



TALLINN UNIVERSITY OF TECHNOLOGY

SCHOOL OF ENGINEERING

Department of Electrical Power Engineering and Mechatronics

DEVELOPMENT OF ATTITUDE DETERMINATION CONTROL SYSTEM FOR TTÜ 100 SATELLITE

TTÜ 100 SATELLIIDI POSITSIONEERIMIS- JA KONTROLLSÜSTEEMI ARENDUS MASTER THESIS

Student: Kristjan Kruuser

Student code: 182889

Supervisors: Eiko Priidel, Early Stage Researcher
Indrek Roasto, Senior Research Scientist

Tallinn, 2020

(On the reverse side of title page)

AUTHOR'S DECLARATION

Hereby I declare, that I have written this thesis independently. No academic degree has been applied for based on this material. All works, major viewpoints and data of the other authors used in this thesis have been referenced.

"....." 202.....

Author:

/ signature /

Thesis is in accordance with terms and requirements

"....." 202.....

Supervisor:

/ signature/

Accepted for defence

"....."202... .

Chairman of thesis defense committee

/ name and signature /

LÕPUTÖÖ LÜHIKOKKUVÕTE

Autor: Kristjan Kruuser

Lõputöö liik: Magistritöö

Töö pealkiri: "TTÜ 100 satelliidi positsioneerimis- ja kontrollsüsteemi arendus"

Kuupäev: 18.05.2020

76 lk

Ülikool: Tallinna Tehnikaülikool

Teaduskond: Inseneriteaduskond

Instituut: Elektroenergeetika ja mehhatroonika instituut

Töö juhendaja(d): Eiko Priidel, Indrek Roasto

Töö konsultant (konsultandid): Martin Rebane, Alar Leibak

Sisu kirjeldus:

Käesolev töö käsitleb endas Tallinna Tehnikaülikooli poolt läbiviidava tudengisatelliidi TTÜ 100 kontroll ja positsioneerimissüsteemi arendust. Töös tuuakse välja projekti ajalugu ja senine käik, satelliidi tööpõhimõtted, töös läbiviidud simulatsioonide jaoks vajalikud põhitõed ja teooria, läbiviidud simulatsioonid ja nende analüüs ning ka esiletulnud probleemid ja nende lahendused.

Märksõnad: Kosmos, Satelliit, Arendus, Positsioon, Orbiit, Simulatsioonid, Positsioneerimine, Kontrollsüsteemid, Magneetiline, Juhtimine, Mudelid.

ABSTRACT

Author: Kristjan Kruuser

Type of the work: Master Thesis

Title: "development of attitude determination control system for TTÜ 100 satellite"

Date: 18.05.2020

76 pages

University: Tallinn University of Technology

School: School of Engineering

Department: Department of Electrical Power Engineering and Mechatronics

Supervisor(s) of the thesis: Eiko Priidel, Indrek Roasto

Consultant(s): Martin Rebane, Alar Leibak

Abstract:

TalTech's TTÜ 100 is a student satellite programme and this thesis shows the development of attitude determination and control system of that satellite. It focuses on developing and simulating the magnetic control system, orbital and earth geomagnetic reference field simulations and adjusting testing hardware. The thesis first discusses the theoretical background on satellites, and then uses simulations to develop a control system for the TalTech satellite.

Keywords: Space, Satellite, Development, Attitude, Orbit, Simulation, Positioning, Control Systems, Earth, Magnetic, Controlling, Models

LÕPUTÖÖ ÜLESANNE

Lõputöö teema:	“TTÜ 100 satelliit: ADCS juhtimissüsteemi arendamine”
Lõputöö teema inglise keeles:	“TTÜ 100 satellite: development of ADCS system”
Üliõpilane:	Kristjan Kruuser 182889 AAAM
Eriala:	
Lõputöö liik:	Magistritöö
Lõputöö juhendaja:	Eiko Priidel, Indrek Roasto
Lõputöö ülesande kehtivusaeg:	
Lõputöö esitamise tähtaeg:	20.05.2020

Üliõpilane (allkiri)

Juhendaja (allkiri)

Õppekava juht (allkiri)

1. Teema põhjendus

Käesolev töö uurib ja näitab Tallinna Tehnikaülikooli satelliidiprogrammi „TTÜ Mektory 100“ arendust ja selle tulemusi. Käesolevas töös on arendatud ja uuritud Satelliidi juhtimissüsteemi, selle arendamiseks läbi viiduid katseid ja teoreetilist poolt. Satelliidi juhtimissüsteem on tudengite poolt ise arendatud ja testitud. Käesolev töö toob välja satelliidi arenduse käigus ilmnunud probleemid, ja suurimad takistused, millele tuleb pöörata tähelepanu.

2. Töö eesmärk

Käesoleva töö peamine eesmärk on töödelda ja simuleerida, ning katsetada üheühikulise kuubiksatelliidi juhtimissüsteemi väljatöötatud osa.

3. Lahendamisele kuuluvate küsimuste loetelu:

Satelliidi simuleerimine

Satelliidi arendamisel ilmnevad suurimad probleemid

Satelliidi juhtimissüsteemi testimine

IGRF koodi analüüsimine ja tulemuste kajastamine

4. Lähteandmed

Käesolevas töös kasutatakse teoreetiliseks lahenduseks kirjandust, vastavaid teaduslikke artikleid ja koostöös teiste satelliidimeeskonna liikmetega leitud lahendusi.

5. Uurimismeetodid

Uurimismeetodid on:

Vastava kirjanduse ja artiklite analüüsimine

Simulatsioon keskkonnas Matlab/Simulink

Praktiline katsetus ja mõõtmised meeskonna ruumides vastavate seadmetega (magnetpuur)

Tarkvara analüüsimine arendusplaadil

6. Graafiline osa

Graafiline osa on nii töö põhiosas kui ka lisades.

7. Töö struktuur

Töö peatükid ja punktid :

Tudengisatelliidi projekti ülevaade ja satelliidi enda tutvustus.

ADCS süsteemi ülevaade ja satelliidi juhtimismeetodite ülevaade

Loodud simulatsioonimudeli ja sooritatavate katsete tööpõhimõte

Katsetes kasutatud riistvara ja seadmed

Katsete tulemused ja analüüs

Järeldused

8. Kasutatud kirjanduse allikad

Peamised kasutatud kirjanduse allikad on raamatud, artiklid ja teemakohased teadustekstid:

1. P.Fortescue, G.Swinerd, J.Stark „Spacecraft Systems Engineering Fourth Edition“
2. A.Sinharay “An Efficient Control Laws Design Approach for Unmanned Aircraft”
3. A. Slavinskis “ESTCube-1 Attitude Determination”
4. T.P.Sarafin ”Spacecraft Structures and Mechanics”

9. Lõputöö konsultandid

Eiko Priidel

Martin Rebane

10. Töö etapid ja ajakava

Teemaga tutvumine, kirjanduse läbitöötamine, töö etappide planeerimine. (03.03.2020)

Teoreetilise osa kirjutamine ja praktiliste katsete sooritamine. (31.03.2020)

Praktilise osa tulemuste analüüsimine ja järelduste kirjeldamine (20.04.2020)

Töö esimene versioon valmis ja juhendaja tagasiside saamine (01.05.2020)

Paranduste sisseviimine ja juhendaja teine tagasiside (10.05.2020)

Lõputöö lõpliku versiooni valmimine ja töö esitamine. (20.05.2020)

TABLE OF CONTENTS

LÕPUTÖÖ LÜHIKOKKUVÕTE	3
ABSTRACT	4
LÕPUTÖÖ ÜLESANNE	5
FOREWORD	10
LIST OF ABBREVIATIONS AND TERMS	11
1. INTRODUCTION.....	12
1.1 TTÜ 100 CUBESAT PROJECT	12
1.2 PROBLEM STATEMENT.....	12
2. BACKGROUND AND OVERVIEW	14
2.1 CUBESATS	14
2.1.1 TalTech 1-Unit Satellite.....	17
2.2 ATTITUDE DYNAMICS AND REPRESENTATIONS.....	18
2.2.1 Direction Cosine Matrix DCM.....	18
2.2.2 Euler angles	19
2.2.3 Quaternions	20
2.3 SATELLITE ATTITUDE DETERMINATION CONTROL SYSTEM.....	21
2.3.1 Attitude Determination and Sensing	21
2.3.2 Attitude Control.....	23
2.4 ORBITS	27
2.4.1 BASICS OF ORBITAL MECHANICS.....	27
2.4.2 ORBITAL ELEMENTS	28
2.4.3 ORBIT TYPES AND CLASSIFICATIONS	29
3. SIMULATIONS	31
3.1 IGRF code and simulation comparison analysis.....	31

3.1.1 International Geomagnetic Reference field.....	31
3.1.2 IGRF-12 analysis.....	32
3.2 Simulating attitude control using magnetorquers.....	34
3.2.1 Magnetorquer Dynamics	35
3.2.2 Controller development and working principal	36
3.2.3 Controller simulation	37
3.3 SIMULATING THE ORBIT	46
4. MODIFYING MAGNETIC CAGE CONTROL SCHEMATIC	55
5. FUTURE WORK	58
SUMMARY	59
KOKKUVÖTE.....	61
REFERENCES	62
APPENDICES.....	65
APPENDIX 1 IGRF COMPARISON TABLES	65
APPENDIX 2 CONTROLLER SIMULATION MATLAB CODE FOR INPUTS.....	69
APPENDIX 3 SIMULINK SYSTEM MODELS.....	70
APPENDIX 4 ECI AND ECEF ILLUSTRATIONS	72
APPENDIX 5 ORBIT SIMULATION CALCULATIONS IN MATLAB	73

FOREWORD

The subject of this thesis came from my membership of TalTech student satellite team. I am part of ADCS development team, and the subject of this thesis was recommended by Martin Rebane who is a senior member of the team and has supervised other members previous theses. Working rooms for ADCS team are on TalTech's campus in cybernetics building and most of the work was done there. Due to my previous background in mechatronics, the control system development was a good subject to focus on. To make it up on theoretical space knowledge I took additional classes and completed online courses to understand the theoretic behind space engineering and astrophysics. Main source of information was found on articles, books and previous CubeSat project from other universities who shared their knowledge and data.

I would like to thank following people: Indrek Roasto and Eiko Priidel for supervising the writing process and helping with the subject. I would also like to thank Martin Rebane, Alar Leibak, Madis Kaal, Shu Taya and Peeter Org who are members of the TalTech ADCS team and Rauno Gordon who invited me to the team and other project team members.

LIST OF ABBREVIATIONS AND TERMS

ADCS – Attitude Determination Control System

OBC – Onboard Computer

LEO – Low Earth Orbit

MEO – Middle Earth Orbit

HEO – High Earth Orbit

IFR – Inertial Frame Reference

RGB – Red Green Blue

NIR – Near Infrared

PCB – Printed Circuit Board

SBF – Satellite Body Frame

IGRF – International Geomagnetic Reference Field

FOV – Field of View

DCM – Direction Cosine Matrix

ECEF – Earth Centered Earth Fixed

ECI – Earth Centered Inertial

TLE – Two Line Elements

WGS84 – World Geodetic System 84

DOF – Degree of Freedom

ACS – Attitude Control System

ADS – Attitude Determination System

EPS – Electronic Power System

NOAA – National Oceanic and Atmospheric Administration

1. INTRODUCTION

1.1 TTÜ 100 CUBESAT PROJECT

TTÜ 100 Satellite is a program carried out by University of TalTech. The main goal of the program is to build a nanosatellite, made by students themselves and launch it into Low Earth Orbit (LEO). The satellite's main functions will be earth observation and testing the technologies used in observation. The satellite uses two onboard cameras: standard three sensor Red Green Blue (RGB) and Near Infrared (NIR) ones to observe earth in colour and in the infrared spectrum. The NIR camera is mainly used to observe climate and flora. Furthermore, there is an image processing software onboard that prepares the images for downloading. The nanosatellite communicates with the ground station at frequencies of 435 MHz and 10.5 GHz. The 435 MHz frequency is a two-way communication frequency band and the 10.5 GHz communication is for downloading bigger packages. The communication protocol for the satellite is AX 25 [1]. Satellite itself is a 1-unit (1U) nanosatellite, built according to CubeSat standards by the mechanical team [2]. Because the goal of the project is to educate students, the used systems are developed in steps. For example, in controlling the satellite the first system developed is magnetic control. When the system is working, then additional systems are developed and added. That way students can get a better understanding how the developed systems work.

1.2 PROBLEM STATEMENT

The main goal of the thesis is to show the development process of the satellite's Attitude Determination Control System (ADCS). When the satellite is sent to orbit it cannot be brought back and all systems developed must guarantee that the satellite is operational and will continue communicating. Meaning all developed systems must be carefully tested in ground station to avoid malfunctions on orbit. Worst case scenario, due to the system's malfunction the satellite will not communicate with the ground station and the system cannot be fixed. Simulations are used to see how the satellite will behave on orbit and to test the system itself. According to the results of simulations, the system itself is tested in magnetic cage built by the members of the team. Due to the fact that the nanosatellite is controlled by magnetorquer, the International Geomagnetic Reference Field (IGRF) model must be analysed and work accordingly. Only then the system can be adjusted accordingly and finally uploaded to the satellite itself. Main problems handled in this thesis are:

- Comparing the IGRF model used in the satellite with model from National Oceanic and Atmospheric Administration (NOAA.)
- Developing control and stabilization system of the satellite.
- Simulating orbits and satellite behaviour in MATLAB.
- Modifying the Helmholtz cage control board schematic for controlling the coil polarity.

This thesis is separated into an introductory part and four main parts. First part is the introduction and problem statement. Second part is background and overview, theoretical work and research of the subject. Third part is simulations of created models and systems and their results. Fourth part shows the work made modifying the Helmholtz cage. Fifth part is the summary of made work and prognosis for future work.

2. BACKGROUND AND OVERVIEW

To have a working CubeSat in space, it is necessary to know how they work, how they are made, what affects them in space, the orbital differences and the effects of different controlling methods. The following chapter will give an overview of CubeSats, their builds as well as controlling methods. Also, it will give an overview of orbits and their differences. In following chapters, the methods and orbits TalTech satellite uses are shown.

2.1 CUBESATS

As every spacecraft launched has a specific task and function, meaning there are different types and classes for them. Mainly nine types of classes are used:

- Remote Sensing Satellite
- Navigation Satellite
- Global positioning systems – GPS
- Drone Satellites
- Geocentric Orbit type satellites – LEO, MEO, HEO
- Ground Satellites
- Polar Satellites
- Nanosatellites
- Geostationary Satellite – GEO

CubeSats belong to nanosatellite class and are mainly used for research. Due to their low manufacturing cost and simplicity of launching these satellites, they are becoming more popular. CubeSats have a standard dimension unit of 10 cm x 10 cm x 10 cm with a mass of 1 to 1,33 kg. Usually their sizes are 1 to 6 units (Figure 2.1) [3]. CubeSats started to appear in the beginning of the 21st century. The first satellite with standardized dimension was Aalborg University's AAUSAT 1 from Denmark. It had a size of 1U, and it was launched in 2003. Compared to regular satellites, the development time and cost needed to make a CubeSat is very small, so they are also being used as educational projects in universities. Moreover, most

CubeSats are planned with relatively short lifespan for about three years. For example, 1U satellite is heavily affected by radiation and the lifespan of the electronic components are greatly reduced and they will stop working after a certain time. Then due to atmospheric drag it will burn in earth's atmosphere. This reduces space debris and collision danger with other spacecrafts.

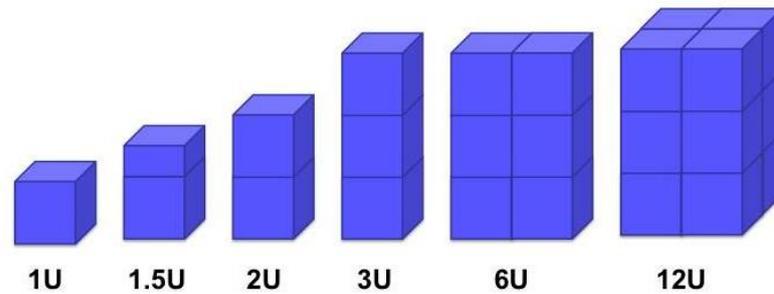


Figure 2.1 . Comparison of most popular CubeSat sizes.

CubeSats are launched into orbit from standardized CubeSat deployer systems called Poly Picosatellite Orbital Deployers (P-POD) (Figure 2.2). It is a rectangular box with a door and a spring mechanism. When the release mechanism is engaged by a signal sent from the launch vehicle (LV) (Figure 2.3), the door is forced open and the CubeSats are deployed by the main spring [1].

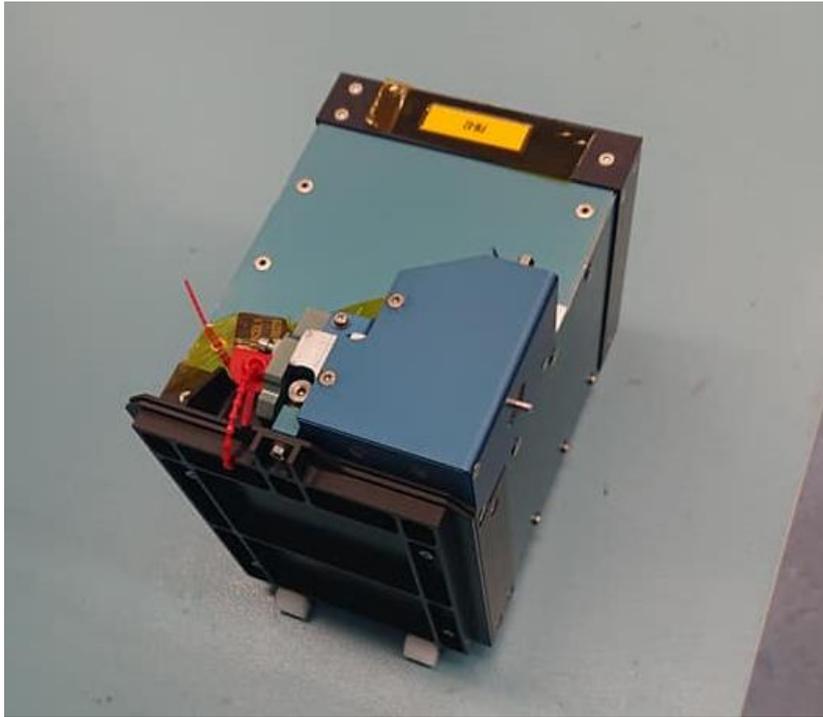


Figure 2.2. TalTech CubeSat in 1U P-POD.

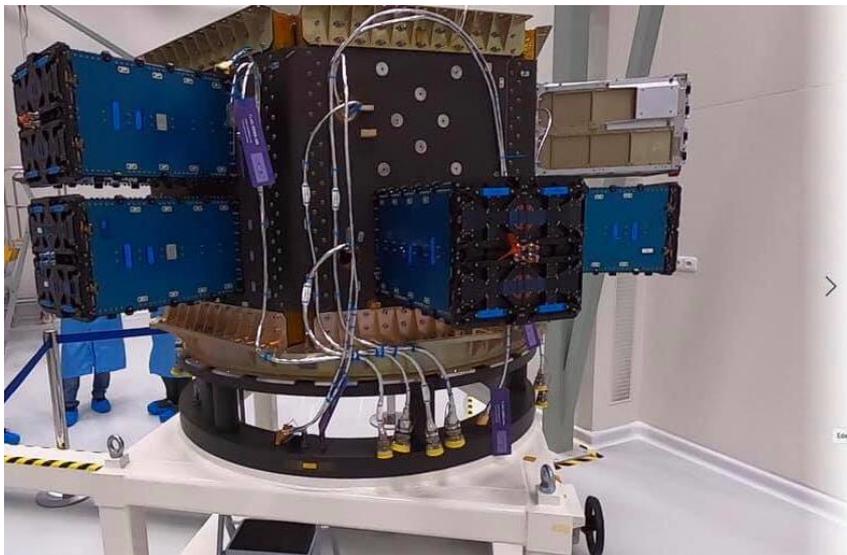


Figure 2.3. Part of LV with P-POD-s attached.

2.1.1 TalTech 1-Unit Satellite

The CubeSat manufactured by TalTech is a 1-unit satellite covered with solar panels to power the nanosatellite. Inside the frame there are Printed Circuit Boards (PCB) designed by the members of the team (Figure 2.4). During the launch the satellite is under a lot of mechanical stress and vibrations. To guarantee working conditions, satellite design must be resilient enough to endure these mechanical forces. It also must endure extreme temperature of outer space. When the satellite reaches orbit, its wings, which are covered by solar panels, are opened by launch mechanism.

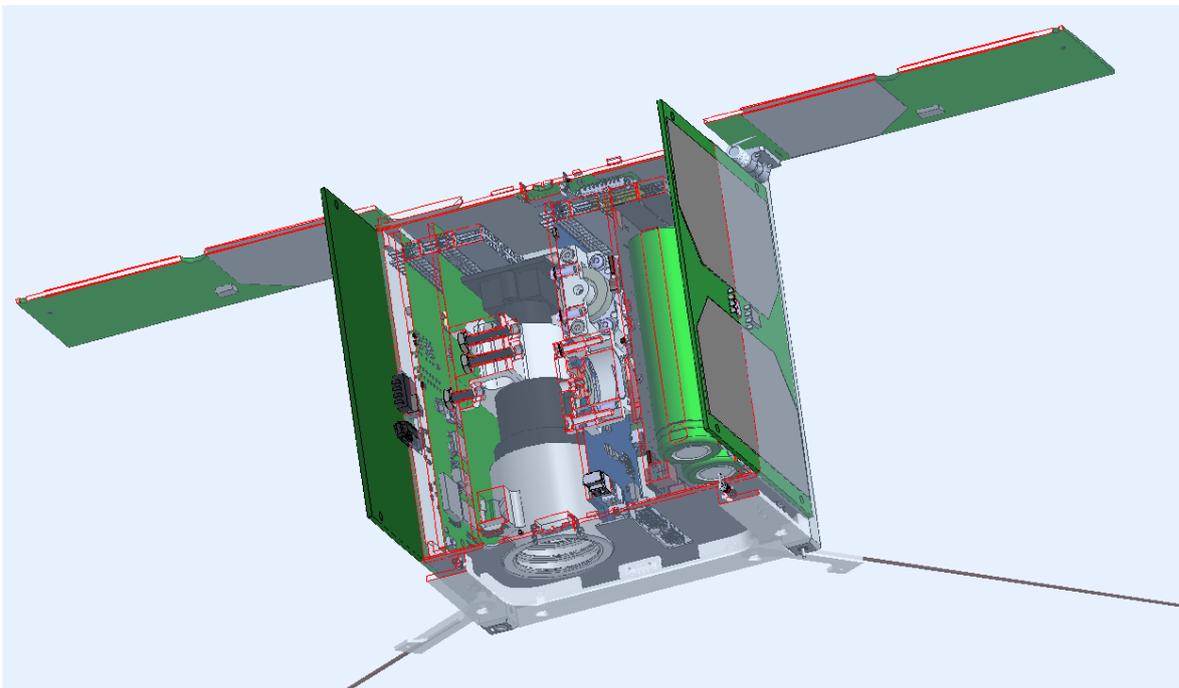


Figure 2.4. Cross section from TalTech CubeSat 3-D model.

