Advancing the Attitude Determination and Control System for the CubeSat MOVE-II

Scientific Thesis for the procurance of the degree M.Sc.
from the Department of Electrical and Computer Engineering at the Technical University of Munich.

Supervised by PD Dr.-Ing. habil. Dirk Wollherr
Dr.-Ing. Marion Leibold
Chair of Automatic Control Engineering (EI)
M.Sc. Jonis Kiesbye
Chair of Astronautics (MW)

Submitted by B.Sc. Maximilian Preisinger

Submitted on Munich, 13.03.2019

TECHNISCHE UNIVERSITÄT MÜNCHEN
LEHRSTUHL FÜR STEUERUNGS- UND REGELUNGSTECHNIK
PRIV.-DOZ. DR.-ING. HABIL. DIRK WOLLHERR

May 23, 2018

M A S T E R ' S T H E S I S
for
Maximilian Preisinger
Degree EI

Advancing the Attitude Determination and Control System for the CubeSat MOVE-II

Problem description:

The CubeSat MOVE-II, built by students of TUM, uses a multi-dimensional statefeedback controller [2] for attitude control. This controller uses only the output of the sun sensor, the gyroscope, and the magnetometer. The attitude control system relies only on magnetic coils for actuation [3]. Therefore, we cannot apply a torque vector parallel to Earths magnetic feld vector, resulting in reduced controllability. Still, the current controller shows satisfying results in Hardware-in-the-Loop simulations.
Recent developments of the MOVE-II team resulted in an Extended Kalman Filter (EKF) for attitude estimation and an approach to compensate the disturbance torque introduced by the parasitic dipole of the satellite. These approaches need to be characterized and tested in an integrated setup before they can be implemented on the satellite flight model.
Alternative controllers might be far better suited for attitude control with nonlinear characteristics and reduced controllability. A variety of control approaches shall be evaluated for implementation on MOVE-II.
Attitude control of satellites relying only on magnetic coils for actuation is widely covered in diferent Ph.D. theses [5, 4] but the in-flight application of the discussed controllers is not covered in many cases. Some satellites do not achieve stable pointing at all [1] and augment the magnetic coils with reaction wheels in subsequent revisions.
The high reliability requirements of a satellite mission necessitate thorough testing in Software-in-the-Loop, and Hardware-in-the-Loop requirements. If these tests are successful, the controller implemented in this thesis may be uploaded on the satellite and evaluated in-flight.

Tasks:

- Evaluate the old controller in critical scenarios
- Extend the current controller with the EKF and parasitic dipole compensation
- Evaluate alternative control strategies in a simulation environment
- Select and implement a suitable controller for use on the satellite
- Verify the implemented controller in SiL and HiL environments
- Comparison of old and new controller

Bibliography:

[1] S. Busch, P. Bangert, S. Dombrovski, and K. Schilling. UWE-3, in-orbit performance and lessons learned of a modular and flexible satellite bus for future picosatellite formations. In *65th IAC*, 2014.
[2] D. Meßman. Development of a sun pointing attitude controller using magnetic actuation for the MOVE-II cubesat mission. Technical report, Chair of Automatic Control, Technische Universität München, Garching, 2017.
[3] D. Meßmann, F. Coelho, P. Niermeyer, M. Langer, and H. Huang. Magnetic attitude control for the MOVE-II mission. In *7th EUCASS*, 2017.
[4] A. Slavinskis. *ESTCube-1 Attitude Determination*. PhD thesis, University of Tartu, Tartu, Estonia, 2015.
[5] R. Wisniewski. *Satellite Attitude Control Using Only Electromagnetic Actuation*. PhD thesis, Department of Control Engineering, Aalborg Universitet, Aalborg, 1997.

Supervisor:	M. Leibold (EI), Jonis Kiesbye (MW)
Start:	28.05.2018
Intermediate Report:	15.10.2018
Delivery:	28.02.2019

(D. Wollherr)
Priv.-Doz.

Abstract

This thesis evaluates an existing state feedback controller designed for the CubeSat MOVE-II. In addition, the control system is further developed and a variety of control strategies are evaluated. The goal is to achieve a pointing accuracy of 5° and a guaranteed convergence to the desired attitude within the domain of the nonlinear system. The evaluated concepts are restricted to low earth orbits for small satellites utilizing magnetic actuation only. Furthermore, a spin stabilization around a non-principal axis of inertia is assumed. The presence of sensor noise and external disturbances, especially the residual magnetic dipole, are considered.
The existing controller is analyzed for different orbits and the convergence rate for randomized initial conditions is obtained in a Monte Carlo simulation. A new gain is developed for the state feedback controller under consideration of the system requirements for MOVE-II. Furthermore, the concepts of residual magnetic dipole moment compensation, the use of an Extended Kalman Filter and a new operation point switching strategy are proposed in order to increase the pointing accuracy and convergence rate. Three nonlinear control laws based on different Lyapunov functions are evaluated. One of them is selected as an implementation candidate. A Software-in-the-Loop and a Hardware-in-the-Loop simulation successfully verifies this candidate for use on the flight hardware as part of the control firmware.
It is found that the existing controller is sensitive to inclination changes of the orbit. Furthermore, with an average pointing accuracy of 5.8° and a corresponding variance of $4.3\,\mathrm{deg}^2$ it does not fulfill the stated goal. In total, 69% of the evaluated cases converge to the desired attitude. The linear controller with the new gain and extensions attains a pointing accuracy of 3° with a variance of $0.38\,\mathrm{deg}^2$. It converges in 80% of the cases. The nonlinear implementation candidate shows convergence in all Monte Carlo simulation runs and achieves a final pointing accuracy of 3.3° with a variance of $2.3\,\mathrm{deg}^2$. Furthermore, a Hardware-in-the-Loop simulation shows that this controller is stable in the presence of large delays and with low numeric precision. It also shows to be compatible with the existing firmware of MOVE-II. Dynamically updateable control parameters render this implementation interesting to satellite operators.

This thesis made it possible to continue my education in an interesting field of study within an exciting space project. But this thesis would not be possible without the support of several people. First and foremost, I thank Dirk Wollherr for accepting my thesis proposal. I extend my thanks to my supervisors Marion Leibold and Jonis Kiesbye for their support during my work, for their organizational efforts and for their valuable input. I express my gratitude to Jonis for his various comments on my work, leading me to new directions, but also for supporting me organizing my work and time schedule, as well as his advice concerning the HiL simulation environment. I appreciate the time he took within our regular meetings.
Further thanks belong to all the current and previous ADCS team members, without whose prior work this thesis would not be possible. Special thanks goes to David Meßmann for introducing me to the extensive Matlab simulation software and for the interesting discussions about Kalman filters and control concepts.
I also thank Tejas Kale for supporting me with the tricky parts of the ADCS firmware and for providing creative solutions to otherwise difficult software or hardware problems.
Last but not least I want to thank the whole MOVE team for the welcoming atmosphere and the good time working together on MOVE-II and the MOVE-ON projects.
Finally I thank my family for their abundant support during writing of my thesis.

Contents

Chapter 1

Introduction

1.1 Motivation

The Munich Orbital Verification Experiment (MOVE) program was initiated in 2006 with the goal to launch the satellite First-MOVE. It is built by TUM students who are part of the student society WARR (Wissenschaftliche Arbeitsgemeinschaft für Raketentechnik und Raumfahrt; English: Scientific Workgroup for Rocketry and Spaceflight). First-MOVE is a one unit CubeSat with the capability to deliver pictures from Earth and features a scientific payload, which evaluates experimental solar cells [LOH+15]. CubeSats are nanosatellite with well defined design specifications [Pro14]. They are partitioned into units, where the maximum mass per unit may not exceed 1.33 kg and the size of one unit is restricted to a volume of 10 cm × 10 cm × 11.35 cm. Up to 6 units can be combined in order to build a larger CubeSat. This standardized specification allows for a simple integration process with the satellite deployer and leads to reduced launch costs [SMA+11]. Therefore they are popular for educational and experimental missions. First-MOVE was launched in 2013 and operated for the duration of a month. Although the mission ended preliminary due to a failure of the on board computer, the satellite's main purpose of student education was achieved [LOH+15].
Due to this success, a second mission started in 2015 with the satellite MOVE-II [LAD+15]. Like First-MOVE, it is a one unit CubeSat launched into a low earth orbit. Its mass of 1.2 kg is close to the CubeSat specification limit. The system design of MOVE-II includes several enhancements compared to the first mission. An on board computer for computationally intensive tasks was designed. The MOVE team also developed a new communication protocol for MOVE-II, which ensures a reliable communication link when the signal quality is weak [ARL16]. The satellite features Flappanels, which form an increased solar cell area after deployment. Figure 1.1 illustrates MOVE-II with deployed Flappanels. The Toppanel, which is located in the middle between the four Flappanels, features several experimental solar cells. This scientific payload of MOVE-II has the the goal to evaluate the novel multi-junction solar cell design with on-orbit verification [RKS+16]. Point-

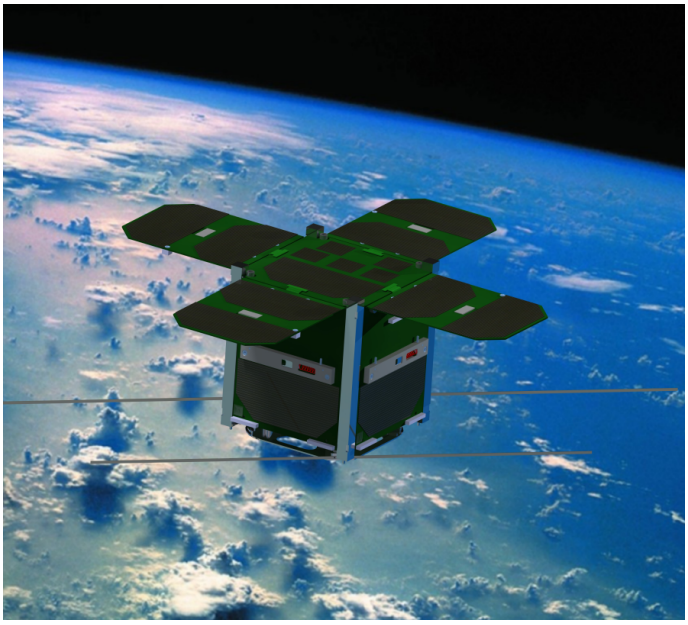

Figure 1.1: Computer generated graphic of MOVE-II with deployed Flappanels [ARL16].

ing the Flappanels and the Toppanel towards the sun is critical for the mission. One reason is that the scientific payload needs to be directed towards sun for meaningful measurements. The other reason is that evaluations of the power system show that the power budget is only slightly positive, if the Flappanels are not facing sun directly [Nag18], and becomes negative in low temperatures.

In order to achieve this, MOVE-II needs an active Attitude Determination and Control System (ADCS). The requirement ADCS-01.1 defined in the Critical Design Review (CDR) of MOVE-II [MBC+16] states that the ADCS must provide a pointing accuracy of at least 10°, and should have a pointing accuracy of 5°. These challenges are new for the MOVE team, as First-MOVE does not have an active ADCS.

The ADCS team decided to use sun sensors, magnetometers and gyroscopes for attitude determination, and to use magnetic actuation for attitude control [MCN+17]. This control concept utilizes electric coils in order to generate a magnetic dipole, which interacts with the Earth magnetic field. This concept has an inherent limitation in controllability, but it is cheap and lightweight and therefore attractive for small satellites. Two control modes are available for MOVE-II. One detumbling mode and one sun pointing mode. For detumbling, a B-dot controller is implemented by the ADCS team, which reduces the initial angular velocity of the satellite to a small magnitude. It reduces the kinetic energy by providing negative feedback from the derivative of the Earth magnetic field. For the sun pointing mode, a linearized state feedback controller for stabilization of the operation point is implemented. Recent evaluation of the ADCS team shows that MOVE-II benefits from spin sta-

bilization, although the pointing axis of MOVE-II does not align with the principal axis of inertia. In the scope of this thesis, only sun pointing controllers are evaluated.

1.2 Goal of the Thesis

This thesis aims for further development of the ADCS. The existing state feedback controller shall be evaluated in critical and non-nominal scenarios and in the presence of disturbances. Particularly the convergence behavior starting from an arbitrary attitude towards the desired attitude is of interest, since the initial attitude after launch cannot be influenced.
The existing controller should be extended with an Extended Kalman Filter (EKF) and a residual magnetic dipole moment compensation, both methods recently developed by the ADCS team. The performance of the controller with these extensions has to be evaluated and compared.
Alternative nonlinear control approaches should be investigated. An evaluation has to be performed in order to show, if the examined approaches provide a benefit in performance or convergence behavior for the special challenges imposed by the design of MOVE-II. One challenge consists of the inherent controllability limitation of the magnetic actuation. The other challenge is the fact that the pointing axis is not a principal axis of inertia, and furthermore is close to the intermediate principle axis, which is known to be unstable for rotations around it. This makes spin stabilization difficult. There is a trade-off between the positive effects of an increased spin rate and the nutation torque, which grows with increasing spin rate around a non-principal axis of inertia.
Eventually, one controller shall be selected and evaluated in a Software-in-the-Loop (SiL) and Hardware-in-the-Loop (HiL) environment. A successful pass of these simulations is a requirement for every piece of software that is uploaded onto the flight hardware.

1.3 State of the Art

The two most popular actuation systems among CubeSats are reaction wheels and magnetic actuation [XSZ+17]. Reaction wheels transfer angular momentum from the satellite to flywheels. Usually there is a reaction wheel mounted on every axis of the satellite, which allows full three axes control. Magnetic actuation, however, can generate torques in a plane perpendicular to the Earth magnetic field only, and therefore has an inherent controllability limitation. Using magnetic actuation instead of reaction wheels has several advantages, however. The electric coils to generate the magnetic dipole are lightweight, cheap to implement and they utilize no moving parts, which are more error prone than electric coils.
Bouwmeester and Guo [BG10] analyze CubeSat missions launched until 2009, which amounts to about hundred satellites in total. The data for 94 of those CubeSats is

publicly available and included in their statistics. They find that in this early stage of CubeSat history, most satellites use a passive attitude stabilization system, or no attitude control at all. Passive attitude stabilization is obtained for example with a permanent magnet, which aligns the satellite along the magnetic field of Earth. They further state that the major number of attitude control systems implement the B-dot controller in order to detumble the satellite. Only 15% of the analyzed satellites feature an ADCS which is capable of pointing the satellite towards a desired direction. However, the pointing accuracy of the reviewed CubeSats is low.
Xia et al. [XSZ+17] provide an extended analysis of CubeSat attitude control systems for satellites launched between 2003 and 2016. Their analysis includes 357 CubeSats. They find that more than half of all analyzed satellites feature reaction wheels for high accuracy pointing, mostly utilizing proportional-derivative control laws. Only 11% utilize a solely magnetic based actuation system, which mostly aims for a pointing accuracy of around 15°. A recent mission relying solely on magnetic actuation is the AntelSat [TMP13]. Its ADCS utilizes a linear state feedback control law for three axes stabilization and aims for a pointing error of less than 10°.
Student groups which start follow-up missions often include reaction wheels in their next-generation satellites. For example the AAU-cubesat, being among the first five launched CubeSats, features a solely magnetic based actuation system utilizing a constant gain control law [ABV+03]. In their next-generation CubeSat AAUSAT-II, however, they include experimental reaction wheels [NLG+09].
One student-built CubeSat needs to be highlighted within this context. The satellite ESTCube-1 is solely based on magnetic actuation and aims for a pointing error of less than 3° [SKK+14]. They use a control law based on a Lyapunov function, which is originally designed for the JC2Sat-FF mission. This accuracy, however, is achieved only with spin stabilization using a high spin rate of about 360 °/s. Such high spin rates are not feasible for satellites, like MOVE-II, where the pointing axis is not equal to a principal axis of inertia. Furthermore, in their next mission, ESTCube-2, they also include reaction wheels and aim for a pointing error of below 0.1° [ESI+16].
Solutions to the problem of high accuracy pointing with magnetic actuation only, which do not rely on high spin rates for spin stabilization or adress satellites where the pointing axis is not equal to a principal axis of inertia, are not found in literature. The topic of attitude control systems for CubeSats is divided into two distinct fields: Magnetic actuation if low pointing accuracy is sufficient, and actuation using reaction wheels, when high accuracy is required. If, however, a high pointing accuracy can be achieved with magnetic actuation only, future missions relying on high pointing accuracy can also benefit from the advantages of magnetic actuation.

