kg$ of xenon [32].

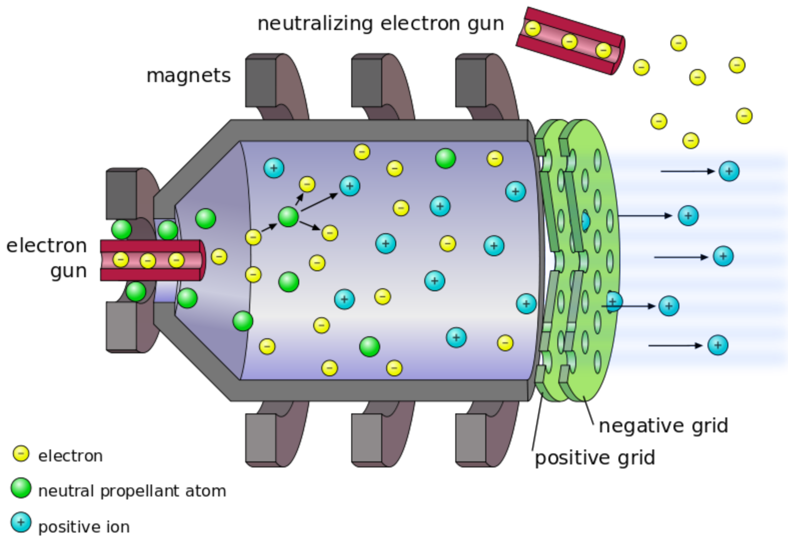


Figure 2.5: Ion thruster and how it produces a torque [31]

Ion thruster are among some of the most efficient propulsion systems, as they offer a high Δv value [32].

2.2.3 Plasma Thrusters

A plasma thruster is an electric propulsion system, which generates thrust from a quasi-neutral plasma, consuming gas in the process. Unlike gridded ion thrusters, plasma thrusters do not use high voltage grids to accelerate the charged particles in the plasma, resulting in a lower exhaust velocity. Quasi-neutral means that in the plasma exhaust, there exists an equal number of ions and electrons.
Just like ion thrusters, one of the main advantages is the high efficiency the system offers, while being able to operate in small time instances many times. Plasma thrusters are still limited to the laboratory, and have only been used on 2 spacecrafts until this day [37].

2.2.4 Propellant-less systems

In the vacuum of space, there are magnetic fields, gravitation fields, solar wind, solar radiation, and electromagnetic waves. These elements can be used to accelerate and orientate a spacecraft in space without the use of propellant. Some examples of such systems, are solar sails, tether propulsion, magnetic sails, and E-sails [42]. This section will focus on solar sails, because it has the most development and have been proven to be applicable.
A solar sail is a propulsion system for spacecrafts, which uses the radiation pressure from sunlight to generate thrust. It uses large mirrors to capture the radiation

pressure and has no moving parts or propellant. It is typically used to raise the orbit of a satellite and for interplanetary travel. NASA launched their first satellite with solar sail in LEO in 2010, with a mission called NanoSail-D2. It consisted of a CubeSat measuring 30x10x10 centimeters weighting 4kg, equipped with a solar sail with an area of $10m^2$. It was expected to remain in orbit for 70 to 120 days, but re-entered the atmosphere after 240 days.

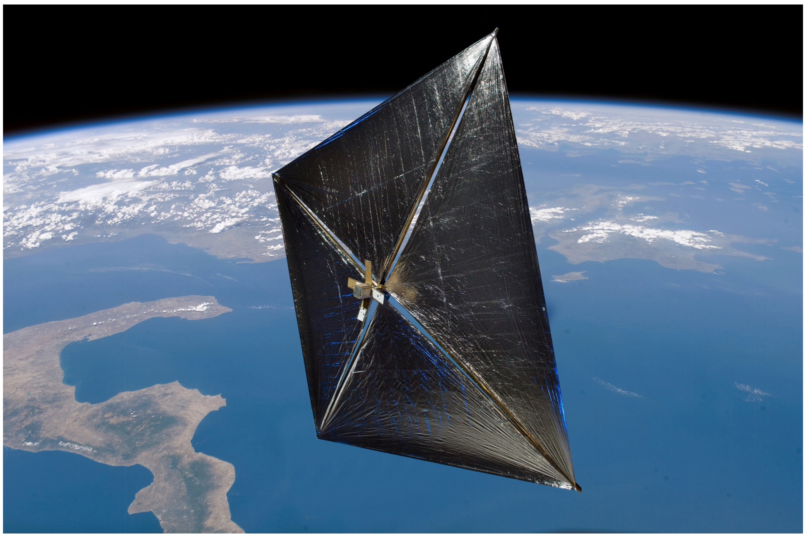


Figure 2.6: The NanoSail-D2 satellite [34]

This project is part of a research project, described in Section 2.4.4, which proposes the use of an ion thruster for orbit control. Therefore, no further analysis will be done within the scope of orbit actuation.
The dynamics of orbit actuation will be investigated in Section 3.2.

2.3 Antennas

Aside from actuators, one of the main parts of a communicative satellite is the antenna. This is the hardware that allows the satellite to receive and broadcast radio signals at a desired radio frequency.
An antenna acts as the interface between electric currents moving through a metal conductor, and electromagnetic waves moving through free space [13]. When receiving electromagnetic waves, the antenna is known as a receiving antenna. When sending electromagnetic waves, the antenna is known as a transmitting antenna. Often an antenna is used for both reception and transmission. Additional information about the basics of antenna theory can be found in Appendix B.
An important thing to consider when designing an antenna is the radiation pattern of the electromagnetic waves emitted or received by the antenna.

2.3.1 Radiation Pattern

For many applications it is often desirable to aim the signal in a given direction especially for satellite applications. In order to aim the signal, it is necessary to shape the pattern in which it radiates from the antenna.
This is done by describing the power radiated from the antenna as a function $U(\theta, \phi)$. Where θ and ϕ is shown in Figure 2.7. In this figure, the antenna is pointing in the z-direction. The direction in which the antenna is pointing, is commonly known as the boresight of the antenna. Hence, θ is the angle from boresight, also known as the elevation, while ϕ is the angle about boresight, also known as the azimuth [6].

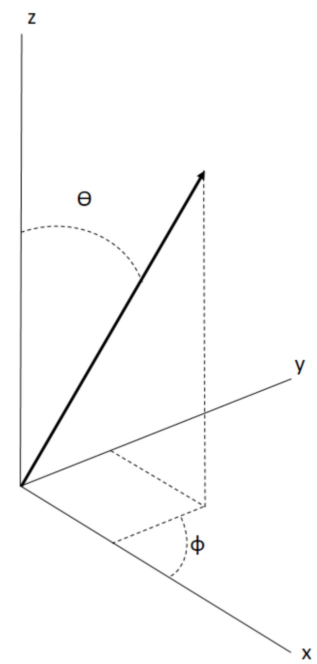


Figure 2.7: Angles of the radiation pattern

The amount of power radiated in any given direction, is dependent on the strength of the signal emitted from the antenna. To get a more useful measure of the antennas ability to direct power, the **directive gain** $D(\theta, \phi)$ is determined. This describes the measure between the power radiated in a particular direction, against the average value over all directions as seen in Equation (2.2):

$$D(\theta, \phi) = \frac{U(\theta, \phi)}{P_{rad}/4\pi} \tag{2.2}$$

Where $P_{rad}/4\pi$ is the total power radiated averaged over a sphere. When describing the specifications of an antenna, the entire directive gain function is not commonly presented. Instead, the maximum value of the directive gain is stated as the **directivity** of the given antenna. Thereby giving a measure of how well the antenna directs the energy in the main direction of use.
While directive gain describes the antennas ability to direct power, this does not give a measure of how much of the energy supplied to the antenna is actually radiated. To get a measure of this the **radiation efficiency** η_e of the system is described as seen in Equation (2.3):

$$\eta_e = \frac{P_{rad}}{P_{input}} \tag{2.3}$$

However, the really useful description of an antenna is these two combined, as this will give a measure of how effectively the antenna converts input power into radio waves headed in a given direction. This is described through the antenna **gain** $G(\theta, \phi)$ as seen in Equation (2.4):

$$G(\theta, \phi) = \eta_e \cdot D(\theta, \phi) \tag{2.4}$$

As with directive gain, the gain stated when commonly specifying an antenna is the maximum gain.
Gain is expressed in $decibels - isotropic$ or dBi, as the antenna is compared to an ideal antenna with an isotropic radiation pattern, where power is radiated equally in all directions about the antenna. Alternatively, the gain can be expressed in dBd if the radiation pattern is compared to that of a lossless half-wave dipole.
To illustrate the gain of an antenna, a polar radiation plot is used. An example of a polar radiation plot of a dipole antenna is seen in Figure 2.8.

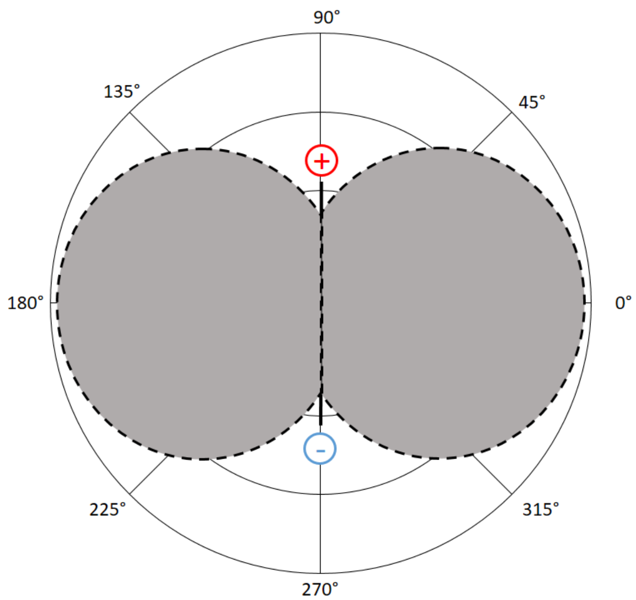


Figure 2.8: The radiation pattern of a simple dipole antenna.

This gives a visual indication of how the antenna directs the signal. For applications where high directivity is needed, such as in satellites, the challenge is to design the antenna, such that this plot is much more directive as seen in Figure 2.9

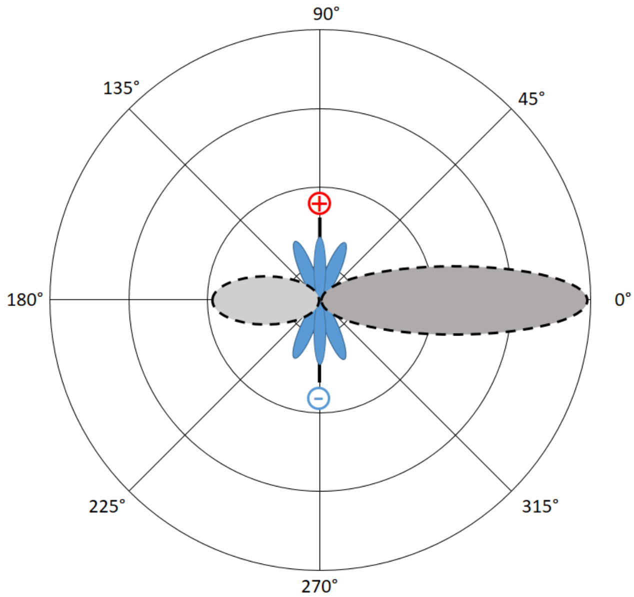


Figure 2.9: A directed radiation pattern

When the pattern is directed such as this, it can be divided into several lobes. The main lobe with the greatest field strength, a secondary back lobe radiating in the opposite direction of the main lobe, and residual side lobes acting perpendicular to the main lobe.
Several types of antennae can be used to achieve high gain, however, one that has given promising results when used in small-sat applications is the Yagi-Uda antenna [5] [3]. A short description of this kind of antenna is found in Appendix B.
Now that some of the main parts of a satellites have been presented, the state of the art of communications satellites will be investigated to get an idea of how satellites can be used to establish VDES coverage.

2.4 State of the art

Several satellite constellations are already providing communication coverage around the world. Most of these focus on providing coverage around the equator and other populated areas.

2.4.1 Inmarsat-4 Constellation

One of these constellations is owned and operated by the British company Inmarsat, who currently has 13 satellites in geostationary orbit [16]. These satellites provide services for maritime and aviation safety, telecommunication, and data transmission over the L-band (1-2 GHz), Ka-band (26.5-40 GHz), and inflight Wi-Fi

over the S-band (2-4 GHz). Hence the satellites from Inmarsat operates in a very broad spectrum, providing a host of services.
As an example, the Inmarsat-4 constellation provides global 3G mobile network using 4 geostationary satellites. Each satellite covers a large area of the planet using 19 wide beams and more than 200 narrow beams. The coverage of the Inmarsat-4 constellation can be seen in Figure 2.10

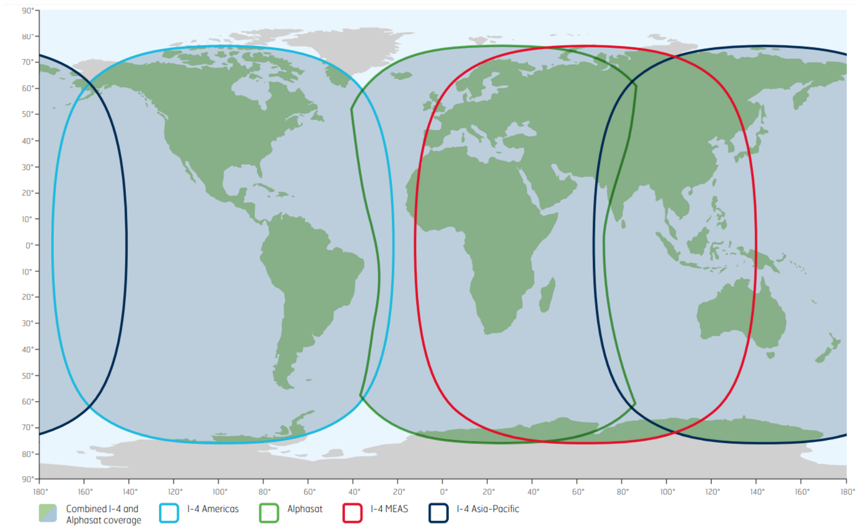


Figure 2.10: The coverage of the Inmarsat-4 constellation [16]

The fourth and latest addition to the constellation, was the Inmarsat-4A F4 satellite, also known as Alphasat, which was launched in 2013. The satellite provides coverage over Europe, the Middle East, and Africa. Fully fueled, it weighs $6650kg$ and operates in an orbit with an altitude of about $35770km$ [10]. It has an orbital period of 23.9 hours making it roughly geostationary.
As seen in Figure 2.10, the coverage of the Inmarsat-4 constellation is not completely global. Even though it does cover most populated areas of Earth, the poles remain without coverage. However, in these areas, coverage is provided by another constellation of satellites, namely the Iridium constellation.

2.4.2 Iridium Next Constellation

The first Iridium satellite constellation was deployed in 1997-2002. The constellation was established to provide coverage in areas with no or limited terrestrial coverage and in disaster areas, where the existing network has been rendered inoperable [12].
The constellation consisted of 95 satellites where 66 of them are operational, and the rest are kept as spares, most of which have now decayed or intentionally been

deorbited since the original Iridium constellation was replaced by the Iridium Next constellation in 2018.

The Iridium Next constellation consists of 66 operational satellites, nine spares in orbit, and six spares on ground [17]. The operational satellites operate in six near polar LEO, with an altitude of $780km$. As compared to the geostationary satellites of Inmarsat being at an altitude of $35770km$, this is much lower. By having a lower orbit, the Iridium satellites achieve a stronger signal and a lower transmission time, reducing the latency of the connection.
Each Iridium Next satellite has 48 spot beams operating in the L-band, each of which has a footprint with a diameter of about $400km$. This results in a total footprint with a diameter of $2500km$ for each satellite [17]. This allows the Iridium Next constellation to cover the entire surface of Earth at all times. However, since the satellites are in near polar orbits, the coverage is best at the poles as can be seen in Figure 2.11.

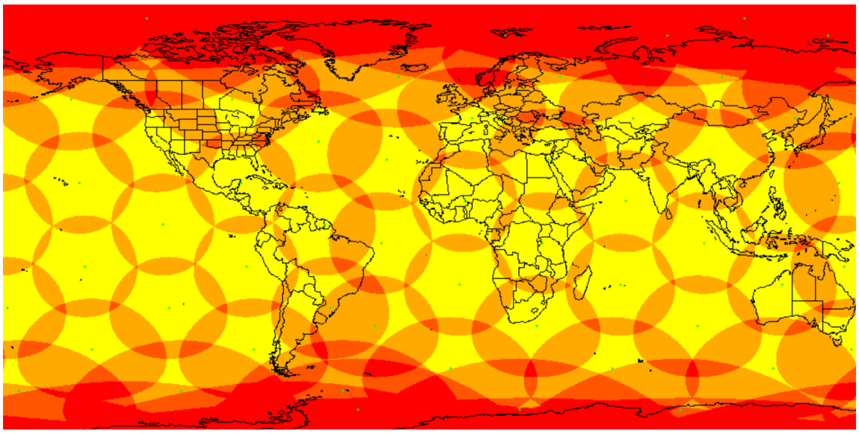


Figure 2.11: A still-image of the coverage provided by the Iridium constellation [43]

The satellites travel at a speed of about $27000km/h$, giving them an orbital period of about 100 minutes. Each satellite is connected to four other satellites across the Ka-band at all times, two adjacent satellites, and the one in front and behind it in the same orbit. Each satellite has a weight of $860kg$ at launch [24].

The Iridium Next constellation will provide a significant upgrade to the Iridium network. However, one of the more interesting additions to the satellites, is a $50kg$ slot assigned for hosted payloads. This allows other companies to "buy a seat" on the constellation, adding their own technology to the Iridium constellation. This opportunity was seized by Canadian company ExactEarth, who in partnership with the Harris Corporation, added modules covering the maritime VHF band to 58 of the satellites in the constellation [20]. Using these satellites, ExactEarth seeks to deliver global AIS connectivity with a revisit time as low as 1 minute. As of now, ExactEarth has presented no plans to implement VDES connectivity to their satellite constellation, however, they do state that the constellation do support future evolution of the AIS network, including VDES [21].

While the Iridium Next constellation might be included in a global VDES network in the future, other companies seek to do the same with much smaller and cheaper satellites.

2.4.3 NorSat-2

Norway holds control of about two million square kilometers of sea. In order to surveil and maintain these waters, for instance in connection with search and rescue missions, Norway is developing satellite communications systems using microsatellites.
Currently Norway has four microsatellites in orbit, with the main objective of tracking ships in the Norwegian waters using the AIS network [25]. In 2017, NorSat-2 was launched as the latest part of this constellation.
NorSat-2 is a microsatellite, seen in Figure 2.12, built by the Space Flight Laboratory (SFL) of the University of Toronto, Canada for the Norwegian Space Center (NSC). Aside from tracking vessels through AIS, NorSat-2 is the first satellite to carry a VDES payload which is being used to test an experimental VDES communications module developed by Kongsberg Seatex [5].

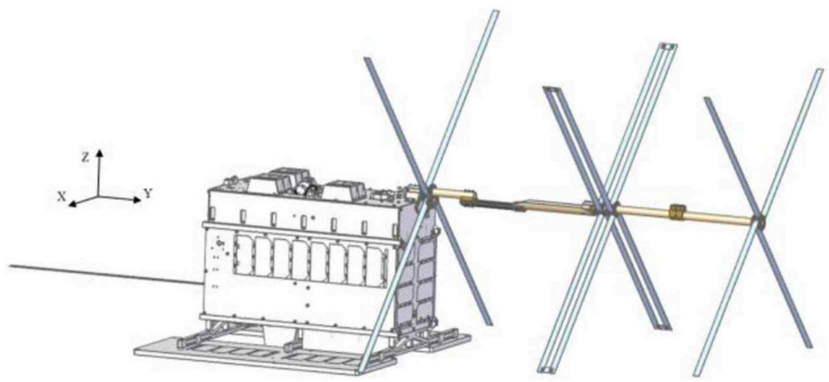


Figure 2.12: The fully deployed NorSat-2 satellite [5]

The satellite has a size of $20x27x42cm$, with two pre-deployed solar panels with a size of $20x50cm$. The communication modules of the satellite consists of an AIS receiver using a deployable AIS antenna, a VDES-transceiver using a deployable three-element eight dBi crossed Yagi antenna, and an S-band feeder uplink. The satellite is mounted with an inspection camera to observe the deployment of the antennae.
The satellite has a total mass of $15.6kg$, and is based on the NEMO (Next-Generation

Earth Monitoring) bus design by SFL. It operates in a sun synchronous polar orbit at an altitude of $600km$.

NorSat-2 has three on-board computers:

- The Housekeeping Computer (HKC)
  The HKC takes care of the basic operations of the spacecraft such as managing the collection of telemetry from the various subcomponents of the satellite, activating the radio communications modules, and sending commands to the other on-board computers.

- The Attitude Determination and Control Computer (ADCC)
  The ADCC is the control module which ensures that the satellite attains the correct attitude, thereby pointing the antennae in the desired direction. This is done by first determining the attitude of the spacecraft by assessing data gathered from on-board sensors. The NorSat-2 has six fine sun sensors found on the hull of the satellite, a three-axis rate sensor, and a three-axis magnetometer.
  Once the current attitude is determined, the spacecraft is actuated in accordance to one of several control laws. For actuation, the NorSat-2 has three orthogonal reaction wheels and three orthogonal magnetorquers, where the magnetorquers are mainly used to desaturate the reaction wheels.