The satellite consists of five subsystems that are divided onboard 5 PCB-s. The frame and the system PCB-s are covered with solar panels. There are also two solar panel wings and communication antennas. Every inner circuit board is for different subsystem. Like camera board, attitude board, battery board and Onboard Computer (OBC) board. Attitude controlling system board is placed in the centre and has reaction wheels used in Attitude Control System (ACS). Camera board has two cameras for observation. Electronic Power System (EPS) houses satellite's battery, charging and power management. The goal of the ADCS system that it is completely autonomous, and it can work even when OBC is not active. In outer PCB are embedded magnetorquers for controlling satellites attitude. Total mass of the satellite is about 1.33 kg. Satellite has three working modes: first is sun

tracking mode so it could charge itself using the solar panels. Second is ground station pointing mode so it could communicate with the ground station and third is observing mode that is used for taking pictures. For observing mode, the satellite's pointing accuracy should be about 3°.

2.2 ATTITUDE DYNAMICS AND REPRESENTATIONS

Satellite attitude can be described with three rotational axes that are perpendicular with each other (Figure 2.5). Attitude can be specified in many ways such as direction cosines, Euler angles or quaternions. The most common way is to use Euler angles. These axes are defined in the same way as is standard practice for aircrafts [4].

2.2.1 Direction Cosine Matrix DCM

The Direction Cosine Matrix (DCM) is a transformation matrix that transforms one reference frame to another reference frame. In this system it represents the attitude of body frame relative to the reference frame. It is specified by 3x3 rotation matrix (R) which has columns to represent the unit vectors in the body axes projected along the reference axes (2.1) [5].

$$R = \begin{bmatrix} A_{11} & A_{12} & A_{31} \\ A_{21} & A_{22} & A_{32} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \quad (2.1)$$

If the two coordinate systems are defined as:

$$\{\hat{r}\} = \begin{Bmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \end{Bmatrix}, \{\hat{b}\} = \begin{Bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{Bmatrix} \quad (2.2)$$

where r – reference coordinate system,

b – body coordinate system.

When \hat{r}_i and \hat{b}_i ($i = 1,2,3$) are three-unit vectors along their axis direction and transformation matrix R_{br} is a 3x3 matrix then:

$$R_{br,ij} = \cos \alpha_{ij} = \langle \hat{b}_i, \hat{r}_j \rangle \quad (2.3)$$

where α_{ij} - angle between body and reference frame ($j = 1,2,3$).

DCM-s are widely used because the inverse matrices can be computed for DCMs, showing rotation between the frames in other direction. Also, they can be described as matrix multiplications of individual matrixes. Attitude control system used in this thesis applies DCM usage.

2.2.2 Euler angles

Euler angles are used to describe the attitude of a rigid body as a sequence of three rotations around the axes of a fixed coordinate system. The axes of the original frame are defined as x, y and z and as X, Y and Z for the rotated frame. The angles are defined with φ , θ and ψ (also called yaw, pitch and roll) and they represent the angle between the original frame and rotational frame [5].

$$[X_\varphi] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \quad (2.4)$$

$$[Y_\theta] = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.5)$$

$$[Z_\psi] = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

- Roll – Rotation about x-axis
- Pitch – Rotation about y-axis
- Yaw – Rotation around the centre z-axis

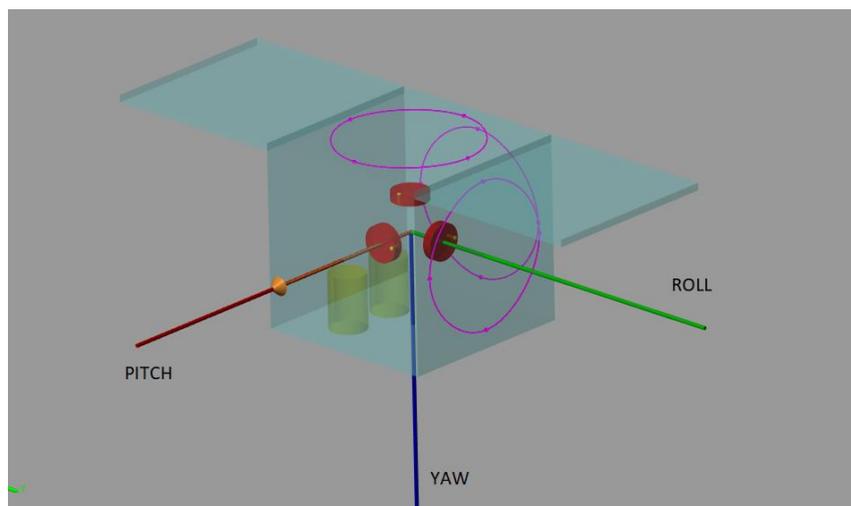


Figure 2.5. CubeSat Rotational axes.

The Euler angles sequence can also be constructed to a DCM by multiplying the rotation matrices for each euler angle (2.7).

$$[C] = [X_\varphi][Y_\theta][Z_\psi] =$$

$$= \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi & \sin \varphi \sin \theta \sin \psi + \cos \varphi \cos \psi & \sin \varphi \cos \theta \\ \cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi & \cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi & \cos \varphi \cos \theta \end{bmatrix} \quad (2.7)$$

Euler angles are widely used in three-axis stabilisation in which small angle approximations are used. They are mainly used to visualize attitude. The largest issues with Euler angles is their singular value and the heavy computational load that is needed to get the attitude matrix from Euler angles. The singularity is related to the physical phenomenon called Gimbal lock that happens in mechanical gyroscopes.

2.2.3 Quaternions

Quaternions are called Euler parameters where attitude is described as a 4-dimensional unit. In general, quaternions are represented in the form of $a + bi + cj + dk$ where a, b, c and d are real numbers and i, j and k are quaternion units. Quaternions are described with a vector part and a scalar part. When assuming the vector describing the Euler axis is $e = [e_1 \ e_2 \ e_3]^T$ and the rotational angle is ϕ then quaternion can be defined as [6]:

$$q = \begin{bmatrix} q_v \\ q_s \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} e_1 \sin \frac{\phi}{2} \\ e_2 \sin \frac{\phi}{2} \\ e_3 \sin \frac{\phi}{2} \\ \cos \frac{\phi}{2} \end{bmatrix} \quad (2.8)$$

where q_v – vector part,

q_s – scalar part.

The benefit of using quaternions is the avoidance of singularities in rotational matrices, plus they have smaller time and computational power consumption. They are simpler to interpolate and easier to work with. The disadvantage of using quaternions is that they are more complex in theory and harder to understand.

2.3 SATELLITE ATTITUDE DETERMINATION CONTROL SYSTEM

Without attitude control the satellite that is in orbit is useless. The prime purpose of ADCS is to orientate the body of the satellite correctly to the required accuracy. Also, it is a momentum management system. Angular momentum itself can be stored, disposed and acquired. Attitude is a three-dimensional orientation of an object compared to a reference frame [4]. To acquire the required attitude measurement, three pieces of information must be determined to relate the spacecraft axes to some datum set. It could be in the form of Euler angles or something else. To get that information the measurement subsystems must include sensors to measure and provide the information. Those measurements must be done in all mission phases. Those sensors can be set in two categories:

- Reference sensors – These sensors measure directions of external objects such as the Sun or stars. These sensors complement inertial sensors and give them the needed data.
- Inertial sensors – These sensors are measured continuously. As they measure attitude changes relative to the gyroscope, they need calibration from reference sensors.

Using those sensors, a measurement system using inertial and reference sensors complementing each other can be formed. Reference sensors can calibrate inertial sensors, which use these calibrating inputs to remember the previous position of reference object [4]. The whole attitude determination control system consists of:

- Sensors - sun sensors, magnetometers and gyroscopes
- Software – models and calculation algorithms on onboard computer (OBC)
- Hardware actuators - magnetorquers and reaction wheels
- Ground support equipment – communication and simulations equipment, plus test to keep the satellite in working condition.

2.3.1 Attitude Determination and Sensing

To control a satellite, it must know its own attitude first. Spacecraft attitude is determined through a process of combining data inputs from available sensors of the spacecraft and combining it with spacecraft dynamics. This results in an attitude state as a function of time, either onboard for immediate use, or post-processing (Figure 2.6). Specifying attitude may be done in many ways such

as Euler angles, direction cosines, quaternions and more [4]. Precise attitude determination results in right calculations that are made to determine how much thrust or torque is needed for the satellite to get into desired position. Spacecraft attitude changes according to the fundamental equations of motion for rotational dynamics (Euler equations) [6].

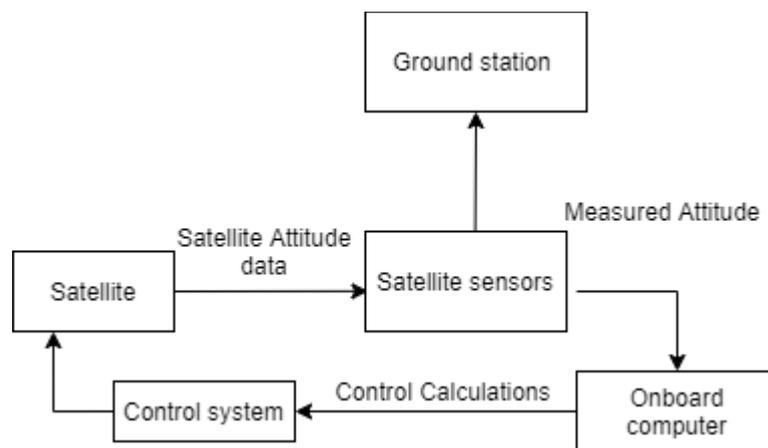


Figure 2.6. Satellite ADS system block diagram.

Attitude determination sensors used in TalTech CubeSat are :

- Sun sensors
- Magnetometers
- Gyroscope

The Sun subtends an angle of ≈ 30 arc minutes at Earth, which provides a well-defined vector, it is unambiguous because of the intensity of the radiation [4]. Sensors measure the current position of the sun vector in satellites coordinate system and when satellite position changes, the sun vector also changes. The drawbacks of sun sensors are that they can only be used when the satellite is positioned in the sun-lit phase of the orbit or it cannot be used.

Magnetometers measure the direction and the strength of the local magnetic field. To gain attitude information, the measured field is compared to a magnetic field model held in the on-board processor. The magnetometer is used in conjunction with magnetorquers. The basis of pre-mentioned inertial sensing system for attitude is formed by a gyroscope that measures angular velocity and the placement of satellites yaw, pitch and roll angles.

2.3.2 Attitude Control

Spacecraft attitude is controlled with actuators that apply or create external thrust or torque to change the spacecraft's orientation. It is a feedback system where the satellite itself is the plant in which the torque affects the attitude. The ACS system is determined by the mission type. The torques that affect the spacecraft around its central mass must be determined as external or internal. External torquers are necessary and must be added to every ACS system. External torquers can be in the form of thrusters or magnetorquers. Also, other external forces must be considered, like solar radiation pressure, gravity-gradient torque and aerodynamic torque. Internal torques, like mass movement inside the spacecraft and momentum storage affects the control system.

TalTech CubeSat uses reaction wheels and magnetorquers to control the attitude. Magnetorquers work by creating a magnetic field (magnetic dipole moment) that interacts with Earth's magnetic field. This effect can be compared with a compass needle attempting to align itself with the local direction of the field. The magnetic torque vector in the satellites body frame can be described as when spacecrafts magnetic dipole moment interacts with Earths magnetic flux (2.9) [4].

$$\vec{T} = \vec{M} \times \vec{B} = \|\vec{M}\| \|\vec{B}\| \sin(\theta) \vec{u} \quad (2.9)$$

where \vec{T} – Produced torque, Nm,

\vec{M} – Magnetic dipole moment, A*m²,

\vec{B} – Earth flux density, T,

θ – Angle between magnetic dipole moment vector and Earth flux density,

\vec{u} – unit vector.

It must be considered that provided torque is at its maximum value when the angle between magnetic dipole moment vector and Earth flux density (θ) is 90°. Magnetic moment created by the magnetorquers is controlled by a current affected by the magnetic coil parameters. (2.10).

$$\vec{M} = I * n * \vec{S} \quad (2.10)$$

where I – current in the coil, A,

n – is the number of turns of the coil,

\vec{S} – cross section areal vector of the coil, m^2 .

TalTech CubeSat magnetorquers are 50mm from its mass center and their maximal dipole moment is 0.12 Am^2 . Earth's magnetic field maximum value is 65 microteslas, maximum torque is created when the degree between the moment vector and Earth's magnetic field is at its maximum will be 7.8 micronewtons. Magnetorquers used in this CubeSat are air core magnetorquers inside the PCB-s. CubeSat has inertial wheels that work by using their momentum to create torque in opposite direction (figure 2.7).

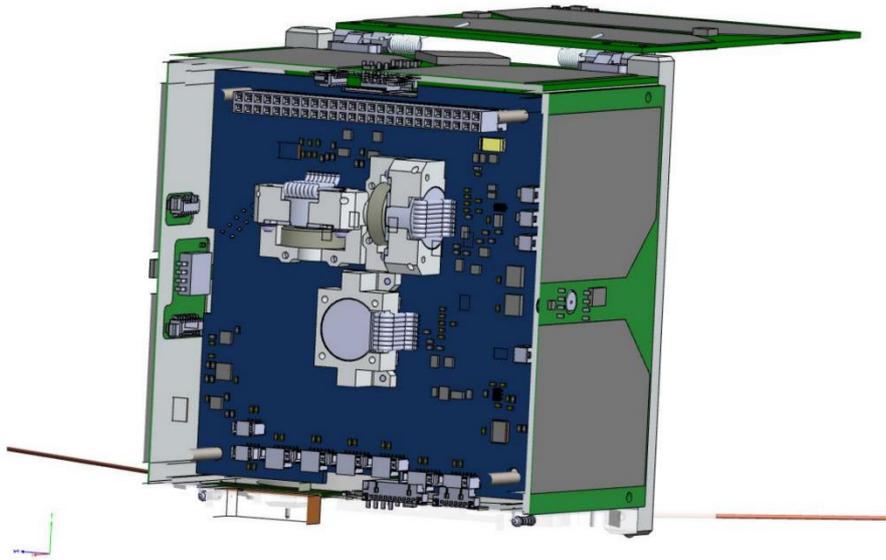


Figure 2.7. Reaction wheels inside CubeSat.

Flywheels are positioned so that they would be perpendicular with each-other. They are controlled with electric motors and are used to change satellites attitude around its mass centre. Reaction wheels are more accurate than magnetorquers for stabilization and correcting attitude, but their drawback is their saturation. ACS system is complemented by ADS system and gives constant output feedback (Figure 2.8).

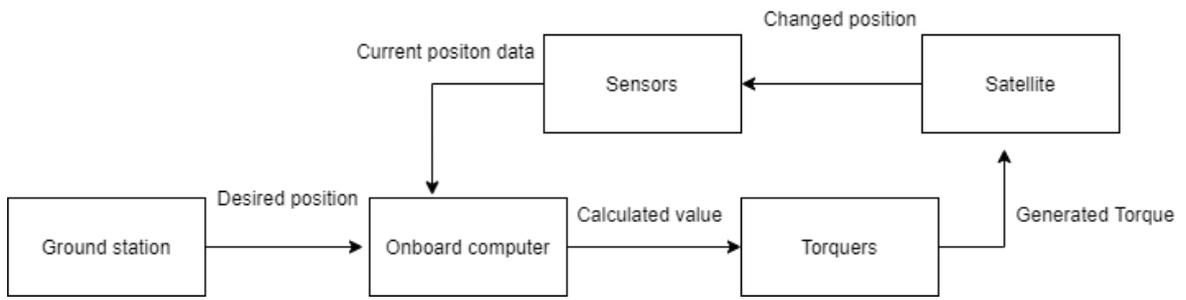


Figure 2.8. ACS system Block diagram.

Whole ADCS system idea diagram and its components in working mode (Figure 2.9) can be seen in the thesis by Peeter Org [7].

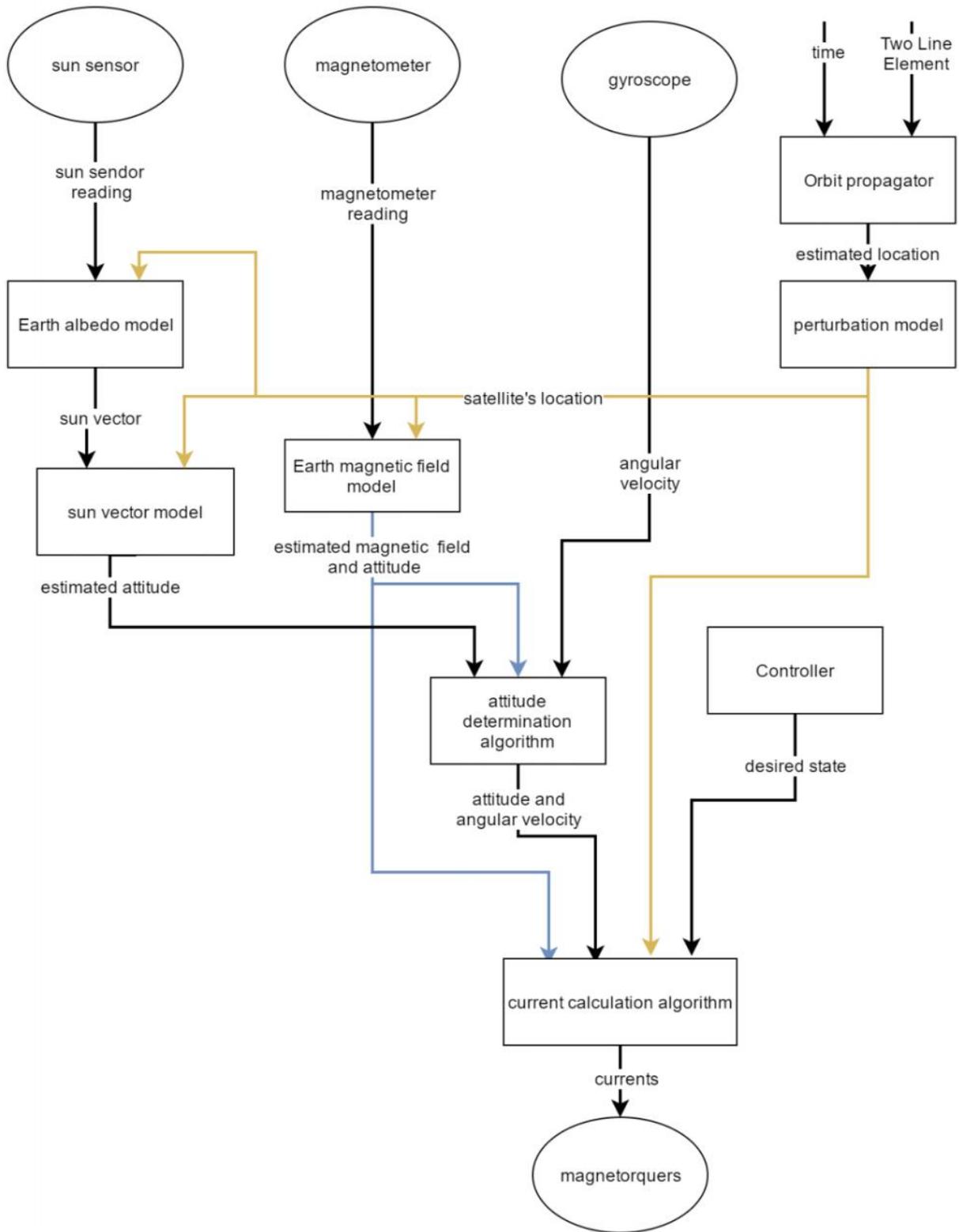


Figure 2.9 ADCS system in normal working mode by Peeter Org [7].

Figure 2.9 shows both hardware and software components: hardware components being circular and software components rectangular. It shows the components in normal working mode, where different software models receive input from both components: software perturbation model giving satellite its location and hardware sensors providing the reading data. Different software models are used to transform data for inputs to other models, with the final output controlling magnetorquers to help detumble and control the satellite.

2.4 ORBITS

Orbits are curved paths in space caused by object's, such as planets, stars etc, gravity. It is a path where one body moves around other body (stars, planets, asteroids). When objects have similar masses, neither object is so called centre object. When one object is larger, smaller object orbits around the bigger one. Easiest examples are our own planet Earth which orbits around the Sun and the Moon that orbits around Earth. Due to the fact that two bodies both influence another, means larger bodies are also influenced by the gravity of smaller bodies. The moon's gravity pulls Earth slightly away from its centre and Earth also influences the Sun. In orbital mechanics this is called two body problem.