1.4 Methodology and Structure

This thesis is structured as follows: Chapter 2 covers the theoretical background of orbital mechanics and introduces the system equations for the attitude of a spacecraft. Furthermore, the limitations of the MOVE-II actuation system are stated and the controllability limitation caused by magnetic actuation is covered. Also, the major disturbance effects for MOVE-II are presented.
Chapter 3 deals with the analysis of the existing controller. The theoretical background of state feedback controllers and the corresponding stability analysis within time varying systems are introduced. The performance of this controller is analyzed with regards to orbit changes and sensitivity to initial conditions. The latter is evaluated with a Monte Carlo simulation.
In Chapter 4, an extended linear controller is developed. The EKF and the residual magnetic dipole moment compensation are included among further extensions. The extended controller is evaluated in similar conditions as in Chapter 3 and the results are compared to the existing controller. An enhancement to the residual magnetic dipole moment compensation is provided and the performance of this extension is analyzed in terms of pointing accuracy and power consumption.
Chapter 5 reviews popular nonlinear control concepts used in space, which results in the decision to focus on Lyapunov based control laws only. Three different control laws are described and analyzed. The sensitivity to initial conditions and the long-term behavior of the nonlinear controllers is evaluated and compared to the results of the linear controllers in Chapters 3 and 4. Furthermore, one appropriate nonlinear controller is selected and evaluated within a SiL and HiL simulation.
Chapter 6 discusses the results and provides a comparison of the different controller characteristics. Chapter 7 summarizes the obtained results, and Chapter 8 states future work to be done.

Chapter 2

Space Environment

2.1 Coordinate Systems

Throughout this thesis we use different coordinate systems to express physical values. The following sections define and explain them.

Earth Centered Inertial Frame

The Earth Centered Inertial Frame (ECI) is an inertial system, thus its orientation is fixed in space. The center of Earth is the origin of the corresponding coordinate system. The Z-axis aligns with the Earth's rotational axis. The X-axis is defined by the line resulting from the intersection of equatorial and ecliptic plane, directed from sun towards the vernal equinox. The vernal equinox describes the point in space where the earth is located when in spring of the northern hemisphere day and night are the same length. The Y-axis completes the orthogonal basis of the right-hand-side coordinate system. See Figure 2.1 for illustration.

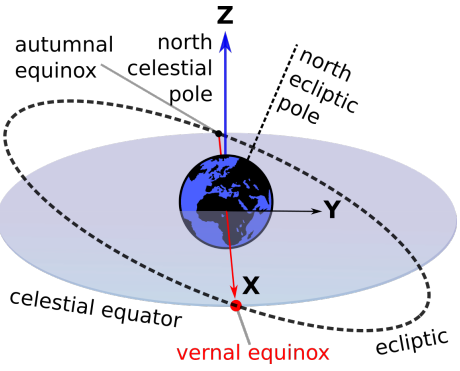

Figure 2.1: Illustration of the ECI and the vernal equinox. [Earth by David Courey from thenounproject.com]

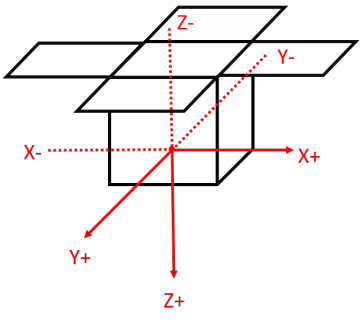

Figure 2.2: Body Frame (BF) of MOVE-II. [Internal graphic of the ADCS team]

Body Frame

The Body Frame (BF) is used to express the variables as the satellite observes them, therefore it is fixed to the structure of MOVE-II. In order to define coordinate axes it is important to know about this structure. As every CubeSat, MOVE-II has a cubical form with a panel at each side of the cube. The panel located on the top is called Toppanel and the four panels on the side are called Sidepanels. Attached to the Toppanel are four additional solar panels. During launch they lie flat against the Sidepanels and during deployment they are folded out to form a plane together with the Toppanel. Due to the deployment mechanism they are called Flappanels. Since these solar cells are the main energy source of the satellite, it is important to have them aligned with the sun. Therefore, the axis of most relevance is the Z-axis, which points away from the Toppanel. The X- and Y-axis point out of an arbitrary but well defined Sidepanel to complete a right-hand coordinate system. The geometric center defines the origin of this coordinate system. For illustration see Figure 2.2.

2.2 Orbit Characteristics

This section describes the classical orbital elements and how they define the movement of a satellite around its celestial body. Furthermore the target orbit of MOVE-II will be defined, which serves as foundation for all further analyses.
The orbit of a satellite around another celestial body is completely defined by six elements. There are several different ways of expressing them, in this thesis the classical orbit elements will be used. They can be divided into two groups. The dimensional elements define the size of the orbit and the position of the satellite on the orbit. The orientation elements define the orientation of the orbit with respect to the ECI. Table 2.1 introduces these elements.

Dimensional Elements	Orientation Elements
Semimajor axis a	Inclination i
Eccentricity e	Right Ascension of the Ascending Node (RAAN) Ω
Mean anomaly M	Argument of perigee ω

Table 2.1: Classical orbit elements.

These elements are always connected with an epoch E_0, which specifies to which point in time these elements refer. An orbit is always subject to changes caused by various disturbances [Cho02]. The meaning of each of these elements will now be discussed briefly, see also Figure 2.3 for illustration.
The **semimajor axis** a defines the size of the orbit. Closed Keplerian orbits around a celestial body always form an elliptical shape, and the largest possible diameter divided by two is called the semimajor axis.

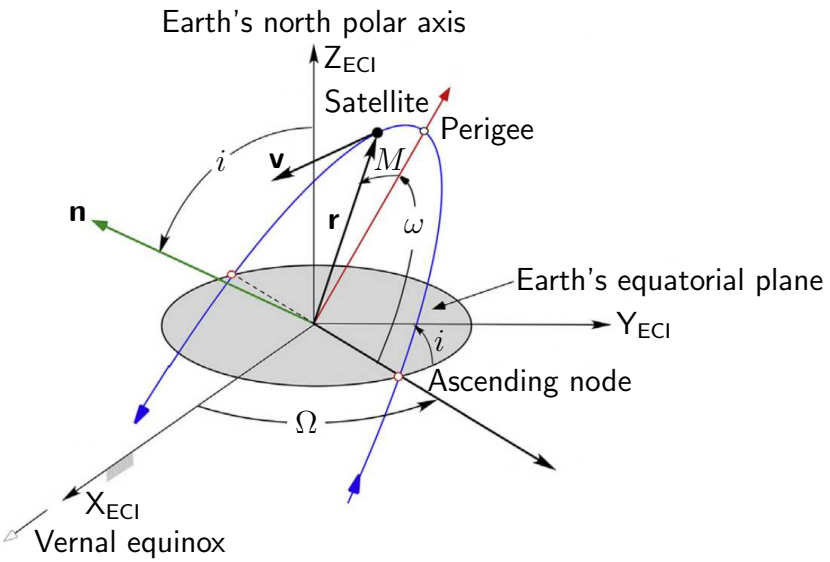

Figure 2.3: Illustrating the classical orbit elements [Cur14, p. 196]. The gray disc illustrates the equatorial plane, the blue trajectory is the orbit, **n** defines the normal of the orbital plane, **r** is the position vector of the satellite and **v** is the instantaneous velocity vector.

The **eccentricity** e defines, how much the orbit plane deviates from a circular shape. It is defined by

$$e = \frac{r_\mathrm{a} - r_\mathrm{p}}{r_\mathrm{a} + r_\mathrm{p}}, \tag{2.1}$$

where r_a is the radius of apogee, the radius at the point on the orbit farthest away from the celestial body, and r_p is the radius of perigee, the radius at the point on the orbit closest to the celestial body. It follows from this formula, that $e = 0$ defines a perfectly circular orbit, and that $0 \leq e < 1$ for any closed Keplerian orbit.
The **mean anomaly** M describes the position of the satellite on its orbit. Let t be the time elapsed since the satellite passed perigee. Then the mean anomaly is defined as the angle between perigee and an imagined point in space where the satellite would be after time t, if it was orbiting the celestial body with the same orbit period on a circular orbit, starting from the perigee. The mean anomaly increases linearly over time. Given an initial mean anomaly M_0 valid at initial time t_0, the mean anomaly M for any other time t can be calculated by

$$M = M_0 + \sqrt{\frac{\mu_\mathrm{B}}{a^3}} \cdot (t - t_0) \mod 2\pi, \tag{2.2}$$

where μ_B is the gravitational constant of the celestial body.
The **inclination** i describes by what angle the orbit plane is tilted with respect to the equatorial plane. The inclination is defined in the range from $-90°$ to $+90°$. An

inclination of 0° describes an orbit around the equator while $|i| = 90°$ describes an orbit passing the poles of the celestial body.
The **Right Ascension of the Ascending Node (RAAN)** Ω is defined as the angle between the X-axis of the ECI and the ascending node, which is the point where the satellite is passing the equator from south to north. For $i = 0°$, the RAAN is not defined and thus set to 0°.
The **argument of perigee** ω describes the location of apogee and perigee. It is defined as the angle between the ascending node and the perigee. For circular orbits where $e = 0$, the argument of perigee is undefined and thus set to 0°.
Another parameter of interest often needed is the orbit period

$$T = 2\pi \cdot \sqrt{\frac{a^3}{\mu_\mathrm{B}}} \,. \tag{2.3}$$

It is not part of the classic orbit elements since it can be determined from the semimajor axis and the gravitational constant of the celestial body. This value describes, how long the satellite needs for one revolution around its orbit. The orbit period defines the orbit rate

$$\omega_\mathrm{o} = 2\pi/T, \tag{2.4}$$

which describes the average angular velocity of the position of the satellite around its orbit.
The Earth is not perfectly round, its equatorial radius is a bit larger than its polar radius. This oblateness disturbs satellites on their orbits around Earth and causes their orbit planes to rotate. When the orbit plane rotates with the same angular velocity in an inertial coordinate system as the Earth rotates around the sun, it is called a sun-synchronous orbit. As a result, the ascending node lies at a fixed local time, and the orbit plane maintains a constant angle relative to the sun [Cur14]. MOVE-II is launched into such an orbit.
The parameters for the target orbit of MOVE-II as provided by the launch provider are summarized in Table 2.2. The epoch E_0 is chosen such that the satellite leaves eclipse in the first 20 s of the simulation. This orbit is assumed in all simulations and analysis of this thesis, if not stated otherwise.

Parameter	Default Value
Semimajor axis	$a = 6953.1\,\mathrm{km}$
Eccentricity	$e = 0.00001$
Mean anomaly	$M_0 = 0°$
Inclination	$i = 97.78°$
RAAN	$\Omega = 97.78°$
Argument of perigee	$\omega = 0°$
Epoch	$E_0 = 21.01.2017\ 00{:}21{:}25\ \mathrm{UTC}$
Orbit period	$T = 5770\,\mathrm{s}$

Table 2.2: Target orbit of MOVE-II.

2.3 System Equations

The behavior of a free-floating spacecraft is determined by its equations of motion and its dynamic equations. The equations of motion depend strongly on the representation of attitude. In application, various different attitude representations can be found. Popular representations are quaternions, Euler angles, Euler angle-axis representation, the direction cosine matrix and modified Rodriguez parameters [OB12].
When using quaternions as attitude representation, the equations of motion are written as [Yan12]

$$\dot{q}_0 = -\frac{1}{2}\boldsymbol{\varrho}^{\mathrm{T}}\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} \tag{2.5}$$

$$\dot{\varrho} = \frac{1}{2}(q_0 \cdot \mathbf{1}_{3\times 3} + [\boldsymbol{\varrho}\times])\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}, \tag{2.6}$$

where $\mathbf{1}_{3\times 3}$ represents the three-dimensional identity matrix, $\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}$ denotes the angular velocity of the BF in relation to the ECI expressed in the BF, the notation $[\mathbf{v}\times]$ for an arbitrary three-dimensional vector $\mathbf{v} = (v_1, v_2, v_3)^{\mathrm{T}}$ represents the cross-product operator

$$[\mathbf{v}\times] \equiv \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{2.7}$$

and

$$\mathbf{q}_{\mathrm{ib}} = (q_0, \boldsymbol{\varrho}^{\mathrm{T}})^{\mathrm{T}} \tag{2.8}$$

identifies the unit quaternion representing a rotation of BF with respect to ECI, where q_0 is the scalar part of the quaternion, and

$$\boldsymbol{\varrho} = (q_1, q_2, q_3)^{\mathrm{T}} \tag{2.9}$$

is the vector part with its three components, such that

$$\mathbf{q}_{\mathrm{ib}} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}. \tag{2.10}$$