- The Payload Computer (POBC)
  The POBC controls the payloads of the satellite, namely the inspection camera, AIS communication, and the VDES payload.

As the NorSat-2 orbits Earth, it receives AIS signals from ships and forwards the signals to ground stations at Vardø and Svalbard in northern Norway. The NorSat-2 carries a new generation AIS receiver, which has shown a significant increase in the ability of the satellite to detect vessels through AIS, as compared to the previous generation found on the AISSat satellites, which are the predecessors of the NorSat satellites [9].
As previously stated, the NorSat-2 also carries a VDES payload. This is specifically designed to test the downlink capabilities of VDES in a realistic setting. To do this, a VDES receiver was installed on the Norwegian Coast Guard inspection vessel the KV Harstad. As of November 2018, VDES signals were received by the KV Harstad 103 times, proving the capability of the satellite component of VDES in a realistic setting. Furthermore, an ice-chart was successfully transmitted to a receiver at the headquarters of Kongsberg Seatex in Trondheim proving the possibility of data transference across satellite VDES [11].
The antenna used by the NorSat-2 for VDES-transmission, is a deployable crossed three-element Yagi-Uda antenna with a size of $62x62x73cm$ developed and tested by the Space Flight Laboratory of the University of Toronto.

Through testing, the antenna was found to have a radiation pattern as seen in Figure 2.13:

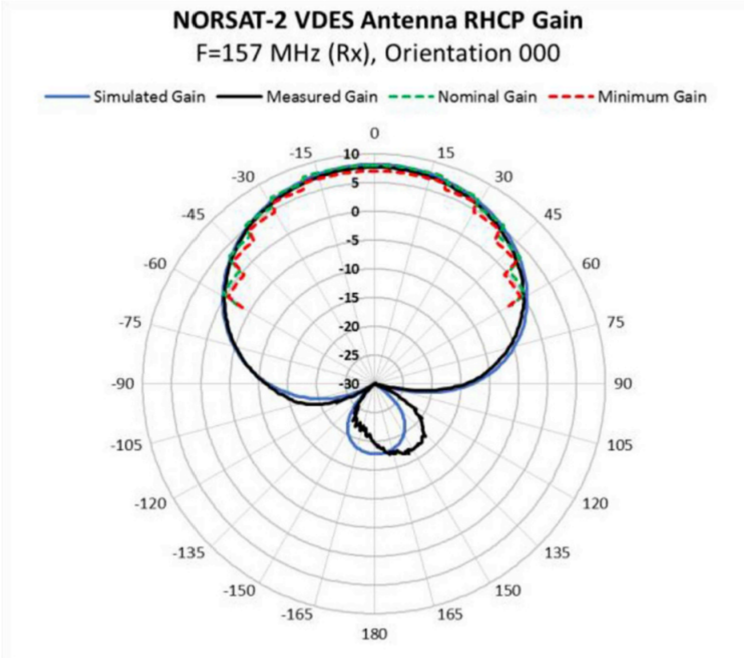


Figure 2.13: The radiation pattern of the Yagi-Uda antenna of the NorSat-2 [5]

Furthermore, tests concluded that the antenna was able to radiate enough power for proper functioning of VDES transmission at offsets in elevation and azimuth of up to 30° [5].

2.4.4 MARIOT

With the promising results of NorSat-2, a host of Danish companies have initiated a research project known as MARIOT. Lead by the company Sternula, MARIOT seeks to develop a satellite based maritime IoT network using VDES communication.

The MARIOT project is partially funded by Innovation Fund Denmark, and initially seeks to provide communications in Danish maritime areas where there are currently little to no existing VHF coverage. These areas mainly exist in the polar regions around Greenland. Hence, the first satellite to be launched in 2022, MARIOT-1, will be placed in a polar orbit to provide coverage in these areas.

MARIOT-1 will be used to demonstrate selected maritime services, such as the transmission of ice charts to, and from ships in the North Artic region.

Following the initial launch of MARIOT-1 the project will launch additional satellites, expanding the constellation with a goal of providing global VDES coverage

for maritime IoT solutions.
Starting with a very limited amount of satellites, Sternula seeks to maximize the resulting VDES coverage. Assuming the satellites of the Sternula constellation will have a satellite footprint like the one of the NorSat 2 satellite, the VDES coverage provided by different constellations will be analysed in the section below.

2.5 Orbit Analysis

A satellite constellation roadmap provided by Sternula, shows four generations of satellite constellations, which each consists of a number of satellites and a corresponding guaranteed revisit time in either the Arctic waters or globally. The roadmap is shown in Figure 2.2. The Arctic waters are defined as Earth between $60°$ to $-90°$ and $-60°$ to $-90°$ latitude. Global revisit time is anywhere on Earth, which will not be investigated.

Generation	**Satellites**	**Guaranteed revisit time**
0	1	95min in Arctic waters
1	4-6	25min in Arctic waters
2	16-20	15min global
3	40-50	Global Realtime

Table 2.2: Satellite constellation roadmap by Sternula.

This section will investigate the revisit time of Arctic waters, using Walker Delta constellations. In order to do so, the program AGI STK is used, where a satellite with a sensor attached will be used as a seed for creating a Walker constellation, and a facility will be placed at 90, 85, 80, 75, 70, 65, and 60 degree latitude. Revisit time will be a measurement of the maximum time between communication between a facility and any sensor in the constellation, in a period of 24 hours. This means that the revisit times presented in this section, is a worst case scenario, and the minimum and average revisit time will not investigated. This is due to the criteria set by the roadmap, which guarantees a revisit time.
The revisit time will be computed for one, two, three, and four orbital planes, with an increasing amount of satellites, until constant coverage is provided for the facility on 90 degree latitude, namely the North pole.

First off, a satellite with a sensor attached is created. The sensor represents an antenna with a half cone angle of 30°, limited to a range of 3500km, as show in Figure 2.14. The sensor is tilted at a 67° angle with respect to nadir pointing, being towards the center of Earth.

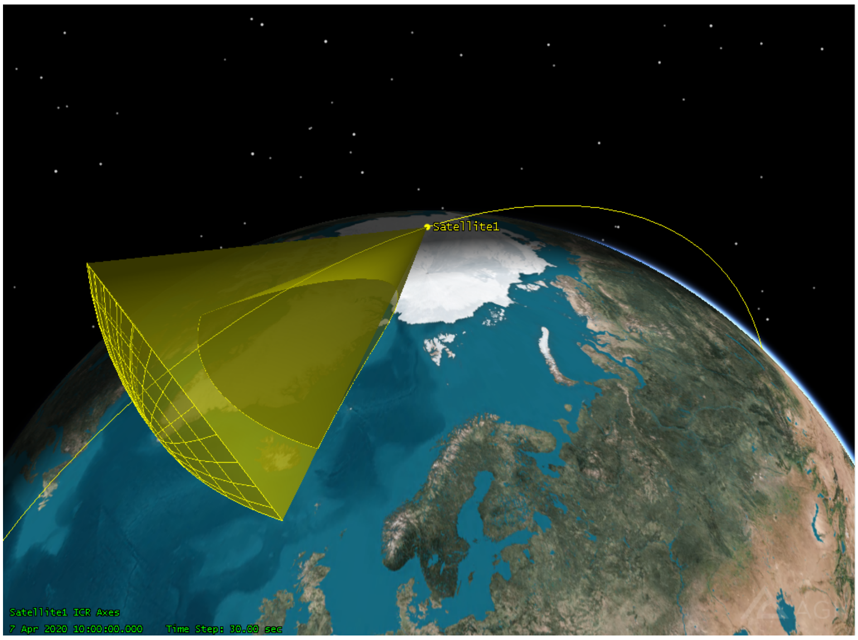


Figure 2.14: Footprint of the sensor, representing an antenna

Now, the points we want to measure the revisit times of are created. This is done by creating facilities at the different latitudes, starting at 90° to 60°, decreasing by 5° latitude for each facility. The facilities are shown in Figure 2.15.

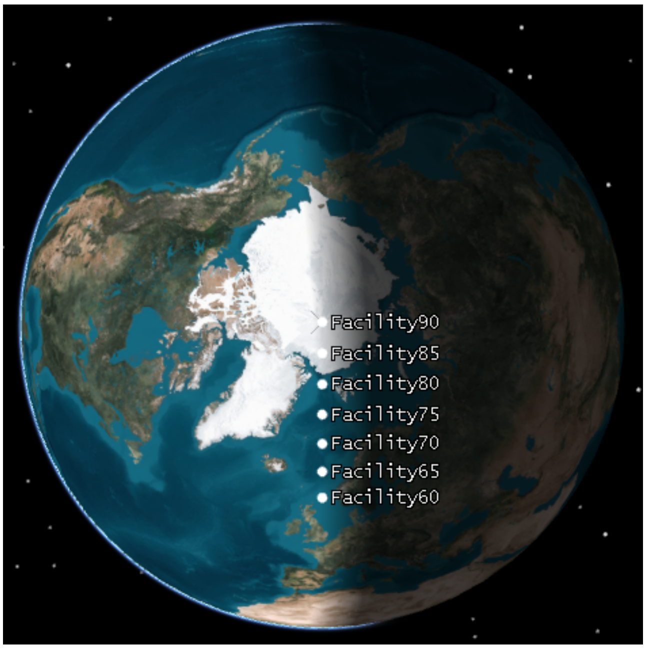


Figure 2.15: Facilities used to determine revisit time at different latitudes

The revisit time can now be computed by using the function called Figure of Merit, which calculates revisit time based on criteria given to the function. The criteria used in this section, are that at least one sensor needs to be in range, and that the maximum revisit time is the output. Revisit time will therefore describe the maximum communication gap between each facility and a constellation in a 24 hour period. Figure 2.16 shows the start of a communication period and Figure 2.17 shows the end of that period.

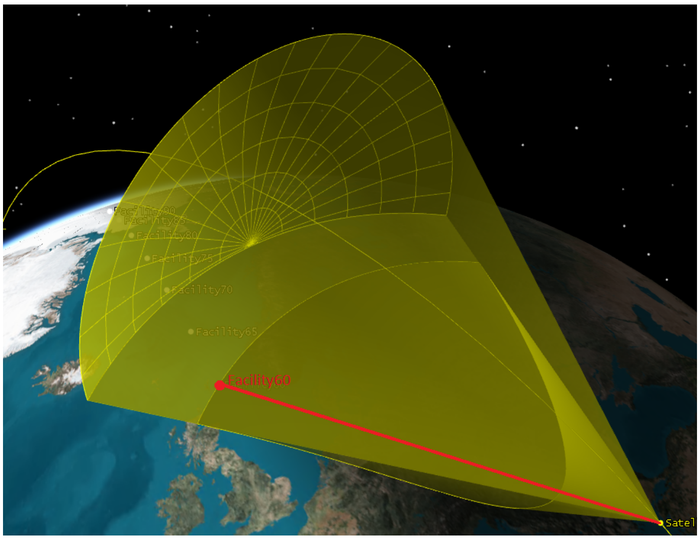


Figure 2.16: Start of a connection to Facility60

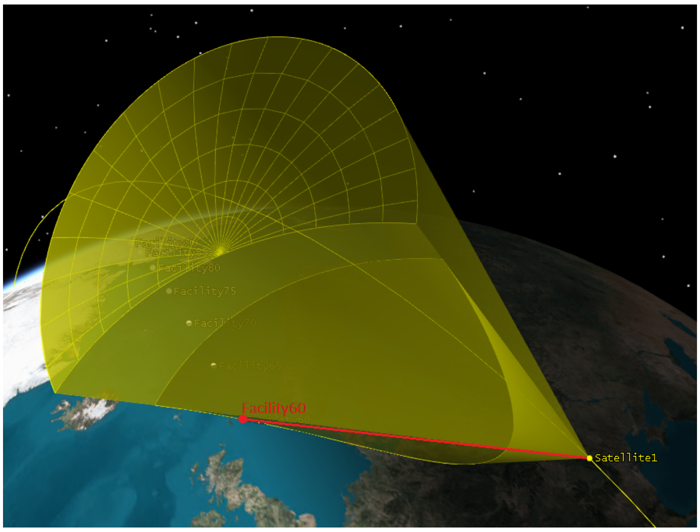


Figure 2.17: End of a connection to Facility60

Now that revisit times can be checked, a Walker constellation is created. In order to create a Walker constellation, a seed satellite is used. The seed satellite will be copied onto all satellites in the Walker constellation. The seed satellite used in this section can be seen in Figure 2.14. To create a Walker constellation, the amount of planes, the amount of satellites pr. plane, and inter plane spacing are needed. Inter plane spacing is set to one to make STK calculate the true anomaly phasing degree automatically. This ensures that all satellites are evenly distributed in each plane. A Walker Delta i:6/1/0 constellation can be seen in Figure 2.18.

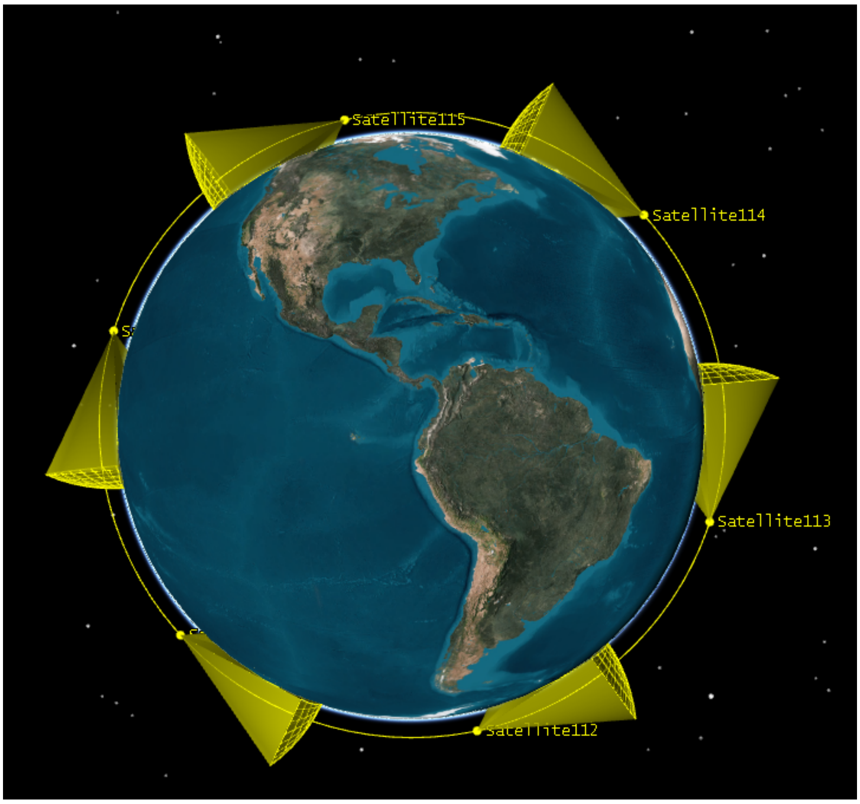


Figure 2.18: A Walker delta i:6/1/0 constellation

The revisit time for the different latitudes can now be computed using walker constellations. This is done by starting with one plane and one satellite, and adding a satellite to each plane until constant coverage is provided for facility 90. By doing this, the planes will always have the same amount of satellites and be evenly spread out. Figure 2.19 shows a Walker Delta i:20/4/1 constellation, which is the minimum number of total satellites required to provide constant coverage for the poles using four planes.

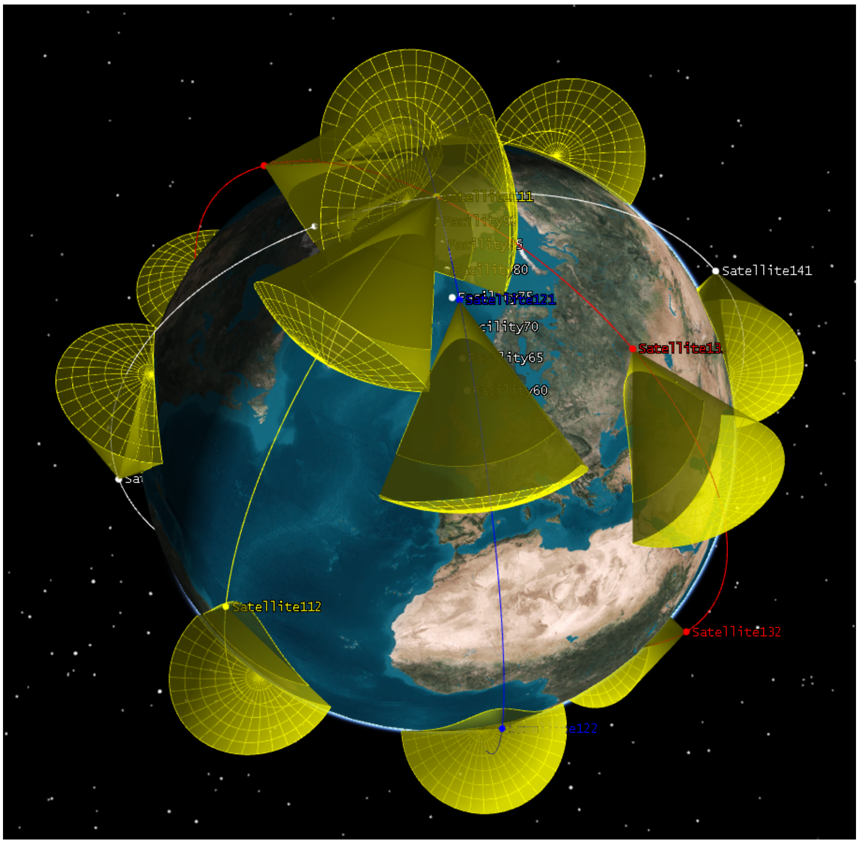


Figure 2.19: A Walker delta i:20/4/1 constellation

The revisit times have been computed and are shown for one plane in Table 2.3, for two planes in Table 2.4, for three planes in Table 2.5, and for four planes in 2.6.

Latitude Satellites	90 [s]	85 [s]	80 [s]	75 [s]	70 [s]	65 [s]	60 [s]
1	5429	5527	5708	22440	29155	29376	40836
2	2559	2643	2810	17487	29155	29376	35212
3	1603	1681	1844	18798	25363	29376	33342
4	1124	1200	1360	17487	26312	29376	32407
5	837	911	1070	18052	24604	29376	31846
6	646	719	876	17487	25363	29376	31472
7	509	581	738	17732	25083	29376	31205
8	407	478	635	16933	24889	29376	31693
9	327	398	554	16929	24724	28739	31461
10	263	335	489	16933	24604	28803	31275
11	211	283	437	16930	25018	28855	31123
12	168	240	393	16933	24884	28898	31472
13	131	203	356	16930	24779	28935	31328
14	100	172	324	16933	24681	28967	31205
15	72	145	296	16931	24604	28994	31465
16	48	121	272	16933	24529	28670	31349
17	27	100	251	16931	24801	28711	31246
18	9	81	232	16620	24410	28430	31155
19	0	65	216	16931	24661	28480	31073

Table 2.3: Revisit time of multiple latitudes using a single plane. Unit of data is seconds.

Latitude Satellites	90 [s]	85 [s]	80 [s]	75 [s]	70 [s]	65 [s]	60 [s]
2	2560	2721	2944	5613	9038	13386	19799
4	1125	1287	1511	2692	7637	12022	13693
6	647	808	1032	1790	5252	9248	12341
8	408	568	792	1296	4976	8576	11576
10	264	425	648	1011	3991	8136	10859
12	168	329	552	806	4698	8336	10094
14	100	261	484	706	3973	7868	10276
16	49	210	432	640	3723	8206	10366
18	9	170	389	565	3617	7376	10484
20	0	139	352	479	3649	7725	10021

Table 2.4: Revisit time of multiple latitudes using a two planes. Unit of data is seconds.

Latitude Satellites	90 [s]	85 [s]	80 [s]	75 [s]	70 [s]	65 [s]	60 [s]
3	1605	1774	1995	5475	5630	8211	12155
6	649	817	1041	2573	2741	4452	5685
9	330	499	722	1611	1763	3198	5349
12	170	339	563	1219	1272	1425	4186
15	75	244	467	946	953	1133	4019
18	11	180	404	725	816	998	3229
21	0	134	358	568	732	951	3446

Table 2.5: Revisit time of multiple latitudes using a three planes. Unit of data is seconds.

Latitude Satellites	90 [s]	85 [s]	80 [s]	75 [s]	70 [s]	65 [s]	60 [s]
4	1125	1293	1515	4074	4724	5548	9741
8	408	577	800	2361	2625	2742	2934
12	169	338	561	1674	1695	1760	1808
16	49	219	441	1168	1153	1230	1359
20	0	147	370	836	859	959	1041

Table 2.6: Revisit time of multiple latitudes using a four planes. Unit of data is seconds.

It can be seen that the highest number of satellites used are not the same for all data sets. This is due to a satellite being added to each plane, so for three planes, three satellites are added each time. From the data sets, it can be seen that the revisit times of latitude 90 are the same for all number of planes, and as the number of planes increase, the revisit time of the lower altitudes decrease.

The revisit times can now be compared to the times on the roadmap to see if they are sufficient, or can be obtained with a lower number of satellites.
Generation 0 states that one satellite can guarantee a revisit time of 95min in Arctic waters. From the tests, it can be concluded that a revisit time of 90min and 681min can be guaranteed for 90° and 60° degrees latitude respectively. This is not sufficient. In order to obtain a revisit time of 95min, one would need three planes with two satellites per plane, so a total of six satellites.
Generation 1 states that four to six satellites can guarantee a revisit time of 25min in Arctic waters. From the tests, it can be seen that a revisit time of 162min at 60° degrees latitude can be guaranteed with a total of four satellites in four orbital planes, one satellite per plane. In order to obtain a revisit time of 25min, one would need a total of 16 satellites in four orbital planes, four satellites per plane.