2.4.1 BASICS OF ORBITAL MECHANICS

In a stable orbit, the main force applied to a satellite is the universal force of gravity. In most cases, the satellite is moving in a vicinity of a single body, which is either a planet, the Sun or a further central body. It means other celestial bodies and their fields affect its motion much less than the gravity of central body. It makes most sense to examine the motion caused by the gravity of central body. Other forces affecting the satellite are regarded as perturbations of higher orders of magnitude. Because most celestial bodies are nearly spherical, Newton's law of gravity (2.11) can be used and non-sphericity can be considered as perturbation [8]. The main equations are Newton's law of gravity and acceleration of the satellite (1.2).

$$F_g = \gamma \frac{Mm_s}{r^2} = \mu \frac{m_s}{r^2} \quad (2.11)$$

And

$$a_g = \gamma \frac{M}{r^2} = \frac{\mu}{r^2} \quad (2.12)$$

where γ – *gravitational constant* $\approx 6,67 * 10^{-11} (m^3 / (kg * s^2))$,

M – mass of the central body, kg,

m_s – mass of the satellite, kg,

r – distance from the centre of the body to the satellite, km,

μ – *gravitational parameter* ($\mu_{Earth} \approx 4,0 * 10^{14} (m^3 / s^2)$), ($\mu_{Sun} \approx 1,3 * 10^{20} (m^3 / s^2)$).

To put the satellite into the orbit, it must have a certain velocity. The minimum velocity the satellite needs to move away from the planet to an infinite distance is defined as escape velocity. Earth's escape velocity is about 11 kilometres per second. For a satellite to maintain a circular orbit, its centripetal acceleration must be same as its gravitational acceleration. For the Earth the acceleration is about 7900 meters per second.

When two objects are revolving, the one with the greater mass is considered primary, and the other one is secondary. If Earth's mass is denoted M , the mass of the satellite is irrelevant m , thus $M + m \approx M$. Therefore, following Kepler's laws, it can be said that the trajectory of the satellite is an elliptical orbit.

2.4.2 ORBITAL ELEMENTS

Orbital elements, also called Keplerian elements, determine the orbit and how satellite places itself in the orbit (Figure 2.10). There are six elements:

- Eccentricity (e) – Defines the shape of an orbit. It shows how much the orbit deviates from a perfect circle. If the orbit is circular $e = 0$, if it is parabolic then $e = 1$.
- Semi-major axis (a) – Defines orbit's size.
- Inclination (i) – Angle between the orbital plane and the reference plane. Zero inclination $i = 0^\circ$ indicates that the orbit is equatorial. If $i = 90^\circ$, the orbit is polar.
- Longitude of the ascending node (Ω) – Angle between the vernal point and the ascending node. Vernal point is the point on the celestial sphere where the Sun crosses the equatorial plane.

- Argument of perigee (ω) - Angle between the ascending node and the orbit's point of closest approach to the Earth.
- True anomaly (ν) – Represents the angle between the perigee and the orbiting body.

Since Keplerian orbits are planar, the first pair of elements (eccentricity and semi-major axis) refer to the parameters of the orbits within its plane. The second pair of elements (inclination and longitude of the ascending node) defines the orientation of the orbit relative to the plane of reference. Usually the equatorial plane of the planet is used as the referential plane. The last pair of elements (argument of perigee and true anomaly) define the position of the orbit in its plane and the current position of the object on the orbit. Besides, these elements show the satellites Field of View (FOV). FOV shows the angle describing the amount of Earth's surface the satellite can see.

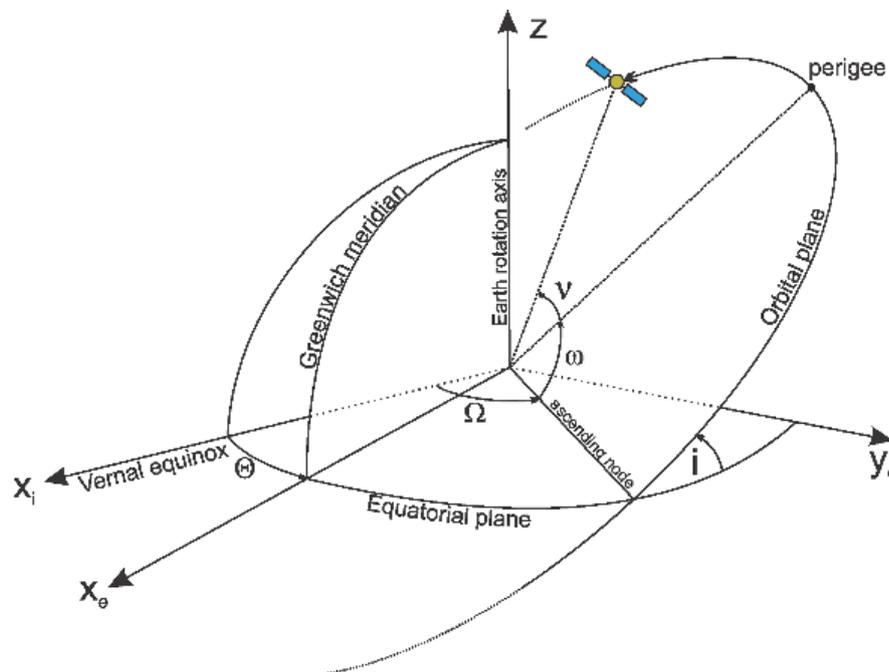


Figure 2.10. Illustration of Keplerian elements.

2.4.3 ORBIT TYPES AND CLASSIFICATIONS

Satellites around Earth perform different tasks defined by their orbit. Certain orbits are better for some purposes. Therefore, every spacecraft has its own orbit. Orbits are also defined by the body that affects them. For example, geocentric orbits mean they are around Earth while heliocentric means the orbits are around the Sun.

Orbit types around Earth are determined by Keplerian elements and the satellites escape velocity. Those factors show the orbital shape. There are four main types of orbits defined by those two factors:

- Circular orbits ($e = 0, V = V_{\text{circ}}$)
- Elliptical orbits ($e < 1, V_{\text{circ}} < V < V_{\text{esc}}$)
- Parabolic orbits ($e = 0, V = V_{\text{esc}}$)
- Hyperbolic orbits ($e > 1, V > V_{\text{esc}}$)

There are different orbit types based on inclination:

- Equatorial orbit (inclination is 0°) – The satellite will always remain above the equator.
- Polar orbit (inclination is 90°) – The satellite passes above the poles.

It is important to take note that Keplerian motion is an idealisation. In real conditions the motion of objects is always perturbed.

One more way to show different orbit types is their orbital altitude. The orbits in use are following:

- Low Earth Orbit (LEO) – Orbits with altitudes < 2000 km. Satellites in orbital heights around 200 kilometres and less cannot sustain their altitude for long due to the atmospheric drag being too powerful. The lower limit of LEO is 160 km.
- Medium Earth Orbits (MEO) – Orbits with altitudes from 2000 km to 36000 km. These orbits are usually used by GPS and communication satellites. Due to atmospheric drag being so low the lifetime of those satellites is very long. They must be heavily shielded due to the radiation from the Van Allen belt that damages the electronics.
- High Earth Orbits (HEO) – Orbits with altitudes more than 36000 km. These orbits are rarely used.

3. SIMULATIONS

This chapter shows the simulations, how they were conducted and their results. Following chapters overview the analysis of IGRF models and the difference between them. Second part gives an overview of the simulation of satellite body and its stabilisation on three axes using magnetorquers. The focus is on magnetorquers as they are used to detumble the satellite and they are the main ACS system. If something happens with reaction wheel control, the system can always fall back to magnetic control. Third part shows orbital simulation and its visualisation. MATLAB and Simulink environments with additional block sets add-ons were used to run the simulations and to visualise the results.

3.1 IGRF code and simulation comparison analysis

Earth's magnetic field also called the geomagnetic field, is the magnetic field extending from Earth's interior. It is generated when heat escapes from Earth's core and moves the molten iron in Earth's outer core. This movement causes electric currents that generate the magnetic field. The strength of Earth's magnetic field at its surface ranges from 25 to 65 microteslas. Current magnetic dipole field is tilted about 11 degrees from Earth's rotational axis. The magnetic poles, north and south, are located near geographic poles but over time they move and switch places in events known as magnetic reversals. The timespan is several hundred thousand years, and the time in which the change happens is irregular.

3.1.1 International Geomagnetic Reference field

The IGRF is a series of mathematical models of Earth's main magnetic field and its annular rate of change. It models the geomagnetic field as a negative gradient of a magnetic scalar potential (3.1) which comes from magnetic scalar potential model (3.2) [9].

$$\vec{B}(r, \varphi, \theta, t) = -\nabla V(r, \varphi, \theta, t) \quad (3.1)$$

$$V(r, \varphi, \theta, t) = a \sum_{l=1}^L \sum_{m=0}^l \left(\frac{a}{r}\right)^{l+1} (g_l^m(t) \cos m\varphi + h_l^m(t) \sin m\varphi) P_l^m(\cos \theta) \quad (3.2)$$

where r – radial distance from earth's center, km,

φ – East longitude,

θ – colatitude (polar angle), degrees,

a – Earth’s radius, km,

$P_l^m(\cos \theta)$ - Schmidt semi-normalized associated Legendre functions (m and l are orders of the function).

IGRF model provides a magnetic field vector in North East Down (NED) frame at any given latitude, longitude and altitude. IGRF model is updated every five years and new versions are released by IAGA (International Association of Geomagnetism and Aeronomy) [9]. In ADCS IGRF model is used, when satellite measures magnetic field with magnetometers and compares them with the model implemented on the OBC.

3.1.2 IGRF-12 analysis

TalTech CubeSat uses IGRF 12-th version which was released in 2015. Model implemented on OBC was written from MATLAB model to C-code by Shu Taya a member of CubeSat team. To find out the difference between the models and estimate allowed tolerance, i simulated the model in MATLAB and compared with NOAA IGRF model [10]. The comparisons are made in all three axes and made in four different main longitude degrees: 0° (figure 3.1), 90° (figure 3.2), -90° (figure 3.3) and 180° (figure 3.4). Difference is calculated in main latitude degrees (-90°, -60°, -30°, 0°, 30°, 60° and 90°) (Appendix 1). Calculations and magnetic field value in nT is calculated at height of 600km. IGRF code does not compute fields at 90° on Y-axis.

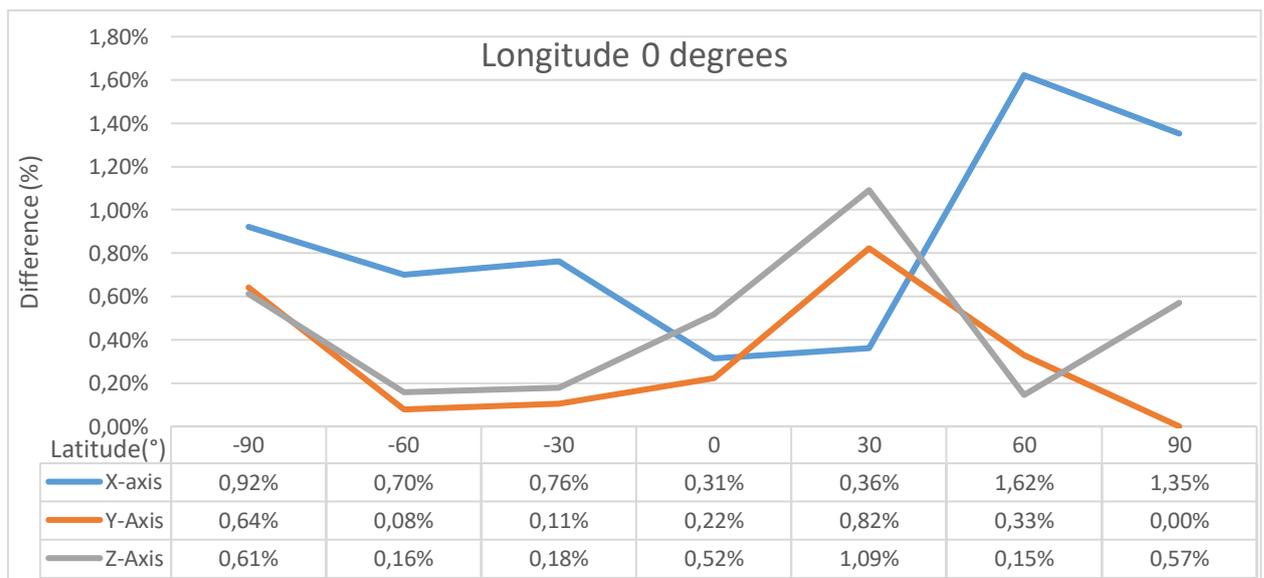


Figure 3.1. Difference % between longitude 0°.

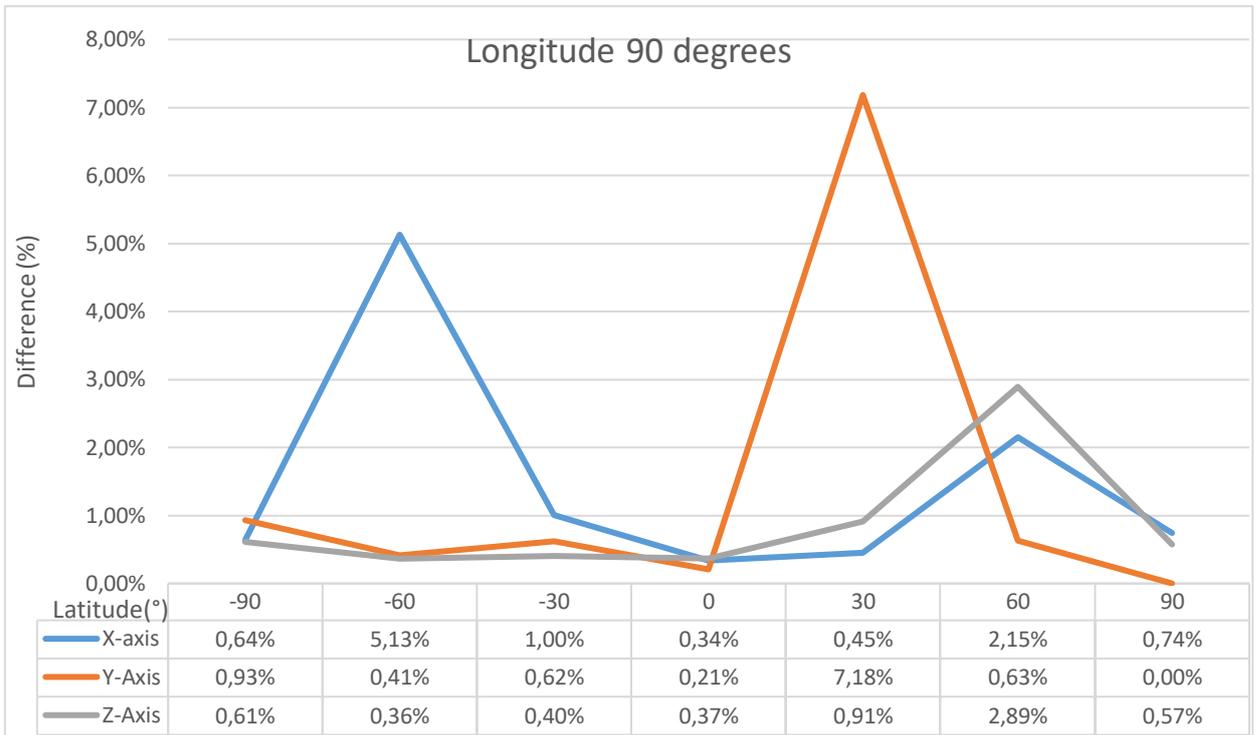


Figure 3.2. Difference % between longitude 90°.

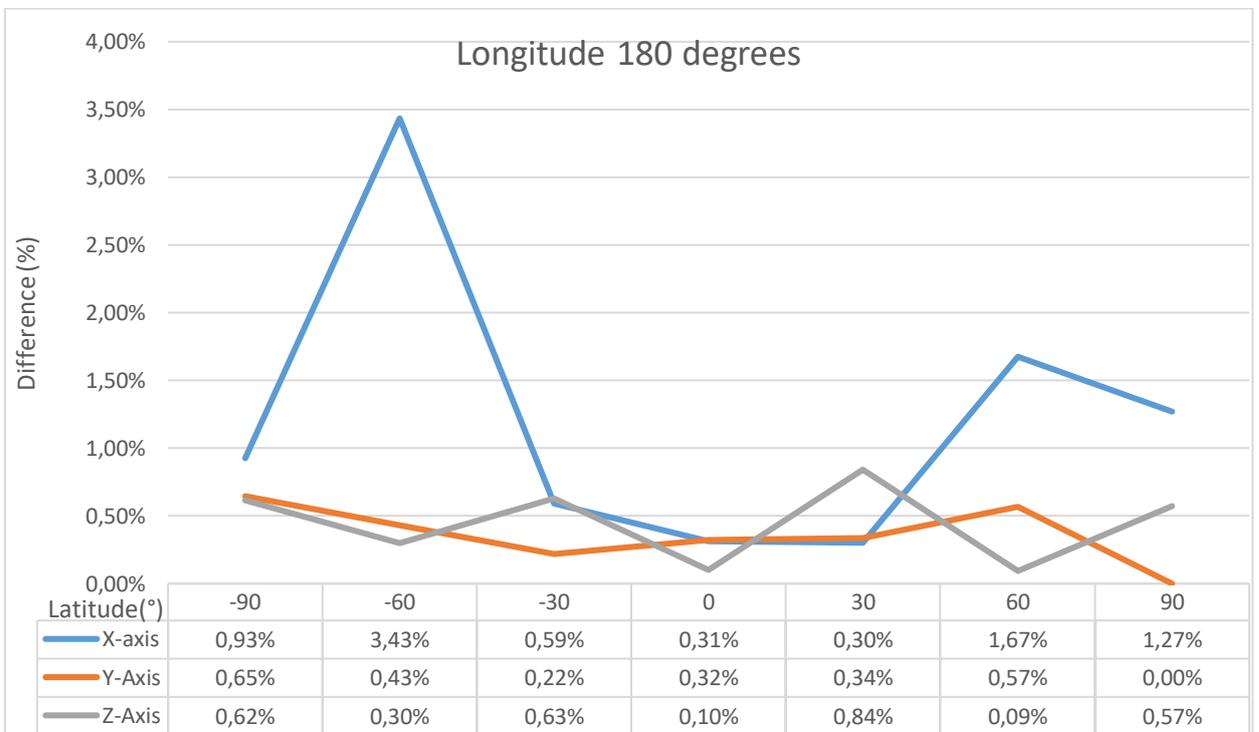


Figure 3.3. Difference % between longitude 180°.

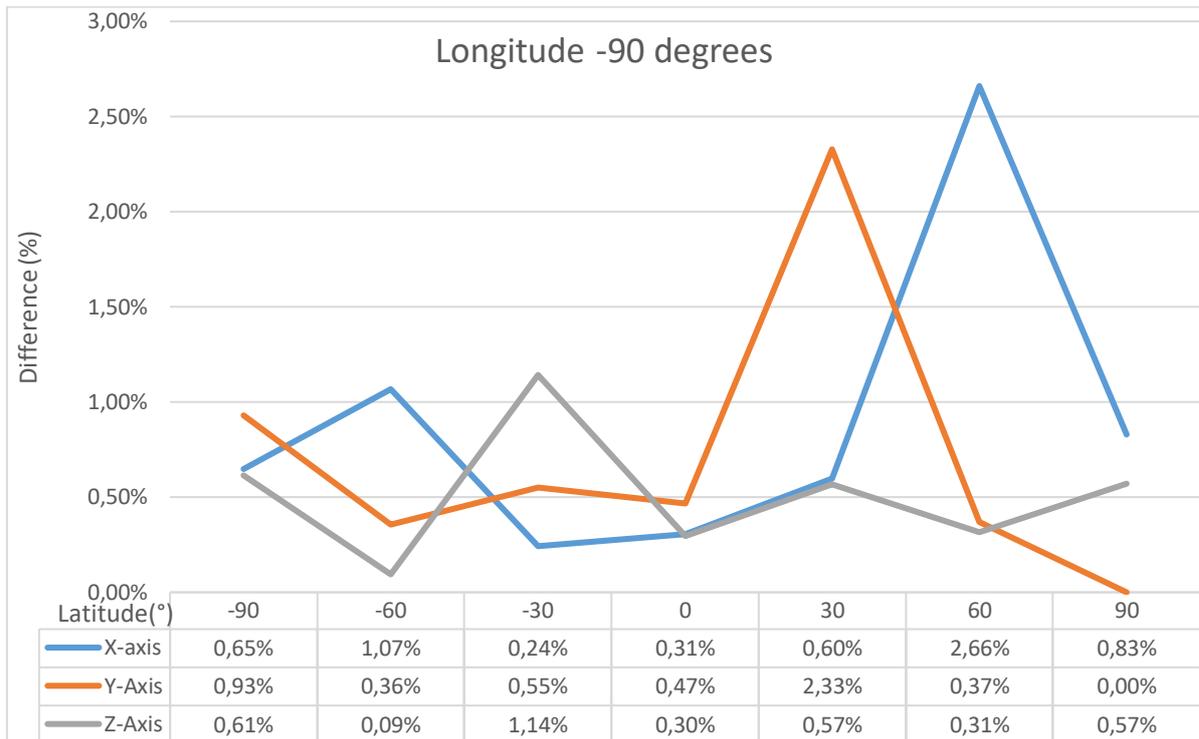


Figure 3.4. Difference % between longitude -90°.

The overall average difference percentage is 0.45%. In 0° longitude maximum difference % is 1.62 % at 60° latitude on X-axis, while other difference percentages remain under 1%. In 90° longitude, biggest difference % is at 30° latitude on Y-axis with 7.18% difference and -60° latitude with 5.13% difference. In other latitudes difference remains mostly under 1%. In 180° longitude biggest difference percentage is at X-axis in -60° latitude with 3.43% difference. Other latitude difference remains mostly under 1 %. With longitude at -90° (270°), biggest difference is at X-axis in 60° latitude with 2.66% difference. Other latitudes remain mostly under 1%. The overall difference calculation results show that the accuracy of the IGRF model onboard the OBC is under 1%. Differences are mostly remaining under 1% and the biggest percentage is at 90° longitude and -60° (+N | -S) latitude and 30° (+E | -W) latitude.

3.2 Simulating attitude control using magnetorquers

Three axis magnetorquers will be used for momentum damping and attitude control. Momentum damping is achieved with detumbling b-dot algorithm, which is being developed by another team

member and is not shown in this thesis. Simulations in this chapter are made with presumption that satellite is already detumbled, desired attitude positions are given by Kalman filter and it needs to achieve given position and also it has to keep to keep it stable.