In general the superscript b indicates that a vector is represented in the BF, while a superscript i indicates vectors that are represented in the ECI.
MOVE-II uses a direction vector for attitude representation. The sun plays an important role for MOVE-II, since the Flappanels have to be aligned to the sun for optimal power generation. Therefore, the attitude is represented as a unit vector $\mathbf{s}^{\mathrm{b}}$ in BF pointing towards the sun. This vector is called sun vector. The equations of motion can be written as

$$\begin{aligned} \dot{\mathbf{s}}^{\mathrm{b}} &= -\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} \times \mathbf{s}^{\mathrm{b}} \\ &= \mathbf{m}_{\mathrm{s}}(\mathbf{s}^{\mathrm{b}}, \boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}), \end{aligned} \tag{2.11}$$

where $\dot{\mathbf{s}}^{\mathrm{b}}$ denotes the differential change of the sun vector. The sun vector does not represent the full attitude, since a rotation of the satellite around the sun vector will be unobservable. For the attitude controller of MOVE-II the representation with the sun vector is advantageous over the quaternion representation, since we are not interested in the rotational angle around the sun vector, but rather in the pointing angle between the sun vector and the Z-axis of the BF [MCN+17].
The equation of dynamics are characterized through the change of angular momentum

$$\dot{\mathbf{h}} = -\mathbf{h} \times \boldsymbol{\omega}_{\mathrm{ib}} + \boldsymbol{\tau} \tag{2.12}$$

where $\mathbf{h}$ is the angular momentum of the satellite, $\boldsymbol{\omega}_{\mathrm{ib}}$ the angular velocity vector and $\boldsymbol{\tau}$ is the vector of torques acting on the satellite [Wer02]. The term $\mathbf{h} \times \boldsymbol{\omega}_{\mathrm{ib}}$ shows that, even if the applied torques are zero, $\mathbf{h}$ is not constant. This term can be interpreted as an additional torque acting on the satellite, called nutation torque

$$\boldsymbol{\tau}_{\mathrm{n}} = -\mathbf{h} \times \boldsymbol{\omega}_{\mathrm{ib}}, \tag{2.13}$$

which causes a tumbling motion. This motion appears as long as the angular velocity vector and the angular momentum vector are not parallel. Knowing that

$$\mathbf{h} = \mathbf{I} \cdot \boldsymbol{\omega}_{\mathrm{ib}}, \tag{2.14}$$

we can replace the angular momentum with the angular velocity, introducing the inertia tensor $\mathbf{I}$. In case the angular momentum and angular velocity vectors are parallel, we call the rotation axis $\boldsymbol{\omega}_{\mathrm{ib}}$ a principal axis of rotation. This happens, if and only if $\boldsymbol{\omega}_{\mathrm{ib}}$ is an eigenvector of $\mathbf{I}$.
We can evaluate (2.14) in any coordinate system. It is convenient to evaluate it in the BF

$$\mathbf{h}^{\mathrm{b}} = \mathbf{I}^{\mathrm{b}} \cdot \boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}, \tag{2.15}$$

where the inertia tensor of the satellite in BF is given as

$$\mathbf{I}^{\mathrm{b}} = \begin{bmatrix} 3.08\text{e-}3 & 4.57\text{e-}5 & 7.58\text{e-}6 \\ 4.57\text{e-}5 & 2.96\text{e-}3 & -1.78\text{e-}5 \\ 7.58\text{e-}6 & -1.78\text{e-}5 & 3.01\text{e-}3 \end{bmatrix} \mathrm{kg\,m^2}. \tag{2.16}$$

This inertia tensor is used for all further analysis and simulations, if not stated otherwise. It is calculated from the computer generated model of MOVE-II provided by the structure team of MOVE-II.
In case a spin stabilizing controller is selected, it is of interest to consider the behavior when the satellite rotates around its Z-axis. This is the axis which should point away from the sun in order to align the Flappanels. Calculating the normalized eigenvectors, which equal the principal axes of inertia

$$\mathbf{p}_1^{\mathrm{b}} = \begin{bmatrix} 0.95 \\ 0.32 \\ 0.02 \end{bmatrix}, \quad \mathbf{p}_2^{\mathrm{b}} = \begin{bmatrix} 0.31 \\ -0.91 \\ -0.27 \end{bmatrix}, \quad \mathbf{p}_3^{\mathrm{b}} = \begin{bmatrix} 0.07 \\ -0.26 \\ 0.96 \end{bmatrix}, \tag{2.17}$$

and the corresponding eigenvalues, which equal the principal moments of inertia

$$m_1 = 3.1\text{e-}3\,\mathrm{kg\,m^2},\;\; m_2 = 2.9\text{e-}3\,\mathrm{kg\,m^2},\;\; m_3 = 3.0\text{e-}3\,\mathrm{kg\,m^2}, \tag{2.18}$$

from $\mathbf{I}^{\mathrm{b}}$, we find that the Z-axis is not a principal axis of inertia. This means that we observe a nutation motion for a spin around this axis. We also find that the principal axis of inertia closest to the Z-axis, $\mathbf{p}_3^{\mathrm{b}}$, belongs to the intermediate eigenvalue m_3. A rotation around a principal axis of inertia with an intermediate eigenvalue is not stable [VDMS17], and thus needs active stabilization to maintain spin around this axis.
Recalling (2.12), we can also split up the torque vector

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{\mathrm{c}} + \boldsymbol{\tau}_{\mathrm{dist}} \tag{2.19}$$

into a control torque $\boldsymbol{\tau}_{\mathrm{c}}$ and a disturbance torque $\boldsymbol{\tau}_{\mathrm{dist}}$. The composition of $\boldsymbol{\tau}_{\mathrm{dist}}$ and the origin of the individual disturbance torques are discussed in Chapter 2.6. Substituting (2.14) and (2.19) into (2.12), we obtain

$$\mathbf{I} \cdot \dot{\boldsymbol{\omega}}_{\mathrm{ib}} = -\mathbf{I} \cdot \boldsymbol{\omega}_{\mathrm{ib}} \times \boldsymbol{\omega}_{\mathrm{ib}} + \boldsymbol{\tau}_{\mathrm{c}} + \boldsymbol{\tau}_{\mathrm{dist}}, \tag{2.20}$$

where $\boldsymbol{\tau}_{\mathrm{n}} = -\mathbf{I} \cdot \omega_{\mathrm{ib}} \times \omega_{\mathrm{ib}}$ is the nutation torque. After solving for $\dot{\boldsymbol{\omega}}_{\mathrm{ib}}$ we end up with the dynamic equations for the spacecraft in the BF

$$\begin{aligned} \dot{\boldsymbol{\omega}}_{\mathrm{ib}}^{\mathrm{b}} &= (\mathbf{I}^{\mathrm{b}})^{-1}(\mathbf{I}^{\mathrm{b}}\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} \times \boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} + \boldsymbol{\tau}_{\mathrm{c}}^{\mathrm{b}} + \boldsymbol{\tau}_{\mathrm{dist}}^{\mathrm{b}}) \\ &= \mathbf{d}(\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}, \boldsymbol{\tau}_{\mathrm{c}}^{\mathrm{b}}). \end{aligned} \tag{2.21}$$

For a linear control approach, these system equations have to be linearized around an operation point. Let

$$\mathbf{x}_{\mathrm{f}} = \begin{bmatrix} \mathbf{s}^{\mathrm{b}} \\ \boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} \end{bmatrix} = \begin{bmatrix} s_{\mathrm{x}} \\ s_{\mathrm{y}} \\ s_{\mathrm{z}} \\ \omega_{\mathrm{x}} \\ \omega_{\mathrm{y}} \\ \omega_{\mathrm{z}} \end{bmatrix} \tag{2.22}$$

be the full state vector, where the first three entries represent the components of vector $\mathbf{s}^{\mathrm{b}}$ and the last three entries represent the components of vector $\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}$. Then we define an operation point for the sun vector $\mathbf{s}^{\mathrm{b}}$ as

$$\mathbf{s}_{\mathrm{OP}}^{\mathrm{b}} = (0, 0, -1) \tag{2.23}$$

such that the Flappanels of MOVE-II point towards the sun in operation mode. The operation point for the angular velocity is defined as

$$\boldsymbol{\omega}_{\mathrm{ib,OP}}^{\mathrm{b}} = (0, 0, \omega_{\mathrm{s}}). \tag{2.24}$$

With the free variable ω_s it is possible to define a spin stabilizing around the Z-axis in operation mode. This gives us the state vector for the operation point

$$\mathbf{x}_{\mathrm{f,OP}} = (0, 0, -1, 0, 0, \omega_\mathrm{s})^\mathrm{T}. \tag{2.25}$$

Calculating the Jacobian of the system equations and applying the operating point $\mathbf{x}_{\mathrm{f,OP}}$, we obtain the full system matrix

$$\mathbf{A}_{\mathrm{f,d}} = \left.\frac{\delta \begin{bmatrix} \mathbf{m}_\mathrm{s}(\mathbf{s}^\mathrm{b}, \boldsymbol{\omega}_\mathrm{ib}^\mathrm{b}) \\ \mathbf{d}(\boldsymbol{\omega}_\mathrm{ib}^\mathrm{b}, \boldsymbol{\tau}_\mathrm{c}^\mathrm{b}) \end{bmatrix}}{\delta \mathbf{x}_\mathrm{f}}\right|_{\mathbf{x}_{\mathrm{f,OP}}} = \begin{bmatrix} 0 & \omega_\mathrm{s} & 0 & 0 & 1 & 0 \\ -\omega_\mathrm{s} & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \omega_\mathrm{s}\sigma_1 & 0 \\ 0 & 0 & 0 & -\omega_\mathrm{s}\sigma_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.26}$$

$$\text{with} \quad \sigma_1 = \frac{\mathbf{I}_{22}^\mathrm{b} - \mathbf{I}_{33}^\mathrm{b}}{\mathbf{I}_{11}^\mathrm{b}}, \quad \sigma_2 = \frac{\mathbf{I}_{11}^\mathrm{b} - \mathbf{I}_{33}^\mathrm{b}}{\mathbf{I}_{22}^\mathrm{b}},$$

where we use a diagonal inertia tensor for simplification, with the principal moments of inertia identifying the diagonal elements $\mathbf{I}_{11}^\mathrm{b} = m_1$, $\mathbf{I}_{22}^\mathrm{b} = m_2$, $\mathbf{I}_{33}^\mathrm{b} = m_3$ and all off-diagonal elements equal zero.
We find that the third row and the third column of the system matrix $\mathbf{A}_{\mathrm{f,d}}$ consists of zeros, which means that the s_z component of $\mathbf{s}^\mathrm{b}$ does not affect nor is affected by any other state variable. Furthermore, since the sun vector is a unit vector, the component s_z is fully determined by the other two components s_x and s_y. Since also no input nor disturbance can influence s_z directly, we can reduce the system matrix and the state vector by one variable to obtain the reduced state vector and the reduced system matrix [MCN+17]

$$\mathbf{x}_\mathrm{r} = \begin{bmatrix} s_\mathrm{x} \\ s_\mathrm{y} \\ \omega_\mathrm{x} \\ \omega_\mathrm{y} \\ \omega_\mathrm{z} \end{bmatrix}, \quad \mathbf{A}_{\mathrm{r,d}} = \begin{bmatrix} 0 & \omega_\mathrm{s} & 0 & 1 & 0 \\ -\omega_\mathrm{s} & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & \omega_\mathrm{s}\sigma_1 & 0 \\ 0 & 0 & -\omega_\mathrm{s}\sigma_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.27}$$

This observation also holds when we use the true inertia tensor instead of the simplified diagonal one. The resulting system matrix

$$\mathbf{A}_\mathrm{r} = \begin{bmatrix} 0 & \omega_\mathrm{s} & 0 & 1 & 0 \\ -\omega_\mathrm{s} & 0 & -1 & 0 & 0 \\ 0 & 0 & 1.52\text{e-}2 \cdot \omega_\mathrm{s} & -1.54\text{e-}2 \cdot \omega_\mathrm{s} & -1.15\text{e-}2 \cdot \omega_\mathrm{s} \\ 0 & 0 & -2.45\text{e-}2 \cdot \omega_\mathrm{s} & -1.52\text{e-}2 \cdot \omega_\mathrm{s} & -4.94\text{e-}3 \cdot \omega_\mathrm{s} \\ 0 & 0 & 5.74\text{e-}3 \cdot \omega_\mathrm{s} & 2.37\text{e-}3 \cdot \omega_\mathrm{s} & -3.16\text{e-}7 \cdot \omega_\mathrm{s} \end{bmatrix} \tag{2.28}$$

features more nonzero entries, expresses the real system dynamics more accurately and still allows the modification for an arbitrary spin stabilizing angular velocity ω_s.

2.4 Actuation Model

MOVE-II relies solely on magnetic actuation for attitude control. This means that the satellite produces a magnetic control dipole moment $\mathbf{m}_\mathrm{c}$. Together with the ambient magnetic field of Earth $\mathbf{b}$ it produces a control torque

$$\boldsymbol{\tau}_\mathrm{c} = \mathbf{m}_\mathrm{c} \times \mathbf{b} = [\mathbf{b}\times]^\mathrm{T}\mathbf{m}_\mathrm{c}, \tag{2.29}$$

which causes a change in the angular momentum $\mathbf{h}$ [SL05].
Three coils aligned with the axis of the satellite's body frame X, Y and Z produce the dipole moment

$$\mathbf{m}_\mathrm{c} = \mathbf{i}_\mathrm{c} \cdot n_\mathrm{c} \cdot A_\mathrm{eff}, \tag{2.30}$$

which depends on the control current $\mathbf{i}_\mathrm{c}$ through the coils as well as the number of windings n_c of each coil and the effective area A_eff of the windings. The product of the latter two quantities is constant and takes the value

$$n_\mathrm{c} \cdot A_\mathrm{eff} = 13 \cdot 6 \cdot 0.0039\,\mathrm{m}^2 = 0.3042\,\mathrm{m}^2 \tag{2.31}$$

for every coil used by the actuation system of MOVE-II [MGC+17].
Magnetic actuation has a severe limitation: The generated torques always lie in the plane perpendicular to the magnetic field of Earth [MC14, p. 307]. This is also clear from (2.29). Thus it is not possible to generate an arbitrary control vector in the three-dimensional space. It seems plausible that this limitation could make the system uncontrollable. However, the satellite is moving around the Earth and so the magnetic field observed by the satellite changes. The change is also periodical with the orbit period. Assuming that the magnetic dipole of Earth aligns with its rotation axis, [BD03] proves, under certain restrictions for the orbit of a satellite, that the 3-axis attitude dynamics are fully controllable for the nonlinear system. The restrictions mentioned above are that the orbital plane must not coincide with the magnetic equatorial plane nor must it include the magnetic poles. With a target inclination of 97.78°, MOVE-II is far away from the magnetic equatorial plane, but the satellite might come close to the magnetic poles. Therefore special attention has to be paid to the points where the satellite approaches the poles.
Since dipole moments parallel to the magnetic field $\mathbf{b}$ produce no torque, it is a waste of energy to generate them. To ensure that the control dipole moment $\mathbf{m}_\mathrm{c}$ has no parallel components to $\mathbf{b}$, it is calculated in a way that enforces its perpendicularity to the magnetic field:

$$\mathbf{m}_\mathrm{c} = \frac{\mathbf{u} \times \mathbf{b}}{||\mathbf{b}||} = \frac{1}{||\mathbf{b}||}[\mathbf{b}\times]^\mathrm{T}\mathbf{u}, \tag{2.32}$$

where $\mathbf{u}$ is the control input [MCN+17]. The physical meaning of $\mathbf{u}$ can be interpreted as the magnitude of the desired dipole moment multiplied by the reverse direction of the desired control torque vector.