In order to analyse which number of planes are the most efficient, the data sets are analysed in Table 2.7, which shows the mean of each data set and the sum of the bottom row. This is used to see the performance in regards to which has the overall lowest revisit time over all the latitudes.

Planes	Mean [s]	Sum of best [s]
1	15239	101426
2	4256	22365
3	1997	6189
4	1666	4212

Table 2.7: Performance analysis of revisit times depending on the number of planes. Mean is the mean of the entire data set, and sum of best is sum of the row which includes a revisit time of 0 seconds for latitude 90 (bottom row)

It can be seen from Table 2.7, that the mean of revisit time as well, as the sum of best, are lowest for four planes. This means that spreading out the satellites on more planes, results in better performance in regards to overall lower revisit times. It can be concluded that using a higher number of planes provides overall lower revisit times in the Arctics waters.
Regardless of which of the proposed constellation will be used in the Sternula constellation, a problem to be addressed is the establishment of the constellation. This problem will be investigated in the section below.

2.6 Constellation Phasing

As previously stated, the Sternula constellation will consist of several small satellites. These satellites will be 6U CubeSats. CubeSats are usually launched in a single launch vehicle, carrying several satellites to keep the cost of launching at a minimum.
Assuming the Sternula satellites will be launched in the same manner, any of the constellations presented in the previous section would have to be established, in orbit, from a common deployment point.
Once a satellite is in a given orbit, it requires a significant amount of force to change the orbit. However, movement within the orbit is possible with even a low amount of force. Hence only satellites occupying the same orbit will be assumed to be launched together.
Once the satellites are placed in the desired orbit a problem will now consist of placing the satellites of a given orbit at a desired angular separation as seen in Figure 2.20.

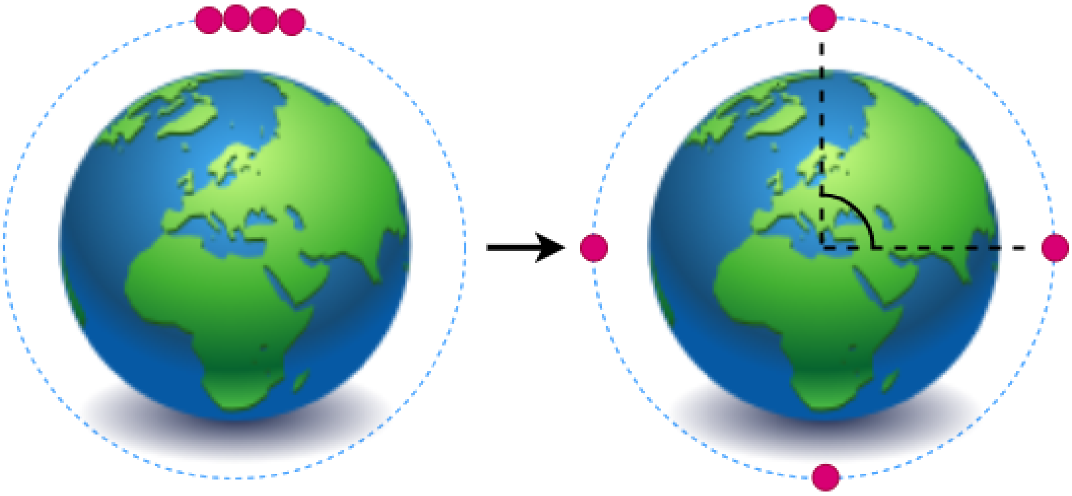


Figure 2.20: Constellation establishment within a single orbit.

Requirements to this problem, specified by Sternula, will now be presented along with a problem statement serving as the basis for the rest of this report.

2.6.1 Requirement Specifications

Requirements concerning the mission of launch and operation have been supplied by Sternula, which states the following:

1. Phasing of grouped satellites is needed.
2. A constant altitude of $550km$ must be maintained once the phasing is complete.
3. The constellation must be operational within six weeks of launch.
4. The constellation must be in a polar orbit.
5. The total thrust actuating the satellite must not exceed $180\mu N$ at any given time

2.6.2 Problem Statement

How can phasing of grouped satellites in a single polar orbital plane be obtained?

Chapter 3

Methods

To solve the problem stated in Section 2.6.2 a kinematic and dynamic model of a satellite is developed and presented in this chapter. Based on this model, Cartesian and polar equations of motion are formed along with a set of Cartesian equations of relative motion.
Following this, three control approaches are presented. One consisting of a simple PD controller based on the relative angle between two satellites, one consisting of a Linear Quadratic Regulator designed on a linearized system and one based on a Lyapunov control approach.
Each of the controllers will seek to place one satellite at a desired position relative to another. This will be considered a proof of concept for a controller to be used in the establishment of a larger constellation.
The control approaches presented will, all be based on the simplified system only affected by uniform gravitational acceleration. The inclusion of perturbations will be discussed in Section 6.

3.1 Kinematics

When describing the dynamics of the system, it must be done with respect to a given coordinate frame. The coordinate frames used in this report are presented in this section.

3.1.1 Earth Centered Inertial Coordinate System

When describing the kinematics of a spacecraft or satellite, the spacecraft is seen as a free floating object positioned in relation to an inertial reference frame placed in the center of Earth [44]. This is known as Earth Centered Inertial Coordinate System (ECI). The Z-axis of the ECI is parallel to Earths rotation axis and points towards the North pole. The x-axis points towards the point of vernal equinox, and the y-axis is computed with the right-hand coordinate system. A sketch of the ECI is seen in Figure 3.1.

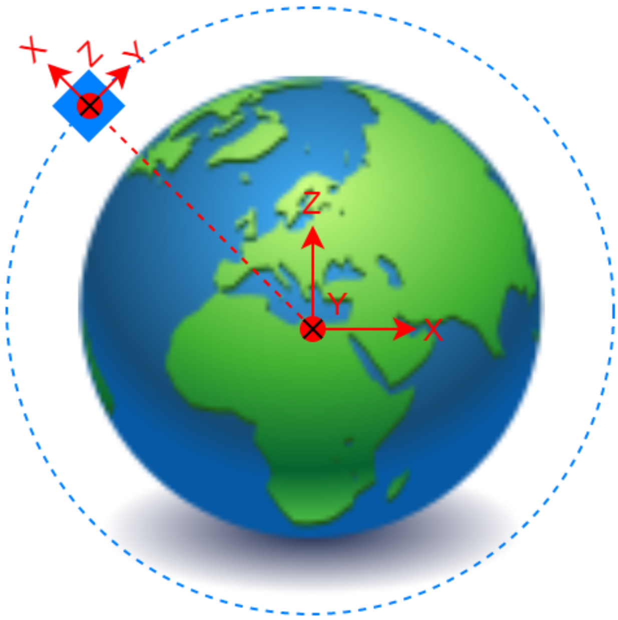


Figure 3.1: A sketch of the Hill coordinate system and ECI of an arbitrary satellite

3.1.2 The Euler-Hill Frame

Another frame used in the description of the dynamics of the system, is called the Euler-Hill frame or simply Hill frame. This is a frame fixed to the satellite, where the x-axis is pointing the opposite direction of Earth, the y-axis points along the track of the orbit, and the z-axis finishes the right-hand rule. A sketch of the Hill frame is seen in Figure 3.1.

3.1.3 Rotation Between ECI & Hill

A rotation matrix defining the rotation between the two frames will later be used. It is defined from the axes of the Hill frame.
If the position of the satellite is given by the vector $\mathbf{r}$ then the x-axis of the Hill frame defined in the ECI frame is found as:

$$\hat{\mathbf{x}}_{\mathcal{H}} = \frac{\mathbf{r}}{|\mathbf{r}|} \tag{3.1}$$

The z-axis is normal to the orbital plane. As discussed in Section 2.5, the satellites will be in a polar orbit. In the rest of the report, the polar orbit concerned will be in the plane of the x- and z-axis of the ECI frame. Hence the z-axis of the Hill frame can be described in the ECI frame as:

$$\hat{\mathbf{z}}_{\mathcal{H}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{3.2}$$

The y-axis of the Hill frame is now found as the cross product of the x- and z-axis:

$$\hat{\mathbf{y}}_{\mathcal{H}} = \hat{\mathbf{z}}_{\mathcal{H}} \times \hat{\mathbf{x}}_{\mathcal{H}} \tag{3.3}$$

A rotation matrix from the satellite fixed Hill frame to the ECI frame can now be formed by arranging these vectors into a matrix as:

$$\mathbf{R}_{\mathcal{H}}^{\mathcal{E}} = \begin{bmatrix} \hat{\mathbf{x}}_{\mathcal{H}} & \hat{\mathbf{y}}_{\mathcal{H}} & \hat{\mathbf{z}}_{\mathcal{H}} \end{bmatrix} \tag{3.4}$$

A rotation matrix in the opposite direction can be found as the transpose of this matrix.

Now that the frames and rotations between them have been defined, the dynamics of the system will be defined in the ECI frame.

3.2 Dynamics

When describing the dynamics of a satellite in orbit about Earth, a classical method used is the Keplerian two-body problem. This will be presented as a way of deriving the equations of motion of a satellite and the variables defining the orbit it follows.

3.2.1 The Keplerian Two-Body Problem

The Keplerian two-body problem describes the motion of a single satellite in relation to Earth under the following assumptions:

- The only external force acting on the system is the gravitational force between the two bodies.
- The bodies are spherical with mass concentrated at the center.
- Earth's mass is much larger than the mass of the satellite.

The gravitational force between the two bodies is modelled through Newton's law of universal gravitation, which in vector notation is given as:

$$\mathbf{F}_{\mathbf{g}} = G\frac{m_1 m_2}{r^2}\frac{\mathbf{r}}{r} \tag{3.5}$$

Where $\mathbf{F}$ is the gravitational force acting on the two bodies, m_1 is the mass of Earth, and m_2 the mass of the satellite. The position of Earth is given by the vector $\mathbf{r_1}$ and the position of the satellite is given by $\mathbf{r_2}$. The distance between the two bodies is given by $\mathbf{r} = \mathbf{r_2} - \mathbf{r_1}$ and the length of the vector is given as $r = |\mathbf{r}|$
Hence the force acting on each of the bodies is defined as:

$$m_1\ddot{\mathbf{r_1}} = \frac{Gm_1m_2}{r^2}\frac{\mathbf{r}}{r} \quad \text{and} \quad m_2\ddot{\mathbf{r_2}} = -\frac{Gm_1m_2}{r^2}\frac{\mathbf{r}}{r} \tag{3.6}$$

Isolating the acceleration in these equations yield the following expressions:

$$\ddot{\mathbf{r_1}} = \frac{Gm_2}{r^2}\frac{\mathbf{r}}{r} \quad \text{and} \quad \ddot{\mathbf{r_2}} = -\frac{Gm_1}{r^2}\frac{\mathbf{r}}{r} \tag{3.7}$$

Subtracting the two equations yields an expression for the second derivative of the distance between the two bodies:

$$\ddot{\mathbf{r}} = -\frac{G}{r^2}m1\frac{\mathbf{r}}{r} - \frac{G}{r^2}m2\frac{\mathbf{r}}{r}$$

$$\Updownarrow$$

$$\ddot{\mathbf{r}} = -\frac{G(m1+m2)}{r^2}\frac{\mathbf{r}}{r}$$

Hence the two-body equation of motion can be written as:

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} \tag{3.8}$$

Where $\mu = Gm_1$. Here the mass of the satellite is ignored due to its relatively small size compared to Earth. This equation can now be seen as the acceleration of a unit mass under assumptions of the Keplerian two-body problem.

Polar Coordinates

As the satellite orbits Earth, it is often useful to present the motion of the satellite in polar coordinates. This is done through the following equations:

$$\mathbf{r} = r\hat{\mathbf{r}} \tag{3.9}$$

$$\dot{\mathbf{r}} = \dot{r}\hat{\mathbf{r}} + r\dot{\theta}\hat{} \tag{3.10}$$

$$\ddot{\mathbf{r}} = (\ddot{r} - r\dot{\theta}^2)\hat{\mathbf{r}} + (2\dot{r}\dot{\theta} + r\ddot{\theta})\hat{} \tag{3.11}$$

Where θ is the argument of latitude given as the sum of the true anomaly and the argument of periapsis, and $\hat{\mathbf{r}}$ is the unit vetor of $\mathbf{r}$.
Substituting Equation 3.8 into Equation 3.11 and evaluating the linear and angular parts separately, the following equations are found:

$$\ddot{r} = r\dot{\theta}^2 - \frac{\mu}{r^2} \tag{3.12}$$

$$\ddot{\theta} = -\frac{2\dot{r}\dot{\theta}}{r} \tag{3.13}$$

From Equation 3.12 the angular velocity required to maintain a constant altitude while subject to the gravity term of Equation 3.8 can be found. For an altitude of $550km$ this is calculated as:

$$\begin{gathered} \dot{\theta} = \sqrt{\frac{\mu}{r^3}} \\ \Downarrow \\ \dot{\theta} = \sqrt{\frac{\mu}{(6.371+550)10^3)^3}} \approx 0.0011 \end{gathered} \tag{3.14}$$

Where r is given as the sum of the mean radius of Earth and the altitude of the satellite.

Equation 3.8 constitutes the main part of the dynamics. However, there are a series of other minor forces also acting upon the system. These will be viewed as perturbations to this main model and will be presented in the following section.

3.2.2 Perturbations

The main perturbations to the model consists of the following forces:

- Gravitational perturbations due to the non-sphericity of Earth.
- Atmospheric drag
- Solar radiation
- Gravitational effects of other celestial objects

While all of these affect the satellite over time, the main ones affecting a satellite in a LEO are the gravitational and atmospheric drag perturbations. Hence, this section will focus on the modelling of these.

Gravitational Perturbations due to the Non-Sphericity of Earth

In Equation 3.5, the gravitational force acting upon the system is found. This is however, done under the assumption of the two bodies being point masses. The satellite being a point mass is a fair assumption given the relative size to Earth. However, assuming Earth to be a point mass would only be completely correct if Earth was a perfect sphere. Since this is not the case, the non-sphericity of Earth has to be taken into account when modelling the gravitational forces acting upon an orbiting satellite.

This is initially done by not assuming Earth to be a single point mass, but as a series of point masses. This means that the gravitational force per unit mass can now be modelled as:

$$\mathbf{F_g} = -\sum_i G\frac{m_i}{r_i^3}\mathbf{r_i} \tag{3.15}$$

Where m_i is the mass of the i'th body and $\mathbf{r_i}$ is the position vector of the i'th body. Ideally, this term should be written in a continuous form. This is done by defining the gravitational forces as a potential function of the form:

$$\mathbf{F} = \triangledown\phi \tag{3.16}$$

Where $\triangledown$ is the gradient operator defined as:

$$\triangledown(\cdot) = \frac{\delta}{\delta x}(\cdot)\mathbf{x} + \frac{\delta}{\delta y}(\cdot)\mathbf{y} + \frac{\delta}{\delta z}(\cdot)\mathbf{z} \tag{3.17}$$

Where $\mathbf{x}, \mathbf{y}, \mathbf{z}$ are the unit vectors of the coordinate frame in which the force is described.
The potential function equivalent to the force described in Equation 3.15 is found to be:

$$\phi = \sum_i \frac{Gm_i}{r_i} \tag{3.18}$$

This function for gravitational potential can now be rephrased in continuous form by exchanging the summation for an integration and the mass for the term $dm = \rho dV$ where ρ is the mass density and dV is an infinitesimally small volume element. Further derivation of the continuous potential function over the volume of Earth can be found in [26], but will not be further addressed in this paper.

Through this derivation, the perturbing gravitational potential function is found to be dependent on three coefficients $J_n, C_{n,m}$ and $S_{n,m}$ describing the deviation from the ideal spherical Earth about the x-, y- and z-axis respectively. These coefficients are determined experimentally and can be found as table values.
The main perturbing effect is caused by the J_2 coefficient. This is the coefficient associated with the oblateness of Earth and is given as:

$$J_2 = 1.083 \times 10^-3 \tag{3.19}$$

By only including the effects of the J_2 perturbations, the perturbing potential function can be found to be:

$$\phi_p = -\frac{\mu}{r^3}J_2R_e^2(\frac{3}{2}sin^2\delta - \frac{1}{2}) \tag{3.20}$$

Where R_e is the radius of Earth at the equator and δ is the latitudinal angle to the spacecraft.
From this potential function, the force per unit mass can be derived as seen in

Equation 3.16. Once again, the derivation will not be presented here, but is found in [26].

$$\mathbf{F_{J2}} = \frac{3\mu J_2 R_e^2}{2r^5}((5\frac{(\mathbf{r}\cdot\mathbf{z_G})^2}{r^2} - 1)\mathbf{r} - 2(\mathbf{r}\cdot\mathbf{z_G})\mathbf{z_G}) \tag{3.21}$$

Where z_G is the unit vector of the z-axis of the ECI frame.

Atmospheric drag

Atmospheric drag is the force caused by the interaction between the satellite and the atmosphere through which it moves. A model of the forces produced by this interaction will now be presented, as described in [23].
Over a given time interval, denoted Δt, the satellite will collide with a mass of atmosphere, denoted Δm, given by the following equation:

$$\Delta m = \rho A v_r \Delta t \tag{3.22}$$

Where ρ is the density of the atmosphere, A is the cross-sectional area of the satellite and v_r is the velocity of the satellite with respect to the atmosphere. From this, the impulse Δp [1] acting on the satellite can be found as:

$$\Delta p = \Delta m v_r = \rho A v_r^2 \Delta t \tag{3.23}$$

Hence, the resulting force is found by $F = \Delta p/\Delta t$. This is written as:

$$\mathbf{F_d} = -\frac{1}{2} C_D A \rho v_r^2 e_v \tag{3.24}$$

A few terms have been added to this equation, which are presented below along with an elaboration of some of the previously presented parameters:

- $\frac{1}{2}$ is included to adhere to the standard notation used in aerodynamics.
- C_D is the drag coefficient which is introduced to describe the interaction between the atmosphere and the surface material of the satellite. This includes how the particles of the atmosphere is deflected off the satellite upon impact, and the air flow around the satellite. This coefficient is determined experimentally and ranges from 1.5 to 3.0. A rough approximation of the coefficient for a satellite with a spherical body is $C_D = 2$, while for non-spherical convex satellites the value is between 2.0 and 2.3. In the further modelling of this satellite, the value will be assumed to be 2.2.

[1] Impulse is given as the integral of force over a given time

- A is the cross sectional area of the satellite as previously stated. This is dependent on the shape of the satellite, as well as the attitude of the system. For this model, the satellite will always be assumed to be nadir pointing, hence the cross sectional area will be constant. As previously stated, the satellite is a 6U Cubesat which measures $10x20x30cm$. Assuming the satellite is designed, such that the smallest cross sectional area is subject to the drag forces during nominal flight, the value of A is found to be $0.1m \cdot 0.2m = 0.02m^2$

- $\mathbf{v_r}$ the relative velocity between the satellite and the atmosphere is found as

$$\boldsymbol{v_r} = \boldsymbol{v} - \boldsymbol{\omega}_{\oplus} \times \boldsymbol{r} \tag{3.25}$$

  Where $\boldsymbol{\omega}_{\oplus}$ is the angular velocity of Earth. Hence, the velocity of the atmosphere is assumed to be equal to the angular velocity of Earth which is given as:

$$\boldsymbol{\omega}_{\oplus} = \begin{bmatrix} 0 \\ 0 \\ 0.7292 \end{bmatrix} 10^{-4} rad/s \tag{3.26}$$

  At an altitude of $550km$ above Earth's surface, this corresponds to a linear velocity of:

$$\mathbf{v}_{\oplus} = 0.7292 \cdot 10^{-}4 rad/s(6371 + 550)km \approx 0.5km/s \tag{3.27}$$

  Satellites at this altitude travel at about $7.5km/s$, hence this is a significant factor.

- $\mathbf{e_v}$ is the unit vector describing the direction in which the force acts. It is found as:

$$\mathbf{e_v} = \mathbf{v_r}/v_r \tag{3.28}$$

- As previously stated, ρ is the density of the atmosphere. Determining this value is a complex task, as it is affected by a series of parameters including, but not limited to, altitude, temperature, solar radiation and winds, geomagnetic storms, and the chemical composition of the atmosphere. Several complicated models of the atmospheric density exists, however a simplified model publicly available by NASA in [7] is stated as:

$$\rho = \frac{p}{0.2869(T + 273.1)} \tag{3.29}$$

  Where p is the air pressure at the given altitude in kPa and T is the temperature in $^{\circ}C$.

Each of these values are estimated with the following expressions:

$$T = -131.21 + 0.00299h \tag{3.30}$$

$$p = 2.488(\frac{T + 273.1}{216.6})^{-11.388} \tag{3.31}$$

At an altitude of $550km$ above Earth, this results in a density of $1.78 \cdot 10^{-13}$. One could argue that given the limited altitude range, the satellite operates in a table value could just as well be used. Such a value is found in [19]. For an altitude of $550km$, the density is found to be $2.21 \cdot 10^{-}13$.
Both the table value and the value found through the NASA model are in the same magnitude, hence either can be used.