3.2.1 Magnetorquer Dynamics

Simulating the control algorithm and the dynamics is not easy due to the constant change of magnetic field where the satellite is moving. The magnetic flux field affecting the body frame $B = [B_x \ B_y \ B_z]^T$ is not constant. By creating its own magnetic field with the three coils inside the body, the satellite can adjust its attitude (3.3) [11].

$$m_b = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} \quad (3.3)$$

Where $m_{x,y,z}$ – magnetic field in axis.

Produced torque given as a cross product in chapter two (2.1) can be derived as the cross product between produced torque and Earth's magnetic field. Writing the magnetic moment matrix based on the control voltage can be derived as:

$$\begin{bmatrix} \frac{N_x S_x V_z}{R_x} \\ \frac{N_y S_y V_z}{R_y} \\ \frac{N_z S_z V_z}{R_z} \end{bmatrix} = \tau_b \times B_b \quad (3.4)$$

where $N_{x,y,z}$ – number of turn in coil,

$S_{x,y,z}$ – surface area of the coil, m²,

$V_{x,y,z}$ – generated voltage, V,

$R_{x,y,z}$ – resistance, Ω.

When separating the voltage the formula can be defined as:

$$\begin{bmatrix} \frac{N_x S_x}{R_x} & 0 & 0 \\ 0 & \frac{N_y S_y}{R_y} & 0 \\ 0 & 0 & \frac{N_z S_z}{R_z} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \tau_b \times B_b \quad (3.5)$$

When solving the equation to derivate voltage the control matrix for the coil M_{coil} can be written as:

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} \frac{N_x S_x}{R_x} & 0 & 0 \\ 0 & \frac{N_y S_y}{R_y} & 0 \\ 0 & 0 & \frac{N_z S_z}{R_z} \end{bmatrix}^{-1} (\tau_b \times B_b) = M_{coil}^{-1} (\tau_b \times B_b) \quad (3.6)$$

This control matrix is derived because magnetorquer coils need voltage inputs and the controller gives outputs as torques.

3.2.2 Controller development and working principal

Simulated control system is based on a Proportional Derivative (PD) controller, which controls the actuation torque on any axis (3.7) [12]. In this thesis when simulating this controller it is presumed the satellite has already been detumbled with the b-dot algorithm and the Kalman filter model is giving the input. The controller is meant to take over when the b-dot has completed the detumbling. The controller input is in the desired attitude in Euler angles and it gets feedback from current satellite position. Desired position is transformed from Euler angles to quaternions. Then the desired attitude quaternion is inversed and multiplied with the quaternion of actual attitude to get the attitude error quaternion (Figure 3.5).

$$\tau_{con}^b = -k_p \varepsilon - k_d \omega_{ob}^b \quad (3.7)$$

where k_p – proportional gain,

ε – error quaternion vector,

k_d – derivative gain,

ω_{ob}^b – satellite's angular velocity.

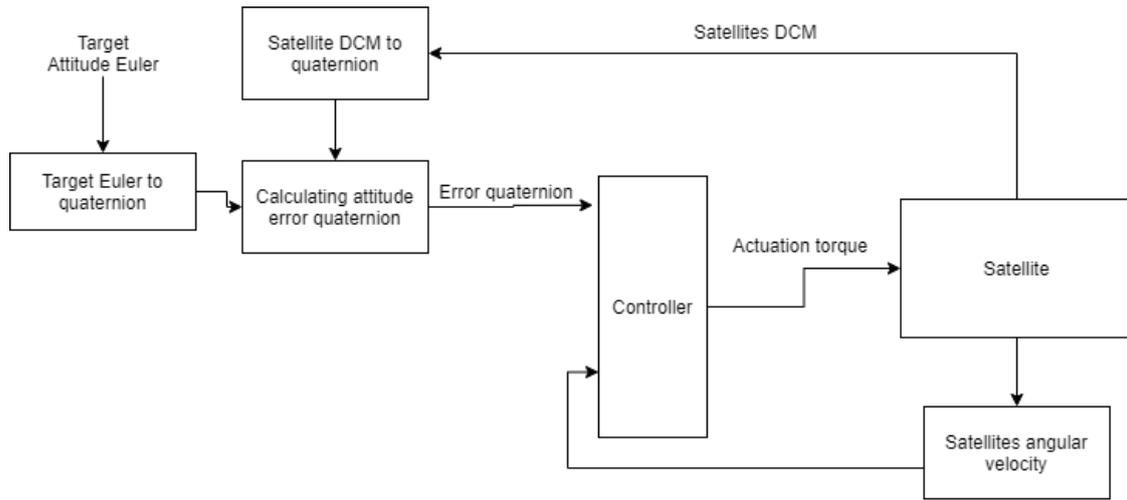


Figure 3.5. Controller IO signals.

Satellites output will be DCM that must be converted to quaternions, so it could be used in calculating the attitude error. Because the controller output will be torque and magnetorquers are controlled with voltage input, an additional part must be added to give an output voltage related to torque. This additional part is derived from control matrix with voltage inputs (3.8).

$$V = -\frac{M_{coil}^{-1}}{|B^b|} [-k_p(B^b \times \varepsilon) - k_d(B^b \times \omega_{ob}^b)] \quad (3.8)$$

where M_{coil}^{-1} – coil control matrix,

B^b – satellite's magnetic flux vector,

ε – error quaternion vector,

ω_{ob}^b – satellite's angular velocity,

$k_{p,d}$ – gains proportional and derivative.

This addition to the controller generates required voltage relating to the torque needed for the attitude change.

3.2.3 Controller simulation

As previously mentioned the controller is simulated with the presumption that the satellite is already detumbled. The simulated model is made in Simulink and MATLAB. Additional satellite

model is also created in Simulink with magnetorquers and reaction wheels to visualize the attitude positioning (Figure 3.6). Created satellite body has two cameras. All additional components can be configured with masses for additional accuracy, to see how it affects the satellites behaviour. Simulations made in this thesis do not have those additional masses.

Simulation gets its input parameters from written MATLAB code (Appendix 3). Simulation parameters were given as following:

- Simulation time for both cases $t = 3000$ seconds
- Initial angular velocity for both cases $\omega_{ob}^b = [0.2 \ -0.1 \ 0.07]$
- Initial attitude in Euler angles $\Theta_{i1} = [156 \ 85 \ 5]$ and for case 2: $\Theta_{i2} = [79 \ 77 \ 56]$
- Target attitude for case 1: $\Theta_{t1} = [0 \ 0 \ 0]$ and for case 2: $\Theta_{t2} = [10 \ 9 \ 44]$
- Magnetic coil matrix $M_{coil} = \begin{bmatrix} 0.12 & 0 & 0 \\ 0 & 0.12 & 0 \\ 0 & 0 & 0.12 \end{bmatrix}$

Due to satellite's magnetic field and Earth's magnetic field not being constant, to calculate the produced voltage, a random number generator with parameters of IGRF minimum and maximum values must be added to the computation logic. Number generator sample time was set for 5 seconds and they were converted into a 1x3 vector.

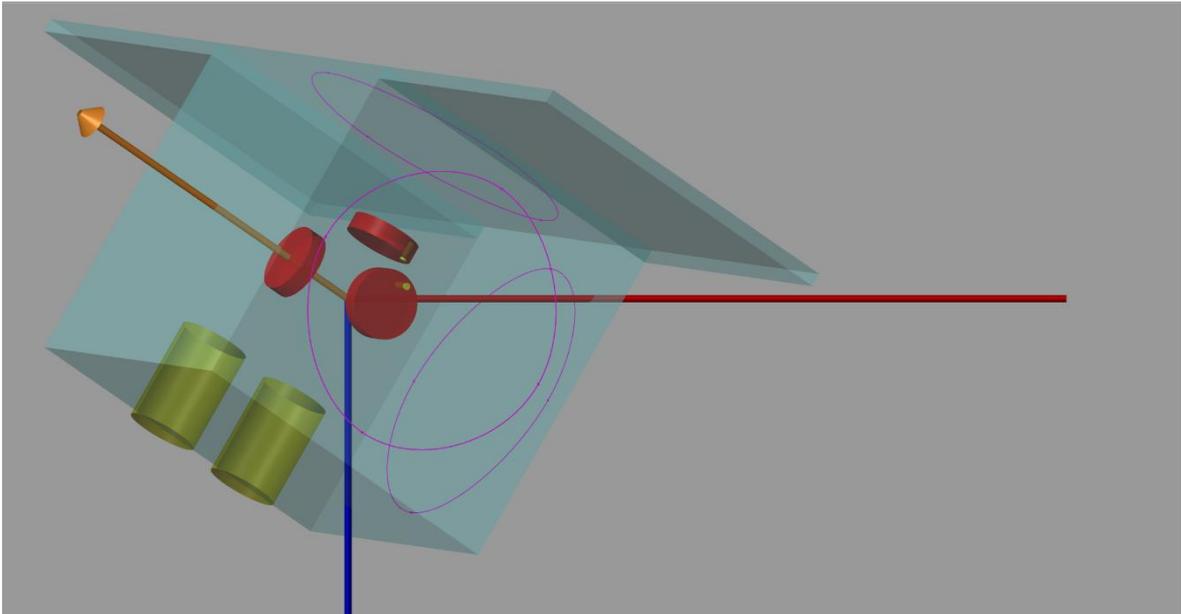


Figure 3.6. Screenshot of created satellite model with 2 cameras, magnetorquers and reaction wheels.

In case one, controller gains were set quite high $k_p = 0.8$ and derivative gain was set to $k_d = 0.5$. Simulation time was set to 1000 seconds, but as a result the controller became unstable and too aggressive and the satellite body model crashed. Then gains for case one were set to $k_p = 0.006$ and derivative gain was set to $k_d = 0.005$. In case one desired position was monitored in Euler angles (Figure 3.7) and also in quaternions (Figure 3.8). Also generated torque (Figure 3.9) and voltage (Figure 3.10) were monitored with angular velocity (Figure 3.11). After tuning the controllers for case two proportional gain was set to $k_p = 0.0006$ for smoother transition and derivative gain was left to $k_d = 0.005$. In case two the monitored values were the same: desired attitude in Euler angles (Figure 3.12) and quaternions (Figure 3.13), torque (Figure 3.14) and needed voltage (Figure 3.15) and angular velocity (Figure 3.16).

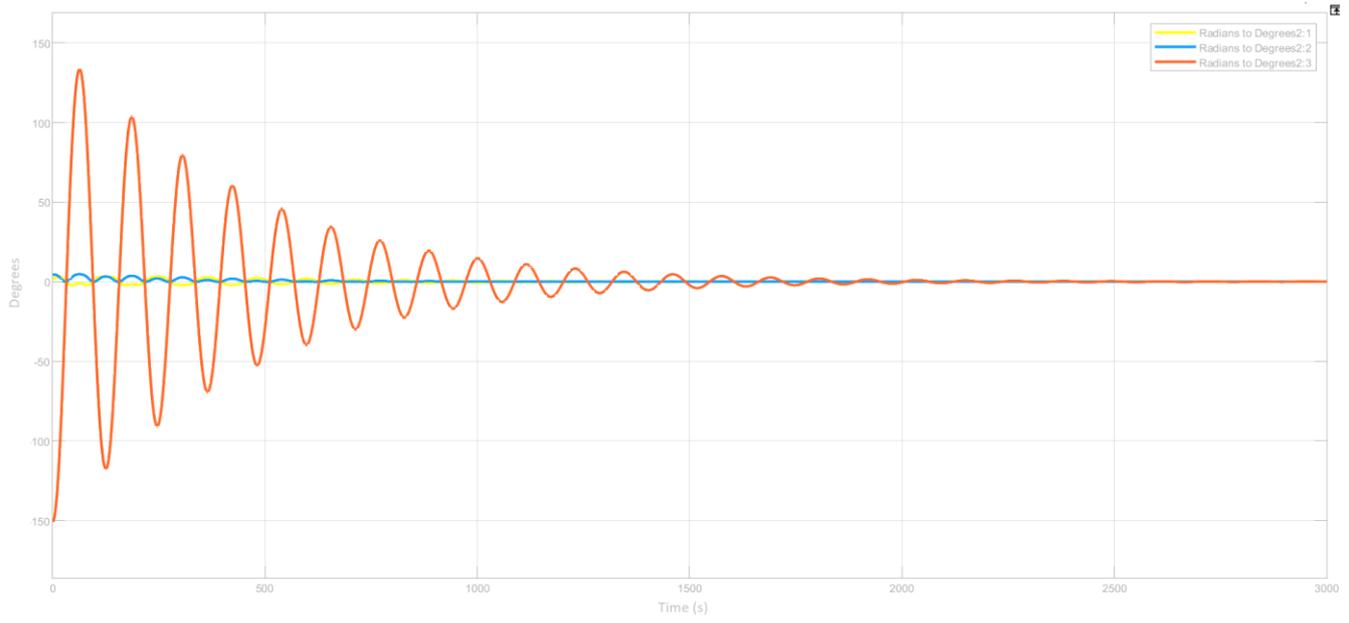


Figure 3.7. Case 1 Euler angles.

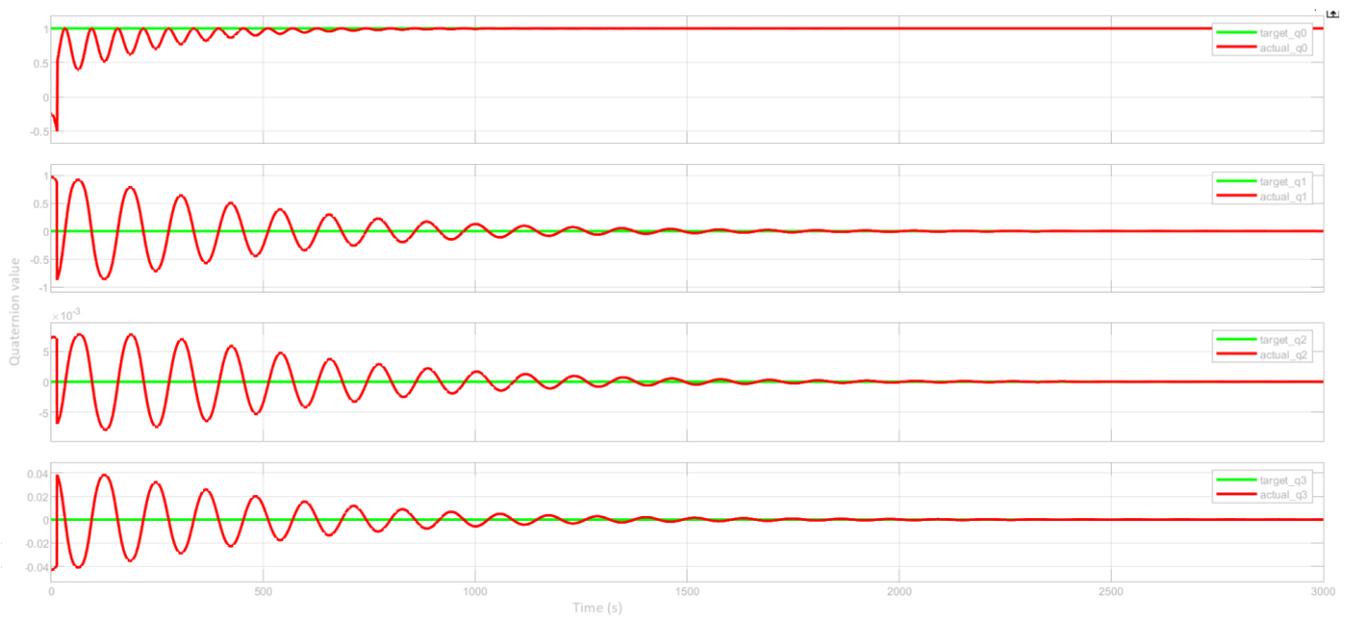


Figure 3.8. Case 1 quaternions.

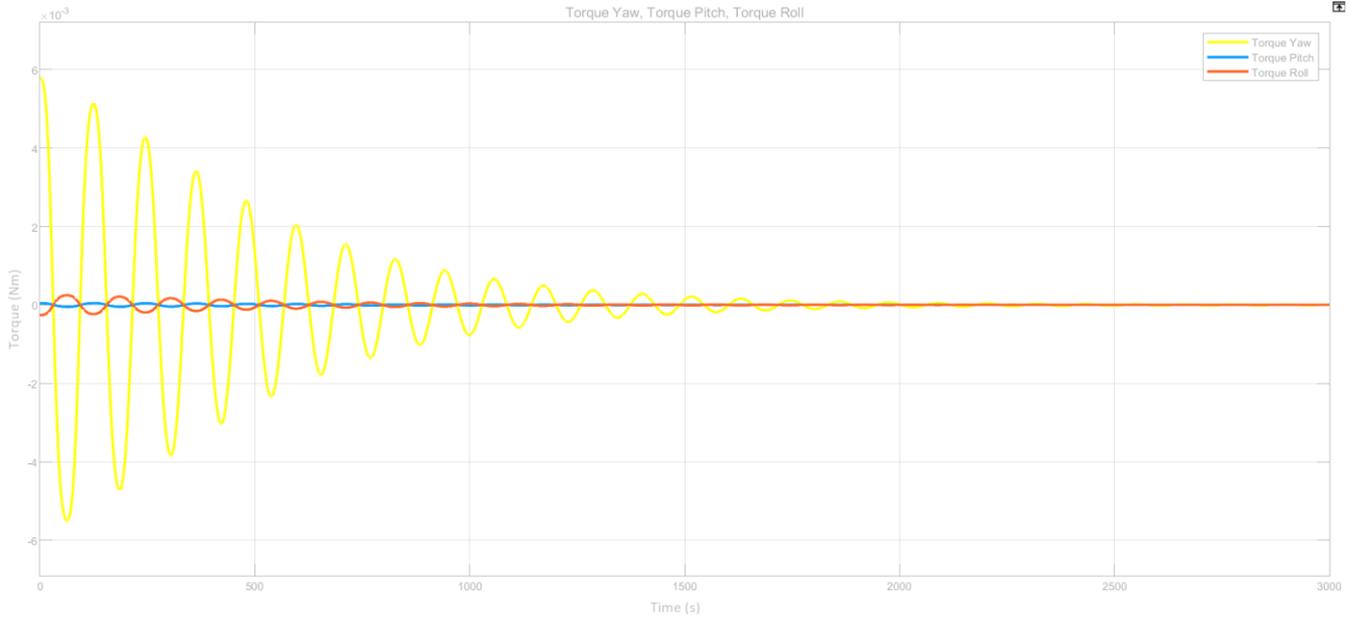


Figure 3.9. Case 1 Torque.

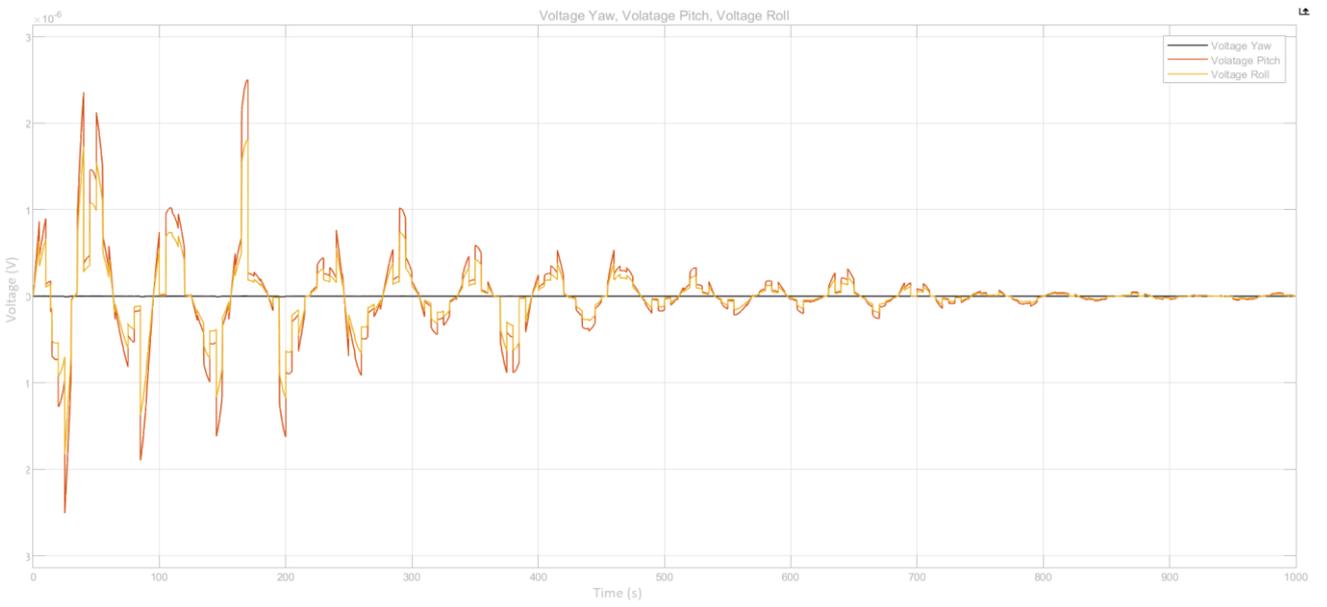


Figure 3.10. Case 1 voltage.