Combining (2.29) and (2.32), we obtain the equation for the control torque as a function of the control input

$$\boldsymbol{\tau}_\mathrm{c} = \frac{[\mathbf{b}\times]^\mathrm{T}[\mathbf{b}\times]^\mathrm{T}}{||\mathbf{b}||}\mathbf{u}. \tag{2.33}$$

This function is valid for any single point in time only, since the magnetic field is time dependent. Considering this and (2.21) we obtain the time dependent input matrix

$$\mathbf{B}_\mathrm{r}^*(t) = \begin{bmatrix} \mathbf{0_{2\times 3}} \\ \mathbf{I}^{-1}\frac{[\mathbf{b}(t)\times]^\mathrm{T}[\mathbf{b}(t)\times]^\mathrm{T}}{||\mathbf{b}(t)||} \end{bmatrix} \tag{2.34}$$

for the reduced state, where $\mathbf{0_{2\times 3}}$ is a matrix of size 2×3 filled with zeros.
To determine the magnetic field of Earth, the International Geomagnetic Reference Field (IGRF) [IAG14], 12th generation can be used. This is an accurate model which suits fine for simulations. For controller design and analysis, a simpler approach is used. Assuming a magnetic dipole $\mathbf{m}_\mathrm{E}$ at the center of Earth with dipole moment $||\mathbf{m}_\mathrm{E}|| = 7.7 \cdot 10^{22}\ \mathrm{A\,m^2}$ pointing to the south celestial pole, the magnetic field

$$\mathbf{b}^\mathrm{i}(\mathbf{r}^\mathrm{i}) = \frac{\mu_0}{4\pi \cdot ||\mathbf{r}^\mathrm{i}||} \cdot [3(\mathbf{m}_\mathrm{E}^\mathrm{i} \circ \mathbf{r}^\mathrm{i}) \cdot \mathbf{r}^\mathrm{i} - \mathbf{m}_\mathrm{E}^\mathrm{i}] \tag{2.35}$$

can be approximated [Wer02], where μ_0 is the vacuum permeability, $(\mathbf{v}_1 \circ \mathbf{v}_2)$ describes the dot product of two vectors and $\mathbf{r}^\mathrm{i}$ the position vector of the point of interest in ECI coordinates.

2.5 Properties of the MOVE-II ADCS

In this chapter, the limitations due to hardware, software and system constraints are presented, as well as the default settings for the controller.
For a desired control dipole moment $\mathbf{m}_\mathrm{c}$ the necessary current

$$\mathbf{i}_\mathrm{c} = \frac{\mathbf{m}_\mathrm{c}}{n \cdot A_\mathrm{eff}} \tag{2.36}$$

can be calculated through rearranging (2.30). Due to design limitations, the current for each coil can either be off or in the range between $I_\mathrm{min} = 50\,\mathrm{mA}$ to $I_\mathrm{max} = 300\,\mathrm{mA}$. If one coil current exceeds the maximum, a firmware algorithm accordingly scales the currents for the other coils down, too. This coil scaling ensures that the desired direction for the control torque is maintained and is not distorted by the current limit of one coil. However if one component of $\mathbf{i}_\mathrm{c}$ is less than I_min, the actuation hardware will ignore this actuation command and the according coil is not actuated. To calculate the desired control moment from (2.32), it is important to accurately measure the Earth's magnetic field. However, this is not possible while the satellite actuates, since the generated magnetic field superposes the Earth's magnetic field. To solve this issue, a duty cycle is introduced for the actuation system of MOVE-II.

Parameter Name	Default Value	Description
$I_{\min}$	0.05 A	Minimum possible coil current.
$I_{\max}$	0.3 A	Maximum possible coil current.
T_{a}	500 ms	Active time of the coils.
T_{am}	1 s	Actuation-measurement period.
Coil scaling	on	Keep direction of control torque, if at least one coil exceeds $I_{\max}$.
Controller mode	Default Controller	B-dot for detumbling or Default Controller (see Chap. 3.2) for sun pointing.
ω_{s}	0.1 rad/s	Spin rate for spin stabilization.

Table 2.3: Standard parameter set for the MOVE-II controller.

During the magnetometer measurement phase of 500 ms, MOVE-II deactivates its coils, so that the measurements are not influenced by the magnetic field produced by the coils. Then there follows an actuation period $T_{\mathrm{a}} = 500\,\mathrm{ms}$. This cycle repeats, and thus the actuation-measurement period is $T_{\mathrm{am}} = 1\,\mathrm{s}$. With these values, a duty cycle

$$D = \frac{T_{\mathrm{a}}}{T_{\mathrm{am}}} \tag{2.37}$$

can be defined. For MOVE-II we obtain a duty cycle of $D = 50\%$.

As sun pointing controller, a linear state feedback controller with gain $\mathbf{K}_{\mathrm{def}}$ called Default Controller is implemented, which transfers the satellite into a sun pointing mode after launch. This gain is designed for a spin stabilized satellite with $\omega_{\mathrm{s}} = 0.1\,\mathrm{rad}$. This control law is analyzed in more detailed in Chapter 3.2.

A dedicated microprocessor runs the controller software and sets the right currents and actuation times for the individual coils. Via a daemon running on the board computer of MOVE-II, this controller software can be configured with various parameters. The relevant parameters as well as the system limitations are summarized in Table 2.3. This parameter set is referred to as the standard parameter set. Every simulation will be run with these parameters as default, unless stated otherwise.

Furthermore it is possible to upload a firmware update during the operation phase of MOVE-II. This makes it possible to extend the controller software with new controller modes to implement new control strategies. It is therefore possible to select one controller evaluated within the scope of this theses and evaluate it on-orbit.

2.6 Disturbances

Three main disturbance torques

$$\boldsymbol{\tau}_{\text{dist}} = \boldsymbol{\tau}_{\text{d}} + \boldsymbol{\tau}_{\text{g}} + \boldsymbol{\tau}_{\text{m}} \tag{2.38}$$

are considered for this low earth orbit, which are caused by atmospheric drag ($\boldsymbol{\tau}_{\text{d}}$), gravity gradient ($\boldsymbol{\tau}_{\text{g}}$) and the residual magnetic dipole moment ($\boldsymbol{\tau}_{\text{m}}$). The following chapters will describe them in more detail.

2.6.1 Drag

The drag torque results of the forces acting on the satellite when it moves through the residual atmosphere of Earth. This torque is determined through the integral of all forces d$\mathbf{f}$ acting on an infinitesimal surface element dA and its corresponding lever arm $\mathbf{l}$, which is the distance vector between the point of attack of the force and the center of mass of the satellite.
The drag torque is not constant and varies significantly over time and with attitude. Sun intensity influences atmospheric density, which introduces time dependency, and the attitude of the satellite defines how many Flappanels are exposed to the atmospheric flow and offer an additional attack surface.
In the simulation models, the time dependency of this disturbance torque is not taken into account. We use a simplified model with a constant air density value of $\rho_{\text{a}} = 2.639\text{e-}13\,\text{kg/m}^3$. However, the attitude dependency is considered in the simulation. A simplified estimation is used, where the surface of the satellite is decomposed into several panels. An algorithm recognizes which panels are exposed to the air flow and at what angle they are exposed to it. The overshadowing effect of the protruding Flappanels is also approximated, but not analytically determined. From this a force $\mathbf{f}_{\text{i}}$ is calculated for each panel, and together with its corresponding lever arm $\mathbf{l}_{\text{i}}$ the drag torque

$$\boldsymbol{\tau}_{\text{i}} = \mathbf{f}_{\text{i}} \times \mathbf{l}_{\text{i}} \tag{2.39}$$

of each panel can be determined. Summing up the torques of all panels we obtain the total drag torque

$$\boldsymbol{\tau}_{\text{d}} = \sum_{\forall i} \boldsymbol{\tau}_{\text{i}} \tag{2.40}$$

acting on the satellite [MCN+17].

2.6.2 Gravity Gradient

The gravity gradient disturbance torque is caused by the gravitational field of Earth. A satellite with not evenly distributed moments of inertia aligns its minimum moment of inertia axis with $\hat{\mathbf{r}}$, which is the normalized position vector of the satellite with respect to the center of Earth. For the gravity gradient disturbance torque

$$\boldsymbol{\tau}_g = 3 \cdot \omega_{\text{o}}^2 \cdot (\hat{\mathbf{r}} \times \mathbf{I}\hat{\mathbf{r}}), \tag{2.41}$$

an analytical formula can be obtained [MCN+17]. Here, ω_o is the orbital rate of the satellite's orbit and $\mathbf{I}$ is the inertia tensor. Recalling the inertia tensor from (2.16), we find that the deviation moments are small compared to the principal moments, and that the principal moments are approximately of the same size in numbers. This means that the gravity gradient torque is expected to be small.

2.6.3 Residual Magnetic Dipole Moment

Due to magnetizable and magnetic materials as well as current loops, the satellite produces its own magnetic field which interacts with the Earth magnetic field. This disturbance is the most significant. Figure 2.4 illustrates this in a semi-logarithmic plot by showing all occurring disturbance torques for the time span of 2 orbits in the simulation of a standard scenario for MOVE-II. In comparison to the maximum possible control torque $\boldsymbol{\tau}_{\mathrm{c,max}}$, which varies along the orbit between 13e-6 N m and 6.4e-6 N m due to the changing magnitude of the magnetic field of Earth, the disturbances are still more than one order of magnitude smaller.

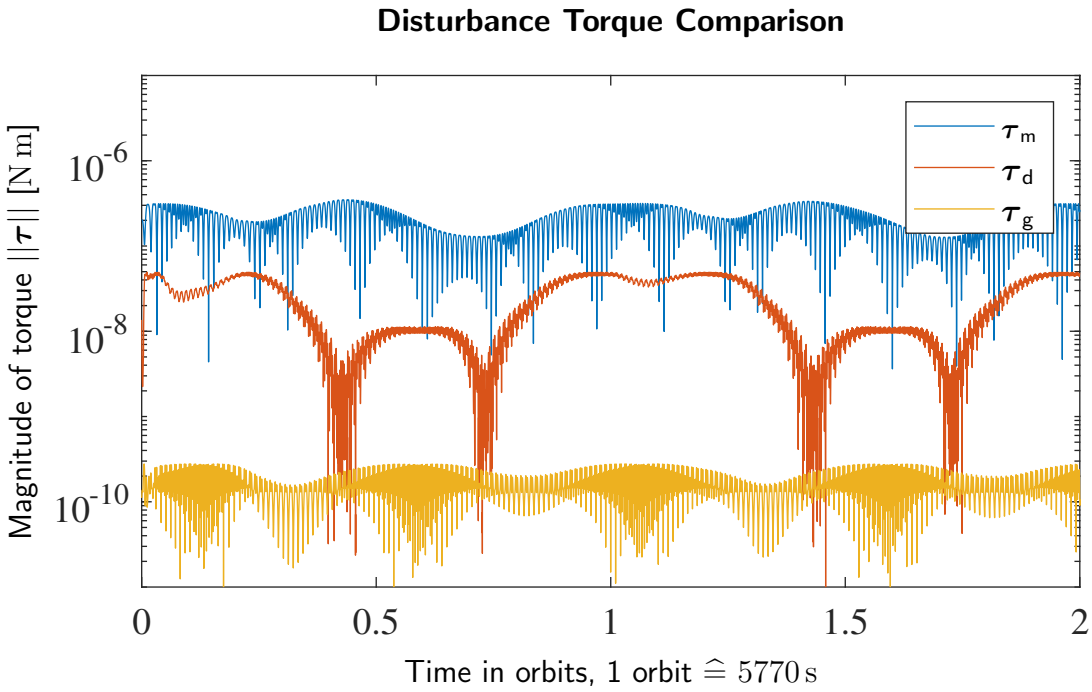

Figure 2.4: Comparison of disturbance torques in the simulation of a standard scenario of MOVE-II in a semi-logarithmic plot. We note that the residual magnetic dipole moment has the greatest influence, while the gravity gradient is negligible.

The residual magnetic field produced by the satellite can be approximated by a constant residual magnetic dipole moment $\mathbf{m}_{\mathrm{res}}$, which is inherent to the satellite and causes a disturbance torque

$$\boldsymbol{\tau}_m = \mathbf{m}_{\mathrm{res}} \times \mathbf{b} \tag{2.42}$$

by interacting with the magnetic field **b** of Earth.

Characterization of the Residual Magnetic Dipole Moment of MOVE-II

The ADCS team measured the residual magnetic dipole moments for MOVE-II at the facility of IABG. In this setup, three magnetometers measure the magnetic field of the satellite in distances of 28, 48 and 68 cm and for every measurement a corresponding magnetic dipole and quadrupole moment is calculated. These three results are averaged to obtain the estimated residual magnetic dipole moment for MOVE-II. Three testcases are analyzed: Testcase 1 is recorded with the satellite switched off, Testcase 2 is recorded with the satellite switched on, but the Communication System (COM) switched off, and Testcase 3 is recorded with all systems on and COM constantly transmitting packages. Table 2.4 shows the results of these measurements: The magnitude of the magnetic dipole and quadrupole moments, as well as the component of the dipole moments in every axis of the BF.

State of Satellite	X_{BF}, di.	Y_{BF}, di.	Z_{BF}, di.	Di. mag.	Quad. mag.
Testcase 1: Off	6.21	2.48	2.58	7.17	0.11
Testcase 2: On, COM off	6.23	2.47	1.79	6.94	0.07
Testcase 3: On, COM constantly transmitting	6.18	2.48	1.47	6.82	0.20

Table 2.4: Components of the magnetic dipole (di.) moments and magnitude (mag.) of dipole and quadrupole (quad.) moments in [mA m^2] for MOVE-II, in the BF.

Mehlem [Meh78] proposes a method to calculate a more accurate magnetic field model for a satellite, consisting of several magnetic dipoles at different positions. He assumes a certain number of dipoles N and places the positions and magnitudes of these dipoles such that the error between the modeled magnetic field and the measured magnetic field becomes minimal. For this process a least squares fit algorithm in a two part iterative process is used. It is considered to use such a dipole model with $N = 2$, to reflect also the quadrupole moment for modeling the magnetic dipole disturbance of MOVE-II. However, since the measurements from IABG indicate a low quadrupole moment, the decision for the scope of this thesis is to neglect the quadrupole moment and stick with the simplified dipole moment model. In the worst case, which is Testcase 3, the ratio of the quadrupole magnitude to the dipole magnitude is below 3%.
All simulations in this thesis, if not stated otherwise, assume a magnetic dipole moment with values according to Testcase 2. This decision is reasonable, since in

nominal mode the satellite only transmits data packages for a duration of 5 seconds every minute.
Another observation one obtains from these measurements is that the dipole moment in the X- and Y-axes is approximately constant over the three testcases, but varies significantly in the Z component. With increasing power consumption, contingent on additionally activated modules, the Z component of the magnetic dipole decreases. One way to explain this behavior is with current loops. When the currents in the loops increase, also the residual magnetic dipole moments increase in the normal direction of the loops. Since most of the satellite's boards are mounted in the X-Y-plane, the magnetic dipole moments generated by such current loops point along the Z-direction. If they predominantly point in the negative Z-direction, it explains why the Z component of the residual magnetic dipole moment decreases with increasing currents, compared to the measurement with the satellite switched off.
For the next generation of MOVE it is therefore suggested to keep possible current loops as small as possible to avoid that the residual magnetic dipole moment becomes depended on the current consumption of the individual hardware boards.