This concludes the modelling of the perturbations that will be included in this model of the spacecraft. The final part of the dynamics will be the actuation. This will be presented in the following section.

3.2.3 Actuation

The spacecraft will be actuated by a single thruster. As stated in Section 2.2, thrusters generate force by accelerating an amount of mass opposite of the direction of movement. Hence, a model of a thruster can simply be given by:

$$\mathbf{F_t} = |\dot{m}|\mathbf{v_e} \tag{3.32}$$

Where $|\dot{m}|$ is the mass expelled by the satellite and v_e is the exit velocity of the given mass [23]. The loss of mass must then of course be included in the model of the satellite. However, the satellite modelled in this report will be using an ion thruster. The mass expelled by such a system is so small that it may be considered negligible for the purposes of this report.
In addition to this, controlling the attitude of the satellite is not within the scope of this report. Hence the thruster will simply be modelled as a force that can be applied in any direction.
This concludes the dynamic modelling of the satellite. All of the forces presented in the previous sections can now be added together, to achieve a complete description of the forces acting upon the satellite as given below:

$$\mathbf{F} = \mathbf{F_g} + \mathbf{F_{J2}} + \mathbf{F_d} + \mathbf{F_t} \tag{3.33}$$

Now that a term for the forces acting upon the system has been found, a set of equations governing the motion of the satellite can be formed.

3.3 Equations of Motion

From the term describing the forces acting upon the satellite, the Cartesian equations of motion can simply be found as:

$$\ddot{\mathbf{r}} = \mathbf{F}/m \tag{3.34}$$

Where m is the mass of the satellite. However, as this report deals with the problem of relative position between satellites, a set of equations describing the relative motion between the satellites are required.

3.3.1 Equations of Relative Cartesian Motion

When considering the relative motion of two satellites, the system is often considered in Cartesian coordinates in the Hill frame as described in Section 3.1. This frame will be placed at one of the two satellites in question, which will be called the "chief" satellite, the position of which is denoted by r_0, while the position of the other satellite is called the "deputy" and is denoted by r_1.
As an example of how these equations are derived, the system only affected by the strict gravitational acceleration of Equation 3.8 will be considered. The derivation is seen in its entirety in [4].

The equations are found by firstly finding a term for the relative acceleration between the two satellites in the ECI frame. The acceleration of the satellites are given by:

$$\ddot{\mathbf{r}}_{\mathbf{0}} = -\frac{\mu}{r_0^3}\mathbf{r}_{\mathbf{0}} \quad , \quad \ddot{\mathbf{r}}_{\mathbf{1}} = -\frac{\mu}{r_1^3}\mathbf{r}_{\mathbf{1}} \tag{3.35}$$

The relative position between the satellites is now found as:

$$\boldsymbol{\rho} = \boldsymbol{r_1} - \boldsymbol{r_0} \tag{3.36}$$

Now the relative acceleration can be found as:

$$\ddot{\boldsymbol{\rho}} = -\frac{\mu(\boldsymbol{r_0} + \boldsymbol{\rho})}{||(\boldsymbol{r_0} + \boldsymbol{\rho}||^3} + \frac{\mu \boldsymbol{r_0}}{r_0^3}\boldsymbol{r_0} \tag{3.37}$$

To express this acceleration in the Hill frame of the chief satellite, the following equation from rigid body kinematics have to be taken into account, describing the acceleration of one point moving on a rigid body:

$$\ddot{\boldsymbol{\rho}} = \ddot{\boldsymbol{\rho}}_{\mathcal{H}} + 2\boldsymbol{\omega} \times \dot{\boldsymbol{\rho}}_{\mathcal{H}} + \dot{\boldsymbol{\omega}} \times \boldsymbol{\rho}_{\mathcal{H}} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{\rho}_{\mathcal{H}}) \tag{3.38}$$

Where $\boldsymbol{\rho}_{\mathcal{H}}$ is the relative position of the satellites in the Hill frame of the chief further defined as:

$$\boldsymbol{\rho}_{\mathcal{H}} = [x, y, z]^T \tag{3.39}$$

Where $\boldsymbol{\omega}$ is the angular velocity vector of the Hill frame of the chief satellite relative to the ECI frame. This velocity vector is normal to the orbital plane and can therefore be denoted as:

$$\boldsymbol{\omega} = [0, 0, \dot{\theta}_0]^T \tag{3.40}$$

Where θ_0 is the argument of latitude of the chief.
The relative acceleration in the ECI frame, can be rotated into the Hill frame of the chief by taken $\mathbf{r_0}$ as:

$$\mathbf{r_0} = [r_0, 0, 0]^T \tag{3.41}$$

Evaluating Equation 3.38 with Equation 3.37 and the above definitions, yields the following equations for the relative Cartesian motion on each axis of the Hill frame of the chief.

$$\ddot{x} - 2\dot{\theta}_0\dot{y} - \ddot{\theta}_0 y - \dot{\theta_0}^2 x = -\frac{\mu(r_0 + x)}{((r_0 + x)^2 + y^2 + z^2)^{\frac{3}{2}}} + \frac{\mu}{r_0^2} \tag{3.42}$$

$$\ddot{y} + 2\dot{\theta}_0\dot{x} + \ddot{\theta}_0 x - \dot{\theta_0}^2 y = -\frac{\mu y}{((r_0 + x)^2 + y^2 + z^2)^{\frac{3}{2}}} \tag{3.43}$$

$$\ddot{z} = -\frac{\mu z}{((r_0 + x)^2 + y^2 + z^2)^{\frac{3}{2}}} \tag{3.44}$$

The additional forces of Equation 3.15 can easily be added to these equations, given that they are defined in the Hill frame of the chief. This can be done through a rotation matrix as presented in Section 3.1.

Together with Equations 3.12 and 3.13, these equations of motion form a 10x1 state vector describing the relative motion between two satellites. These equations will later be used when simulating the system in Section 4.

3.3.2 Equations of Relative Motion in a Polar Frame

Another approach to the description of the relative position of two satellites, is a simpler approach where the relative argument of latitude is simply considered.
The relative angle between two satellites, denoted θ, are found as:

$$\theta = arccos(\frac{\mathbf{r_0} \cdot \mathbf{r_1}}{r_0 r_1}) \tag{3.45}$$

The relative angular velocity and acceleration can then be found as the first and second derivative of this expression respectively. This is computed using Maple, which can be seen in Appendix C

This approach to defining the angle between the satellites is not useful in all settings, as the *arccos* function only returns useful results in the range of $0 - 180$ degrees. In spite of this, the approach will be used in a control setting seen in Section 3.4.1.

3.4 Control Methods

Now that the equations of motion of the system have been defined, several approaches to the control of the system will be presented, starting with a simple PD controller.

3.4.1 PD Control

The PD controller is based on the expression for relative argument of latitude as presented in Equation 3.45. The structure of the controller is seen in Figure 3.2

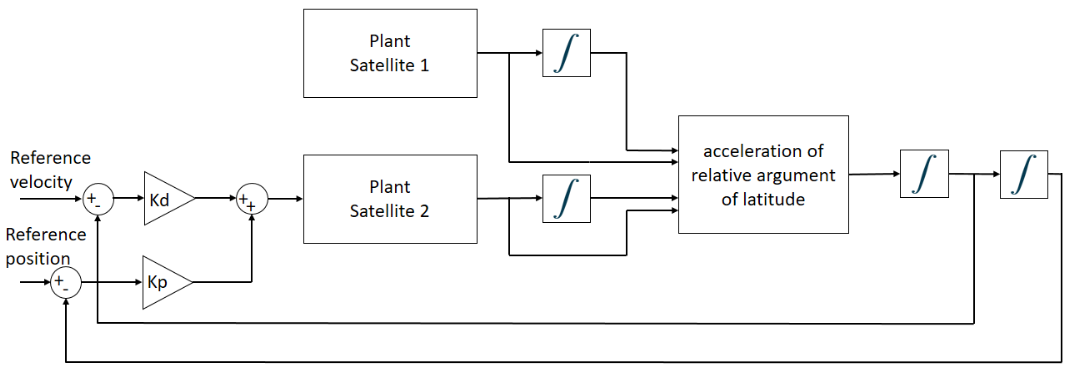


Figure 3.2: The structure of the PD controller

The controller consists of two plants defining the two satellites. These are based on the equations of motion of Equation 3.34. Only the second satellite is actuated. The states of the satellites, along with the derivative of these, are used to calculate the acceleration of the relative argument of latitude between the two satellites. By differentiating the output of this function twice, the angular velocity and position is found.
By comparing these to a reference, two error terms are found and multiplied by

two separate gains. The summation of these is the control input to the second satellite. This is a single value actuating the satellite along the direction of the orbit.

3.4.2 Linear Quadratic Regulator

Another approach to the control of the position between two satellites, is to linearize the Cartesian relative equations of motion, and then perform linear control on this simplified system. This will result in a control law which will stabilize the system around the point where it is linearized. The development of such a control law is presented below, following the approach presented in [4].

Circular Chief Orbit

The first step of linearizing the equations of motion, is to assume a circular orbit of the chief.
This assumption is only true if the satellite is moving at a velocity high enough to overcome the gravitational acceleration. For a system that is only affected by the gravity of Earth, this velocity is found through the term given by Equation 3.14. By assuming this velocity to be constant, the following variables of the system in a circular orbit can be defined:

$$\dot{\theta}_0 = \sqrt{\frac{\mu}{r^3}} = constant = n_0 \quad , \quad \ddot{\theta}_0 = 0 \quad , \quad r_0 = constant = a_0 \tag{3.46}$$

Under these assumptions, the equations of relative motion are:

$$\ddot{x} - 2n_0\dot{y} - n_0^2 x = -\frac{\mu(a_0 + x)}{((a_0+x)^2 + y^2 + z^2)^{\frac{3}{2}}} + \frac{\mu}{a_0^2} \tag{3.47}$$

$$\ddot{y} + 2n_0\dot{x} - n_0^2 y = -\frac{\mu y}{((a_0+x)^2 + y^2 + z^2)^{\frac{3}{2}}} \tag{3.48}$$

$$\ddot{z} = -\frac{\mu z}{((a_0+x)^2 + y^2 + z^2)^{\frac{3}{2}}} \tag{3.49}$$

This removes some of the nonlinearities of the system, however the ones of the gravitational accelerations on the right hand side of the equations remain. These are addressed below.

Clohessy-Wiltshire Linearized Equations of Motion

The Clohessy-Wiltshire (CW) equations were derived in the 1960s for use in satellite rendezvous maneuvers. There are several ways of performing this linearization, however, only the results of a simple linearization will be presented here.

The right hand side of the equations of motion of a circular chief orbit, are expanded around the origin through a Taylor series. By only taking the first order terms of the resulting equations, the expressions are reduced to the following:

$$-\frac{\mu(a_0+x)}{((a_0+x)^2+y^2+z^2)^{\frac{3}{2}}}+\frac{\mu}{a_0^2}\approx n_0^2(2x-a_0) \tag{3.50}$$

$$-\frac{\mu y}{((a_0+x)^2+y^2+z^2)^{\frac{3}{2}}}\approx -n_0^2 y \tag{3.51}$$

$$-\frac{\mu z}{((a_0+x)^2+y^2+z^2)^{\frac{3}{2}}}\approx -n_0^2 z \tag{3.52}$$

Inserting this into Equations 3.47 - 3.49, the following set of linear equations of motion is found:

$$\begin{aligned} \ddot{x}-2n_0\dot{y}-3n_0^2x&=0\\ \ddot{y}+2n_0\dot{x}&=0\\ \ddot{z}+n_0^2z&=0 \end{aligned} \tag{3.53}$$

As these equations are the results of a linearization around the origin, they will only be valid when the two satellites are positioned closely together.

Based on the linearized equations of motion, the following state space equation can be formed. Since only motion in the orbit plane is considered, only the x- and y-components of the equations will be included in this equation.

$$A=\begin{bmatrix} 0 & 1 & 0 & 0\\ 3n_0^2 & 0 & 0 & 2n\\ 0 & 0 & 0 & 1\\ 0 & -2n & 0 & 0 \end{bmatrix}\quad,\quad B=\begin{bmatrix}0\\0\\0\\1\end{bmatrix} \tag{3.54}$$

In this formulation, the only actuation considered is a force in the direction of motion as seen in the B-vector.

This system can be controlled using linear control algorithms. As an example of this, an Linear Quadratic Regulator (LQR) control approach will be presented below.

LQR control

In LQ control, a cost function is used to find the optimal gain to control the system. This cost function is given as:

$$\int_0^\infty (\mathbf{x}^T\underline{\mathbf{Q}}\mathbf{x}+\mathbf{u}^T R\mathbf{u})dt \tag{3.55}$$

Where $\underline{\mathbf{Q}}$ and R are values weighting the cost to the state deviation and control signal respectively. $\underline{\mathbf{Q}}$ and R are freely chosen to shape the cost function in accordance

with the system in question. For the purpose of satellite control, R will usually be chosen to be much larger than $\underline{\mathbf{Q}}$ as actuation is very costly in these systems.
A feedback control law is found to minimize this cost function as:

$$u = -Kx \tag{3.56}$$

Further explanation of how the minimizing control law is found is not within the scope of this project.

Using the control law found through the LQR method, the CW linearized system is stabilized. Since the linearized system is a valid representation of the nonlinear system around the origin, this control law will also stabilize the nonlinear system, as long as the two satellites are within the limits of the linearization.
This control law which stabilizes the system around the origin, can be used in the stabilization and control of the full nonlinear system through Lyapunov control.

3.4.3 Lyapunov Control

The Lyapunov control approach is based on the theorem of Lyapunov stability stating the following [18].

Lyapunov Stability

For a function given by:

$$\dot{x} = f(x) \tag{3.57}$$

With $x = 0$ as an equilibrium point and $D \subset R^n$ being a domain containing this equilibrium point. Then a function $V : D \rightarrow R$ is a Lyapunov function if it is continuously differentiable and satisfy the following:

$$V(0) = 0 \quad , \quad v(x) > 0 \tag{3.58}$$

The system is then stable if:

$$\dot{V}(x) \leq 0 \tag{3.59}$$

And asymptotically stable if:

$$\dot{V}(x) < 0 \tag{3.60}$$

The problem is now to derive a control law that ensures that the derivative of the Lyapunov function is strictly less than zero, namely negative definite.

Deriving the Control Law

The first step of deriving the control law, is to define the system. The equations of motion on the x- and y-axis are together denoted as:

$$\ddot{\mathbf{x}} = \mathbf{N_k}(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{N_u}(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{u} \tag{3.61}$$

Where $\mathbf{N_k}$ is the known nonlinear model of the system as described in Equations 3.42 and 3.43. Even if this model were to be expanded with the additional forces derived in Section 3.2.2, it would still contain inaccuracies in addition to the forces not modelled, such as solar radiation. The inaccuracies and unmodelled dynamics are gathered in the term N_u, which denotes the unknown nonlinear effects. Finally u is the control input.
The control law will be used to track a reference or reference trajectory, denoted $\mathbf{x_r}$. The tracking error is given as:

$$\mathbf{e} = \mathbf{x} - \mathbf{x_r} \tag{3.62}$$

The dynamics of which can be stated as:

$$\ddot{\mathbf{e}} = \mathbf{N_k}(\mathbf{e}, \dot{\mathbf{e}}) + \mathbf{N_u}(\mathbf{e}, \dot{\mathbf{e}}) + \mathbf{u} \tag{3.63}$$

A Lyapunov function is now chosen as:

$$V = n_0^2 \frac{1}{2} \mathbf{e} T \mathbf{e} + \dot{\mathbf{e}}^T \dot{\mathbf{e}} \tag{3.64}$$

From this equation, it is easy to see that the requirements of Equation 3.58 are satisfied. Hence, this is a valid Lyapunov function. The function is now differentiated as:

$$\dot{V} = n_0^2 \dot{\mathbf{e}}^T \mathbf{e} + \dot{\mathbf{e}}^T \ddot{\mathbf{e}} \tag{3.65}$$

This function is now evaluated with Equation 3.63, yielding the following expression:

$$\dot{V} = n_0^2 \dot{\mathbf{e}}^T \mathbf{e} + \dot{\mathbf{e}}^T (\mathbf{N_k}(\mathbf{e}, \dot{\mathbf{e}}) + \mathbf{N_u}(\mathbf{e}, \dot{\mathbf{e}}) + \mathbf{u}) \tag{3.66}$$

To ensure that the system is stable, a control law must be found that makes this expression negative definite. Such a control law can be given as:

$$u = -\mathbf{N_k}(\mathbf{e}, \dot{\mathbf{e}}) - n_0^2 \mathbf{e} - k\dot{\mathbf{e}} \tag{3.67}$$

Where k is a positive constant. Using this as the control law yields the following function:

$$\dot{V} = \mathbf{N_u}(\mathbf{e}, \dot{\mathbf{e}}) - k\dot{\mathbf{e}}^T \dot{\mathbf{e}} \tag{3.68}$$

Hence, as long as the $k\dot{\mathbf{e}}$ is larger than the unmodelled dynamics, the derivative of the Lyapunov function is negative definite resulting in a stable system. Therefore, Equation 3.68 can be considered a stabilizing control law for the nonlinear system.

The control algorithms presented in this chapter will now be implemented and tested in a simulation environment.

Chapter 4

Implementation & Testing

In this chapter, the forces of the model derived in Section 3.2 will be compared to the forces of the model of AGI STK, as to validate both the model derived and the simulation which is based upon it.
Furthermore, the control algorithms described in Chapter 3 will be implemented in Simulink. Each of the controllers will be tested in this simulated environment as a proof of concept for the controllers developed.

4.1 Model Validation Through STK

This section will provide information about how STK is used to compute the gravitational and atmospheric drag perturbations of a satellite, through the use of Matlab. The aim is to be able to compute forces applied on the satellite, for a position specified by the controller. The first step is to establish a connection between Matlab and a running instance of STK. This is done with a single line of code, which gives a handle to the root of STK. From this handle, everything within the current scenario of STK is accessible.
Next, a handle to the satellite is obtained. This allows for changing and reading the parameters of the satellite. In order to propagate the satellite, a starting date and orbit parameters are needed. The simulation can perform calculations along the propagation, and can be evaluated at times using the following format:

```
2 May 2020 03:04:05.000
```

The starting date used in STK is set to 1/5 00:00, which means that the satellite has been moved for 1 day, 3 hours, 4 minutes, and 5 seconds along its propagation in the date stated above. When reading the simulation running time through Matlab, the feedback is a time in elapsed seconds since the starting time. A script converting the elapsed seconds to the date format has been developed.
The propagator for the satellite, has been set to include gravity, J2, J4, drag, and

solar radiation pressure. This results in when reading gravity from STK, it will include the J2 and J4 perturbations. In order to compute gravitational forces in the x-,y-, and z-axis from STK, the following commands are used:

```
gravity = vgtSat.Vectors.Item('force_model_gravity').
FindInAxes(time_current_date,vgtSat.
WellKnownAxes.Earth.Inertial).Vector;
gravity_x = sat_force_gravity.X;
gravity_y = sat_force_gravity.Y;
gravity_z = sat_force_gravity.Z;
```

This evaluates the vector force_model_gravity in Earth inertial frame at the date specified by the variable time_current_date. The vector force_model_gravity, is a vector created in the vector geometry of the analysis workbench in STK, which represents the force model components of gravity. The same is done for drag.
The orbit of the satellite is specified to have a velocity only along the x-axis of the Earth inertial frame. The frame is static in space, and does not follow the rotation of Earth. This allows for simpler control and analysis of the orbit, due to the dependency on only two axis instead of three, and having velocity on only one axis. The satellite model in STK is set to "cubesat_6u.dae", which is a 6U CubeSat with a grid of solar cells attached. The smallest side of the satellite points towards the direction of motion, namely the x-axis, the z-axis points towards Earth's center, and the y-axis finishes the right-hand rule. This can be seen on Figure 4.1.

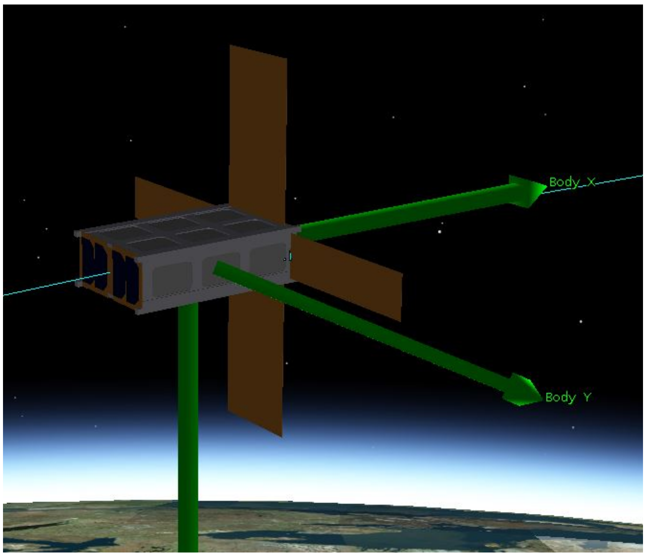


Figure 4.1: Satellite model "cubesat_6u.dae" with inertial frames

Forces applying on a satellite can now be computed. The next step is to implement this in the Simulink model, and compare the forces from STK with the

modelled forces. A Simulink S-function block is used, which takes the states of the satellite as input, the same used in the plant, and outputs forces in the Earth inertial frame. The input states are the Cartesian position and velocity in the x-,y-, and z-axis. These are used as orbit parameters for the satellite in STK, which is then propagated to generate a orbit. This allows for computing the forces through STK, by using the same states as in the plant. The forces from STK is considered to be the true forces, since they are assumed to be significantly more accurate than the forces modelled in this paper. This is based on the software being under development for 30 years and have been used for missions such as the Galileo satellites and many more [2] [1]. By comparing the forces from STK with the modelled forces, a validation of the modelling can be done.
Figure 4.2 and 4.3 shows the gravity from modelling and STK respectively, and Figure 4.4 shows the error between the two. The figures show 26000 samples, which is a about four orbital periods.