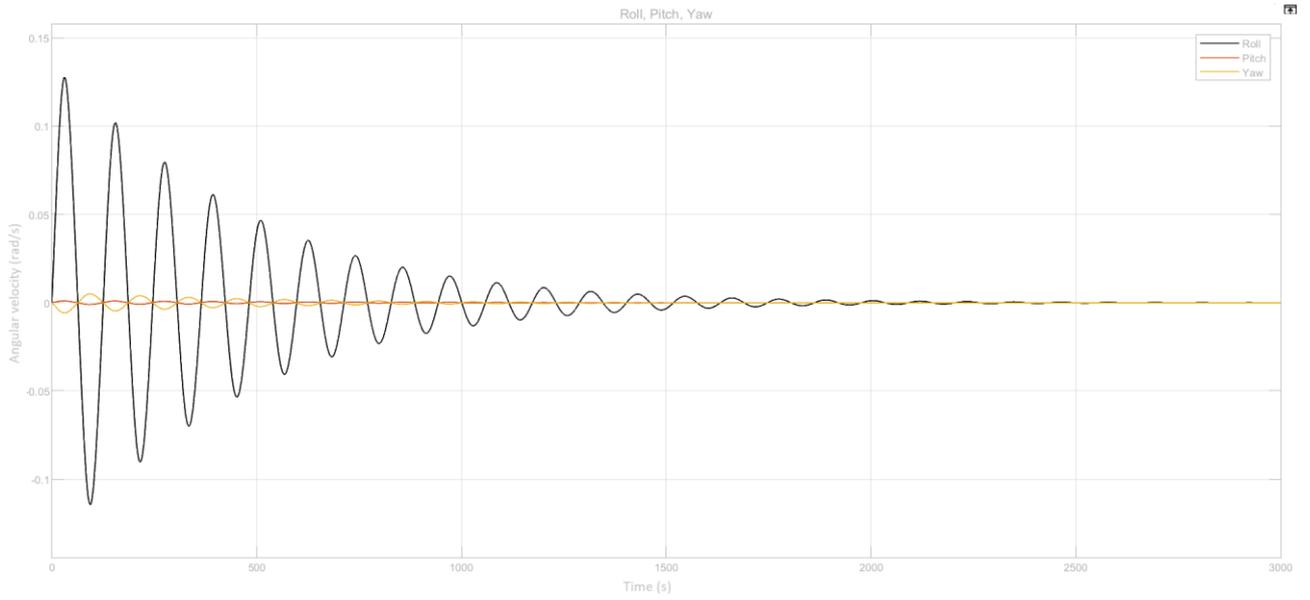


Figure 3.11. Case 1 angular velocity.

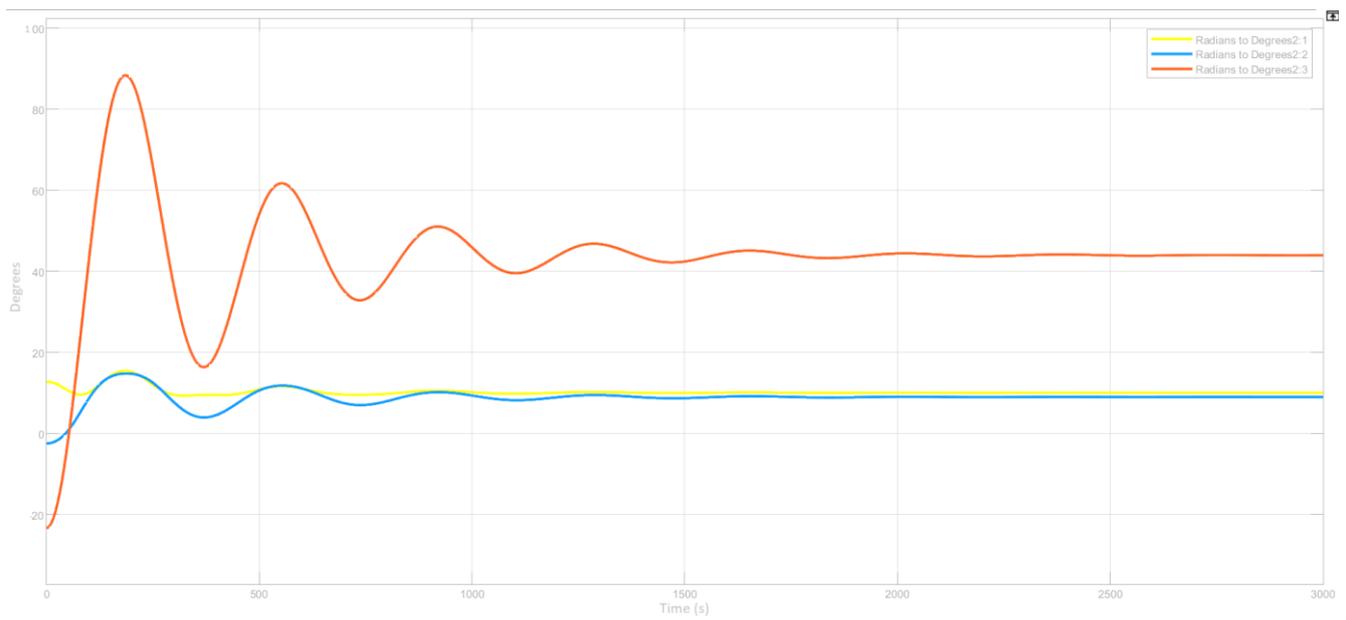


Figure 3.12. Stabilisation of Euler angles case 2.

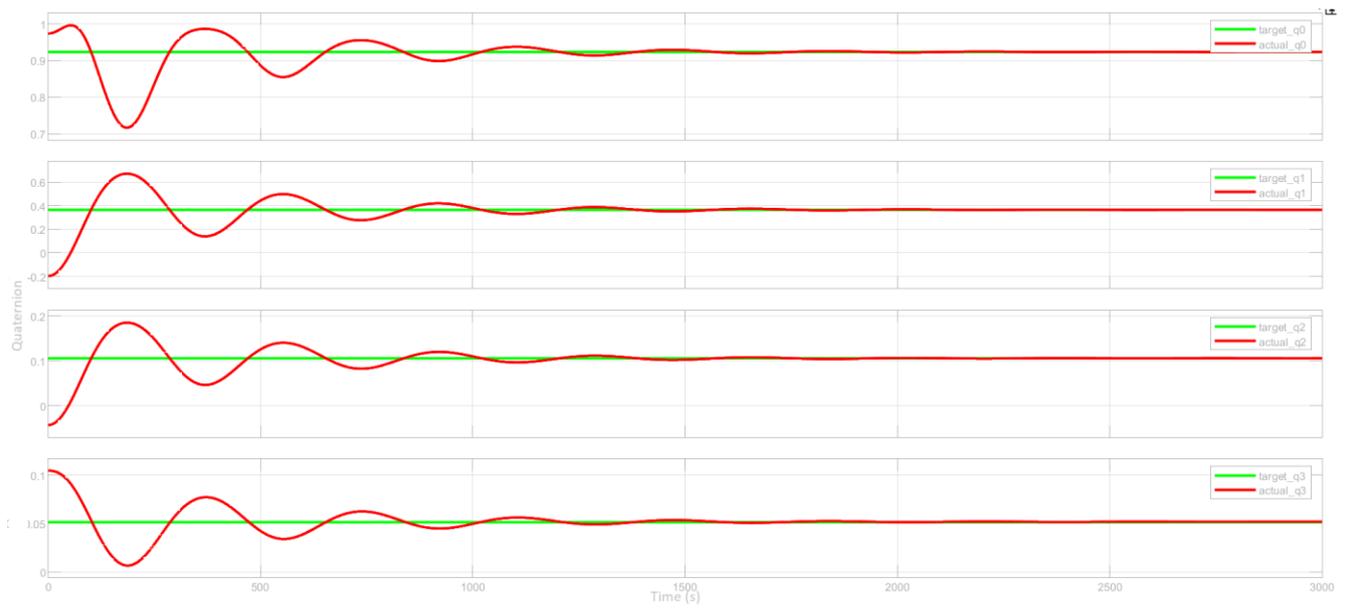


Figure 3.13. Stabilisation of quaternions case 2.



Figure 3.14. Voltage related to actuation torque case 2.

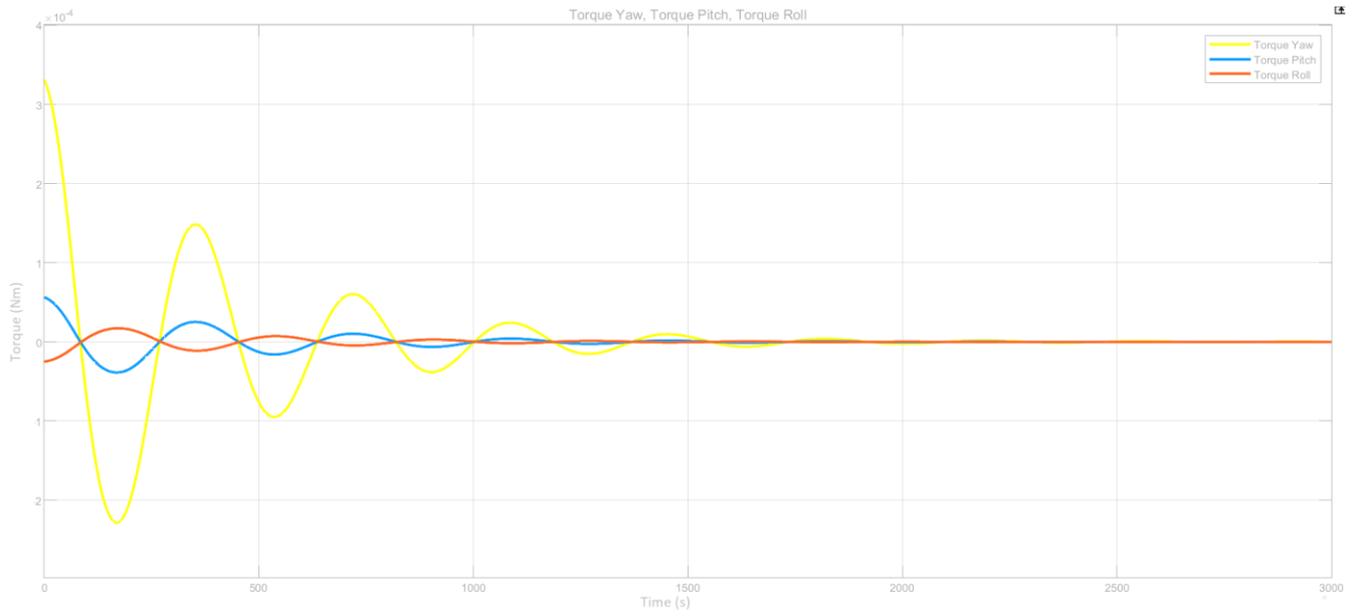


Figure 3.15. Actuation torque case 2.

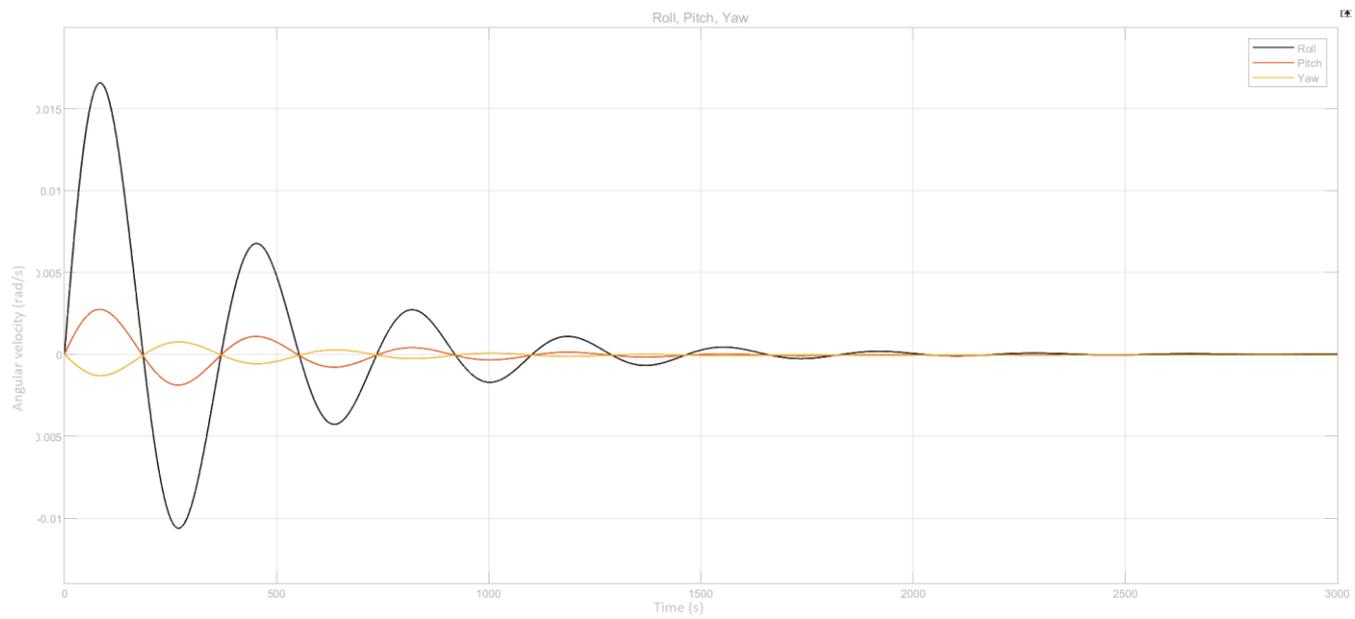


Figure 3.16. Case 2 angular velocity.

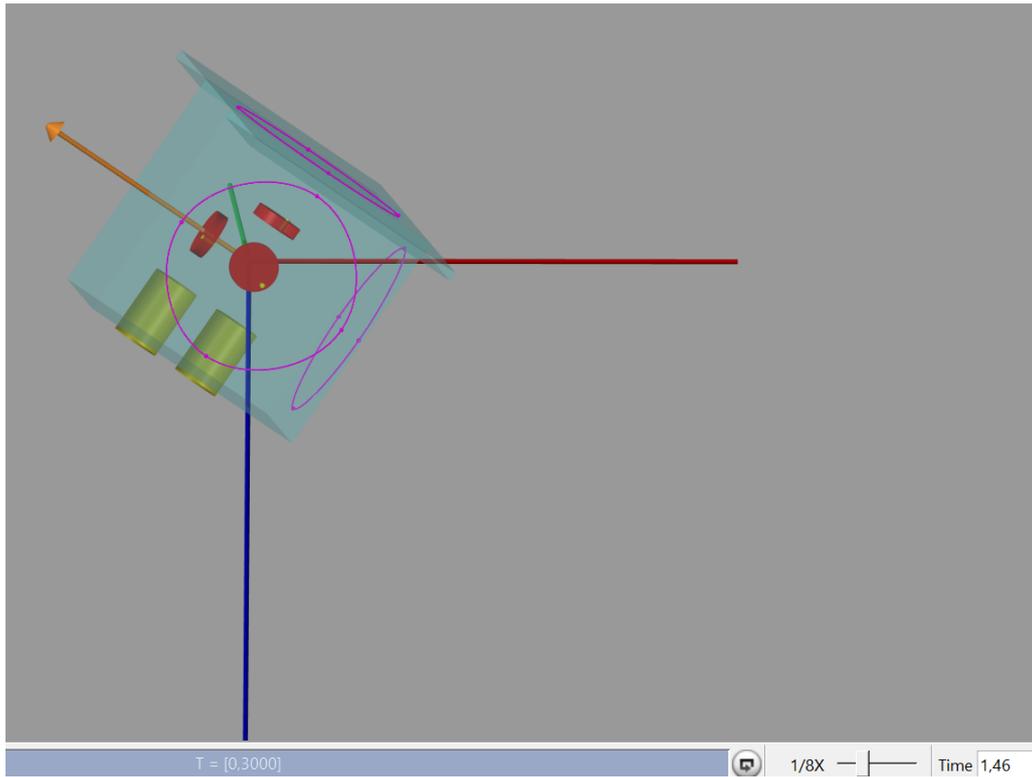


Figure 3.17. Case 2 satellite body position at 1.46 seconds.

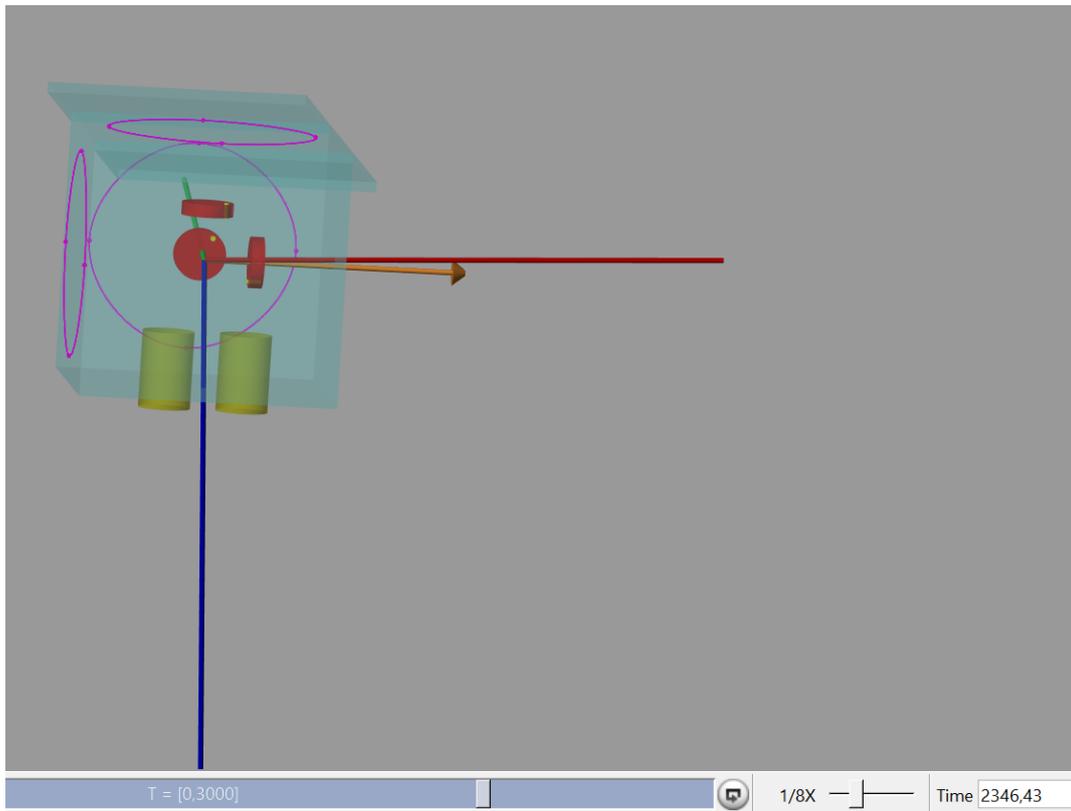


Figure 3.18. Case 2 satellite body at 2346.43 seconds.

From the satellite's body visualisation it can be seen that satellite's position is close to desired attitude.

Stabilising the satellite with magnetorquers and using an PD controller with added model for transforming the torque to voltage was simulated. First case scenario:

- Initial attitude given in Euler angles [156 85 5] and target attitude of [0 0 0].

Attitude change and stabilizations was achieved about 2000 seconds after the initialization, which is about the time of half an orbit. Second case scenario:

- Initial attitude [79 77 56] and target attitude of [10 9 44].

Second case proportional gain was lower, while derivative gain was left the same. Second case scenario achieved desired attitude and stabilisation about 2000 seconds after initialization. While the stabilisation time was about same, the needed voltage and generated torques were lower. Case two stabilisation was overall smoother and the timeframe was about the same ≈ 0.5 orbits.

3.3 SIMULATING THE ORBIT

TalTech satellite is in Low Earth Orbit (LEO) with altitude of 500 km and inclination of 98 degrees. Using this information orbital period can be calculated using Newton's form of Kepler's third law (3.9) [13]. In this thesis orbit is simulated in two different versions. One is simulated in MATLAB and visualised. Other version is simulated in Simulink Aerospace Blockset CubeSat Simulation Library [14]. TLE file format was acquired from Estcube-1 satellite project and for simulation purposes, the elements were modified [12]. Source code was provided by supervisor Eiko Priidel and modified accordingly with additional parts by author.

$$T = \sqrt{\frac{4\pi^2 R^3}{GM}} \quad (3.9)$$

where T – orbital period, sec,

R – average radius of orbit for the satellite = $R_{\text{earth}} + \text{height}$, m,

$R_{\text{earth}} = 6.37 \times 10^6$, m,

G – *gravitational constant* $\approx 6,67 \times 10^{-11} (m^3/(kg * s^2))$.

M – mass of Earth = $5,98 \times 10^{24}$ kg

Calculated period is $T = 5677$ sec = 96.36 minutes. It makes 15 orbits per day but only two out of those 15, are over the ground station. Meaning ground station has two opportunities per day to communicate with the satellite. Before simulating the orbits, its Keplerian elements are needed to know the shape of the orbit. These elements are given by TLE file and calculated in simulations in the MATLAB simulation version. Semi-major axis defines the size and shape of the orbit and is defined by the distance between apogee and perigee divided by two (3.10) [13].

$$a = \sqrt[3]{\frac{T^2 GM}{4\pi^2}} \quad (3.10)$$

where T – orbital period, sec,

M – combined mass of primary and secondary body.

Calculated semi-major axis $a = 6879.68$ km.

Earth's eccentricity defines the shape of the orbit. Earth's eccentricity is about $e = 0.0167$ and its vector can be calculated (3.11):

$$\vec{e} = \frac{\vec{v} \times \vec{h}}{\mu} - \frac{\vec{r}}{|\vec{r}|} \quad (3.11)$$

where \vec{v} – orbital velocity vector,

\vec{h} – angular momentum vector,

μ – gravitational constant $\approx 6,67 * 10^{-11}$ ($m^3 / (kg * s^2)$)

r – orbital radius

Right ascension of ascending node in degrees is the angle from the vernal equinox to the ascending node (3.12). If $n_j < 0$ then $\Omega = 360 - \Omega$.

$$\cos(\Omega) = \frac{\hat{i} * \vec{n}}{|\hat{i}| |\vec{n}|} \quad (3.12)$$

Where \hat{n} – vector pointing towards the ascending node.

Argument of Perigee is the angle from the ascending node to the eccentricity vector measured in the direction of the satellites motion (3.13). If $e_k < 0$ then $\omega = 360 - \omega$.

$$\cos(\omega) = \frac{\vec{n} \cdot \vec{e}}{|\vec{n}| |\vec{e}|} \quad (3.12)$$

Where \vec{n} – vector pointing towards the ascending node,

\vec{e} – eccentricity vector pointing towards the perigee.

True anomaly is the fraction of an orbit period which has elapsed since perigee. It is expressed as an angle [15].

$$M = E - e \sin(E) \quad (3.13)$$

where E – eccentric anomaly,

e - eccentricity.