Methods for On-Orbit Estimation

The residual magnetic dipole moment is not only influenced by the current draw of the satellite. Busch et al. [BBDS15] observe, that the constant part of the dipole still can change after launch of the satellite. They assume that certain materials are magnetized during the launch process. Therefore it is important to measure and verify the residual magnetic dipole moment after launch with on-orbit methods. In literature, different approaches to this task can be found.
Huang and Jing [HJ06] present a method where a magnetometer attached on a pole measures the magnetic far field of the satellite. From this measurement, an estimation of the residual dipole moment is calculated. This method has several drawbacks: First, the satellite needs a mechanism or construction to place a magnetometer far from the center of the satellite. Second, the sensor bias must be low in comparison to the magnetic field produced by the residual magnetic dipole. Third, this method assumes the dipole to be in the center of the satellite, which is not true for every case. These three restrictions make this method unattractive for CubeSats, whose requirements often are geometric compactness and cheap production costs, which result in cheap sensors with a higher sensor noise and bias.
However, Busch et al. [BBDS15] present a different method to estimate the residual magnetic dipole on-orbit. They evaluated the satellite's rotational rate $\boldsymbol{\omega}_{\mathrm{ib}}$ and measure the Earth magnetic field $\mathbf{b}$ at the corresponding time points. The only requirement for this method is, that the data must be sampled at sufficient resolution compared to the spin rate of the satellite. They then calculate the rotational acceleration $\dot{\boldsymbol{\omega}}_{\mathrm{ib}}$ with numerical methods and ascertain the torque $\boldsymbol{\tau}$ acting on the satellite with help of Euler's equation

$$\boldsymbol{\tau} = \mathbf{I} \cdot \dot{\boldsymbol{\omega}}_{\mathrm{ib}} + \boldsymbol{\omega}_{\mathrm{ib}} \times (\mathbf{I} \cdot \boldsymbol{\omega}_{\mathrm{ib}}). \tag{2.43}$$

With a least square fit algorithm for (2.43) in combination with (2.42), the residual magnetic dipole moment $\mathbf{m}_{\text{res}}$ can be calculated. When the satellite is not actuated, one can assume the magnetic disturbance torque to be the dominant one, and therefore this method is a good estimation of the real residual magnetic dipole moment.

Chapter 3

Linear State Feedback Controller

This chapter describes the sun pointing controller as is currently implemented in the ADCS firmware of MOVE-II. Section 3.1 presents the necessary theory about state feedback controllers, the Linear Quadratic Regulator (LQR) design algorithm and the Floquet theory. Section 3.2 defines the Default Controller which is implemented in the firmware at the time of launch. It is also shown how the feedback gain for the Default Controller is derived. Section 3.3 evaluates the Default Controller in various different scenarios and its limitations are shown. In Section 3.4 these limitations are discussed.
In this and the following chapters, the pointing error e_p of the satellite is of great interest. This is the angle between the negative Z-axis of the satellite and the sun vector $\mathbf{s}$, usually expressed in degree. It is defined with help of the dot product

$$\cos(e_\mathrm{p}) = \mathbf{s}^\mathrm{b} \circ (0,0,-1)^\mathrm{T} = (s_\mathrm{x}, s_\mathrm{y}, s_\mathrm{z})^\mathrm{T} \circ (0,0,-1)^\mathrm{T} = -s_\mathrm{z} \tag{3.1}$$

of the sun vector in the BF and the negative Z-axis. Therefore we define the pointing error as

$$e_\mathrm{p} = \cos^{-1}(-s_\mathrm{z}). \tag{3.2}$$

3.1 Theoretical Background

Linear state feedback controllers utilize the full state vector $\mathbf{x} = \mathbf{x}(t)$ to calculate a control input $\mathbf{u} = \mathbf{u}(t)$. Linear state feedback control laws

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t) \tag{3.3}$$

consist of a matrix multiplication with a control gain $\mathbf{K}$, which is assumed to be constant over time for the scope of this thesis.
A popular approach to find such a feedback gain in the field of aerospace is the use of the LQR algorithm [Yan12, Yan17, SAS16]. The reason for its popularity is the fact that one can penalize high input signals and system state deviations from the operation point with individual weights, which allows to custom tailor a gain to a

specific use case. Another advantage of gains developed this way is the insensitivity of the closed loop system to disturbances and parameter variations, as Kwakernaak and Sivan [KS72, Chap. 3.9] show. The LQR algorithm returns a gain matrix $\mathbf{K}$, which guarantees to minimize the cost functional

$$J(\mathbf{u}, \mathbf{x}_0) = \int_{t_0}^{\infty} (\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + \mathbf{u}^{\mathrm{T}}\mathbf{R}\mathbf{u})\, \mathrm{d}t, \tag{3.4}$$

where $\mathbf{x}_0 = \mathbf{x}(t_0)$ is the initial state of the system [KS72]. The weight matrices $\mathbf{Q} \geq 0$ and $\mathbf{R} > 0$ can be used to specify to what extent a deviation from the equilibrium state and to what extend a system input is penalized respectively. The off-diagonal elements of both matrices are usually set to 0, but this is not necessarily required. It is required however, that $\mathbf{Q}$ and $\mathbf{R}$ are positive definite and positive semi-definite respectively. Another requirement for the LQR algorithm to be successful is that the linear time-invariant system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{3.5}$$

with system matrix $\mathbf{A}$ and input matrix $\mathbf{B}$ must be stabilizable.

Definition Stabilizability. *The linear time-invariant system* (3.5) *is* ***stabilizable****, if its unstable subspace is contained in its controllable subspace. That is, any vector* $\mathbf{x}$ *in the unstable subspace is also in the controllable subspace* [KS72, p. 62].

This definition implies that a completely controllable system is also stabilizable. Therefore it is sufficient to show that a system is controllable in order to apply the LQR algorithm successfully.

Definition Controllability. *The linear system with state differential equation* (3.5) *is said to be* ***completely controllable*** *if the state of the system can be transferred from the zero state at any initial time* t_0 *to any terminal state* $\mathbf{x}(t_1) = \mathbf{x}_1$ *within a finite time* $t_1 - t_0$ [KS72, p. 54].

Controllability of a system can be checked with the controllability matrix

$$\mathbf{S} = (\mathbf{B}, \mathbf{A}\mathbf{B}, \mathbf{A}^2\mathbf{B}, ..., \mathbf{A}^{n-1}\mathbf{B}), \tag{3.6}$$

where n is the dimension of the state vector $\mathbf{x}$. If the rank of $\mathbf{S}$ is equal to n, the system is completely controllable.
Without loss of generality one can set $t_0 = 0$. The feedback gain matrix $\mathbf{K}$ which fulfills this equation is obtained by

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P}, \tag{3.7}$$

where $\mathbf{P}$ is the unique positive definite solution of the algebraic Riccati equation

$$\mathbf{P}\mathbf{A} + \mathbf{A}^{\mathrm{T}}\mathbf{P} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P} + \mathbf{Q} = 0. \tag{3.8}$$

Within this context, both the system and the input matrices are time invariant. In the case of a time periodic input matrix like it is the case for a magnetically actuated spacecraft, the input matrix $\mathbf{B}^*(t)$ from (2.34) can be averaged over one period system T_s to obtain a time invariant system with a time invariant input matrix

$$\mathbf{B} = \frac{1}{T_\mathrm{s}} \int_{t_0}^{t_0+T_\mathrm{s}} \mathbf{B}^*(t)\, \mathrm{d}t, \tag{3.9}$$

as Wisniewski [Wis97] and Graversen et al. [GFV02] demonstrate. However, this time averaged system is not equivalent to the periodic system. This means that we must check for stability of the closed loop periodic system with the feedback gain obtained by solving the Riccati equation using the time averaged input matrix.
Stability of a periodic system can be analyzed by the Floquet theory [BC09]. Consider a closed loop system with the system matrix

$$\mathbf{A}_\mathrm{c}(t) = \mathbf{A} - \mathbf{B}(t)\mathbf{K}. \tag{3.10}$$

Then a solution to the homogeneous differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{A}_\mathrm{c}(t)\mathbf{x}(t) \tag{3.11}$$

with the initial condition $\mathbf{x}_0 = \mathbf{x}(t_0)$ is obtained as

$$\mathbf{x}(t) = \mathbf{\Phi}_{\mathbf{A}_\mathrm{c}}(t, t_0)\mathbf{x}_0, \tag{3.12}$$

where $\mathbf{\Phi}_{\mathbf{A}_\mathrm{c}}(t, t_0)$ is the transition matrix of the closed loop system, which describes how the state vector changes moving from time t_0 to time t. The state transition matrix $\mathbf{\Phi}_{\mathbf{A}_\mathrm{c}}(t, t_0)$ is obtained by the solution $\mathbf{X}(t)$ of the differential matrix equation

$$\dot{\mathbf{X}}(t) = \mathbf{A}_\mathrm{c}(t)\mathbf{X}(t_0) \tag{3.13}$$

with initial condition $\mathbf{X}_0 = \mathbf{X}(t_0) = \mathbf{1}_{n\times n}$, where $\mathbf{1}_{n\times n}$ describes the $n\times n$-dimensional square identity matrix.
The transition matrix $\mathbf{\Phi}_{\mathbf{A}_\mathrm{c}}(T_\mathrm{s} + t, t)$, which transfers a state vector $\mathbf{x}$ exactly one system period T_s forward in time, is also called monodromy matrix

$$\mathbf{\Psi}(t) = \mathbf{\Phi}_{\mathbf{A}_\mathrm{c}}(T_\mathrm{s} + t, t). \tag{3.14}$$

Bittanti and Colaneri [BC09] show that the eigenvalues of $\mathbf{\Psi}(t)$ are not time dependent. They are called characteristic multipliers of $\mathbf{A}_\mathrm{c}$. If all characteristic multipliers lie within the unit circle of the complex plane, the system state approaches zero as time approaches infinity, and thus the system is stable. When at least one characteristic multiplier lies outside the unit circle, the state vector grows without bounds, thus the system is unstable.

3.2 The Default Controller of MOVE-II

This section derives the gain for the state feedback controller used for sun pointing, which is active at the time of launch of MOVE-II. Together with the parameter table for the ADCS, namely Table 2.3, the gain developed in this section defines the Default Controller.
The results of Sections 2.3, 2.4 and 3.1 are used to obtain the system and input matrices $\mathbf{A}$ and $\mathbf{B}$, as they are needed to construct the gain $\mathbf{K}_{\text{def}}$, which is the gain for the Default Controller. It is also shown how the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ are designed. The decisions to achieve this gain are made entirely by the then ADCS team. The author merely presents these decisions in a comprehensible way.
The system matrix $\mathbf{A} = \mathbf{A}_{\text{r,d}}$ is the reduced system matrix from (2.27), calculated with the diagonal inertia tensor. The advantage of this decision is the easy computation and simple structure of the system matrix. The downside of this simplification is that the nutation torque introduced in Section 2.3 is not included in the model. Therefore, the LQR algorithm does not take into account this torque and it appears as an additional disturbance to the controller.
For spin stabilization an angular velocity $\omega_{\text{s}} = 0.1\,\text{rad/s}$ is chosen, which equals approximately ninety times the orbital angular velocity $\omega_{\text{o}} = 2\pi/T$. This choice shows promising results in several simulations [Kie17].

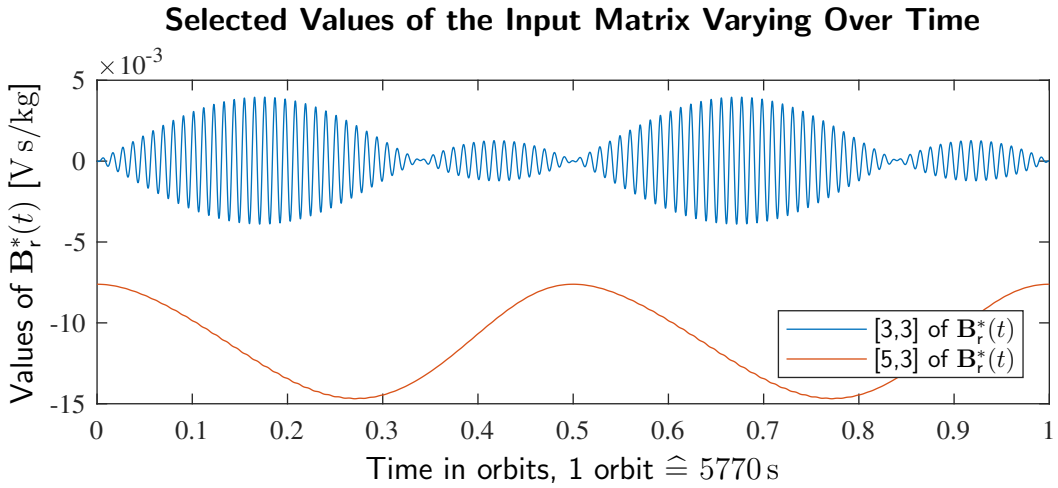

Figure 3.1: Visualizing two selected elements of the input matrix $\mathbf{B}_{\text{r}}^{*}(t)$ over the time of one orbit.

The input matrix $\mathbf{B}$ is time averaged as in the Section 3.1 presented. For this averaging process, the reduced input matrix $\mathbf{B}_{\text{r}}^{*}(t)$ is calculated from (2.34) with the diagonal inertia tensor, with the Earth dipole magnetic field model from (2.35) and with the satellite orbit described in Table 2.2. Since this magnetic field model is symmetric with respect to the Z-axis of Earth, the system period $T_{\text{s}} = 2885\,\text{s}$ is half of the orbit period T. This is also evident from Figure 3.1, where the elements $[3, 3]$

and $[5, 3]$ of the time dependent input matrix $\mathbf{B}_{\mathrm{r}}^{*}(t)$ are visualized over the time of one orbit. The resulting constant input matrix is

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -6.0\text{e-}3 & 5.0\text{e-}6 & 1.7\text{e-}6 \\ 4.5\text{e-}6 & -8.2\text{e-}3 & -5.5\text{e-}7 \\ 1.5\text{e-}6 & -5.7\text{e-}7 & -8.5\text{e-}3 \end{bmatrix}. \tag{3.15}$$

The controllability matrix associated with $\mathbf{A}$ and $\mathbf{B}$ is of rank 5, therefore the system is completely controllable and the requirement for using the LQR algorithm is fulfilled.
In order to obtain a feedback gain, the input weighting matrix $\mathbf{R} = \mathbf{1}_{3\times3}$ is fixed as identity matrix, while the state weighting matrix

$$\mathbf{Q}(q) = \begin{bmatrix} q\cdot\text{e-}10 & 0 & 0 & 0 & 0 \\ 0 & q\cdot\text{e-}10 & 0 & 0 & 0 \\ 0 & 0 & q\cdot\text{e-}5 & 0 & 0 \\ 0 & 0 & 0 & q\cdot\text{e-}5 & 0 \\ 0 & 0 & 0 & 0 & q\cdot\text{e-}5 \end{bmatrix} \tag{3.16}$$

is parameterized by q, which is swept through a range from 100 to 3e6. For each realization of this matrix, a corresponding gain matrix $\mathbf{K}$ and the eigenvalues of its monodromy matrix are calculated. For $\mathbf{Q} = \mathbf{Q}(2.28\text{e}6)$, a minimum value of the largest normed eigenvalue among all the corresponding monodromy matrices is found. This process of finding appropriate weighting matrices is adopted from Graversen et al. [GFV02]. Choosing a feedback gain, which produces a monodromy matrix with low eigenvalues, means that a disturbed state vector will approach zero faster compared to a monodromy matrix with higher eigenvalues.
With these weighting matrices, the LQR algorithm provides a feedback gain. Since the LQR algorithm assumes a continuous actuation, but the MOVE-II ADCS actuates only 50% of the time due to the duty cycle, the resulting matrix is multiplied by 2 in order to compensate for the time when the coils are disabled. The resulting gain matrix with minimal maximum eigenvalue is

$$\mathbf{K}_{\min} = \begin{bmatrix} -2.72\text{e-}2 & 9.00\text{e-}3 & -11.0 & -1.22 & -2.88\text{e-}5 \\ -1.56\text{e-}2 & -3.45\text{e-}2 & -1.68 & -11.0 & -2.94\text{e-}4 \\ 3.32\text{e-}6 & -6.07\text{e-}6 & 1.71\text{e-}4 & 2.93\text{e-}5 & -10.6 \end{bmatrix}. \tag{3.17}$$