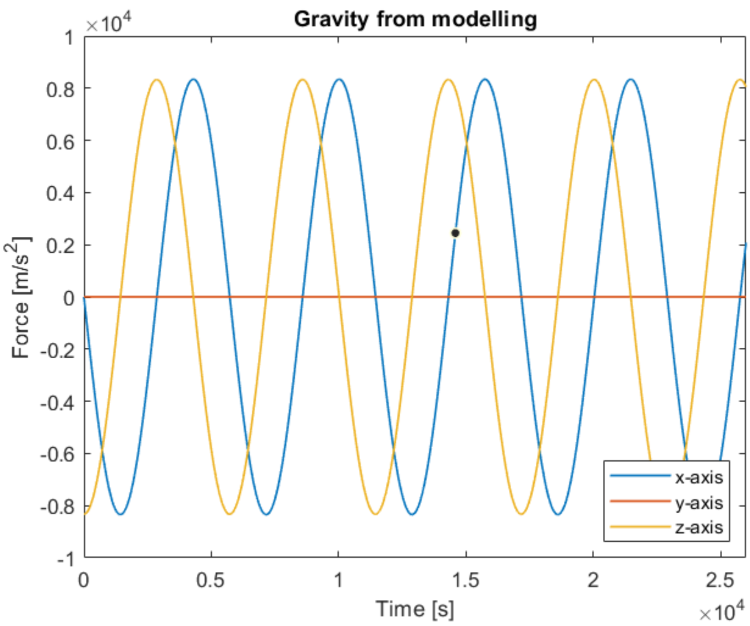


Figure 4.2: Gravity from modelling

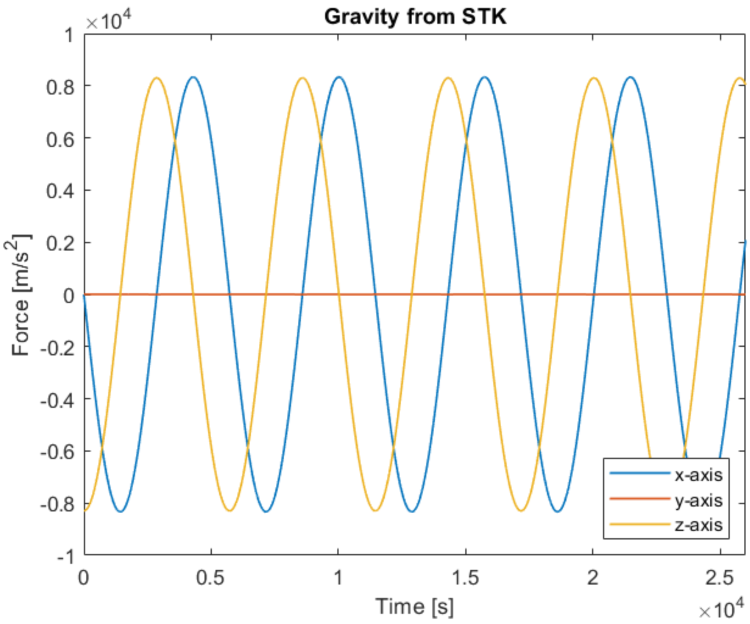


Figure 4.3: Gravity from STK

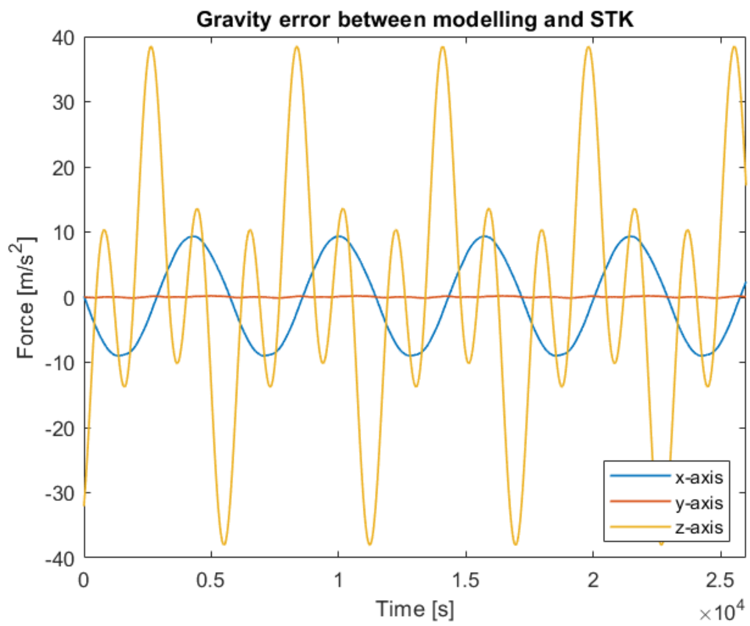


Figure 4.4: Gravity error between modelling and STK

From the figures, it can be seen that the gravity from modelling and STK are very similar, and looking at the axis representing force of the error, it has been reduced to 0.5% of the original axis. The error on the x-axis is symmetric and repeating spikes on the z-axis is present. The high spikes on Figure 4.4, are when the satellite is directly above the South- and North pole respectively. This points towards that the difference lies in the J2 perturbation, as well as STK includes the J4 perturbation, however J4 is approximately 1000 times smaller than the J2.

Next, the drag from STK and modelling are compared. The drag modelled in STK, uses the parameters described in Section 3.2.2, but uses the atmospheric density model Jacchia Roberts, which is valid within an altitude of $90km$ to $2500km$. Figure 4.5 and 4.6 shows the drag from modelling and STK respectively, and Figure 4.7 shows the error between the two.

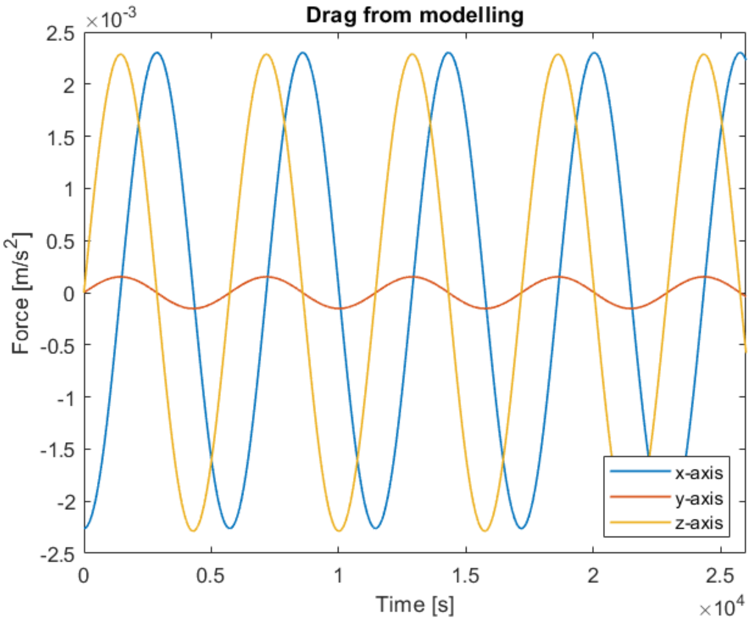


Figure 4.5: Drag from modelling

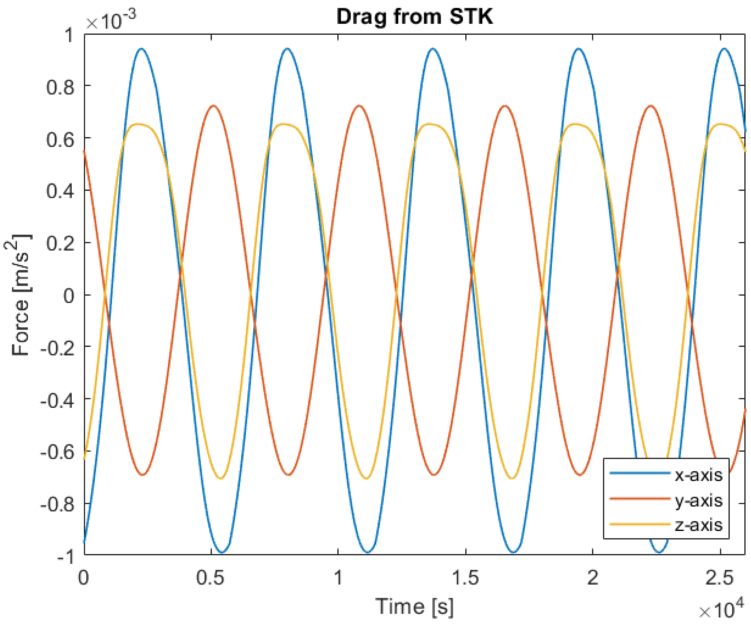


Figure 4.6: Drag from STK

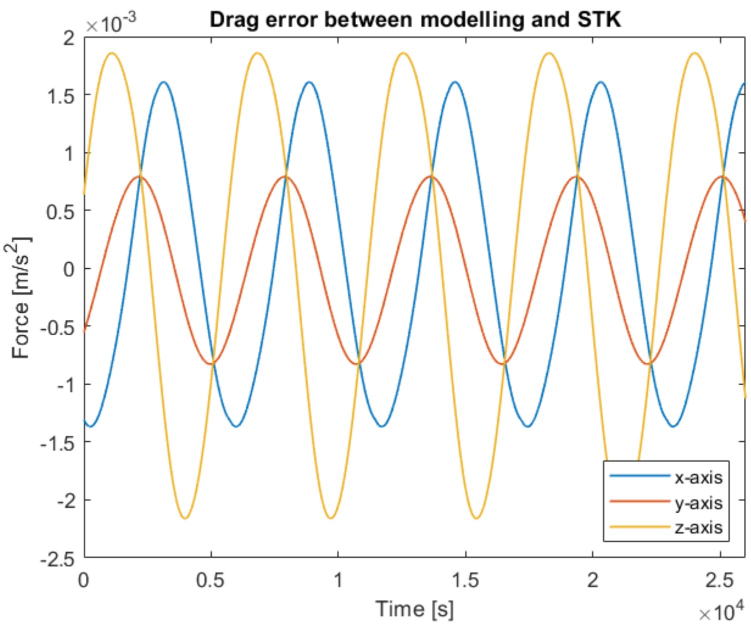


Figure 4.7: Drag error between modelling and STK

From the figures, it can be seen that both drag models are symmetric, where the x- and z-axis from modelling are similar, only with a shift in time, and the drag on the y-axis is very small. Looking at the drag from STK, the scale of the axis' are very different than from the modelled drag. It is approximately a factor two smaller than from modelling, even though a solar panel is considered in the model. This is visible by looking at the error, where the axis representing force of the figure has not changed scale, compared to Figure 4.5. It can be concluded, that the drag from modelling and STK are very different, however, from a control point of view, the drag from modelling is conservative since the true forces are much smaller.

In conclusion, it is found that the model of the system derived in this report is representative of an actual satellite in orbit about Earth. This conclusion is taken under the assumption that the model in STK can be considered a ground truth.
Now that the dynamics have been verified, the first controller to be addressed is the PD controller presented in Section 4.2.

4.2 PD Controller

The PD controller is implemented in Simulink, as seen in Figure 4.8. Here, the phasing of one satellite with respect to another is simulated. The dynamics of the two satellites are defined in the two central plants containing the gravitational dynamics of a spherical Earth, as seen in Equation 3.5, and the integration of these. Only satellite 2 is actuated. This actuation is only found in the direction of the track of the orbit.
The acceleration of the relative angle between the two satellites are calculated in the "Relative_angle" block, using the equations presented in Section 3.3.2. The position and velocity term of these equations are used to find the error between the references. These are now multiplied by a P and D gain and added to find the control signal.

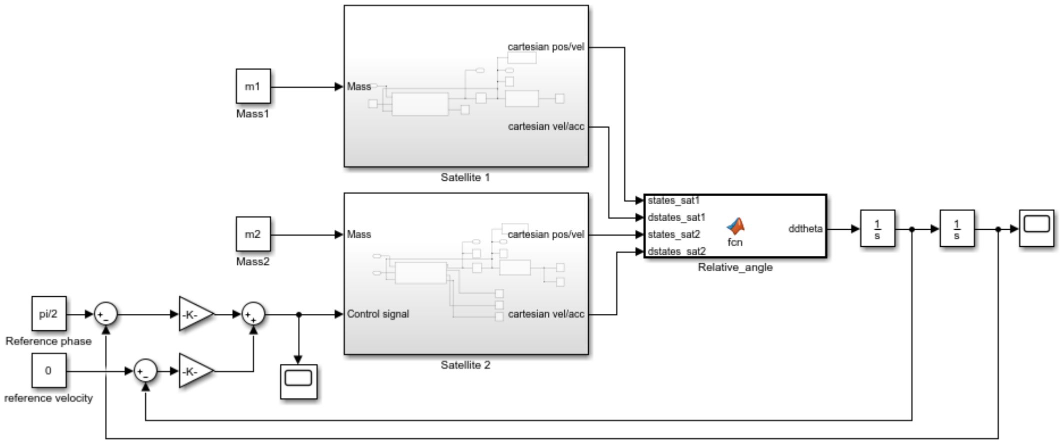


Figure 4.8: The implementation of the PD controller in the Simulink environment.

A test of the PD controller is now conducted. The conditions and results of this test are shown below.

Both satellites start at an initial phasing of 0. Satellite 2 will now seek to reach a reference phasing of $\pi/2$ radians or 90^{circ}.
The initial conditions for the satellites are seen in Table 4.1, where the vectors are given in the ECI frame.

		Satellite 1	Satellite 2
Position	x	0	0
	y	0	0
	z	$6921 \cdot 10^3 m$	$6921 \cdot 10^3 m$
Velocity	x	$7.593 \cdot 10^3 m/s$	$7.593 \cdot 10^3 m/s$
	y	0	0
	z	0	0

Table 4.1: Initial conditions of the satellites for PD control

This corresponds to an altitude of $550km$ and a velocity ensuring a stable circular orbit, found through Equation 3.14. A test will now be performed with the gains seen in Equation 4.1. The simulation is run for 20 days, corresponding to 1728000 time steps. Figure 4.9 shows the Hill x-, y-, and z-axis and control signal of the PD controller.

$$P = -0.001 \quad , \quad D = -100 \tag{4.1}$$

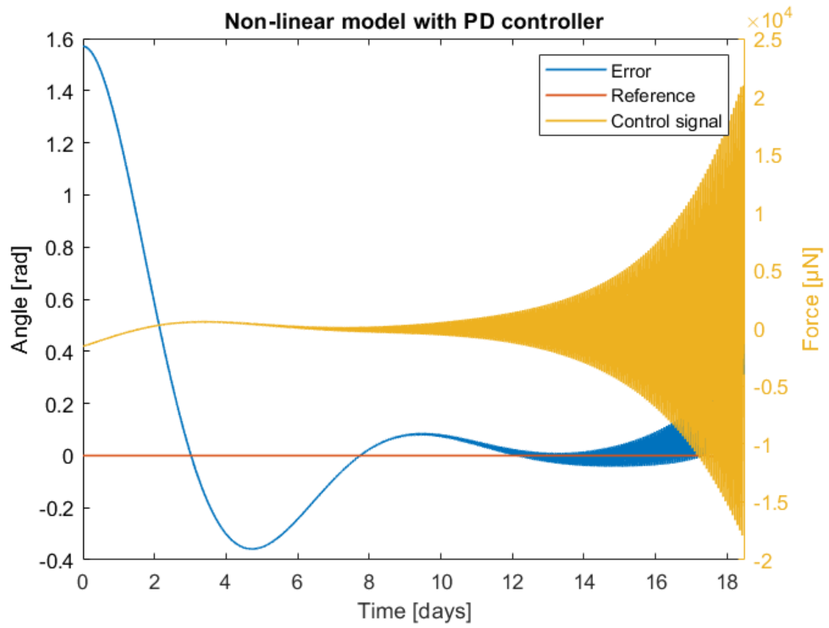


Figure 4.9: System response of the PD controller on the non-linear model

It can be seen that the system converges to the reference, but becomes unstable once too close. The control signal starts to oscillate around day five, which results in the system becoming unstable and overshooting the reference. Next, a test for LQR is conducted similarly.

4.3 LQR

The LQR controller is implemented and tested on two systems; the CW linearized system and the nonlinear system of relative equations of motion. This is done to firstly prove that the LQR is able to control the linearized system, and to show that this control law is also usable on the nonlinear system within certain bounds.

4.3.1 LQR Control of the CW Linearized System

The LQR implemented on the CW linearized system in the Simulink environment is seen in Figure 4.10. The CW equations of motion are used to define the plant seen in the center of the figure. The x- and y terms of the integral of the output of these equations, are used to compute the error to the reference. The error is multiplied with the optimal gain of the LQR, and the control signal is thereby determined. The control signal is a scalar, as the system is only actuated along the orbit in the y-direction of the local Hill frame.

The CW equations describe the relative motion between an actuated satellite and a reference satellite which is located at a phase of 90° from an unactuated satellite. The control system will seek to place the actuated satellite at the reference.

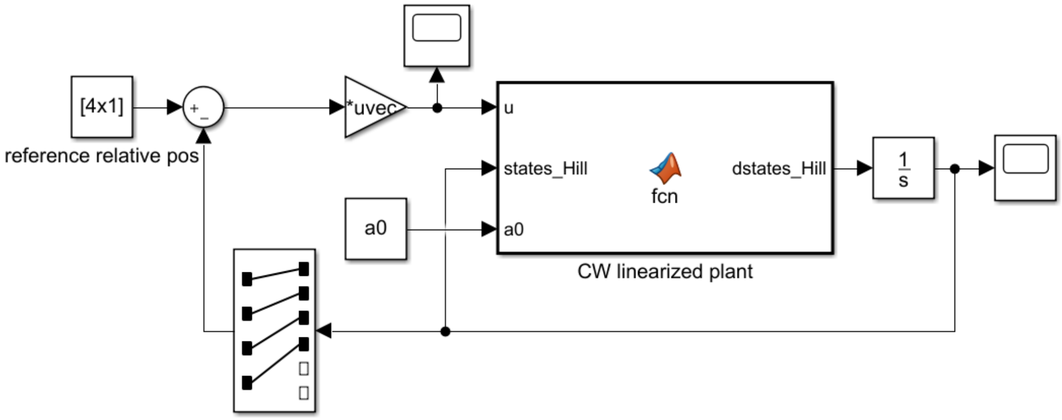


Figure 4.10: The LQR implemented on the CW linearized system in Simulink

A test is conducted to asses the control law found through the LQR approach. The initial conditions of the reference satellite and the actuated satellite is seen in Table 4.2. The position and velocity are given in a Hill frame fixed in the reference satellite.

		Actuated Satellite	Reference Satellite
Phasing		$\pi/2.001$	$\pi/2$
Position	x	-2.1 m	0
	y	-5433 m	0
	z	0	0
Velocity	x	0	0
	y	0	0
	z	0	0

Table 4.2: Initial conditions of the satellites for LQR control

The LQR is now used to place the actuated satellite at the reference. The cost function is defined with the following weights:

$$\underline{\mathbf{Q}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad , \quad R = 10^{19} \tag{4.2}$$

The results of this test is seen in Figure 4.11

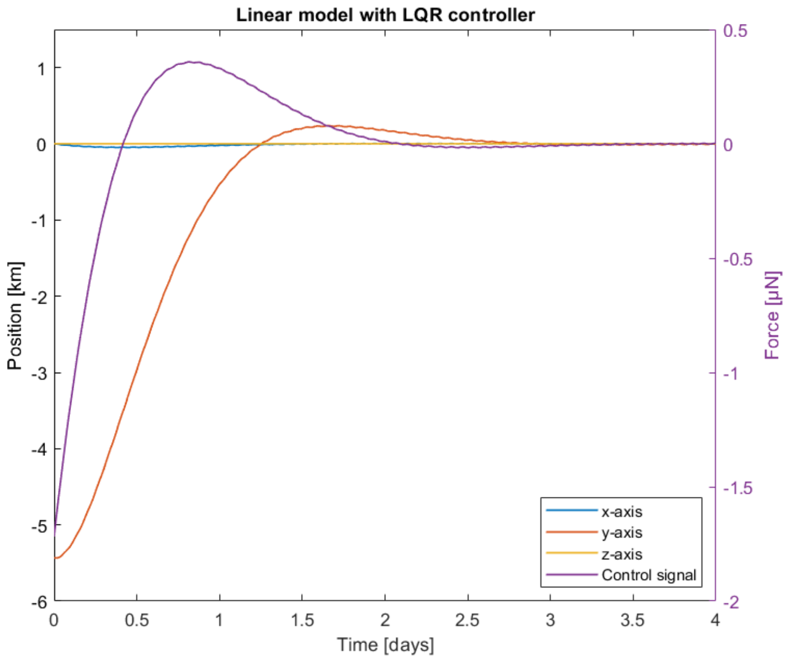


Figure 4.11: System response of the LQR controller on the linear model in Hill frame

It can be seen that the system converges to the reference and remains stable.