Calculated orbital elements were implemented to the TLE file and run in simulations where other needed elements for positioning data were calculated with MATLAB. In Simulink library following orbital elements were inserted:

- Semi-major axis: $a = 6879.68$
- Eccentricity: $e = 0.0010357$
- Inclination: $i = 98^\circ$
- Right ascension of the ascending node in degrees: $RAAN = 165.0210$
- Argument of Apoapsis in degrees: $AoP = 146.7093$
- True Anomaly in degrees: $Ta = 213$

Simulation time was 86400 second which is ≈ 15 orbits. Simulation run on Simulink gives data both in Earth – Centred Inertial (ECI) (Figure 3.23) and Earth – Centred, Earth – Fixed (ECEF) (Figure 3.23) (Appendix 4) coordinate frames. ECI centred coordinate system's origin point is at the centre of mass and is an inertial frame while ECEF is fixed with respect to the surface of the Earth. ECI x-axis

goes through the point where vernal equinox and the equatorial plane cross, z-axis goes through the Geographic North Pole and the y-axis is the cross product of z-and x-axis [13].

ECEF x-axis crosses the point where the Greenwich meridian crosses the equatorial plane, z-axis crosses the Geographical North Pole and y-axis is the cross product between the x-and z- axis [13]. Simulated orbital tracks (Figure 3.19) were also projected to the ground, which produce a ground track (Figure 3.20) from where it can be observed when the satellite will be in ground stations FOV (Figure 3.21). Also, satellite's geodetic longitude and latitude and altitude were monitored (Figure 3.24). In MATLAB simulation satellite's position in ECI frame (Figure 3.26) and ECEF frame (Figure 3.27) were monitored to compare them with Simulink results. Also, satellite's velocity in ECEF frame was plotted (Figure 3.28).

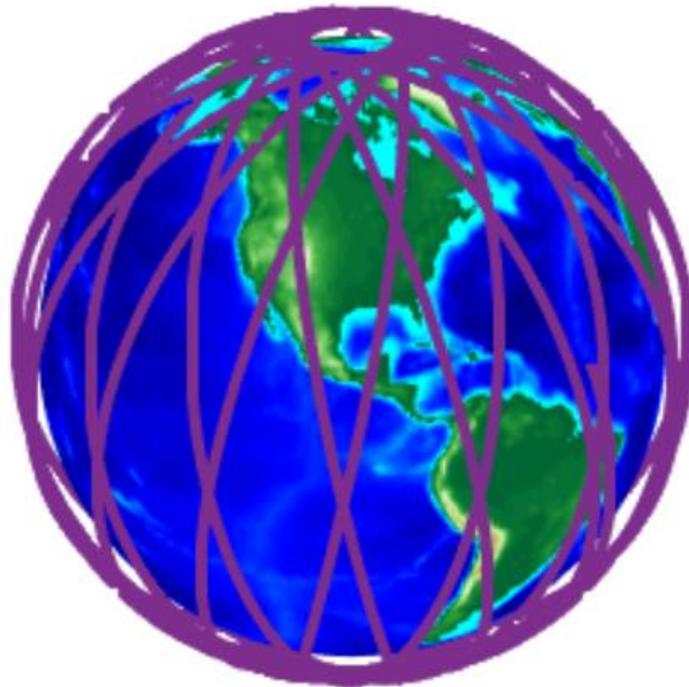


Figure 3.19. Orbital Track generated in Simulink.

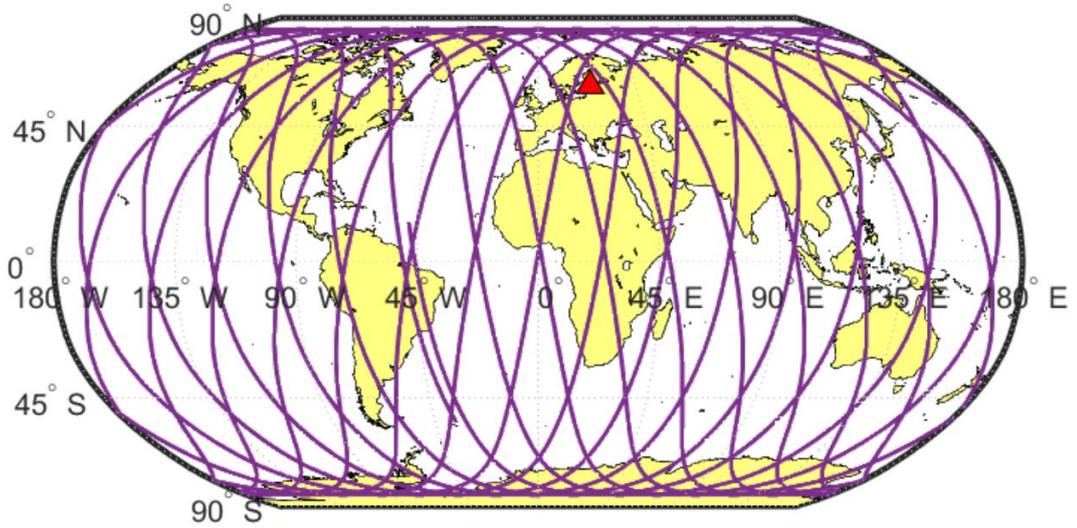


Figure 3.20. Orbital ground track generated in Simulink where the red triangle is the ground station.

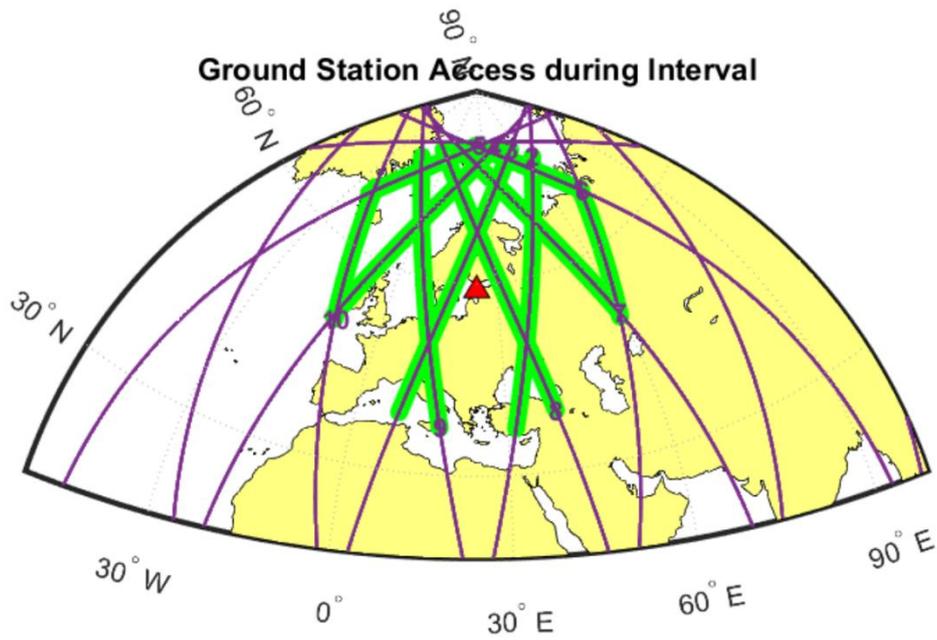


Figure 3.21. Generated orbits with ground station in reach.

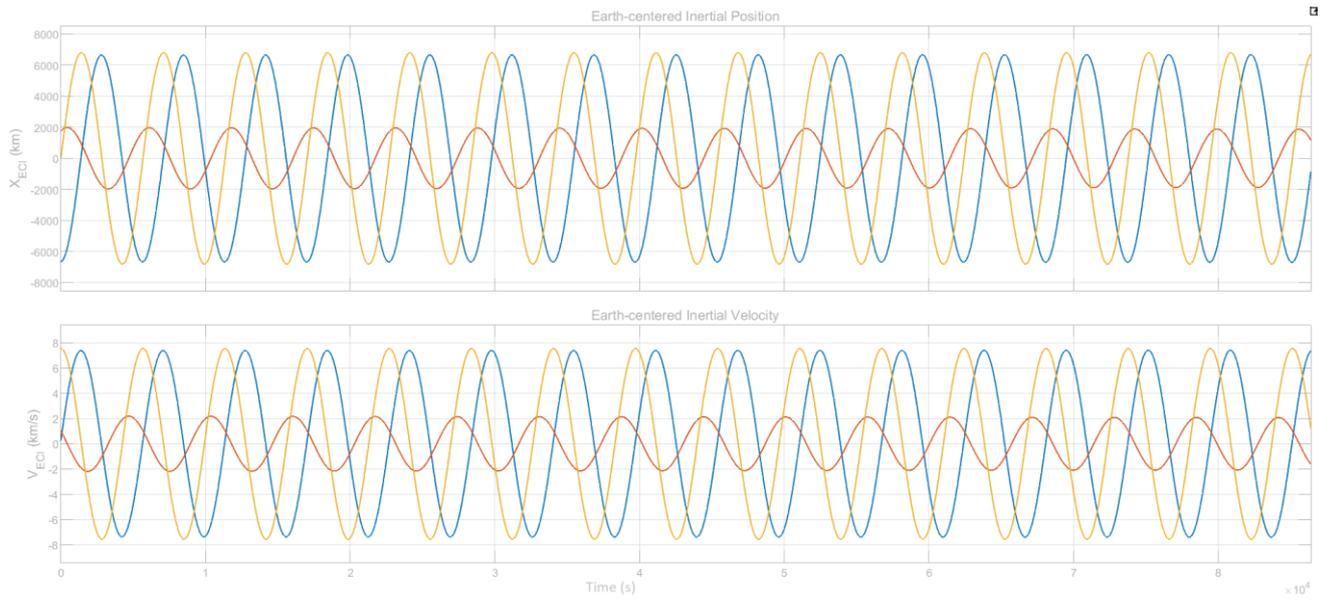


Figure 3.22. Satellite Position and Velocity in ECI frame where x – yellow, y – blue, z – red.

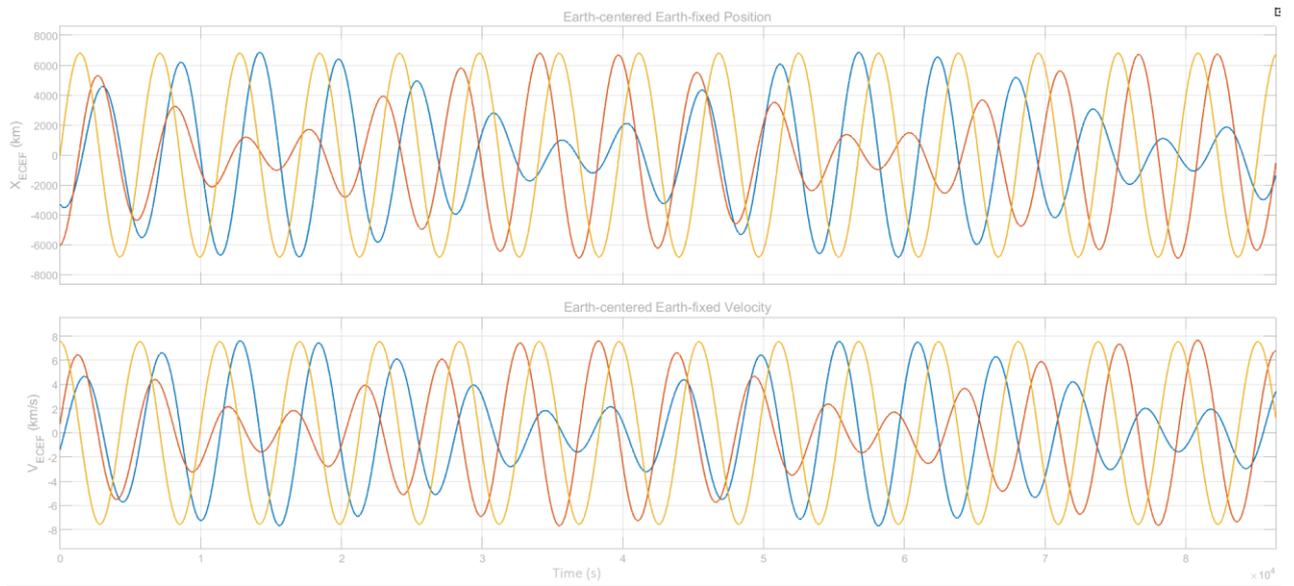


Figure 3.23. Satellites position and velocity in ECEF frame where where x – yellow, y – blue, z – red.

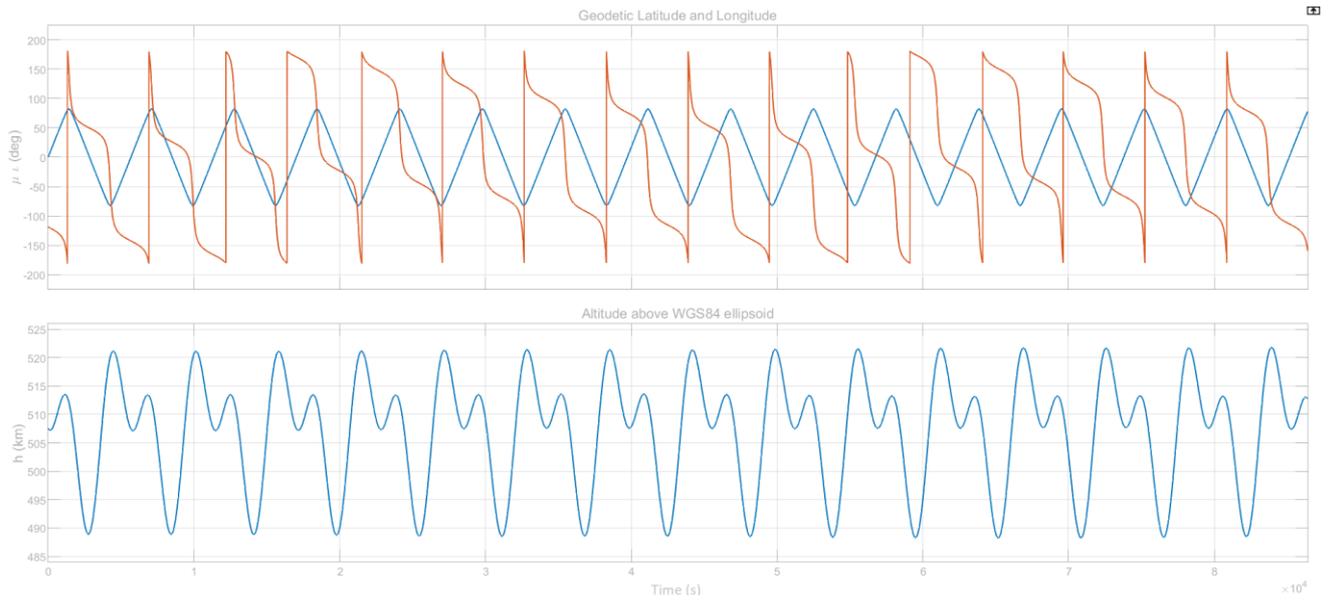


Figure 3.24. Satellite's Latitude – blue , Longitude – yellow and Altitude.

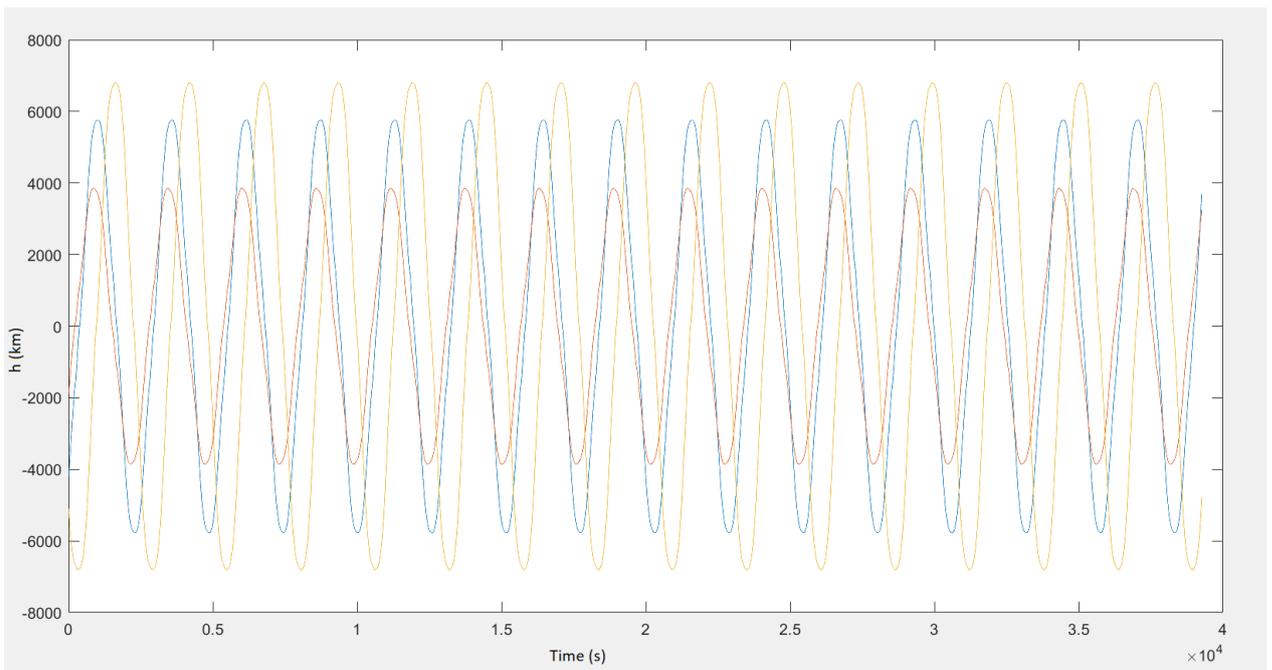


Figure 3.26. Satellite position in ECI frame with MATLAB where axes are: yellow – x, blue – z, red – y.

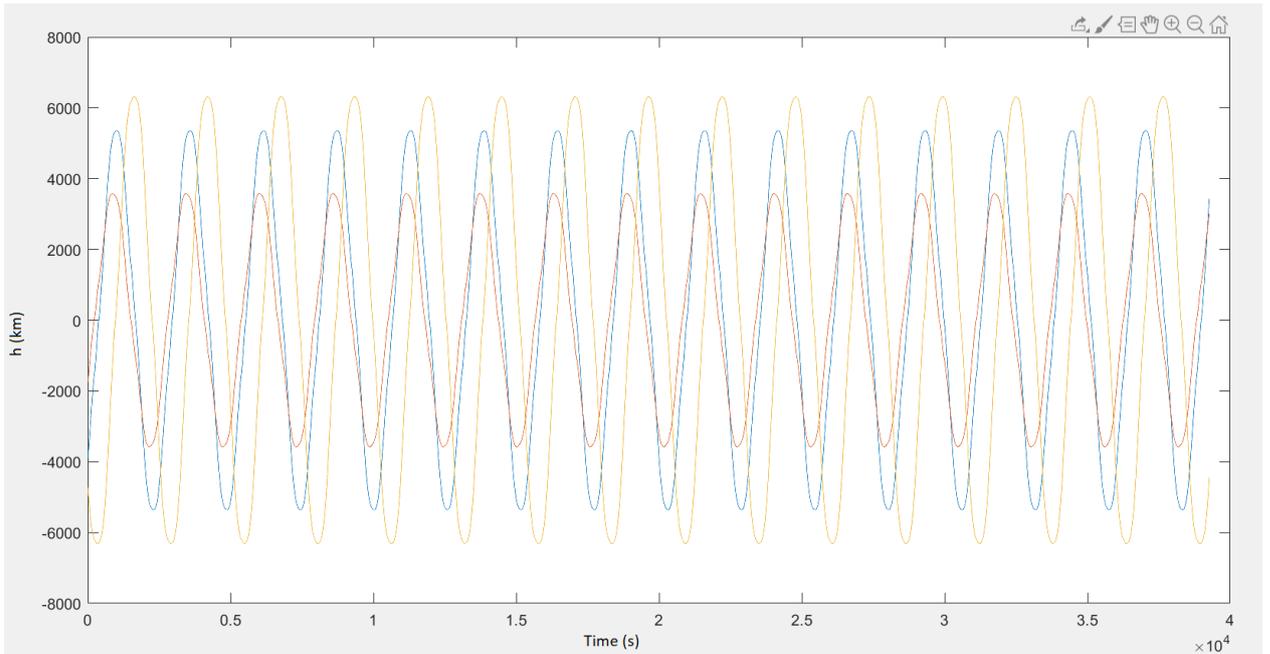


Figure 3.27. Satellite position in ECEF frame MATLAB where axes are: yellow – x, blue – z, red – y.

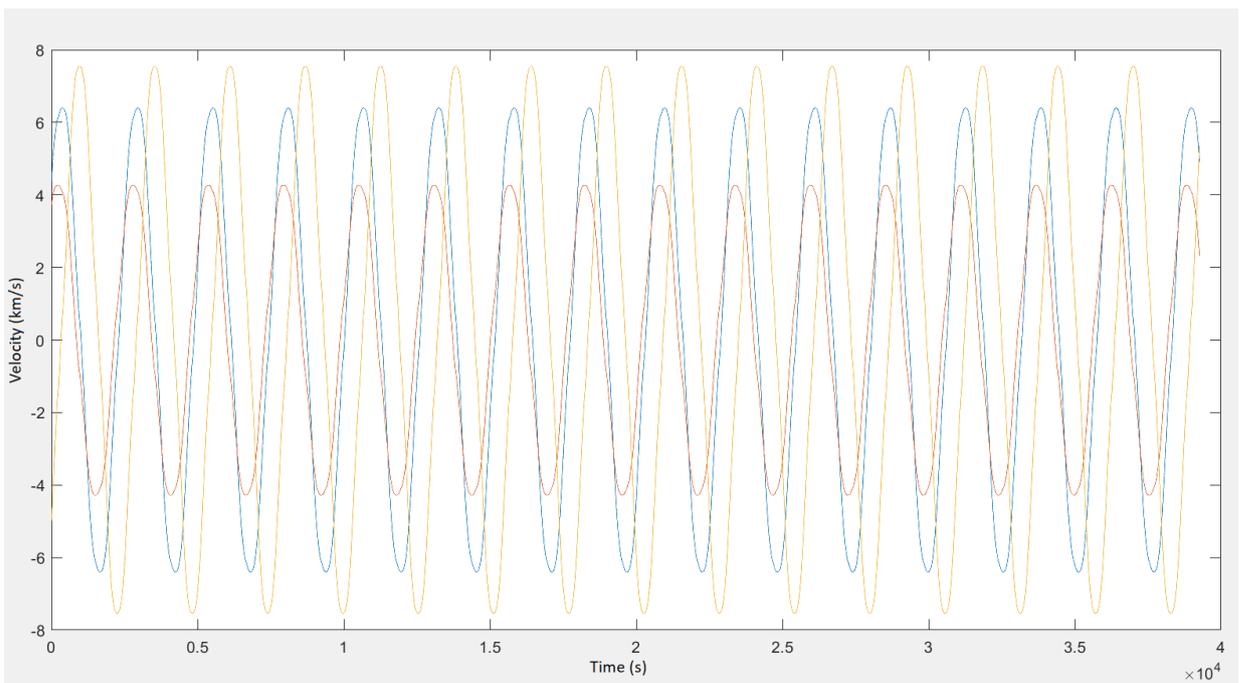


Figure 3.28. Satellite velocity in ECEF frame where yellow – x, blue – z, red – y.