The only non zero eigenvalues of its monodromy matrix is the complex conjugated pair $\lambda_{1,2} = 6.56\text{e-}2 \pm 2.8\text{e-}3\,\mathbf{i}$ with a magnitude of $|\lambda_{1,2}| = 6.6\text{e-}2$.
However, in simulations this feedback gain does not perform as desired. Especially with high residual magnetic disturbance torques, this gain does not achieve

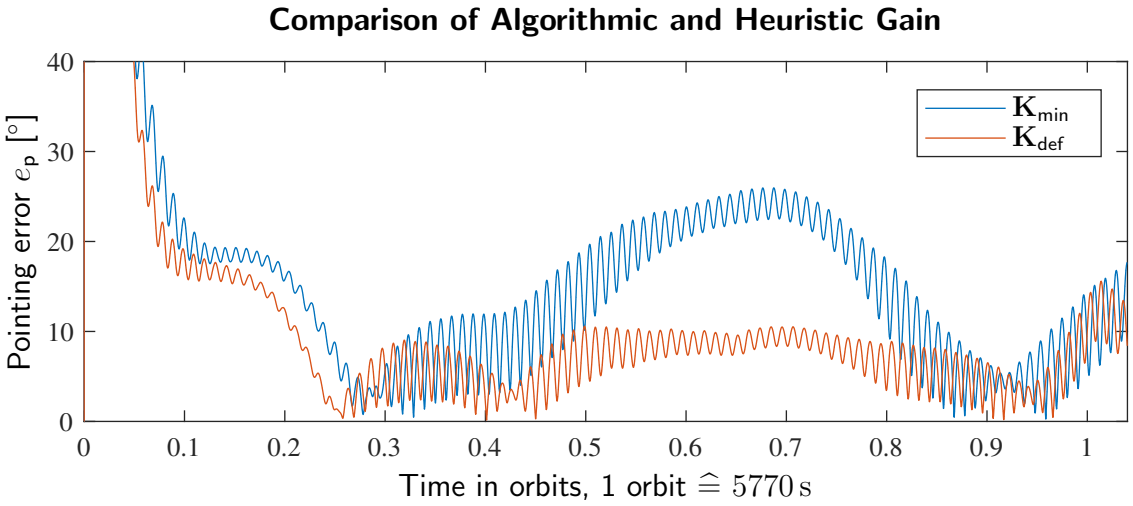

Figure 3.2: Comparing the pointing error of the algorithmically derived gain $\mathbf{K}_{\text{min}}$ to the heuristically modified gain $\mathbf{K}_{\text{def}}$.

a pointing error within the system requirements. Therefore, the gain is modified heuristically by iteratively changing and simulating, until a gain

$$\mathbf{K}_{\text{def}} = \begin{bmatrix} -6.8\text{e-}2 & 0 & -20 & -1.0 & 5.0\text{e-}3 \\ 0 & -6.2\text{e-}2 & -1.4 & -20 & 4.0\text{e-}3 \\ 0 & 0 & 8.0\text{e-}4 & 4.0\text{e-}4 & -3.4 \end{bmatrix} \tag{3.18}$$

is found, which has a lower pointing error under the presence of high magnetic dipole moments in simulations. This gain is configured as default gain into the ADCS firmware for the launch of MOVE-II.

Figure 3.2 shows a comparison between these two gains for the duration of one orbit, where the residual magnetic disturbance is multiplied by 3 to amplify the effect of this disturbance. One obtains that the gain $\mathbf{K}_{\text{min}}$, after settling, diverges again until it reaches a pointing error of approximately 25°, while the gain $\mathbf{K}_{\text{def}}$ is able to hold the pointing error within the 10° bounds for most of the time. Calculating the monodromy matrix of $\mathbf{K}_{\text{def}}$, the only non zero eigenvalues one obtains is the complex conjugated pair $\lambda_{1,2} = 8.0\text{e-}3 \pm 2.6\text{e-}2\,\mathbf{i}$ with magnitude $|\lambda_{1,2}| = 2.7\text{e-}2$. This proves that the heuristically found gain is also stabilizing. The magnitude of the largest eigenvalue is even lower compared to $\mathbf{K}_{\text{min}}$, which makes the improved performance reasonable. However, the resulting gain is not an LQR gain any more, therefore there is no guarantee that it provides a good disturbance rejection to other disturbances than the constant residual magnetic dipole disturbance, for which it is custom tailored.

3.3 Simulation Results of the Default Controller

This section evaluates the Default Controller. First it is analyzed how the controller performs in orbits different from the target orbit. Then, a Monte Carlo simulation is performed with random initial attitude and angular velocity. The results show the extent to which the controller is suitable to acquire the operation point from an arbitrary initial state.
For the simulations other than the Monte Carlo simulation, two different sets of initial conditions are used throughout this thesis. They are presented in Table 3.1.

Name	$\mathbf{q}_{\mathrm{ib},0}$	$\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib},0}$
Standard initial conditions	$(0,0,0,1)^{\mathrm{T}}$	$(5,5,5)^{\mathrm{T}} \cdot$ 1e-3 rad/s
Optimal initial conditions	$(0.22,0.53,0.47,-0.67)^{\mathrm{T}}$	$(0,0,\omega_{\mathrm{s}})^{\mathrm{T}}$ rad/s

Table 3.1: Initial condition sets.

The standard initial conditions are applied if not stated otherwise. They are constructed such that the satellite starts with an initial pointing error of approximately 70°. This allows to observe the settling behavior of the satellite. The initial velocities in each direction are small non-zero values. Therefore, the satellite has to spin up to reach its operation point.
The optimal conditions are designed for special benchmarks. The satellite starts in the operation point which implies a pointing error of 0°. When one is not interested in the settling behavior but only in the average pointing error during the simulation run, the optimal initial conditions are advantageous.
The simulation is implemented as a Matlab Simulink model. It is described in more detail in Appendix A. All relevant disturbances are included in this simulation model, as well as the characteristic sensor noise for the sensors of MOVE-II.

3.3.1 Sensitivity to Orbit Changes

This section analyzes the sensitivity of the Default Controller with respect to orbit parameter variations. It is not guaranteed that the satellite is launched into the desired orbit. Indeed, the satellite can end up in an entirely unexpected and unsuitable orbit. The two Galileo satellites FOC-FM1 and FOC-FM2 are good examples [GAC+15]. Thus it is of interest, how the satellite performs in different orbits and if it is robust to parametric changes.
Figure 3.3 shows the pointing error of four simulations over the time of more than 3.5 orbits with different inclinations i. The eclipse time is illustrated as gray area. During this time no information about the sun position is available to the satellite and the sun sensors return a zero-vector as sun vector. Therefore, the controller assumes the pointing error to be zero during eclipse. A moving average filter of window size 5e3 is applied to the results of this and the following figure,

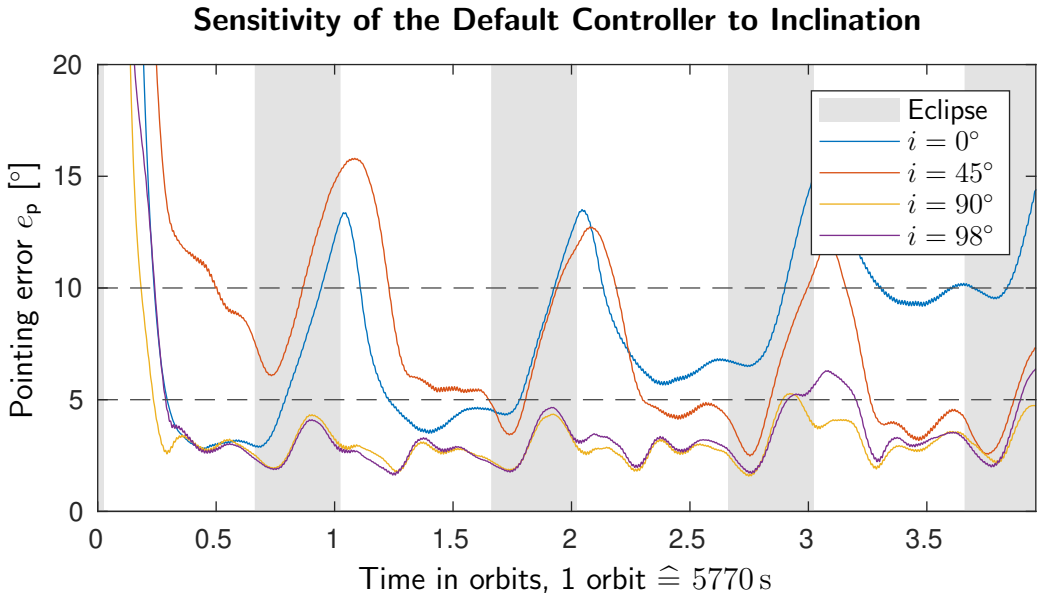

Figure 3.3: Pointing error achieved with the Default Controller for differently inclined orbits. The graphs are smoothed with a moving average filter.

where the data is sampled at 10 Hz. This smooths out high frequency oscillations and illustrates the longterm behavior of the pointing error. One obtains that the controller keeps the pointing error within the 5° boundary most of the time for high inclination orbits.

However, for orbits with inclination 0° and 45°, the pointing error increases significantly. Especially during eclipse, where no sun vector information is available and the attitude can be attained only through spin stabilization and driving the state components ω_x and ω_y to zero. The must requirement for the pointing error to be less than 10° is therefore violated during eclipse. For the equatorial orbit with inclination 0°, the controller is not able to decrease the pointing error by the same amount it increases during eclipse. The pointing becomes greater in numbers every orbit and eventually violates the pointing error requirement also when not in eclipse.

Figure 3.4 shows the results of simulations in different orbit heights and with a highly eccentric orbit. The semimajor axis is set to $a = R_E + h_E$, where $R_E = 6378$ km is the radius of Earth and h_E is the height of the orbit above ground for the circular orbits. The forth simulation is performed in a highly elliptical orbit with eccentricity $e = 2/3$. The semimajor axis is defined such that the apogee is at 2000 km and the perigee is at 400 km above ground. The time axis cannot be expressed in terms of orbits this time, since the different semimajor axes also imply different orbit periods of $T = 5770$ s, 5553 s, 7632 s, 6565 s respectively. The results show that the performance of the controller is not significantly affected by orbit height or shape. The orbit with height 2000 km above ground shows a slightly higher peak pointing

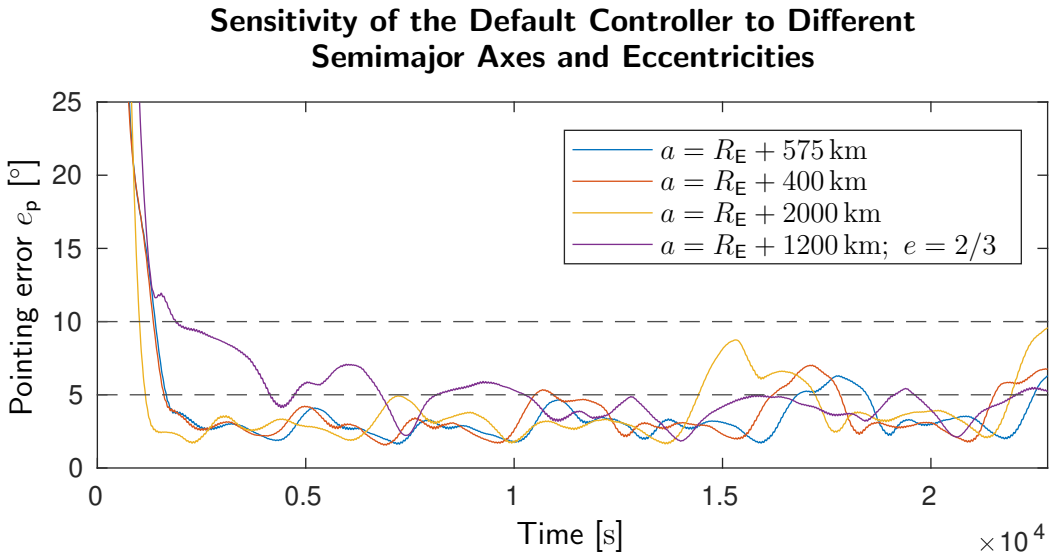

Figure 3.4: Pointing error achieved with the Default Controller for different orbit sizes and shapes. The graphs are smoothed with a moving average filter.

error than the other orbits. This is related to the longer orbit period, which implies that the satellite also spends more time consecutively in eclipse. Other orbit parameters variations show no significant impact on the controller behavior.

In Section 2.4 we state that controllability might be limited for equatorial orbits and for orbits including the magnetic poles of Earth. While we can observe an unstable behavior for the equatorial orbit, passing by close at the poles with high inclination orbits does not affect the pointing error in these simulations.

3.3.2 Sensitivity to Initial Values

This section analyzes the sensitivity of the Default Controller with respect to initial conditions. After launch or after reactivation of the ADCS after a blackout, the controller has to deal with an arbitrary initial attitude $\mathbf{q}_{\text{ib},0}$ and an arbitrary initial angular velocity $\boldsymbol{\omega}^{\text{b}}_{\text{ib},0}$. A Monte Carlo simulation with a number of $n = 100$ different testruns with simulation time 2500 s is performed to investigate the behavior of this controller with respect to random initial values of $\mathbf{q}_{\text{ib},0}$ and $\boldsymbol{\omega}^{\text{b}}_{\text{ib},0}$. The initial attitude takes any possible value, restricted to a unit quaternion, while the initial angular velocity is restricted to a range from $-0.085\,\text{rad/s}$ to $+0.085\,\text{rad/s}$ in each axis. This means a maximum magnitude of the angular velocity of $||\boldsymbol{\omega}^{\text{b}}_{\text{ib},0}|| = \sqrt{3 \cdot (0.085\,\text{rad/s})^2} = 0.15\,\text{rad/s}$ is possible. This assumption is based on the fact that the ADCS features a detumbling controller, which reduces the magnitude of the angular velocity vector to this level, before the sun pointing controller is activated [MCN+17].

Behavior	Description	Times Observed	In Percent
b_1	Nominal behavior	49	49%
b_2	Slow convergence	20	20%
b_3 & b_4	Oscillation	24	24%
b_5	Anti-pointing	7	7%

Table 3.2: Frequentness analysis of the five distinct behaviors in the Monte Carlo simulation of the Default Controller with $n = 100$ runs.

In Figure 3.5 we see the result of this Monte Carlo simulation. In total, five different behaviors are observed, which are categorized in Table 3.2. This table also states the frequentness of each behavior that is observed.