4.3.2 LQR Control of the Nonlinear System

The LQR is implemented on the nonlinear system as seen in Figure 4.12. The plant consists of the 10x1 state vector found in Section 3.3.1. In the same way as the linearized plant, the x- and y terms of the equations of motion are used to calculate the error term. This is multiplied by the optimal gain found through the LQR and used as control input. Once again, the system is only actuated along the orbit.

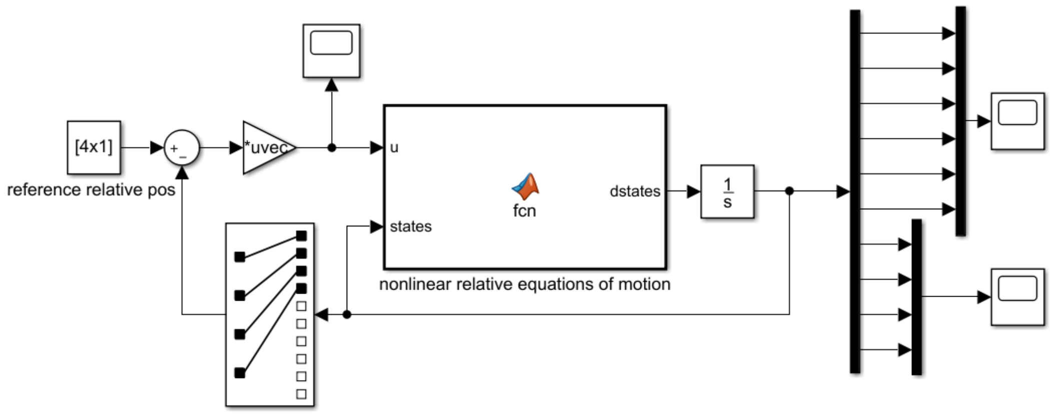


Figure 4.12: The LQR implemented on the nonlinear plant.

Two tests will now be presented showing the ability of the LQR to control the nonlinear system.

The first test of the LQR on the nonlinear system, will be conducted when the actuated satellite is only very slightly phase shifted from the reference satellite. The initial conditions of the two satellites will be the same as in the test of the LQR on the linearized system, seen in Table 4.2. The weights of the LQR will also be the same as the ones seen in Equation 4.2.

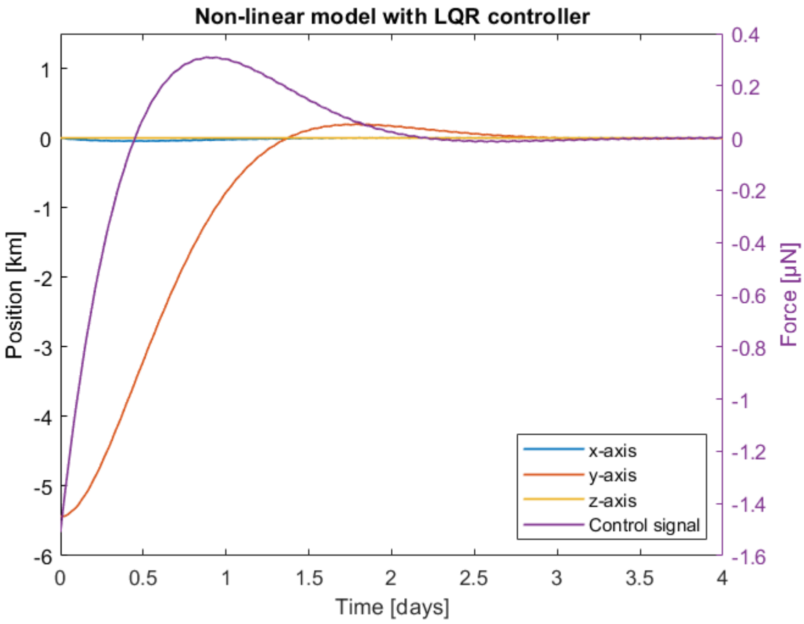


Figure 4.13: System response of the LQR controller on the non-linear model in Hill frame

It can be seen that the system converges to the reference and remains stable. The behavior of the system is very similar to the linear model in Figure 4.11, with only the axis representing force changing slightly. Hence, the initial conditions are within the area where the CW linearized equations still remain a valid approximation of the nonlinear system.

The second test is performed with a larger initial phase as to determine the performance of the LQR when the two satellites are further away from each other. The initial conditions of the two satellites are seen in Table 4.3. Once again, the position and velocity are presented in a Hill frame fixed to the reference satellite.

		Actuated Satellite	Reference Satellite
Phasing		$\pi/2.009$	$\pi/2$
Position	x	-171 m	0
	y	-48702 m	0
	z	0	0
Velocity	x	0	0
	y	0	0
	z	0	0

Table 4.3: Initial conditions of the satellites for LQR control, second test

The LQR is now used to control this system using the same weights as the previous test. The results are seen in Figure 4.14.

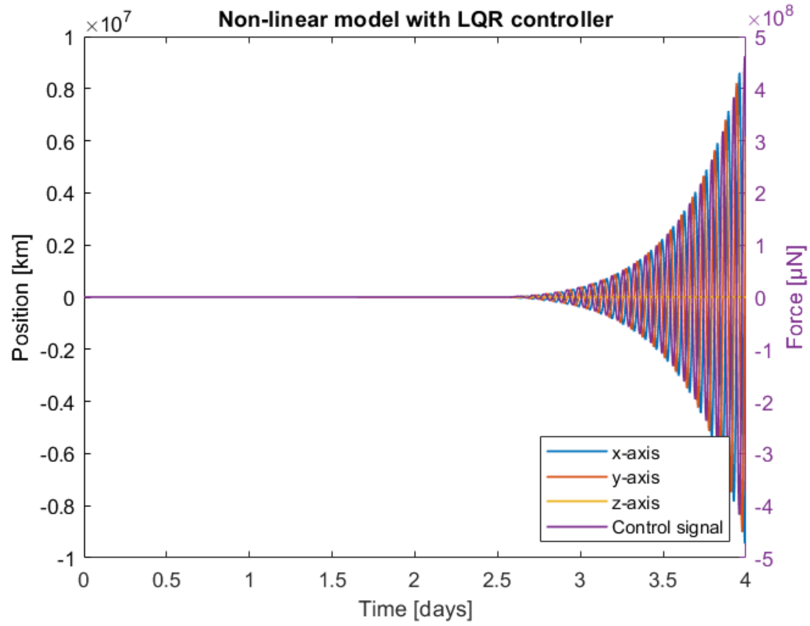


Figure 4.14: System response of the LQR controller on the non-linear model in Hill frame, with a reference further away

It can be seen that the system is unstable right from the initial state, and both the control signal and system states oscillates more and more as time goes. Hence, even at this small shift in phase, the CW linearization is no longer a valid approximation of the system and the LQR no longer works.

4.4 Lyapunov Controller

The Lyapunov controller is implemented in the simulation in much the same way as the LQR controller. One of the main differences is, how the reference is implemented. Instead of just giving a constant reference as for the LQR, the Lyapunov controller is given a reference trajectory from the initial state to the desired state of the system.
There are many ways of deriving such a reference trajectory, however, a straight forward approach would be to use the linearized system found through the CW equations controlled by the LQR.

The Lyapunov controller used here needs to actuate the system in both the x- and y-axis of the Hill frame, located at the actuated satellite to ensure stability. Therefore, the CW equations will now also include the possibility to actuate the system in the x-axis. Because of this, the B-vector of the state space equation will now be given as:

$$B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.3}$$

In addition to this, the R weight of the LQR will be given by a 2x2 matrix with the value of R on the diagonal.

4.4.1 Reference Generation Using CW Equations

In Figure 4.11, it is shown that the LQR stabilized CW equation produces a useful reference trajectory, given that the satellites start relatively close. However, the purpose of the Lyapunov controller is to control the system at any given initial phase of the two satellites.
The behaviour the LQR stabilized CW linearized system for larger phase shifts are therefore investigated, by testing with the initial conditions seen in Table 4.4. The weights of the LQR are given by:

$$\underline{\mathbf{Q}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \underline{\mathbf{R}} = \begin{bmatrix} 10^{21} & 0 \\ 0 & 10^{21} \end{bmatrix} \tag{4.4}$$

		Actuated Satellite	Reference Satellite
Phasing		$\pi/3$	$\pi/2$
Position	x	-927.2 km	0
	y	-3460.5 km	0
	z	0	0
Velocity	x	0	0
	y	0	0
	z	0	0

Table 4.4: Initial conditions of the satellites for Lyapunov control

The resulting reference trajectory is seen in Figure 4.15. Even though the system is stable and the reference is reached, this can not considered a useful reference trajectory, as there is a significant overshoot on the y-axis. Further tuning of the weighting parameters of the LQR were attempted, but did not mitigate the problem.
As another approach, the initial error in the x-axis were omitted when generating

the trajectory. The trajectory generated in this way is shown in Figure 4.16. This trajectory seems much more useful as a reference for the Lyapunov controller, hence it will be tested with the nonlinear system.

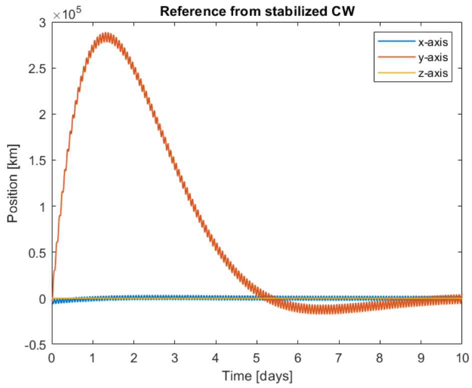


Figure 4.15: Reference generated by the stabilized CW system under an initial offset in both the x- and y direction.

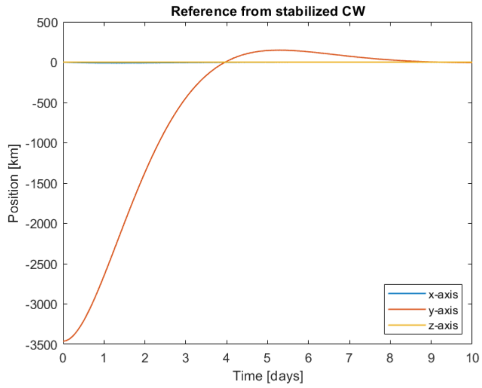


Figure 4.16: Reference generated by the stabilized CW system under an initial offset in only the y direction.

4.4.2 Lyapunov Controller with CW Reference

To test the reference generated using the CW equations, the Lyapunov controller is implemented in Simulink as seen in Figure 4.17.

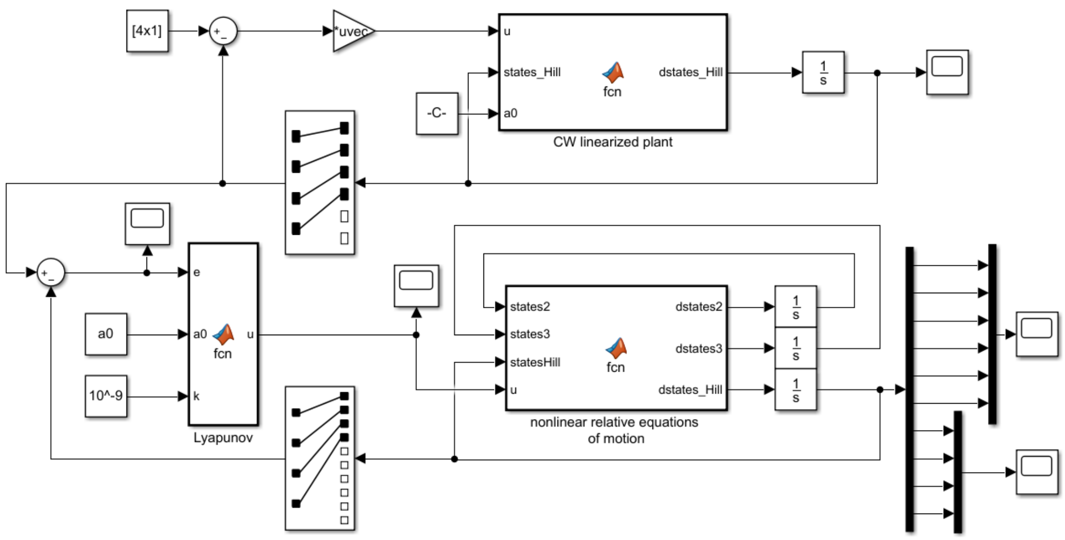


Figure 4.17: The implementation of the Lyapunov controller

Here, the plant is defined through the nonlinear equations of relative motion

given in Section 3.3.1. The error is calculated as the difference between the x- and y-term of the plant and of the reference trajectory. The reference is generated using the linearized CW plant as seen in the top of Figure 4.17. This is stabilized using the LQR controller.
Based on the error, the control signal is calculated using the control law given in Equation 3.67. The output of this function is the actuation of the plant in the x- and y-axis respectively.

For the test, the constant k of the Lyapunov control law is given as:

$$k = 10^{-9} \tag{4.5}$$

The initial conditions of the nonlinear plant are the same as used in the generation of the trajectory in Section 4.4.1. However, the reference is generated with the initial offset in x omitted. The result of the test is given in Figure 4.18.

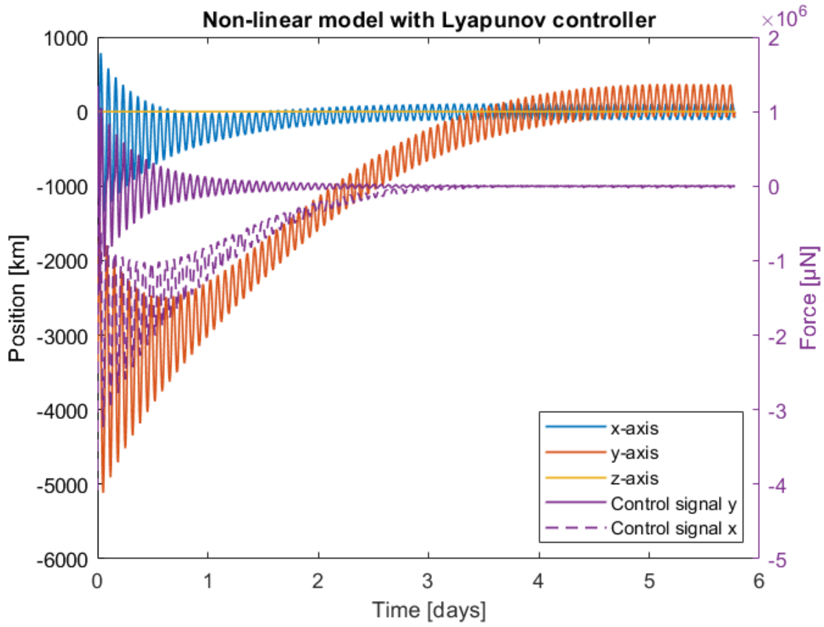


Figure 4.18: System response of the Lyapunov controller on the non-linear model in Hill frame

It can be seen that the system follows the reference and somewhat reaches the reference. However, oscillations are seen throughout the motion. These are largest at the start of the simulation, and somewhat dissipates as the system nears the reference. This oscillation is caused by the difference in initial offset in the x-direction between the reference and the nonlinear model. Initially, the reference assumes no offset along the x-axis, while the nonlinear system actually has an offset of $-927.2km$.

As the control law tries to correct this, oscillation are introduced to the system. Aside from causing an unwanted use of actuation, this oscillation also causes the simulation to crash after just less than six days.
Hence, this reference is not useful in the control of the nonlinear system.

As a way to address this problem, another reference is generated using a reduced set of CW equations.

4.4.3 Reference Generation Using Reduced CW Equations

In an attempt to generate a reference trajectory with no overshoot, even when subject to an initial offset in x, the CW equations of motion have been reduced to the following set of equations:

$$\begin{aligned} \ddot{x} - 2n_0\dot{y} &= 0 \\ \ddot{y} + 2n_0\dot{x} &= 0 \\ \ddot{z} + n_0^2 z &= 0 \end{aligned} \tag{4.6}$$

By removing some of the dynamics on the x-axis, the equations can no longer be considered a representative linearization of the nonlinear model, however, the system may still generate a useful reference trajectory.

As a test of whether a useful reference can be generated for the requirements set in Section 2.6.1, the system will start with a phase offset of $\pi/4$ and is tuned to reach 0 within 1.5 weeks. This is done as the maximum phase to be shifted is π and this must be done within six weeks, specified by the requirements ser in Section 2.6.1.

The initial conditions under which the reference is generated, is given by Table 4.5.

		Actuated Satellite	Reference Satellite
Phasing		$\pi/4$	$\pi/2$
Position	x	-2027.1 km	0
	y	-4893.9 km	0
	z	0	0
Velocity	x	0	0
	y	0	0
	z	0	0

Table 4.5: Initial conditions of the satellites for Lyapunov control, second test

The weights of the LQR are given by Equation 4.7 and the results are seen in Figure 4.19.

$$\underline{\mathbf{Q}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad , \quad \underline{\mathbf{R}} = \begin{bmatrix} 5 \cdot 10^{15} & 0 \\ 0 & 5 \cdot 10^{15} \end{bmatrix} \tag{4.7}$$

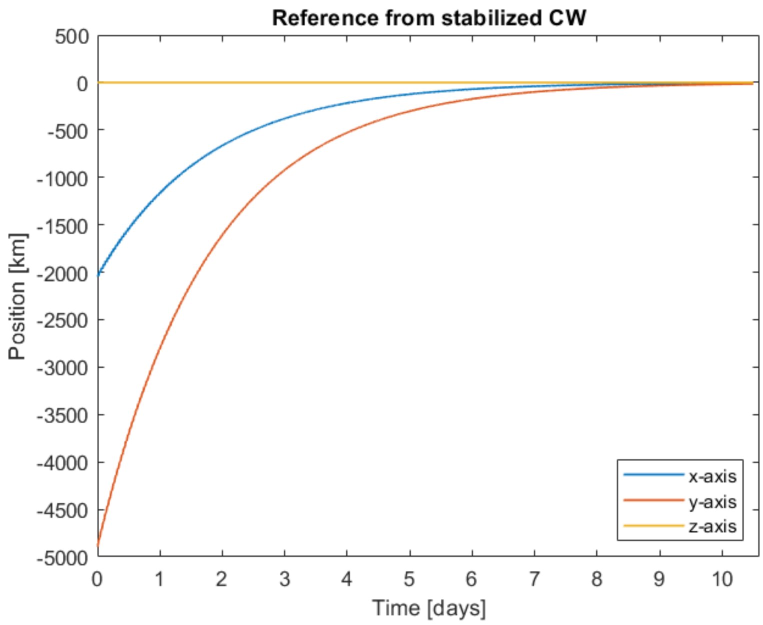


Figure 4.19: Reference trajectory generated from the reduced set of CW equations.

The trajectory looks promising and is therefore tested on the nonlinear system with $k = 10^{-9}$. The initial conditions remain the same. The results are seen in Figure 4.20.

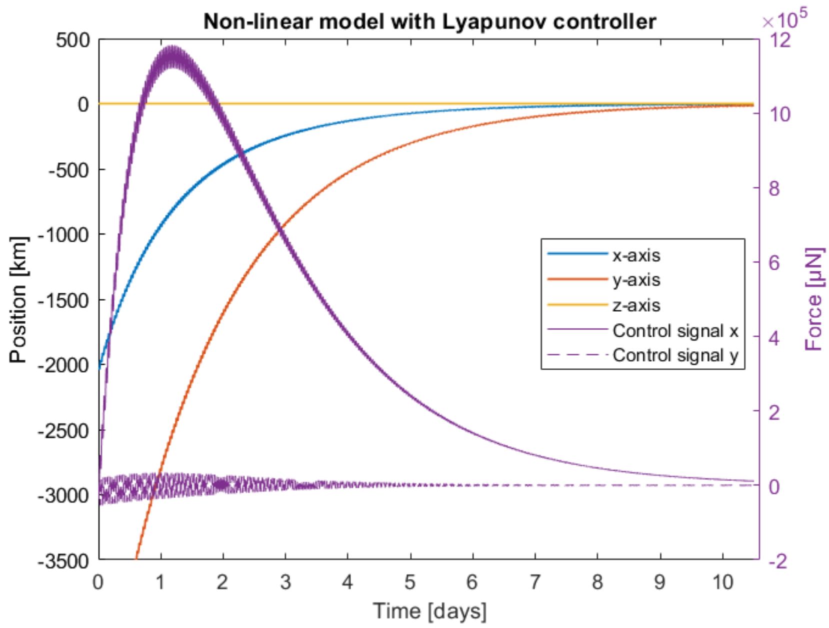


Figure 4.20: The nonlinear system tracking the reference from Figure 4.19

In the figure, it can be seen that the system tracks the reference and reaches the position of the reference satellite. However, in the process of doing this, a large amount of force is used along the x-axis. Further discussion of the results seen in this figure will be presented in Section 6.
From Figure 4.20, the system appears to be stable, however, to confirm this, an x-y plot of the state of the nonlinear system is plotted in Figure 4.21. Here, it can be seen that the system is stable as it follows the reference. A zoom of this plot is seen in Figure 4.22.