First try of simulating was unsuccessful due to wrong insertion and calculation method of semi major axis, which put off the major axis for about 400 km. After fixing the method and implementing the right inputs and pathways to other calculations depending on the semi major axis, simulations started working and data could be plotted and visualised. Overall, the simulation orbits were both

similar. Numerical values of axes' positions were mostly the same, however there is a difference between ECI and ECEF frame data. The difference comes from the origin of coordination points. ECEF position values are smaller in both the MATLAB and Simulink simulations. While the coordinate values are mostly the same in both simulations there are decreases in ECEF frame in Simulink model. These changes come from implemented additional forces by Simulink satellite body model that position the satellite while on orbit and also the fixed frame point, which is changing due to ECEF frame. This has not been implemented in MATLAB simulation and while the satellite performs the orbits it does not move its own body frame and the axes are the same. Satellites altitude changes are taken from World Geodetic System (WGS84) model, already implemented in Simulink model. From these results, it can be presumed that while both simulations results were largely similar orbits, the analyses show that Simulink model shows more details. These details can be modified and implemented to MATLAB simulation model.

4. MODIFYING MAGNETIC CAGE CONTROL SCHEMATIC

Magnetic cage, also called as Helmholtz cage, used in this thesis was built by Nika Mukbaniani as his master's thesis to simulate Earth's magnetic field [16]. During the testing it was discovered that the H-bridge drivers that were controlling the relay board with Arduino signal did not work as described on the datasheet, and that they must be replaced. Measurements are inaccurate and the schematic and the board layout has to be changed for the next revision of the cage control PCB. A temporary fix was to control the switching signals manually, but the result was not satisfactory in long term. In the old solution DRV8836 Dual Low- Voltage H-Bridge IC [17] was used. It controlled a relay board receiving signals from Arduino board and with the combination of signals controlled the magnetic cage and simulated the fields. The problematic drivers were removed from the board (figure 4.1)

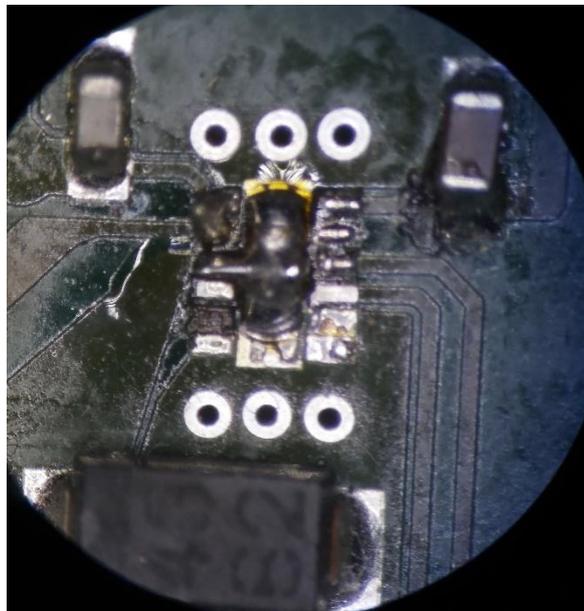


Figure 4.1. PCB under the microscope with the driver removed.

New offered solution (Figure 4.2) is based on MIC5018 high-side MOSFET driver [18] which is used as a logic level driver to control the signal. It has an input of +3.3V, but it can be exceeded and, in this solution, a +5V input is used. Control signal input is received from the Arduino. An inverter is placed on the low side to control the signal without the need to use additional Arduino signals. Offered solution to control both inputs with one signal was simulated in Falstad Circuit Applet and two switching cases were simulated to test polarity switching (Figure 4.3).

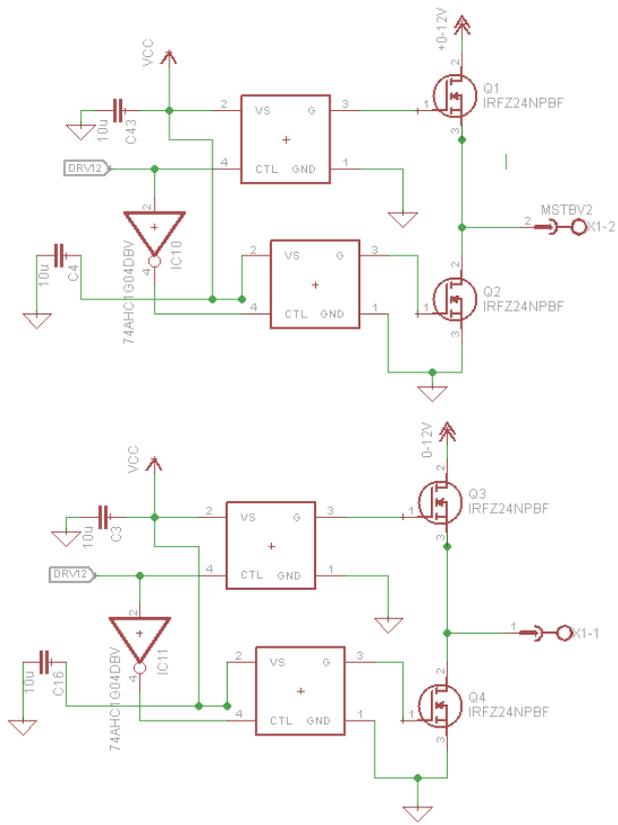


Figure 4.2. MIC 5018 FET driver control scheme.

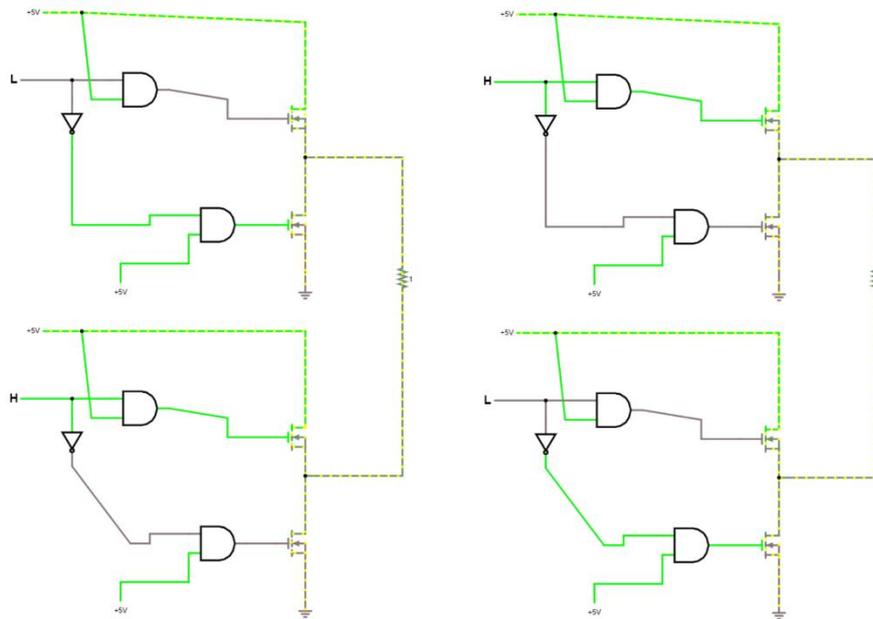


Figure 4.3. Switching logic simulated in Falstad: case 1- left, case 2 -right.

Simulated schematic logic worked as intended. The use of inverter eliminated the need for additional signals for low side controls. Using additional signals to control the drivers can and should be used when the switching frequency is high, but because the magnetic field simulator simulates earth's magnetic fields the switching frequency is low. This also means that the switching times do not affect the simulator's work. If the switching frequency would be high then the switching time should be considered with due to the MOSFET's charge and discharge times and that they would not overlap and shortcircuit the scheme. This would be fixed by using resistors in gate inputs. Also with using the MIC5018 drivers the relays were removed and the coil polarity could be controlled without the need of the relay board. The MIC5018 drivers used in this solution also have a built-in charge pump that eliminates the previous driver faults that occurred on low voltages.

Next step is to create a PCB with offered solution and then to test it on the Helmholtz cage. For schematic and future PCB development EAGLE software is used.

5. FUTURE WORK

Subjects brought out in this thesis have definite potential for improvement. After PD controller is tested on orbit, it can be developed further and real life feedback can be taken to consideration. Controller simulation model could be implemented into other models and vice versa to get more accurate readings and results. Control system working principle can be applied and modified to reaction wheels and use the ACS by angular momentum to merge it with magnetic control method. Also, IGRF model can be implemented to magnetic control. Future control system development next stage should be to develop it to work with two magnetic actuators and one reaction wheel. Orbital simulation models can be implemented in other models and should be done to simulate and research the satellite's behaviour in orbit, with every aspect simulated and overviewed. TalTech satellite project has not ended and with one satellite yet to be launched the future work is still ahead.

The magnetic field simulator control board is worked on outside this thesis, with the need to order the modified PCB and to test the solution in real time.

Furthermore, other ground testing stations can be worked on. With the control system testing station under additional development, new and more effective control and testing methods can be researched and developed. For instance, right now the cage is lacking a mechanism for satellite placement so that it would be stable. Additional testing equipment could be developed and built for other control systems like reaction wheel control station.

As this project is a first of a kind in TalTech, there is lack of previous work, equipment and experience, the university is opening itself to a whole new branch of science and research. The members are the people who, in the lack of better terms lay the groundwork and their work can be developed in the future.

SUMMARY

In this thesis four main issues were handled:

- Comparing the IGRF model used in the satellite with the model from NOAA.
- Developing control and stabilization system of the satellite.
- Simulating orbits and satellite behaviour in MATLAB.
- Modifying the Helmholtz cage control board schematic for controlling the coil polarity.

Tasks set in this thesis were achieved and the results were satisfactory. In the first main chapter a theoretical background and research on the subject was presented. The research materials used were mainly books, articles and reports on other projects in the same category. Main focus was on Cubesats themselves, how does an ADCS system work and what does it include, theoretics on orbits and astrophysics to know how the satellite will behave in space and what is needed to know to control it. Also theoretical overview on the mathematical elements used in the development of the control system. The second chapter was focused on simulations and development. First, the IGRF model developed for the satellite was simulated and compared with the current model, to show the difference in accuracy. The comparisons were made in four main longitude degrees and the difference is brought out in latitude. In 0° longitude the difference % remained under 1 %. In 90° the biggest difference percentage was 7.18 %. In 180° longitude biggest difference percentage 3.43 % and at -90° (270°) longitude the biggest difference is 2.66 % . The overall result was satisfactory with main differences remained mostly under 1%.

Next part was the development and simulation of the magnetic control and stabilisation system. The focus was on magnetic control as it is mainly used. If something happens with reaction wheels, the magnetic control is there to fall back on. PD controller based on quaternions controlling the needed torque was developed and simulated to change satellites attitude and keep it. Simulation inputs were given as Euler angles to make it easier for user. Simulated cases were succesful with the first case the satellite changing it's position from [156 85 5] to [0 0 0] and the second case with more tuned controller gains from [79 77 56] to [10 9 44]. Also satellite body model was developed in Simulink to visualise the behaviour and in the future to add more elements to the simulations. Third simulations were made for orbits and the ways to calculate Keplerian elements were brought out. Simulations were made in MATLAB and Simulink, the differences in satellites position and velocity were brought out. The results showed Simulink simulation being more precise in both ECI and ECEF frame systems. With Simulink having more elements in the model, it brought out the

difference of positioning in frame systems more accurately. Simulation timeperiod was 15 orbits and also orbital tracks and ground station FOV were brought out.

The third main part was to modify the ground station testing equipment, mainly the magnetic field simulator being unaccurate and having problems in controlling system. On the control board the controlling H-bridge system was improved and MIC5018 high-side FET drivers were implemented to control the polarity of the coils. Offered solution is in need of testing and it can be done in the future either by the author or other team members.

The goals set for this thesis were achieved and developed with the knowledge to be improved in the future. Value of this work is the control system for the CubeSat, testing of the IGRF, simulation and confirmation of orbital tracks and satellite's main behaviour. The paper also offered modification solution for ground station testing and simulating equipment.

KOKKUVÕTE

Käesolev magistritöö annab ülevaate teemadest mis on autori poolt valitud. Nendeks ostutusiid

- Maa magnetvälja mudeli analüüs
- Positsioneerimissüsteemi magneetilise juhtimise arendus ja simuleerimine.
- Satelliidi orbiidi simuleerimine ja analüüs
- Maa juhtimiskeskuse testimisvarustuse täiendamine

Käesolev töö on jaotatud neljaks põhipeatükiks. Esimeses põhipeatükis tuuakse välja teoreetiline taustauuring ja teemaga tutvumine. Põhilistesks uurimisallikateks olid teaduslikud raamatud, artiklid, ajakirjad ja aruanded sarnaste teemade ja projektide kohta. Peatükis tehakse ülevaade kuubiksatelliitidest, nende juhtimis- ja positsioneerimissüsteemidest - mida nad endas sisaldavad ja kuidas töötavad. Samuti tehakse ülevaade orbiitidest ja neid määravatest elementidest. Samuti tehakse ülevaade astrofüüsilisest taustast ja matemaatilistest elementidest mida töös kasutatakse. Töö teine põhipeatükk keskendub arendatud süsteemidele ja simulatsioonidele. Esimesena tuuakse välja maa magnetvälja mudeli simulatsiooni tulemuste võrdlus ja nende analüüs. Teisena näidatakse arendatud magneetilist juhtimissüsteemi, kus on kasutatud kvaternioonidel töötavat PD kontrolleri. Simulatsioonis on kasutatud Euleri nurki, et teha süsteem kasutajasõbralikumaks. Samuti on arendatud satelliidi enda mudel, et süsteemi visualiseerida. Arendatud süsteem ostutus töökindlaks, esimeses stsenaariumis saadi soovitud positsioon ja suudeti seda stabiilselt hoida. Samuti suudeti ettemääratud positsioon saavutada teises stsenaariumis. Kolmandana simuleeriti orbiiti, ning toodi välja kahe erineva simulatsioonimudeli tulemused. Simuleerimiskeskonnaks kasutati MATLAB-i ja Simulinki. Tulemuste võrdlusel selgus, et Simulinki simulatsioon on täpsem, kuid MATLAB-i oma on hõlpsasti täiendatav. Samuti sai visualiseeritud orbiidi jälg maal, ning samuti maajaama nähtavusraadius orbiitidel. Neljandas peatükis toodi välja maajaamas testimiseks kasutatava magnetpuuri kontrollplaadil olev elektriskeemi täiendus. Eelnev H-silla driver asendati MIC5018 võtme FET high-side draiveritega, et kontrollida mähiste polaarsust. Pakutud lahendust simuleeriti Falstad keskkonnas ning lahendust on vaja veel riistvaraliselt testida.

Antud töö eesmärgid said täidetud, ning panust projekti võib hinnata rahuldavaks ja realselt kasutatavaks. Töös arendatud lahendustel on potentsiaali, et neid tulevikus edasi arendada ja täiendada.

REFERENCES

- [1] TUT Mektory Nanosatellite programme homepage [WWW]
<https://www.ttu.ee/projektid/mektory-est/satelliidiprogramm-4/satelliidiprogramm/>
- [2] CubeSat Design Specification Rev. 13. The CubeSat Program. California Polytechnic State University, SLO. [WWW]
https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/56e9b62337013b6c063a655a/1458157095454/cds_rev13_final2.pdf
- [3] Cubesats overview [WWW] https://www.nasa.gov/mission_pages/cubesats/overview/
- [4] Peter W, Fortescue. Graham G, Swinerd. John P W, Stark. (2011) Spacecraft Systems Engineering Fourth Edition.
- [5] Kraft, M. White, M, N. (2013) MEMS for Automotive and Aerospace applications. Woodhead Publishing.
- [6] Zheng, You. (2017). Space Microsystems and Micro/Nano Satellites. [WWW]
<http://www.sciencedirect.com/science/article/pii/B9780128126721000047>
- [7] Org, P. (2016). Software Architecture and Development Tools API for Attitude Control System of TUT Nanosatellite: BSc thesis. Tallinn University of Technology, Tallinn.
- [8] Bate, Roger R. Mueller, Donald D. White, Jerry E. (1971) Fundamentals of Astrodynamics
- [9] International Geomagnetic Reference Field
[WWW]<https://www.ngdc.noaa.gov/AGA/vmod/igrf.html>
- [10] NOAA Magnetic field calculator [WWW]
<https://www.ngdc.noaa.gov/geomag/calculators/magcalc.shtml#igrfwmm>
- [11] Alvenes, F. (2012) Attitude Controller-Observer Design for the NTNU Test Satellite: Master's thesis. Norwegian University of Science and Technology.
- [12] Castillo, P. Hernandez, L. Gil, P. (2017) Indoor Navigation Strategies fo Aerial Autonomous Systems.

- [13] Chessab Mahdi, M. (2015) Orbit Design and Simulation for Kufasat Nanosatellite. Artificial Satellites The Journal of Space Research of Polish Academy of Sciences. Sciendo.
- [14] Simulink Aerospace Blockset Cubesat Simulation Library [WWW]
<https://www.mathworks.com/matlabcentral/fileexchange/70030-aerospace-blockset-cubesat-simulation-library>
- [15] Curtis, D, H. (2010) Orbital Mechanics for Engineering Students, Second Edition.
- [16] Mukbaniani, N. (2019) Earths magnetic field simulator: Master's thesis, Tallinn University of Technology, Tallinn.
- [17] DRV8836 datasheet [WWW]
<http://www.ti.com/lit/ds/symlink/drv8836.pdf?ts=1588781866670>
- [18] MIC5018 datasheet
 [WWW]<http://ww1.microchip.com/downloads/en/DeviceDoc/mic5018.pdf>
- [19] [WWW] <https://airtable.com/shrafcwXODMMKeRgU/tbldJoOBP5wINOJQY?blocks=hide>
- [20] Mahdi, M. (2018). Attitude Stabilization for CubeSat: Concepts and Technology
- [21] Nanosatellite and CubeSat database [WWW] <https://www.nanosats.eu/database>
- [22] Starin, S, R. Eterno, J. (2011) Attitude Determination and Control Systems. [WWW]
<https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20110007876.pdf>
- [23] Li, J. Post, M. Wright, T. Lee, R. (2013) Design of Attitude Control Systems for CubeSat-Class Nanosatellite. Journal of Control Science and Engineering
- [24] Noor, S. Ouhocine, C. Filipski, M. Ajir, M. Hamzah, N. (2004) Small Satellite Attitude Control and Simulation [WWW]
https://www.researchgate.net/publication/242370040_SMALL_SATELLITE_ATTITUDE_CONTROL_AND_SIMULATION
- [25] Todd, E Humphreys. (2002) Attitude Determination for Small Satellites with Modest Pointing Constraints [WWW]

https://pdfs.semanticscholar.org/05a5/43e58780f0fa6e4fe84f526eed760127e18.pdf?_ga=2.238385542.743983365.1586023473-268235458.1586023473

[264] Markley, F, Landis. Crassidis, John, L. (2014) Fundamentals of Spacecraft Attitude Determination and Control.

[27]Alvenes, F. (2012) Attitude Controller-Observer Design for the NTNU Test Satellite:Master's thesis. Norwegian University of Science and Technology.

[28] Ploom, I. (2014) Analysis of variations on orbital parameters of Cubesats: M.Sc thesis. University of Tartu Faculty of Science and Technology, Tartu.

[29] Slavinskis, A. (2015). EstCube-1 attitude determination. Estonia: University of Tartu Press.