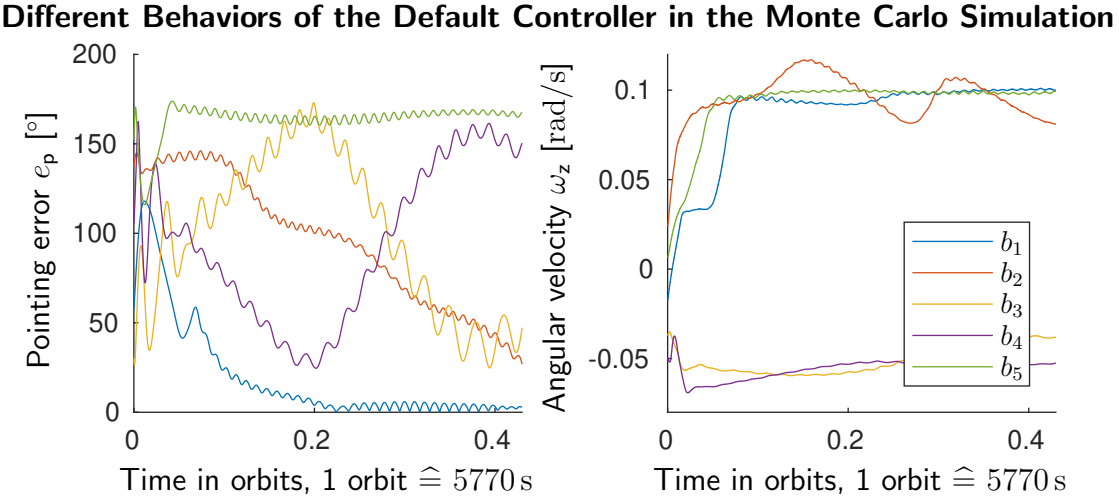

Figure 3.5: Results of the Monte Carlo simulation with five selected outcomes for the Default Controller.

The most often occurring behavior is the nominal behavior, labeled with b_1. It is characterized through a fast convergence. The pointing error must decrease below 10° within a time limit of 1500 s, which equals 0.26 orbits, to be characterized as nominal behavior. As Table 3.2 shows, nearly half of the simulation runs converge within this time.
An example for a simulation run where the pointing error shows a converging behavior, but where it takes longer than 1500 s to reach the 10° pointing error margin, is labeled with b_2. In simulation runs of this category, the pointing error can still be more than 20° after nearly half an orbit.
A special case is observed especially for initial conditions where the Z component of the initial angular velocity $\omega_{z,0}$ is negative. In this case, the satellite shows an oscillating behavior. Two examples can be seen in Figure 3.5 labeled as b_3 and b_4. One observes that unlike the other behaviors, with this starting condition the satellite is

not able to change ω_z from the negative initial value to the desired 0.1 rad/s. It is important to note that this behavior is observed without exception only when the initial value of ω_z is negative. However, there are also scenarios where the initial value of ω_z is negative and the satellite is not trapped in an oscillating behavior, but succeeds in achieving a low pointing error and the desired spin rate of 0.1 rad/s without difficulty. Therefore, we assume that the condition $\omega_{z,0} < 0$ is a necessary, but not a sufficient condition to observe such an oscillating behavior.

The behavior b_5 prevents the satellite from working as expected and leads to a blackout of the satellite sooner or later due to lack of power supply from the solar cells. The satellite spins up to $\omega_z = 0.1$ rad/s and is spin stabilized, but it points away from the sun by about 160°, the exact pointing error varying slightly within this category of behaviors. In this anti-pointing mode, the satellite cannot charge the batteries. After the batteries are drained, the systems shuts down and the satellite starts to tumble without control. Due to the reduced power consumption with shut off systems, the batteries charge again while the satellite is tumbling, as simulations performed by Nagy [Nag18] show. When the batteries reach a voltage threshold value, the satellite is activated again and it gets another chance to acquire the sun pointing attitude.

In order to evaluate, if the non-nominal behaviors b_2 to b_5 eventually converge to a low pointing error, the initial conditions that produce such behaviors are selected and a simulation run with length 3.5 orbits is performed. Figure 3.6 illustrates the results.

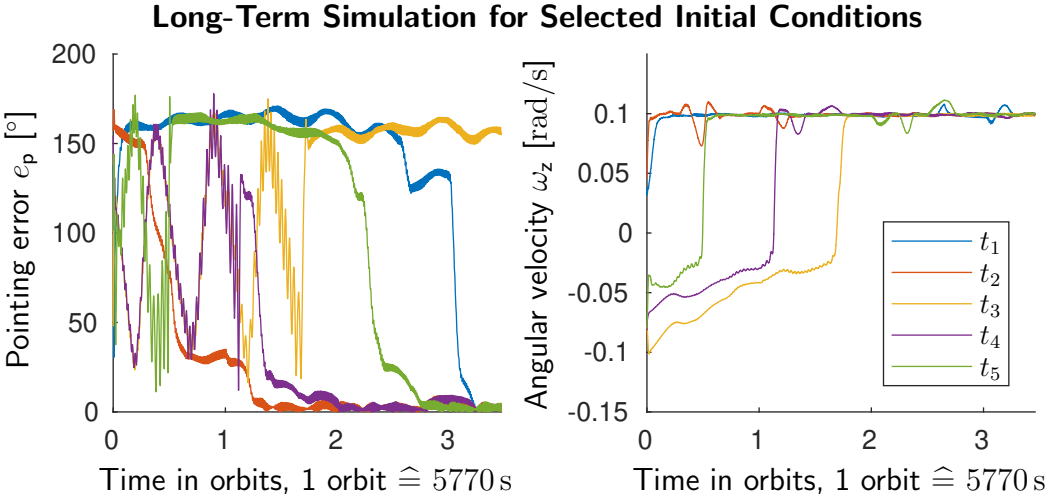

Figure 3.6: Long-term simulation with initial conditions that cause anti-pointing, oscillating behavior or slow convergence with the Default Controller.

Testcase t_1 starts with an initial condition which causes anti-pointing. It can be seen that eventually the satellite changes from anti-pointing to sun pointing. However, this happens only close to the end of the simulation after a time of about 3 orbits.

Testcase t_2 shows a scenario with a slow converging pointing error. One obtains that it can take more than one orbit for the pointing error to converge to less than 10°. Compared to the definition of nominal behavior, this convergence time is nearly four times greater. However, since the pointing error decreases continuously, the light intensity onto the Flappanels and therefore the power generation increases during this time. This increases the chance that the batteries stay charged until the pointing error converged.
The Testcases t_3 to t_5 are examples for the oscillating behavior. One observes that it is possible for an oscillating behavior to end up either in a sun pointing mode, like Testcase t_4, or in an anti-pointing mode, like Testcases t_3 and t_5. For Testcase t_5, the satellite performs a second transition from anti-pointing to sun pointing within the simulation time, whereas for Testcase t_3, the satellite is still stuck in anti-pointing after 3.5 orbits. The time how long the satellite spends in an oscillating mode varies greatly, and during this time the Z component of the angular velocity converges only slowly towards the desired value. However, at some point the satellite acquires the desired angular velocity fast, which is seen in Figure 3.6 as a jump in ω_z, and ends up either in sun pointing or anti-pointing mode.

3.4 Discussion of the Results for the Default Controller

The Default Controller converges to a sun pointing state with a probability of only 49% within half an orbit. Therefore, the controller is not reliable for convergence from an arbitrary initial condition. By adding the probabilities of the nominal behavior and the slow convergence behavior we obtain a probability of 69%. This is the probability for a successful convergence from an arbitrary initial condition. However, the slow convergence runs show a high pointing error for half an orbit and therefore the solar cells generate not much power during this time.
There is a chance with a probability of 24% that the satellite is stuck in an oscillating behavior. This behavior only occurs when the satellite starts with a negative $\omega_{z,0}$. In this case the operation point for ω_z with a positive value is not optimal, since the controller has to reduce the spin of the Z-axis and then generate a spin in the opposite direction.
There is also a chance that the controller ends up in anti-pointing. This means that the batteries almost certainly are drained and the satellite experiences a black out. This state produces less power than a free tumbling satellite which occasionally points the Flappanels with the solar cells towards the sun. Furthermore, additional coil power is consumed while attaining the anti-pointing attitude.
The conclusion is that the Default Controller is not able to converge to a sun pointing state satisfyingly for arbitrary initial conditions. However, in most cases the controller achieves sun pointing eventually. Furthermore, the controller is insensitive to orbit parameter variations, as long as the orbit is far from an equatorial orbit.

Chapter 4

Developing an Extended LQR Controller

This chapter proposes a new controller design for MOVE-II called Extended LQR Controller. The first section proposes a new algorithm for calculating an LQR state feedback gain. New design criteria are presented and the gain matrix for the Extended LQR Controller is defined. A first comparison to the gain of the Default Controller is performed.

Sections 4.2 to 4.5 propose specific extensions to the ADCS which are included in the Extended LQR Controller. Section 4.2 suggests to switch the operation point and invert the spin around the Z-axis, if the initial Z component of the angular velocity is negative. Section 4.3 proposes a magnetic dipole correction method adapted to the ADCS of MOVE-II. Section 4.4 describes how low currents calculated by the controller can be applied to the satellite, considering the hardware limitation of the coils. Section 4.5 summarizes the results of the EKF developed by the ADCS team and how it is used together with the Extended LQR Controller.

Section 4.6 features simulations of the Extended LQR Controller with similar settings as in Section 3.3. The simulation results are compared to those of the Default Controller. Section 4.7 summarizes the results and discusses their impact on the MOVE-II mission.

4.1 New Algorithm for Developing an LQR Gain

In Section 3.2 we state that $\mathbf{K}_{\text{def}}$ cannot be guaranteed to show the properties of a gain designed with the LQR algorithm. Although this gain is stabilizing the satellite, it is advantageous to utilize a gain constructed with the LQR algorithm, due to its insensitivity to disturbances and parameter changes. Therefore a new algorithm is

proposed and elaborated in this section that is used to design a gain with new design criteria and the LQR technique. We use a state weighting matrix

$$\mathbf{Q} = \begin{bmatrix} q_{\mathrm{sx}} & 0 & 0 & 0 & 0 \\ 0 & q_{\mathrm{sy}} & 0 & 0 & 0 \\ 0 & 0 & q_{\omega\mathrm{x}} & 0 & 0 \\ 0 & 0 & 0 & q_{\omega\mathrm{y}} & 0 \\ 0 & 0 & 0 & 0 & q_{\omega\mathrm{z}} \end{bmatrix}, \quad \mathbf{x}_{\mathrm{r}} = \begin{bmatrix} s_{\mathrm{x}} \\ s_{\mathrm{y}} \\ \omega_{\mathrm{x}} \\ \omega_{\mathrm{y}} \\ \omega_{\mathrm{z}} \end{bmatrix}, \quad \mathbf{x}_{\mathrm{r,OP}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \omega_{\mathrm{s}} \end{bmatrix} \tag{4.1}$$

with diagonal weight entries only and the corresponding state vector $\mathbf{x}_{\mathrm{r}}$ whose deviation of the components from the operating point $\mathbf{x}_{\mathrm{r,OP}}$ is to be weighted. The weights define how serious we consider the deviation of each state component with respect to the system requirement. The weights influence the cost functional in the minimization process. It is therefore important to design the weighting matrix carefully with regards to the system requirements, which call for a pointing error of 10° or less as a must requirement and a pointing error of 5° or less as should requirement. These requirements are directly connected with the sun vector components s_{x} and s_{y} in the state vector. They define by how much the pointing is off in the X- and Y-direction respectively. Therefore we like to have high values for the weight components q_{sx} and q_{sy}.
There is no requirement which demands spin stabilization for the satellite, or a certain angular velocity around the Z-axis. However, simulations show that spin stabilization is advantageous for MOVE-II [Kie17]. Hence, we are not interested in a high precision of the value ω_{z} or its settling time, but we want the satellite to reach a spin stabilized state eventually. Thus a low weight $q_{\omega\mathrm{z}}$ is targeted for this component. The remaining two components of the state vector, ω_{x} and ω_{y}, do not contribute directly to these goals. Furthermore, they have to approach zero as the pointing accuracy improves, since an increasing rotation around the X- or Y-axis reduces the pointing accuracy. Since we are not interested in any specific deviation boundary for these two components, and since they are indirectly forced to approach zero, we set the corresponding weights $q_{\omega\mathrm{x}} = q_{\omega\mathrm{y}} = 0$ to zero.
Based on these considerations, we introduce two design parameters

$$\alpha = 5^\circ \quad \text{and} \tag{4.2}$$

$$\beta = 50\% \cdot \omega_{\mathrm{s}}, \tag{4.3}$$

which specify the weight ratio between pointing error and angular velocity for spin stabilization. With the values for α and β we express that a 5° pointing error is as undesirable as a deviation of 50% from the desired spin stabilization rate. These are the only parameters needed for this proposed algorithm, and they may be fine-tuned to redefine the importance between sun pointing and spin keeping.
Since the pointing error is not directly part of the state vector, we have to identify a deviation of the state vector that corresponds to α. The definition of the pointing error in (3.1) shows that the pointing error is completely defined by the Z component

of the sun vector. This component is not part of the reduced state vector $\mathbf{x}_\mathrm{r}$, but it is part of the full state vector $\mathbf{x}_\mathrm{f}$. For this reason it would be helpful to utilize the full system description with the full state vector for designing a new gain matrix. However, when analyzing this system one obtains an uncontrollable subspace with the eigenvalue zero. Thus the system described by the full system description is not stabilizable and we must find a way to express the pointing error in terms of s_x and s_y.
The sun vector is defined as a unit vector, therefore we have

$$\sqrt{s_\mathrm{x}^2 + s_\mathrm{y}^2 + s_\mathrm{z}^2} = 1. \tag{4.4}$$

We make the assumption that whenever we observe a pointing error, this error is related to an equal deviation s_xy in the X- and Y-direction. This serves as a worst case estimate. Combining this assumption with (4.4) and (3.1), we obtain

$$\sqrt{2s_\mathrm{xy}^2 + \cos^2(\alpha)} - 1 = 0, \tag{4.5}$$

which we solve for s_xy. This is the deviation for the components s_x and s_y that should be treated equally as a deviation of β for the component ω_z. The solution to this equation can be found numerically by the bisection algorithm in the interval $[0, 1]$. Similar to the example from Kwakernaak and Sivan [KS72, p. 227], we define a ratio for the weights

$$\sigma_\mathrm{r} = \frac{s_\mathrm{xy}^2}{\beta^2} = \frac{q_\mathrm{s}}{q_{\omega\mathrm{z}}} \tag{4.6}$$

where q_s stands for both q_sx and q_sy. Choosing $q_\mathrm{sx} = q_\mathrm{sy} = \sigma_\mathrm{r}$, $q_{\omega s} = 1$ and reducing the weighting matrix to this three components results in a reduced weighting matrix

$$\mathbf{Q}_\mathrm{r} = \begin{bmatrix} q_\mathrm{sx} & 0 & 0 \\ 0 & q_\mathrm{sy} & 0 \\ 0 & 0 & q_{\omega\mathrm{z}} \end{bmatrix} \tag{4.7}$$