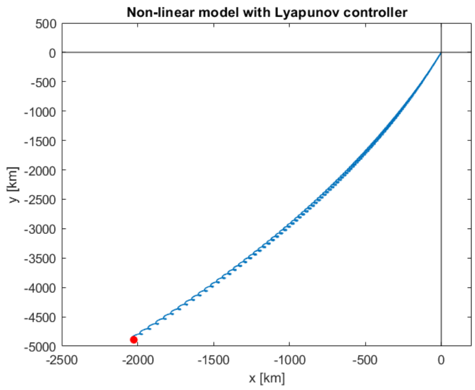


Figure 4.21: x-y plot of the states of the nonlinear system

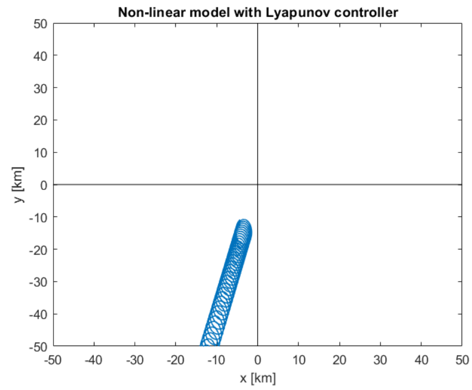


Figure 4.22: Zoomed version of the x-y plot of the states of the nonlinear system

4.4.4 Lyapunov Control with Nonlinear System Affected by Drag

The Lyapunov control law is designed to handle model inaccuracies, however, in the tests performed so far none have been present. As a proof of concept of the control laws ability to handle these, a test will be performed on the nonlinear equations of motion under the influence of drag as modelled in Section 3.2.2.
The initial conditions and reference will be the same as in the previous test, however the value of k is now given as $k = 10^{-5}$. The results of the test are shown in Figures 4.23, 4.24 and 4.25.

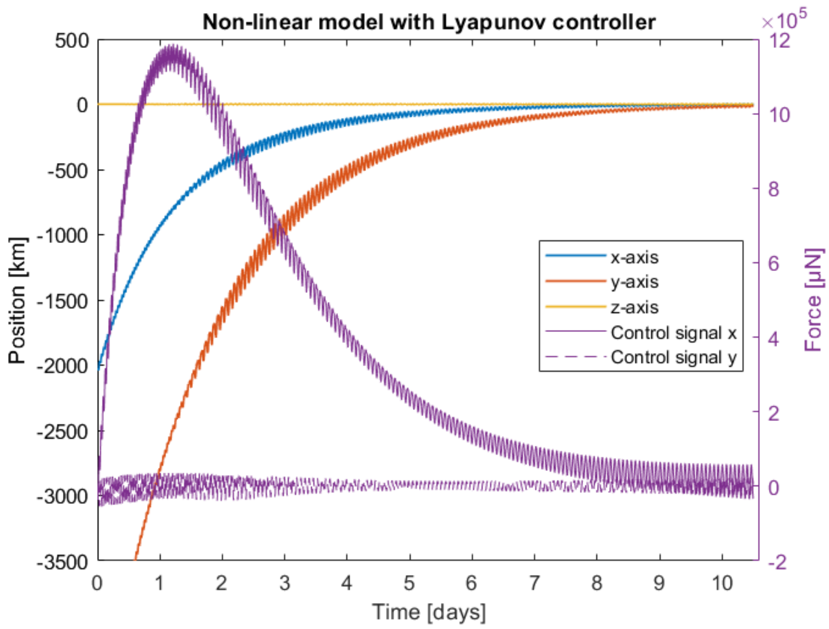


Figure 4.23: The nonlinear system affected by drag tracking the reference from Figure 4.19

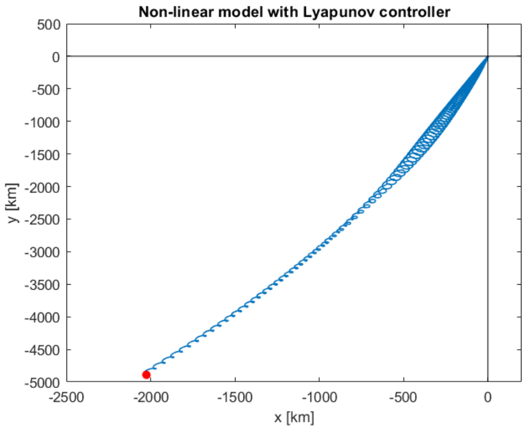


Figure 4.24: x-y plot of the states of the nonlinear system affected by drag.

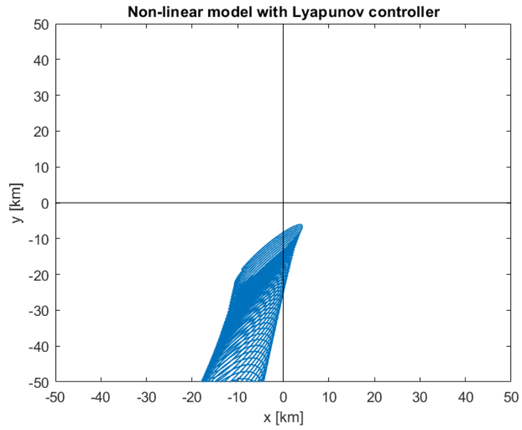


Figure 4.25: A zoom of the x-y plot.

From these figures, it can be concluded that the Lyapunov control law can handle the unmodelled dynamics introduced by the inclusion of the drag model in the plant.

Chapter 5

Conclusion

In this report, a problem proposed by Sternula and the research project MARIOT was investigated. MARIOT seeked to develop a satellite based maritime IoT network using VDES communications.

A problem analysis was done, containing how an orbit is defined, how a satellite is actuated, specifications of an antenna, and a presentation of existing satellite constellations which provide communication coverage around the world.
A satellite constellation roadmap was provided and investigated. It was found that in order to obtain a guaranteed revisit time of $95min$ in Arctic waters, one would need three planes with six satellites in total in a polar orbit. These results were obtained using the program AGI STK, where the satellites and antennas were modelled accordingly.
From this analysis, a control problem were identified in the phasing of satellites occupying the same orbit. A problem statement defining this problem were found, and a series of requirements to the solution of this were listed.
To solve this problem, the kinematics and dynamics of a satellite were described. Section 3.1 showed the Earth Centered Inertial Coordinate System (ECI) and Hill frame. Section 3.2 described the motion of a single satellite in polar coordinates through the Keplarian two-body problem. Gravitational perturbations and atmospheric drag was modelled due to their effect over time on a satellite in LEO. The gravitational perturbation modelled the J2 coefficient, however this was not used in the implementation. The atmospheric drag was modelled with the use of a density model publicly available by NASA.
Section 3.3 presents two approaches of describing the relative motion between two satellites. The first approach considers relative Cartesian motion in the Hill frame, and the second considers relative motion in a polar frame.
After the equations of motion had been described, several control approaches was presented using these. This included PD-, LQR-, and Lyapunov control, all of which was implemented and tested in Chapter 4. The modelled gravitational per-

turbation and atmospheric drag was verified in Section 4.1, where it was concluded that the gravity model could be considered accurate to a certain degree and the drag model was conservative.
A model was made in Simulink containing all kinematics, dynamics, equations of motion, and control approaches described earlier, which was used to test the reference following capabilities of the different approaches.
The PD controller showed that the satellite could move to a reference of 90° offset on the true anomaly, but would become unstable once the satellite got too close to the reference.
The LQR controller showed that by having the initial states of the satellite in close proximity to the reference, the system was stable and able to track the reference. However, the system became unstable once the initial states were further away from the reference.
The Lyapunov controller was aimed to have reference tracking capabilities both in close proximity and further away from the reference. The final test performed was with a reference of 45° offset on the true anomaly with and without drag, which showed that the system was stable and able to track the reference, however small oscillations were present when drag was introduced.
The problem statement:
How can phasing of grouped satellites in a single orbital plane be obtained?
is therefore fulfilled.

Overall, the implementation in a simulated environment was deemed successful.

Chapter 6

Discussion

In this chapter, the results presented in this paper will be discussed. Firstly the results of the testing of the PD-, LQR-, and Lyapunov controller will be discussed. The J2 perturbations and how they could be included in a control law will be addressed next. Finally, a few comments on different approaches that could have been investigated will be presented.

6.1 Implementation of Controllers

In this section, the results of the tests of the controllers will be discussed starting with the PD controller.

6.1.1 PD Controller

The results of the PD controller are seen in Figure 4.9. As previously stated the system seems to converge to zero, but then becomes unstable. One might argue that this problem could be addressed by further tuning of the PD controller, however, attempts at doing so showed no improvement in the results of the controller.

Another thing to keep in mind when considering the results of this controller, is the fact that it is tested on a system only affected by the gravity of a spherical Earth. Hence, any disturbances or other force acting upon the system could lead to increased instability of the system.
With that in mind, it should be noted that the PD controller does show promising results in the start of the simulation. Even though the two satellites are very far apart, the controller does close the gap in a very smooth fashion with limited overshoot. In addition to that, the forces required to do this are well within the limits of the actuator specified for the system.

Hence it would be interesting to continue development of a controller based on the PD controller. Especially using the polar equations of motion presented in Section 3.3.2, where a nonlinear control approach could prove much more suitable for the problem addressed in this report.

6.1.2 LQR Controller

The LQR controller is tested both on the linearized system as well as the nonlinear system. The results of the test on the linearized system is presented in Figures 4.11 and 4.15. From these figures, it can be concluded that the LQR controller does work for the linearized system of equations. However, the response is subject to both overshoot and oscillations as the system is moved away from the origin.
When tested on the nonlinear system, the same behaviour is seen. The LQR works when the satellite is close to the origin, but fails as it moves away.
For the purposes in which we want to use this controller, this is of course not suitable, however, for systems that operate in close proximity to one another, the CW equations and LQR stabilization is definitely a useful control approach.

When combining this with the nonlinear controller as a reference generator, the results were not acceptable either, however in other nonlinear control approaches, this linearized controller could be suitable.
The nonlinear control technique "Backstepping" is based on having a known stabilizing control law at the origin. With this as a starting point, the control law is extended to handle the nonlinearities introduced to the system as it moves away from the origin [18].
Given that the LQR is a stabilizing control law at the origin, Backstepping would be an obvious approach in the development of a control law.

6.1.3 Lyapunov Controller

The Lyapunov controller was tested with both the reference generated from the CW equations in Figure 4.18, and from the reduced CW equations in 4.20. The control law follows the reference and stabilizes the system in both cases, however, the smoothest motion is found with the trajectory generated by the reduced CW equations. This is to be expected as the one from the CW equations only generates a useful reference along one of the axes.
This does prove that the control law works, given a reasonable reference trajectory such as the one from the reduced CW equations. With that in mind, the performance of the controller is reliant on the trajectory which it follows. Since the trajectory is generated from a somewhat arbitrary system, the performance of the controller is not optimal. This is apparent when considering the control signal of Figure 4.20 together with the altitude of the satellite during this motion as seen in

Figure 6.1.

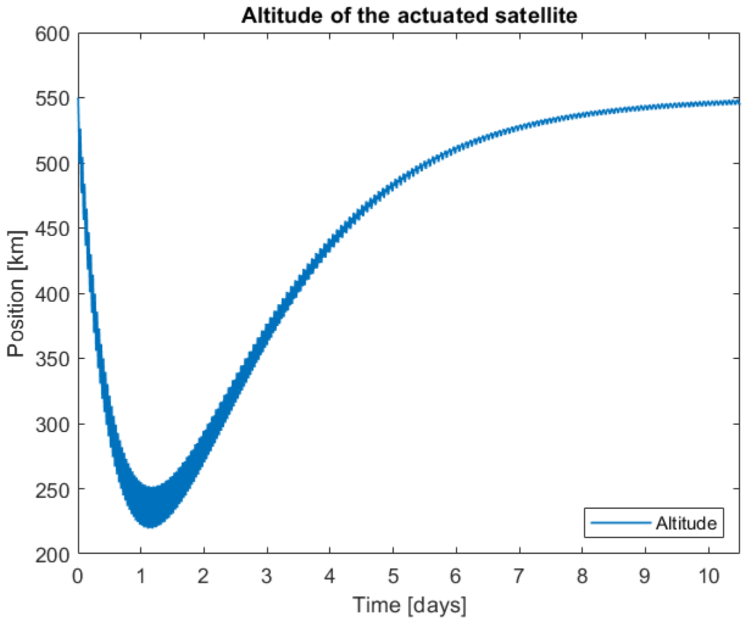


Figure 6.1: The altitude of the actuated satellite during the motion shown in Figure 4.20

The actuated satellite reaches the reference satellite by forcing itself into a lower orbit.
Equation 3.14 states that given a certain altitude, the satellite must orbit Earth with a certain angular velocity in order to maintain that altitude. This angular velocity is dependent on the linear velocity of the satellite. When the satellite is forced into a lower orbit while maintaining this linear velocity, it will attain a correspondingly higher angular velocity since the circumference of the orbit is now smaller than for a higher altitude.
This means that the satellite travels faster around Earth, closing the gap between itself and the reference satellite. As the acceleration forcing the satellite closer towards Earth is removed, the angular velocity will result in the satellite moving to a higher orbit, until Equation 3.14 is once again fulfilled.

From Figure 4.20, it can be seen that this approach works, however, it is not very efficient when considering the amount of thrust used. Hence, this control approach would not be recommended for practical use on an actual satellite.

6.2 The Effect of J2 Perturbations on the System

The J2 effect is described in Section 3.2.2, but is not included in any of the further work of the report. This was initially intended, but proved more difficult to account for than expected. However, a few approaches were investigated.
One of which were the Schweighart-Sedwick linearized equations of relative motion, which expands the CW equations to include the J2 perturbations [27]. These were implemented in the simulation, and a stabilizing LQR were developed. This was able to stabilize the linearized system, however, when tested on a nonlinear system which included the J2 effects, the LQR was not even stable at the origin.
Because of these disappointing results, further work with the J2 perturbations were abandoned, but future work with this project should take these into account.

Bibliography

[1] AGI. *Galileo Adds Accuracy and Reliability to IOT Operational Planning with STK*. https://www.agi.com/resources/case-studies/galileo. 2020.

[2] AGI. *THE AGI STORY*. https://www.agi.com/about/history. 2020.

[3] Waleed Alomar et al. "An Extendable Solar Array Integrated Yagi-Uda UHF Antenna for CubeSat Platforms". In: *IEEE International Symposium on Antennas and Propagation (APSURSI)* (2011). DOI: https://doi-org.zorac.aub.aau.dk/10.1109/APS.2011.5997166.

[4] Kyle T. Alfriend et al. *Spacecraft Formation Flying - Dynamics, control and navigation*. 1st ed. Elsevier Ltd., 2010.

[5] Laura M. Bradbury et al. "NorSat-2: Enabling advanced maritime communication with VDES". In: *Acta Astronautica* 156 (2019), pp. 44–50. DOI: https://doi.org/10.1016/j.actaastro.2018.10.030.

[6] Teresa M. Braun. "Satellite Communications Payload and Systems". In: John Wiley and Sons Incorporated, 2012. Chap. 3 Antenna, pp. 33–79.

[7] NASA Glenn Research Center. *Earth Atmosphere Model*. https://www.grc.nasa.gov/WWW/k-12/airplane/atmosmet.html. Accessed: 2020-05-14. 2015.

[8] Wikimedia Commons. *Dipole xmting antenna animation*. https://commons.wikimedia.org/wiki/File:Dipole_xmting_antenna_animation_4_408x318x150ms.gif. [Online; accessed March 2020]. 2015.

[9] T. Eriksen et al. "In-orbit AIS performance of the Norwegian microsatellites NorSat-1 and NorSat-2". In: *CEAS Space Journal* (2019). DOI: https://doi.org/10.1007/s12567-019-00289-1.

[10] ESA. *Alphasat: a double-decker in space*. https://esamultimedia.esa.int/docs/technology/Alphasat_factsheet.pdf. 2013.

[11] ESA. *VDE-SAT Downlink Verification*. https://artes.esa.int/projects/vde-sat-downlink-verification. [Online; accessed March 2020]. 2018.

[12] Geoborders. *Iridium - How it works*. http://www.iridium.it/en/iridium.htm. 2008.

[13] Wen Geyi. "Foundations of Applied Electrodynamics". In: John Wiley and Sons Incorporated, 2010. Chap. 4 Antenna Theory, pp. 153–202.

[14] IALA. *VHF DATA EXCHANGE SYSTEM (VDES) OVERVIEW*. `www.iala-aism.org`. 2017.

[15] IMO. *E-navigation*. `http://www.imo.org/en/OurWork/Safety/Navigation/Pages/eNavigation.aspx`. 2020.

[16] Inmarsat. *Our satellites*. `https://www.inmarsat.com/about-us/our-technology/our-satellites/`. 2020.

[17] Iridium. *Did you know?* `https://www.iridiumnext.com/`. 2020.

[18] Hassan K. Khalil. *Nonlinear Systems Third Edition*. 3rd ed. Prentice Hall, 2002.

[19] Wiley J. Larson and James R. Wertz. *Space Mission Analysis and Design*. Microcosm Press and Kluwer Academic Publishers, 2005.

[20] exactEarth Ltd. *exactEarth and Harris Corporation Form Strategic Alliance to Provide Real-Time Global Maritime Tracking and Information Solutions*. `http://investors.exactearth.com/2015-06-08-exactEarth-and-Harris-Corporation-Form-Strategic-Alliance-to-Provide-Real-Time-Global-Maritime-Tracking-and-Information-Solutions`. 2018.

[21] exactEarth Ltd. *exactView RT Powered by Harris*. `https://www.exactearth.com/technology/exactview-rt-powered-by-harris`. 2020.

[22] Francisco Lázaro et al. "VHF Data Exchange System (VDES): an enabling technology for maritime communications". In: *CEAS Space Journal* 11 (2019), pp. 55–63. DOI: `https://doi.org/10.1007/s12567-018-0214-8`.

[23] Oliver Montenbruck and Eberhard Gill. *Satellite Orbits Models, Methods, and Applications*. Springer, 2000.

[24] Earth Observation Portal. *Iridium NEXT*. `https://directory.eoportal.org/web/eoportal/satellite-missions/i/iridium-next`. 2020.

[25] Norsk Romsenter. *NORWAY'S SATELLITES*. `https://www.romsenter.no/eng/Norway-in-Space/Norway-s-Satellites`. [Online; accessed March 2020]. 2016.

[26] Anton H. de Ruiter, Christopher Damaren, and James R. Forbes. "Spacecraft Dynamics and Control: An Introduction". In: John Wiley and Sons, Ltd, 2013. Chap. 7: Orbital Perturbations, pp. 149–181.

[27] Samuel A. Schweighart and Raymond J. Sedwick. "Development and Analysis of a High Fidelity Linearized J2 Model for Satellite Formation". In: *American Institute of Aeronautics and Astronautics (AIAA)* (2001).

[28] UNCTAD. *Review of Maritime Transport 2019*. `https://unctad.org/en/PublicationsLibrary/rmt2019_en.pdf`. 2020.

[29] Wikipedia. *Attitude control*. https://en.wikipedia.org/wiki/Attitude_control. 2020.

[30] Wikipedia. *Cold gas thruster*. https://en.wikipedia.org/wiki/Cold_gas_thruster. 2020.

[31] Wikipedia. *Electrostatic ion thruster*. https://da.wikipedia.org/wiki/Fil:Electrostatic_ion_thruster-en.svg. 2020.

[32] Wikipedia. *Ion thruster*. https://en.wikipedia.org/wiki/Ion_thruster. 2020.

[33] Wikipedia. *Magnetorquer*. https://en.wikipedia.org/wiki/Magnetorquer. 2020.

[34] Wikipedia. *NanoSail-D2*. https://en.wikipedia.org/wiki/NanoSail-D2. 2020.

[35] Wikipedia. *Orbit*. https://en.wikipedia.org/wiki/Orbit. 2020.

[36] Wikipedia. *Orbital elements*. https://en.wikipedia.org/wiki/Orbital_elements. 2020.

[37] Wikipedia. *Plasma propulsion engine*. https://en.wikipedia.org/wiki/Plasma_propulsion_engine. 2020.

[38] Wikipedia. *Reaction wheel*. https://en.wikipedia.org/wiki/Reaction_wheel. 2020.

[39] Wikipedia. *Rocket engine*. https://en.wikipedia.org/wiki/Rocket_engine. 2020.

[40] Wikipedia. *Satellite constellation*. https://en.wikipedia.org/wiki/Satellite_constellation. 2020.

[41] Wikipedia. *Spacecraft propulsion*. https://en.wikipedia.org/wiki/Spacecraft_propulsion. 2020.

[42] Wikipedia. *Spacecraft propulsion Without internal reaction mass*. https://en.wikipedia.org/wiki/Spacecraft_propulsion#Without_internal_reaction_mass. 2020.

[43] Wikipedia, the free encyclopedia. *Iridium Coverage Animation*. https://commons.wikimedia.org/wiki/File:Iridium_Coverage_Animation.gif. [Online; accessed March 2020]. 2011.

[44] Rafal Wisniewski. "Lecture Notes on Modeling of a Spacecraft". Aalborg University, Department of Proceskontrol. 2000.

Appendix A

Attitude Actuators

A magnetorquer, see Figure A.1, is an electromagnet coil, which uses Earth's electromagnet field to control the attitude of a satellite. This is done by switching current flow through the coil on and off, producing counter forces which provide torque in the desired direction. This means that no mass is consumed, and there are no moving parts, making it very reliable, and it can in theory work indefinitely. It is lightweight and energy efficient, however, it has a slow response time and is dependent on Earth's magnetic field, meaning it will have a maximum output torque, limited by the position of the satellite around Earth.