[30] Arshad, H. (2013) Orbit Propagator and Geomagnetic Field Estimator for NanoSatellite: The ICUBE Mission[WWW]

https://www.researchgate.net/publication/319685850_Orbit_Propagator_and_Geomagnetic_Field_Estimator_for_NanoSatellite_The_ICUBE_Mission

APPENDICES

APPENDIX 1 IGRF COMPARISON TABLES

Table 1 Difference calculations in Longitude 0

Longitude 0	Latitude	X	Y	Z
MATLAB	-90	9536,66	-6328,85	-39767,00
NOAA	-90	9625,50	-6369,80	-40012,20
Difference	-90	88,84	-40,95	-245,20
Percentage	-90	0,92%	0,64%	0,61%
MATLAB	-60	11161,7	-4416,15	-22233,9
NOAA	-60	11240,5	-4419,6	-22198,6
Difference	-60	78,80	-3,45	35,30
Percentage	-60	0,00701	0,000781	-0,16%
MATLAB	-30	8967,26	-3239,54	-18426,4
NOAA	-30	9036,1	-3236,1	-18393,5
Difference	-30	68,84	3,44	32,90
Percentage	-30	0,76%	-0,11%	-0,18%
MATLAB	0	20728,6	-2153,29	-9965,58
NOAA	0	20663,7	-2148,5	-9914,2
Difference	0	-64,90	4,79	51,38
Percentage	0	-0,31%	-0,22%	-0,52%
MATLAB	30	22911,2	-522,961	19764,3
NOAA	30	22994,3	-527,3	19550,9
Difference	30	83,10	-4,34	-213,40
Percentage	30	0,36%	0,82%	-1,09%
MATLAB	60	11905,1	-585,111	37568,8
NOAA	60	12101,4	-583,2	37623,5
Difference	60	196,30	1,91	54,70
Percentage	60	1,62%	-0,33%	0,15%
MATLAB	90	1056,21		44087,2
NOAA	90	1070,7		44340,4
Difference	90	14,49		253,20
Percentage	90	1,35%		0,57%

Table 2 Difference calculations in Longitude 90

Longitude 90	Latitude	X	Y	Z
MATLAB	-90	-6328,85	-9536,66	-39767
NOAA	-90	-6369,60	-9626,2	-40013
Difference	-90	-40,75	-89,54	-246,00
Percentage	-90	0,64%	0,93%	0,61%
MATLAB	-60	3252,62	-8955,38	-43021,7
NOAA	-60	3428,5	-8992,6	-43179
Difference	-60	175,88	-37,22	-157,30
Percentage	-60	5,13%	0,41%	0,36%
MATLAB	-30	16310,2	-4512,33	-36476,2
NOAA	-30	16475,1	-4484,5	-36330,5
Difference	-30	164,90	27,83	145,70
Percentage	-30	1,00%	-0,62%	-0,40%
MATLAB	0	29563,7	-1218,02	-9639,12
NOAA	0	29464,5	-1215,5	-9603,9
Difference	0	-99,20	2,52	35,22
Percentage	0	-0,34%	-0,21%	-0,37%
MATLAB	30	25442,1	-72,211	26563,2
NOAA	30	25557,8	-77,8	26323,8
Difference	30	115,70	-5,59	-239,40
Percentage	30	0,45%	7,18%	-0,91%
MATLAB	60	10080,7	833,41	44286,5
NOAA	60	10302,6	838,7	45605,6
Difference	60	221,90	5,29	1319,10
Percentage	60	2,15%	0,63%	2,89%
MATLAB	90	462,81		44087,2
NOAA	90	459,4		44340,6
Difference	90	-3,41		253,40
Percentage	90	-0,74%		0,57%

Table 3 Difference calculations in Longitude 180

Longitude 180	Latitude	X	Y	Z
MATLAB	-90	-9536,66	6328,85	-39767
NOAA	-90	-9625,8	6370	-40013,4
Difference	-90	-89,14	41,15	-246,40
Percentage	-90	0,93%	0,65%	0,62%
MATLAB	-60	6551,87	7044,48	-45547,1
NOAA	-60	6784,9	7075,1	-45682,7
Difference	-60	233,03	30,62	-135,60
Percentage	-60	3,43%	0,43%	0,30%
MATLAB	-30	20219,6	5932,43	-30382,2
NOAA	-30	20340,1	5919,4	-30191,7
Difference	-30	120,50	-13,03	190,50
Percentage	-30	0,59%	-0,22%	-0,63%
MATLAB	0	25391,5	4234,33	-2931,91
NOAA	0	25312,5	4220,7	-2929
Difference	0	-79,00	-13,63	2,91
Percentage	0	-0,31%	-0,32%	-0,10%
MATLAB	30	20923,1	2423,57	20158,1
NOAA	30	20986,5	2431,8	19989,8
Difference	30	63,40	8,23	-168,30
Percentage	30	0,30%	0,34%	-0,84%
MATLAB	60	13195,1	980,904	38979,8
NOAA	60	13419,8	986,5	39016,4
Difference	60	224,70	5,60	36,60
Percentage	60	1,67%	0,57%	0,09%
MATLAB	90	-1056,21		44087,2
NOAA	90	-1069,8		44340,6
Difference	90	-13,59		253,40
Percentage	90	1,27%		0,57%

Table 4 Difference calculations in Longitude -90

Longitude -90	Latitude	X	Y	Z
MATLAB	-90	6328,85	9536,66	-39767
NOAA	-90	6370,2	9626,1	-40012,5
Difference	-90	41,35	89,44	-245,50
Percentage	-90	0,65%	0,93%	0,61%
MATLAB	-60	14284,5	7013,85	-27197,4
NOAA	-60	14438,4	7038,9	-27171,8
Difference	-60	153,90	25,05	25,60
Percentage	-60	1,07%	0,36%	-0,09%
MATLAB	-30	17819,6	3668,64	-12009,3
NOAA	-30	17862,9	3648,6	-11873,7
Difference	-30	43,30	-20,04	135,60
Percentage	-30	0,24%	-0,55%	-1,14%
MATLAB	0	21481,8	1165,63	7893,59
NOAA	0	21416,4	1160,2	7870,2
Difference	0	-65,40	-5,43	-23,39
Percentage	0	-0,31%	-0,47%	-0,30%
MATLAB	30	18046,4	-177,232	30727,5
NOAA	30	18154,6	-173,2	30554,1
Difference	30	108,20	4,03	-173,40
Percentage	30	0,60%	-2,33%	-0,57%
MATLAB	60	7144,15	-741,654	43624,8
NOAA	60	7339,4	-744,4	43762,6
Difference	60	195,25	-2,75	137,80
Percentage	60	2,66%	0,37%	0,31%
MATLAB	90	-462,81		44087,2
NOAA	90	-459		44340,5
Difference	90	3,81		253,30
Percentage	90	-0,83%		0,57%

APPENDIX 2 CONTROLLER SIMULATION MATLAB CODE FOR

INPUTS

```
%Target attitude euler angles
target_attitude_euler = [10 9 44];
% yaw pitch roll

%Init attitude euler angles
initial_attitude_euler = [79 77 56];
% yaw pitch roll

%magnetorquer 3X3 controllmatrix
magn_coil_K = [ 0.12 0 0; 0 0.12 0; 0 0 0.12];

%Conversions
initial_attitude_euler = deg2rad(fliplr(initial_attitude_euler));

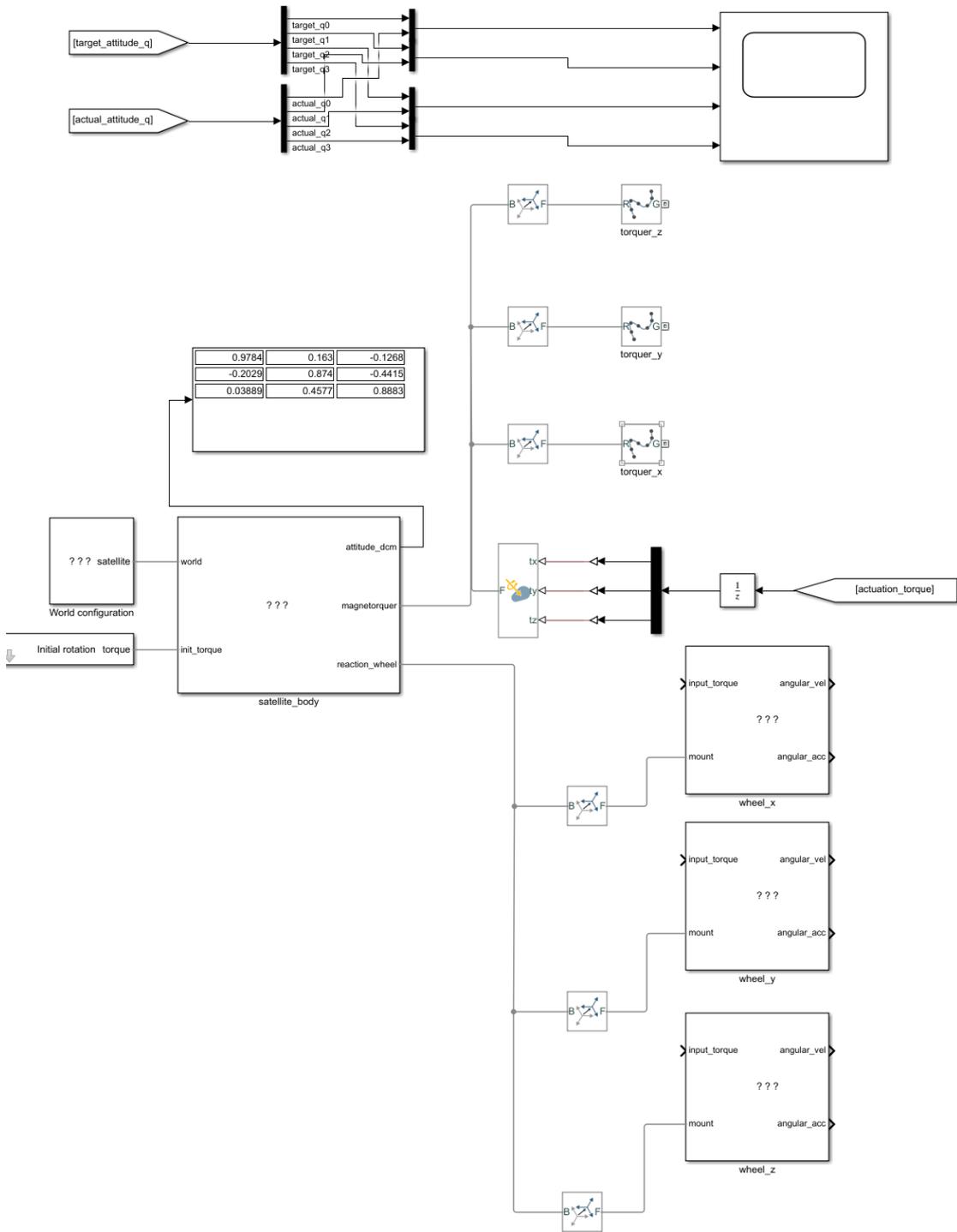
initial_attitude_quat = eul2quat(initial_attitude_euler);

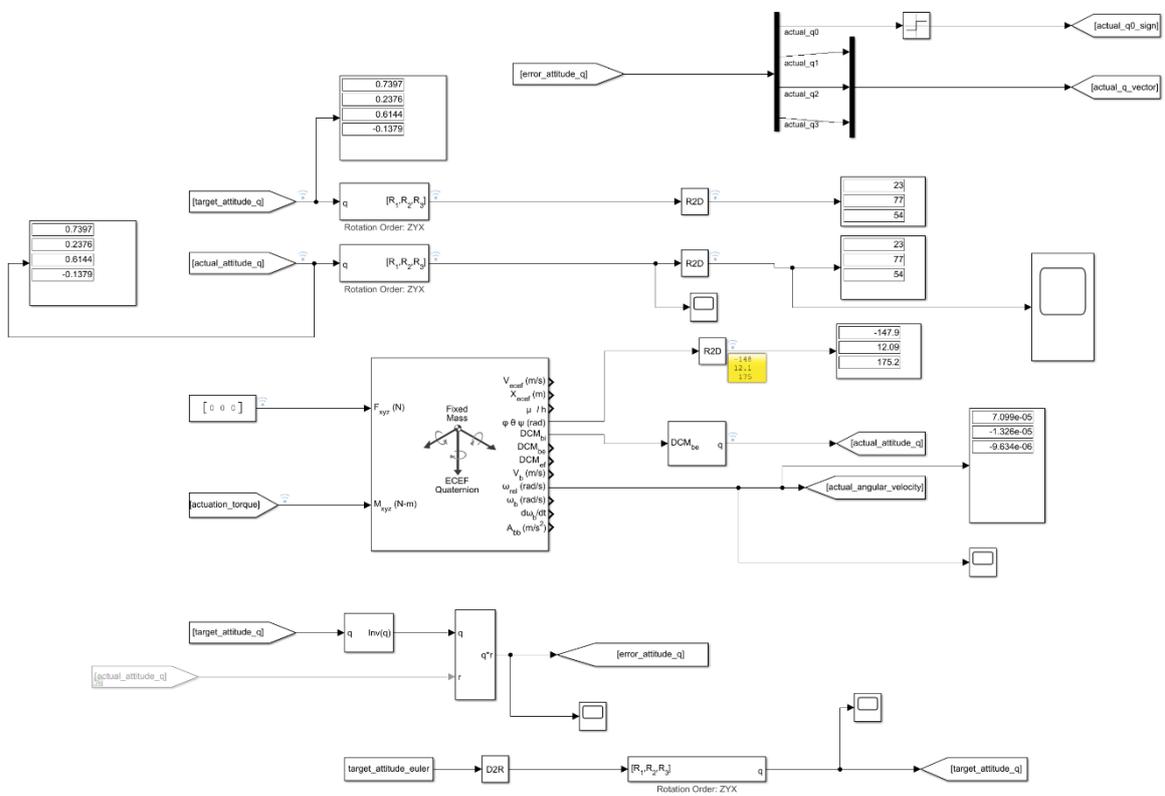
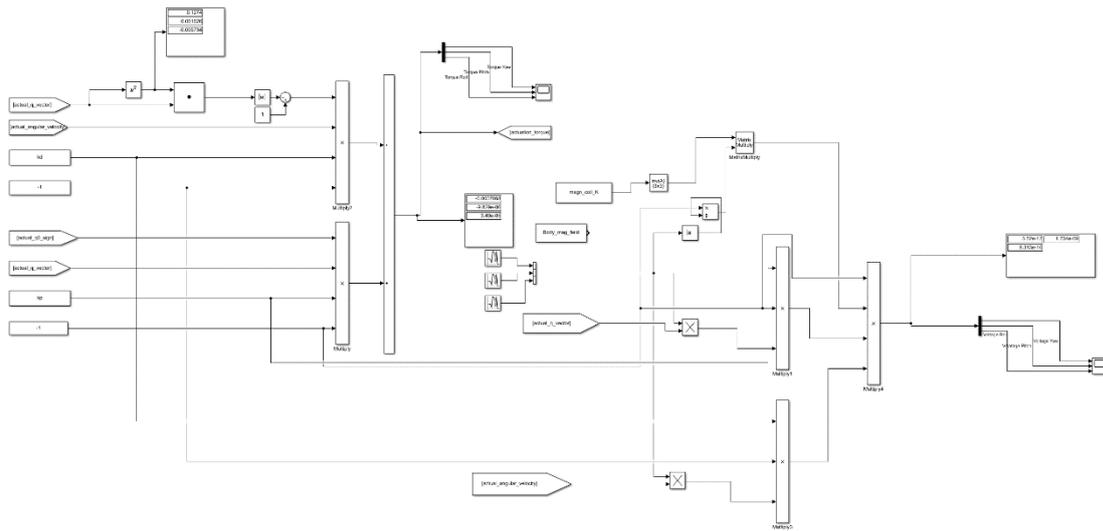
target_attitude_quat = eul2quat(target_attitude_euler);

%Testimise kvaternioon
%tesss = [0 1 0 0];
%sda = quat2eul(tesss);

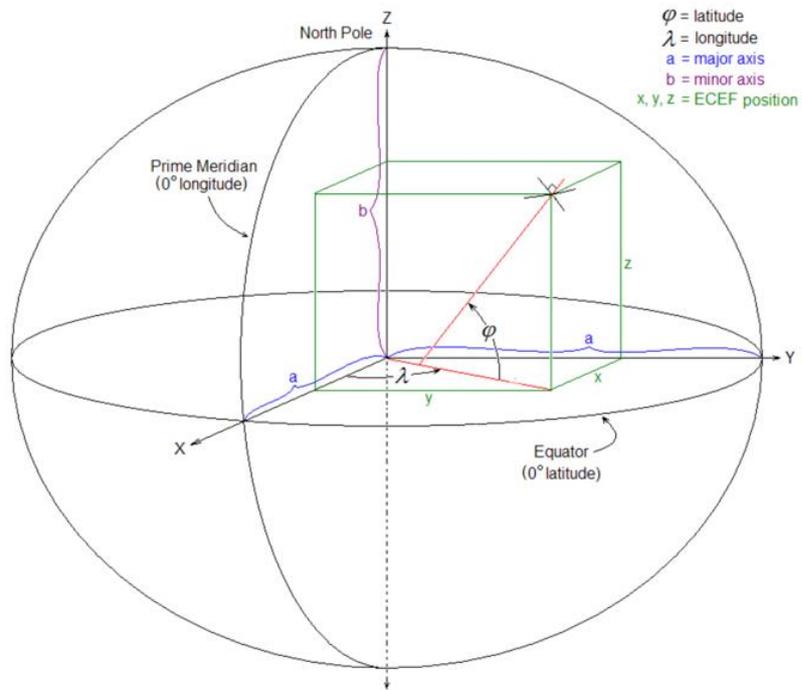
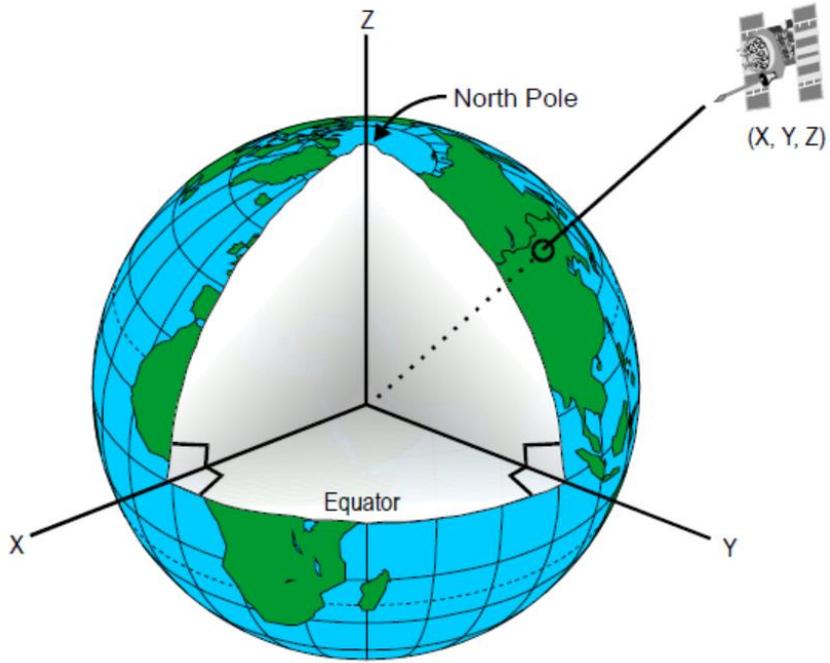
%PD parameters %Future PID
kp = 0.0006*[1 1 1]; %Applied to all 3 axes
kd = 0.005*[1 1 1];
%ki = 0 *[1 1 1];
```

APPENDIX 3 SIMULINK SYSTEM MODELS





APPENDIX 4 ECI AND ECEF ILLUSTRATIONS



APPENDIX 5 ORBIT SIMULATION CALCULATIONS IN MATLAB

```

RE = 6378;      % Earth's radius          [km]
muE = 398600.44; % Earth gravitational parameter [km^3/sec^2]
wE = (2*pi/86164); % Earth rotation velocity around z-axis [rad/sec]
time_mult = 1000;
framerate = 50;
fname = 'h2marik.tle';
[tle_a, tle_e, tle_inc, tle_RAAN, tle_w, tle_nu, tle_T] = tle(fname);
%Time
year = (tle_T(1:2));
epoch = (str2num(tle_T(3:14))-1)*24*3600*1000;
time_reference = datenum(year,'yy');
time_matlab = time_reference + epoch / 8.64e7;
time_matlab_string = datestr(time_matlab, 'yyyy-mm-dd HH:MM:SS.FFF');
tim = datevec(time_matlab_string);

% ORBIT COMPUTATION
a = tle_a;
e = tle_e;
p = a*(1-e^2);      % semi-latus rectus      [km]
rp = a*(1-e);      % radius of perigee      [km]
ra = a*(1+e);      % radius of apogee      [km]
h1 = sind(tle_inc)*sind(tle_RAAN); % x-component of unit vector h
h2 = -sind(tle_inc)*cosd(tle_RAAN); % y-component of unit vector h
h3 = cosd(tle_inc); % z-component of unit vector h
n1 = -h2/(sqrt(h1^2+h2^2)); % x-component of nodes' line
n2 = h1/(sqrt(h1^2+h2^2)); % y-component of nodes' line
n3 = 0; % z-component of nodes' line
N = [n1,n2,n3]; % nodes' line (unit vector)
theta = tle_nu + tle_w; % argument of latitude [rad]
r = (a.*(1-e.^2))./(1+e.*cosd(tle_nu)); % radius [km]

```

```

n = sqrt(muE./a^3);    % mean motion          [rad/s]
T = 2*pi/n;          % period                [s]
v0 = tle_nu*pi/180;   % True anomaly at the departure [rad]
w = tle_w*pi/180;    % Argument of perigee      [rad]
% Satellite coordinates
% vector of time     [s]
%
% True anomaly, Argument of latitude, Radius
sin_v = (sqrt(1-e.^2).*sin(E))./(1-e.*cos(E)); % sine of true anomaly
cos_v = (cos(E)-e)./(1-e.*cos(E));           % cosine of true anomaly
v = atan2(sin_v,cos_v);                       % true anomaly      [rad]
theta = v + w;                                % argument of latitude [rad]
r = (a.*(1-e.^2))./(1+e.*cos(v));            % radius              [km]
% Satellite coordinates
% "Inertial" reference system ECI (Earth Centered Inertial)
xp = r.*cos(theta);                          % In-plane x position (node direction) [km]
yp = r.*sin(theta);                          % In-plane y position (perpendicular node direct.) [km]
xs = xp.*cosd(tle_RAAN)-yp.*cosd(tle_inc).*sind(tle_RAAN); % ECI x-coordinate SAT [km]
ys = xp.*sind(tle_RAAN)+yp.*cosd(tle_inc).*cosd(tle_RAAN); % ECI y-coordinate SAT [km]
zs = yp.*sind(tle_inc);                      % ECI z-coordinate SAT [km]
rs = p./(1+e.*cos(theta-w));                 % norm of radius SAT [km]
% Constants
r_init = [6879 0 0]; %km
r_earth = 6378; %km
mew = 398600;
R = norm(r_init);
h = sqrt(mew*tle_a*(1-tle_e^2))./sqrt(mew*R); %km2/s ----- Angular momentum
ecc = tle_e;%0;                               -----Eccentricity
inc = tle_inc;%47; %degrees                   -----Inclination

```

```
RAAN = tle_RAAN;%0; ----- Right Ascension
AP = tle_w;%90; -----perigee
TA = tle_nu;%atan2d((sind(tle_E)*(1-ecc^2)^.5),(cosd(tle_E)-ecc));%0; ----True Anomaly
```