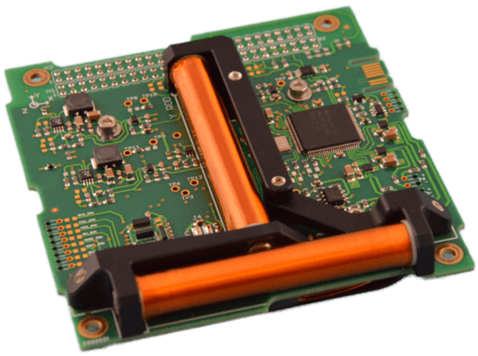


Figure A.1: A magnetorquer for attitude control of a satellite [33]

A table showcasing the advantages and disadvantages of magnetorquers is seen in Table A.1.

Magnetorquers	
Pros	No expendable propellant
	Energy efficient
	Reliable
	Lightweight
Cons	High power for quick response
	Limited torque
	Slow response time
	Dependency on strength of Earth's magnet field

Table A.1: Advantages and disadvantages of a magnetorquer as attitude actuator

Having 3 magnetorquers, each on their own axis in the inertial frame, it is possible to orientate a satellite in any way.

Reaction Wheel

A reaction wheel, see Figure A.2, is a wheel that can be rotated to produce a torque in a desired direction. It consists of an electrical motor attached to a flywheel, which the speed of can be controlled. Changing the rotational speed of the flywheel, causes the satellite to counter rotate. This is done through conservation of angular momentum. It uses electricity to produce a torque, and can be used as a momentum wheel to store rotational energy. This is done by operating the wheel near a constant speed, altering the rotational dynamics of a satellite, so that the disturbances perpendicular to one axis of the satellite, do not result directly in angular motion about the same axis as the disturbance. A reaction wheel is useful for rotating small amounts, and is very good for maintaining orientation and account for disturbances. It does not consume mass to actuate.

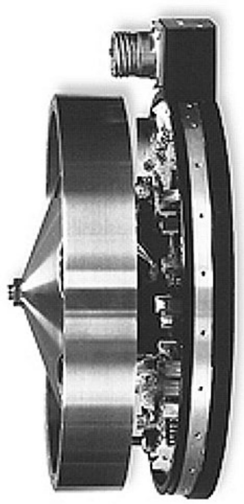


Figure A.2: A reaction wheel for attitude control of a satellite [38]

A table showcasing the advantages and disadvantages of reaction wheels are shown in Table A.2.

Reaction wheel	
Pros	No expendable propellant
	Energy efficient
	Wide range of torques
	Fast response time
	Lightweight
Cons	Moving parts
	Saturation of stored energy

Table A.2: Advantages and disadvantages of a reaction wheel as attitude actuator

Having 3 reaction wheels, each on their own axis in the inertial frame, it is possible to orientate a satellite in any way. However, this will store the rotational energy in the flywheel, which has a physical limit of how much energy can be stored, due to the maximum rotational speed of the wheel. This is called saturation, and other attitude actuators can help cancelling this effect.

Cold Gas Thruster

A cold gas thruster generates torque with the use of the expansion of a pressurized gas. Compared to a combustion engine, a cold gas thruster does not house any combustion, resulting in a lower thrust and efficiency. This propulsion type is the most cheap, reliable, and simple available for maneuvering and attitude control. A

cold gas thruster, consists only of a fuel tank, a regulating valve, and a propelling nozzle.
Furthermore, they are used for smaller space missions, and specifically focused on CubeSats due to them having strict regulations against pyrotechnics and hazardous materials. A cold gas thruster can be used for both attitude actuation and orbit actuation.

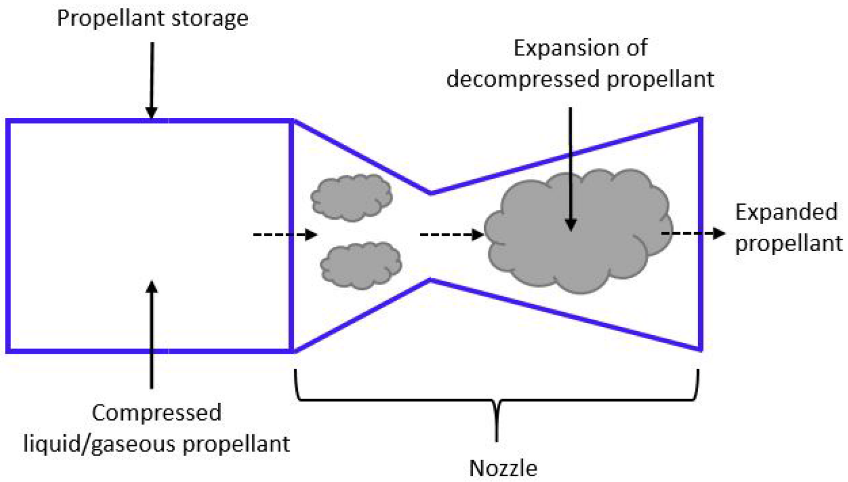


Figure A.3: A cold gas thruster and how it produces a torque [30]

A table showcasing the advantages and disadvantages of cold gas thrusters are shown in Table A.3.

Cold Gas Thrusters	
Pros	Quick response
	Reliable and cheap
	Simple and small
	Low electricity to operate
Cons	Expendable propellant
	Low thrust and efficiency
	Thrust decreases over time

Table A.3: Advantages and disadvantages of a cold gas thruster as attitude actuator

Appendix B

Basic Antenna Theory

A basic antenna consists of a dipole which essentially is a metal rod with a gap in the middle and a positive charge in one end and a negative charge in the other. When transmitting, an oscillating voltage is applied to the rod moving the charges from one end of the rod to the other. As this happens the electric field varies and when the negative charge crosses the positive, the electric field is separated from the rod and propagates into free space. An illustration of this behaviour is seen in Figure B.1.

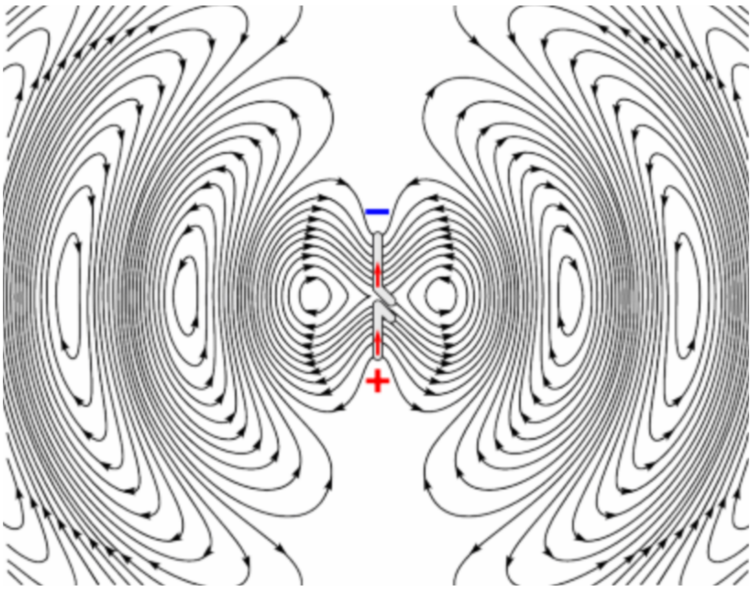


Figure B.1: The electrical field surrounding a dipole during transmission [8]

When receiving a signal, the electromagnetic wave is transferred to an electric current as the wave passes the dipole. The wave induces a current in the rod as the wave passes over it.

When transmitting electromagnetic waves, the frequency of the oscillating voltage of the antenna determines the frequency of the wave being generated which in turn determines the wavelength of signal. To get an optimal radio wave, the physical size of the antenna must be designed in accordance with the wavelength. This is also important when designing an antenna for reception of a signal with a given wavelength.

B.1 Yagi-Uda Antenna

A Yagi-Uda antenna is a high gain, directional antenna consisting of a single driven element and several parasitic elements. As the electromagnetic wave generated by the driven element passes over the parasitic elements, a current is induced in these. This results in the parasitic elements also radiating electromagnetic waves. The elements known as directors in front of the driven element are placed at a distance where the waves generated by these elements interact with the waves of the driven element, amplifying the wave through constructive interference. In the opposite manner, the reflector element placed behind the driven element is placed such that the waves emitted cause destructive interference dampening the signal. A sketch of a simple Yagi-Uda antenna is seen in Figure B.2:

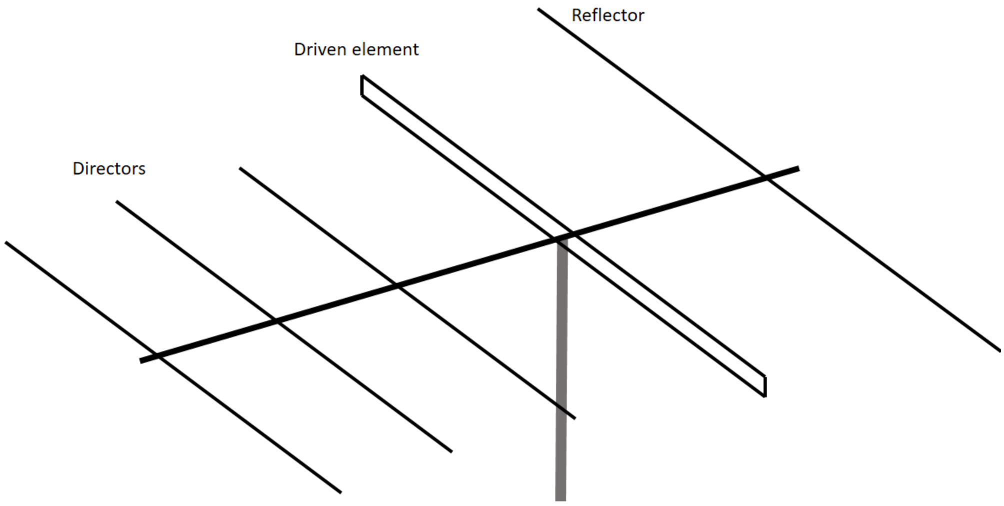


Figure B.2: Sketch of a Yagi-Uda antenna

The positioning and length of the parasitic elements of the Yagi-Uda antenna are crucial in the performance of the antenna, however, this will not be further adressed in this report.

Appendix C

First and Second Derivative of Relative Angle

$\dot{\theta}$ is equal to:

```
thetadot = -(((diff(q1(t), t) * p1(t) + q1(t) * diff(p1(t), t) +
diff(q2(t), t) * p2(t) + q2(t) * diff(p2(t), t) + diff(q3(t), t) *
p3(t) + q3(t) * diff(p3(t), t)) * (q1(t) ^ 2 + q2(t) ^ 2 + q3(t) ^
2) ^ (-0.1e1 / 0.2e1) * (p1(t) ^ 2 + p2(t) ^ 2 + p3(t) ^ 2) ^
(-0.1e1 / 0.2e1)) - ((q1(t) * p1(t) + q2(t) * p2(t) + q3(t) *
p3(t)) * (q1(t) ^ 2 + q2(t) ^ 2 + q3(t) ^ 2) ^ (-0.3e1 / 0.2e1) *
(p1(t) ^ 2 + p2(t) ^ 2 + p3(t) ^ 2) ^ (-0.1e1 / 0.2e1) * (2 * q1(t) *
diff(q1(t), t) + 2 * q2(t) * diff(q2(t), t) + 2 * q3(t) *
diff(q3(t), t))) / 0.2e1 - ((q1(t) * p1(t) + q2(t) * p2(t) +
q3(t) * p3(t)) * (q1(t) ^ 2 + q2(t) ^ 2 + q3(t) ^ 2) ^
(-0.1e1 / 0.2e1) * (p1(t) ^ 2 + p2(t) ^ 2 + p3(t) ^ 2) ^
(-0.3e1 / 0.2e1) * (2 * p1(t) * diff(p1(t), t) + 2 * p2(t) *
diff(p2(t), t) + 2 * p3(t) * diff(p3(t), t))) / 0.2e1) *
((-(q1(t) * p1(t) + q2(t) * p2(t) + q3(t) * p3(t)) ^ 2 / (q1(t) ^
2 + q2(t) ^ 2 + q3(t) ^ 2) / (p1(t) ^ 2 + p2(t) ^ 2 + p3(t) ^ 2) +
1) ^ (-0.1e1 / 0.2e1));
```

$\ddot{\theta}$ is equal to:

```
thetadotdot = -(((diff(diff(q1(t), t), t) * p1(t) + 2 *
diff(q1(t), t) * diff(p1(t), t) + q1(t) * diff(diff(p1(t), t), t) +
diff(diff(q2(t), t), t) * p2(t) + 2 * diff(q2(t), t) *
diff(p2(t), t) + q2(t) * diff(diff(p2(t), t), t) +
diff(diff(q3(t), t), t) * p3(t) + 2 * diff(q3(t), t) * diff(p3(t), t) +
q3(t) * diff(diff(p3(t), t), t)) * (q1(t) ^ 2 + q2(t) ^ 2 +
q3(t) ^ 2) ^ (-0.1e1 / 0.2e1) * (p1(t) ^ 2 + p2(t) ^ 2 + p3(t) ^ 2) ^
```

```
(-0.1e1 / 0.2e1)) - ((diff(q1(t), t) * p1(t) + q1(t) *
diff(p1(t), t) + diff(q2(t), t) * p2(t) + q2(t) *
diff(p2(t), t) + diff(q3(t), t) * p3(t) + q3(t) *
diff(p3(t), t)) * (q1(t) ^ 2 + q2(t) ^ 2 +
q3(t) ^ 2) ^ (-0.3e1 / 0.2e1) * (p1(t) ^ 2 + p2(t) ^ 2 +
p3(t) ^ 2) ^ (-0.1e1 / 0.2e1) * (2 * q1(t) * diff(q1(t), t) +
2 * q2(t) * diff(q2(t), t) + 2 * q3(t) * diff(q3(t), t))) -
((diff(q1(t), t) * p1(t) + q1(t) * diff(p1(t), t) +
diff(q2(t), t) * p2(t) + q2(t) * diff(p2(t), t) +
diff(q3(t), t) * p3(t) + q3(t) * diff(p3(t), t)) *
(q1(t) ^ 2 + q2(t) ^ 2 + q3(t) ^ 2) ^ (-0.1e1 / 0.2e1) *
(p1(t) ^ 2 + p2(t) ^ 2 + p3(t) ^ 2) ^ (-0.3e1 / 0.2e1) *
(2 * p1(t) * diff(p1(t), t) + 2 * p2(t) * diff(p2(t), t) +
2 * p3(t) * diff(p3(t), t))) + 0.3e1 / 0.4e1 *
(q1(t) * p1(t) + q2(t) * p2(t) + q3(t) * p3(t)) *
((q1(t) ^ 2 + q2(t) ^ 2 + q3(t) ^ 2) ^ (-0.5e1 / 0.2e1)) *
((p1(t) ^ 2 + p2(t) ^ 2 + p3(t) ^ 2) ^ (-0.1e1 / 0.2e1)) *
((2 * q1(t) * diff(q1(t), t) + 2 * q2(t) * diff(q2(t), t) +
2 * q3(t) * diff(q3(t), t)) ^ 2) + ((q1(t) * p1(t) +
q2(t) * p2(t) + q3(t) * p3(t)) * (q1(t) ^ 2 + q2(t) ^ 2 +
q3(t) ^ 2) ^ (-0.3e1 / 0.2e1) * (p1(t) ^ 2 + p2(t) ^ 2 +
p3(t) ^ 2) ^ (-0.3e1 / 0.2e1) * (2 * q1(t) *
diff(q1(t), t) + 2 * q2(t) * diff(q2(t), t) + 2 * q3(t) *
diff(q3(t), t)) * (2 * p1(t) * diff(p1(t), t) + 2 *
p2(t) * diff(p2(t), t) + 2 * p3(t) * diff(p3(t), t))) /
0.2e1 - ((q1(t) * p1(t) + q2(t) * p2(t) + q3(t) *
p3(t)) * (q1(t) ^ 2 + q2(t) ^ 2 + q3(t) ^ 2) ^
(-0.3e1 / 0.2e1) * (p1(t) ^ 2 + p2(t) ^ 2 + p3(t) ^ 2) ^
(-0.1e1 / 0.2e1) * (2 * diff(q1(t), t) ^ 2 + 2 * q1(t) *
diff(diff(q1(t), t), t) + 2 * diff(q2(t), t) ^ 2 + 2 *
q2(t) * diff(diff(q2(t), t), t) + 2 * diff(q3(t), t) ^
2 + 2 * q3(t) * diff(diff(q3(t), t), t))) / 0.2e1 +
0.3e1 / 0.4e1 * (q1(t) * p1(t) + q2(t) * p2(t) +
q3(t) * p3(t)) * ((q1(t) ^ 2 + q2(t) ^ 2 + q3(t) ^ 2) ^
(-0.1e1 / 0.2e1)) * ((p1(t) ^ 2 + p2(t) ^ 2 +
p3(t) ^ 2) ^ (-0.5e1 / 0.2e1)) * ((2 * p1(t) *
diff(p1(t), t) + 2 * p2(t) * diff(p2(t), t) + 2 *
p3(t) * diff(p3(t), t)) ^ 2) - ((q1(t) * p1(t) + q2(t) *
p2(t) + q3(t) * p3(t)) * (q1(t) ^ 2 + q2(t) ^ 2 +
q3(t) ^ 2) ^ (-0.1e1 / 0.2e1) * (p1(t) ^ 2 +
p2(t) ^ 2 + p3(t) ^ 2) ^ (-0.3e1 / 0.2e1) * (2 *
```

```
diff(p1(t), t) ^ 2 + 2 * p1(t) *
diff(diff(p1(t), t), t) + 2 * diff(p2(t), t) ^ 2 +
2 * p2(t) * diff(diff(p2(t), t), t) + 2 *
diff(p3(t), t) ^ 2 + 2 * p3(t) * diff(diff(p3(t), t), t))) /
0.2e1) * ((-(q1(t) *
p1(t) + q2(t) * p2(t) + q3(t) * p3(t)) ^ 2 / (q1(t) ^ 2 +
q2(t) ^ 2 + q3(t) ^ 2) / (p1(t) ^ 2 + p2(t) ^ 2 + p3(t) ^ 2) +
1) ^ (-0.1e1 / 0.2e1)) - ((q1(t) * p1(t) + q2(t) * p2(t) +
q3(t) * p3(t)) * ((q1(t) ^ 2 + q2(t) ^ 2 + q3(t) ^ 2) *
(p1(t) * (q2(t) * p2(t) + q3(t) * p3(t)) - (p2(t) ^ 2 +
p3(t) ^ 2) * q1(t)) * diff(p1(t), t) - (q1(t) ^ 2 + q2(t) ^
2 + q3(t) ^ 2) * (p1(t) ^ 2 * q2(t) - p1(t) * p2(t) * q1(t) -
p2(t) * p3(t) * q3(t) + p3(t) ^ 2 * q2(t)) * diff(p2(t), t) -
(q1(t) ^ 2 + q2(t) ^ 2 + q3(t) ^ 2) * (p1(t) ^ 2 * q3(t) -
p1(t) * p3(t) * q1(t) + p2(t) * (p2(t) * q3(t) - p3(t) *
q2(t))) * diff(p3(t), t) + (p1(t) ^ 2 + p2(t) ^ 2 + p3(t) ^ 2) *
(((-q2(t) ^ 2 - q3(t) ^ 2) * p1(t) + q1(t) * (q2(t) * p2(t) +
q3(t) * p3(t))) * diff(q1(t), t) + (p1(t) * q1(t) * q2(t) +
(-q1(t) ^ 2 - q3(t) ^ 2) * p2(t) + p3(t) * q2(t) * q3(t)) *
diff(q2(t), t) + diff(q3(t), t) * (p1(t) * q1(t) * q3(t) +
p2(t) * q2(t) * q3(t) - p3(t) * (q1(t) ^ 2 + q2(t) ^ 2)))) *
(((-diff(q1(t), t) * p1(t) - diff(q2(t), t) * p2(t) -
diff(q3(t), t) * p3(t) - q1(t) * diff(p1(t), t) - q2(t) *
diff(p2(t), t) - q3(t) * diff(p3(t), t)) * (p1(t) ^ 2 + p2(t) ^
2 + p3(t) ^ 2) ^ (-0.1e1 / 0.2e1) + (p1(t) ^ 2 + p2(t) ^ 2 +
p3(t) ^ 2) ^ (-0.3e1 / 0.2e1) * (p1(t) * diff(p1(t), t) + p2(t) *
diff(p2(t), t) + p3(t) * diff(p3(t), t)) * (q1(t) * p1(t) +
q2(t) * p2(t) + q3(t) * p3(t))) * (q1(t) ^ 2 + q2(t) ^ 2 +
q3(t) ^ 2) ^ (-0.1e1 / 0.2e1) + (q1(t) ^ 2 + q2(t) ^ 2 + q3(t) ^
2) ^ (-0.3e1 / 0.2e1) * (p1(t) ^ 2 + p2(t) ^ 2 + p3(t) ^ 2) ^
(-0.1e1 / 0.2e1) * (q1(t) * diff(q1(t), t) + q2(t) *
diff(q2(t), t) + q3(t) * diff(q3(t), t)) * (q1(t) * p1(t) +
q2(t) * p2(t) + q3(t) * p3(t))) * (-(q1(t) * p1(t) + q2(t) *
p2(t) + q3(t) * p3(t)) ^ 2 / (q1(t) ^ 2 + q2(t) ^ 2 + q3(t) ^
2) / (p1(t) ^ 2 + p2(t) ^ 2 + p3(t) ^ 2) + 1) ^ (-0.3e1 / 0.2e1) /
(q1(t) ^ 2 + q2(t) ^ 2 + q3(t) ^ 2) ^ 2 / (p1(t) ^ 2 + p2(t) ^ 2 +
p3(t) ^ 2) ^ 2);
```