with determinant equal σ_r^2. In order to compare the impact of this weighting matrix to the input weighting matrix $\mathbf{R}$, it is advantageous to keep the determinant of $\mathbf{Q}_\mathrm{r}$ constant at 1, regardless of the tuning parameters α and β. This is done by multiplying all three diagonal entries with the factor

$$\zeta = \sqrt[3]{\frac{1}{\sigma_\mathrm{r}^2}}. \tag{4.8}$$

Therefore, we obtain the weights

$$q_\mathrm{sx} = q_\mathrm{sy} = \sqrt[3]{\sigma_\mathrm{r}} \quad \text{and} \tag{4.9}$$

$$q_{\omega\mathrm{z}} = \sigma_\mathrm{r}^{-\frac{2}{3}}. \tag{4.10}$$

The input weighting matrix $\mathbf{R} = \gamma \mathbf{1}_{3\times 3}$ is again defined as identity matrix, only scaled by a scalar factor γ. With this factor, the influence of both weighting matrices can be shifted. If the generated control inputs exceed their limits, γ must be increased, if the control inputs are too small, γ must be decreased. Simulations show that for MOVE-II, $\gamma = 1$ generates control inputs in the desired range. It is sufficient to use the identity matrix instead of weighting the individual entries of $\mathbf{R}$ differently, since the utilized coils all have the same characteristics.
Using the parameters α, β and γ as defined above and $\omega_\mathrm{s} = 0.1\,\mathrm{rad/s}$, we calculate a feedback gain with the LQR method. After multiplying the gain by the factor 2 for the same reasoning as in Section 3.2, we obtain the LQR gain

$$\mathbf{K}_\mathrm{LQR} = \begin{bmatrix} -1.54 & 1.46 & -31.6 & 3.49\text{e-}1 & 3.15\text{e-}1 \\ -1.52 & -1.59 & 3.71\text{e-}1 & -31.2 & 1.90\text{e-}1 \\ -3.22\text{e-}3 & -1.94\text{e-}2 & 7.49\text{e-}2 & 2.37\text{e-}2 & -1.76 \end{bmatrix}, \tag{4.11}$$

which is the state feedback gain associated with the Extended LQR Controller. For the LQR design, the system matrix $\mathbf{A} = \mathbf{A}_\mathrm{r}$ from (2.28) with true inertia tensor from (2.16) is used. The input matrix $\mathbf{B}$ is obtained in the same way as in Section 3.2, with the difference that again the true inertia tensor instead of the diagonal approximation is used.
The only non-zero eigenvalue of the monodromy matrix for $\mathbf{K}_\mathrm{LQR}$ is $\lambda = 1.06\text{e-}11$, which is an order of 9 magnitudes smaller than for the gain $\mathbf{K}_\mathrm{def}$ from Section 3.2. This is an indicator that the settling time with the LQR gain is lower than with the gain of the Default Controller.
This paragraph analyzes the effect of different spin rates ω_s onto the resulting control gain. Spin rates lower than $0.072\,\mathrm{rad/s}$ result in monodromy matrices with eigenvalues, whose norm is greater than one. Those gains are not stabilizing. From (2.12) and (2.14) it is clear that increasing the spin rate results in an increased nutation torque $\boldsymbol{\tau}_\mathrm{n} = \boldsymbol{\omega}_\mathrm{ib} \times \mathbf{I}\boldsymbol{\omega}_\mathrm{ib}$, as MOVE-II does not spin around a principal axis of inertia. Therefore, the value $\omega_\mathrm{s} = 0.1\,\mathrm{rad/s}$ selected by the then ADCS team is found to be reasonable for MOVE-II.
Figure 4.1 compares the performance of the default gain $\mathbf{K}_\mathrm{def}$ to the LQR gain $\mathbf{K}_\mathrm{LQR}$. The pointing error with the LQR gain settles twice as fast as with the default gain. The default gain is able to maintain a pointing error of less than 10°, but it crosses the 5° line occasionally. With the LQR gain the pointing error stays within the 5° bounds from settling time for nearly one orbit, except for one moment at time $t = 0.28$ orbits, where the controller corrects a large angular velocity error in a short time. It stays within the 10° bounds during this maneuver though.
With regards to the angular velocity, the Default Controller reaches the operation point fast. When using the LQR gain, the ω_x and ω_y components are driven towards zero fast while the ω_z component has a settling time of about 0.6 orbits. The corrections of this component take place stepwise at time points 0.3 and 0.5 orbits. The Default Controller settles fast with respect to the angular velocity and minimizes the pointing error afterwards. With the LQR gain in contrast first the pointing error

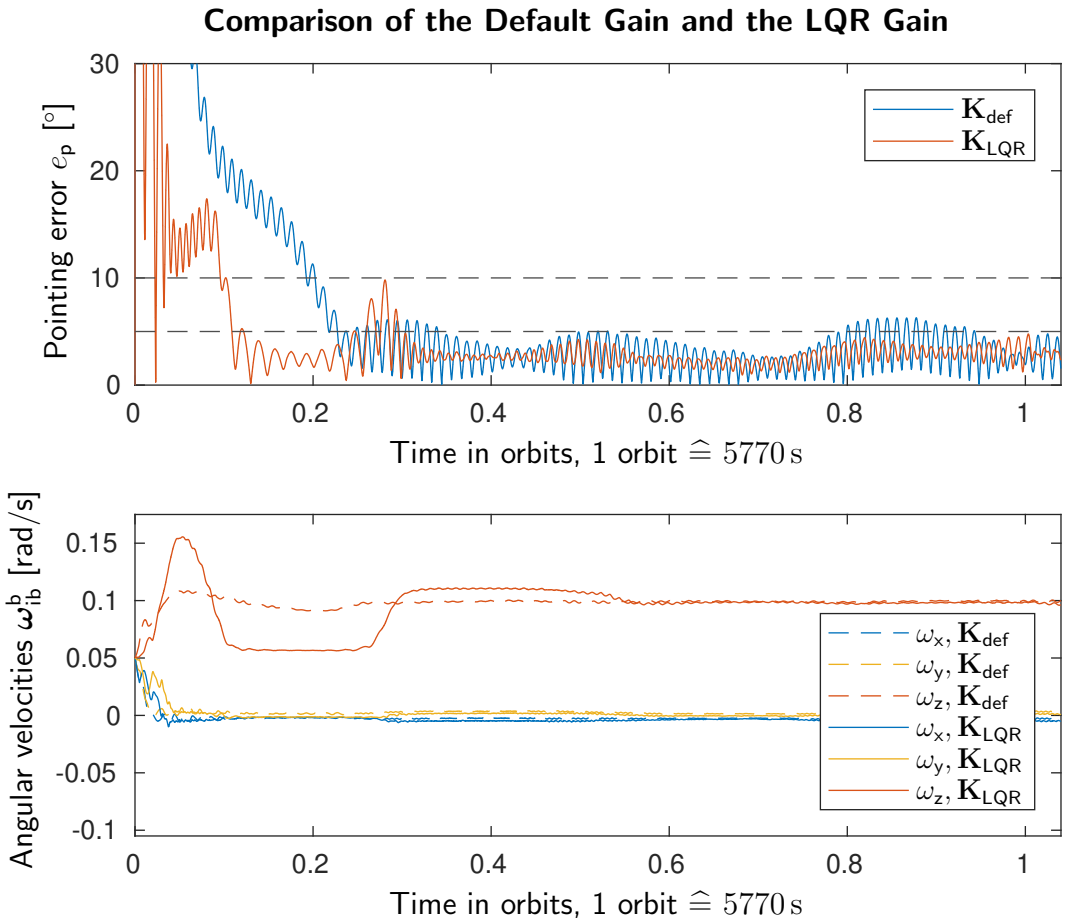

Figure 4.1: Comparing the pointing error and the angular velocities with the default gain and the LQR gain.

is reduced below the must requirement limit. While maintaining the sun pointing, the controller then achieves the desired angular velocity.
Figure 4.2 shows the reason for the stepwise change in the spin rate. Here the Z components of the angular velocity and of the normalized magnetic field

$$\hat{\mathbf{b}}^{\mathrm{b}} = \frac{\mathbf{b}^{\mathrm{b}}}{||\mathbf{b}^{\mathrm{b}}||} \tag{4.12}$$

are compared, both expressed in the BF. In order to drive ω_z towards ω_s, an angular momentum in Z-direction must be generated. Since the required torque is perpendicular to the desired angular momentum and the magnetic field, the controller can only generate such torques, if the Z component of the normalized magnetic field is close to zero. The corresponding regions are highlighted in gray and the change of ω_z becomes big where this is the case.
The negative Z-axis of the satellite is directed towards sun in nominal mode, it rotates with respect to the Earth only slowly. The Z-axis is also the rotation axis

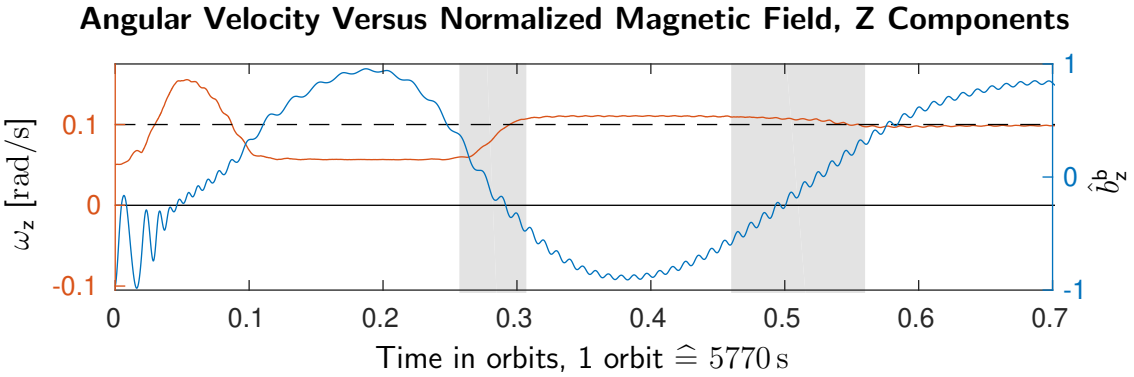

Figure 4.2: Comparing the Z components of the angular velocity and the normalized magnetic field, both expressed in body frame coordinates. The gray areas highlight time spans of special interest.

in nominal mode, therefore the Z component of the magnetic field in the BF is not dependent on the satellite spin and it is periodic with half the orbit period T. This means there are four points per orbit where the Z component becomes zero. This results in four possibilities per orbit for the controller to correct the spin rate, if it deviated from the operating point due to disturbances.

The pointing error, on the other hand, can be controlled at any time. For correction, torques in both X- and Y-directions are necessary. This requires a magnetic field perpendicular to these directions. Since the satellite rotates in the X-Y-plane with angular velocity ω_z, it is ensured that each of those axes observe a perpendicular magnetic field twice per rotation, regardless of the direction of the magnetic field in the ECI.

4.2 Operation Point Switching Strategy

In Section 3.3.2 we observed unstable and oscillating behaviors for the case where the initial value of state ω_z is negative. This is reasonable since the controller is linearized around the positive value ω_s. Furthermore, it needs additional energy to turn around the spin direction around the Z-axis from clockwise to counterclockwise. This applied energy is spent unnecessarily, it provides no benefit if the satellite is spinning clockwise or counterclockwise. Consequently, to save energy and to develop a controller which is able to deal with a broad variety of initial conditions, a control strategy is suggested for spinning around the Z-axis in either clockwise or counterclockwise direction, whatever state is easier to achieve.

In order to realize this, we need to linearize the system around a second operation point

$$x_{r,OP-} = (0, 0, 0, 0, -\omega_s)^T \tag{4.13}$$

with negative target spin. By doing so as described in Section 2.3 and by applying the algorithm described in Section 4.1 using this new operation point, we obtain the gain

$$\mathbf{K}_{\mathrm{LQR}-} = \begin{bmatrix} 1.58 & 1.45 & -31.0 & -3.34\text{e-}1 & -3.55\text{e-}1 \\ -1.51 & 1.53 & -3.59\text{e-}1 & -31.5 & -1.02\text{e-}1 \\ -1.65\text{e-}2 & 1.09\text{e-}2 & -5.37\text{e-}1 & -5.27\text{e-}1 & -1.79 \end{bmatrix}, \tag{4.14}$$

which stabilizes the satellite for a spin around the Z-axis in clockwise direction with magnitude ω_{s}. The Floquet analysis presented in Section 3.1 shows a single non-zero eigenvalue of the monodromy matrix in the same order of magnitude as for the gain $\mathbf{K}_{\mathrm{LQR}}$.
For applying this strategy, we must perform a case differentiation. Depending on the desired direction of rotation, the operation point and the gain must be switched. A simple approach would be to apply the operation point $x_{\mathrm{r,OP}-}$ and gain $\mathbf{K}_{\mathrm{LQR}-}$ when ω_{z} is negative, and switch to $x_{\mathrm{r,OP}}$ and $\mathbf{K}_{\mathrm{LQR}}$ respectively when ω_{z} becomes positive. This case differentiation has one drawback. When ω_{z} is close to zero and the controller improves pointing accuracy or compensates disturbances, ω_{z} could start oscillating between positive and negative values, which would switch from one operating point to its contrary operating point back and forth. With this method it is likely to introduce an oscillating behavior, which we are trying to prevent in the first case.

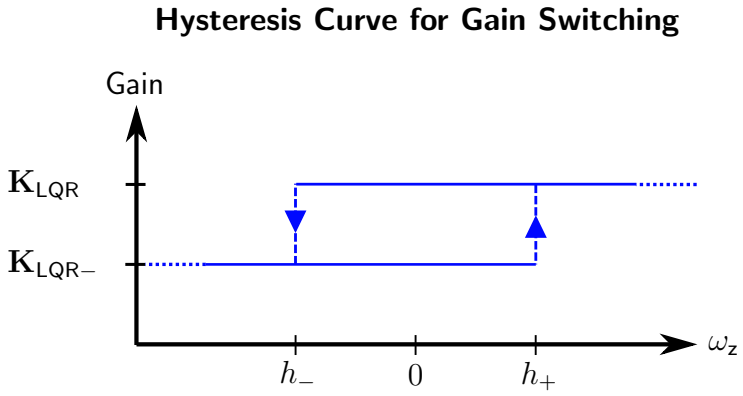

Figure 4.3: This figure illustrates the hysteresis curve of the operation point switching strategy. The blue horizontal lines represent the domains of ω_{z} for the corresponding gain. Whenever ω_{z} leaves the domain of a gain, the gain and the operation point is switched.

To solve this problem, a hysteresis is introduced. We chose the hysteresis values

$$h_{+,-} = \pm 20\% \cdot \omega_{\mathrm{s}} = \pm 0.02\,\mathrm{rad/s}, \tag{4.15}$$

which are determined empirically. This means that whenever ω_{z} crosses either of this values with increasing magnitude, the gain and operation point is switched accordingly, depending on the current sign of ω_{z}. This allows for a margin of 0.04 rad/s

in which the state ω_z can oscillate without switching the gain. Figure 4.3 illustrates this process. This strategy is evaluated together with the other extensions in Section 4.6.2 with a Monte Carlo simulation for various initial conditions.

4.3 Residual Magnetic Dipole Compensation

In Section 2.6.3 we state the residual magnetic dipole moment for MOVE-II measured on Earth, and how to estimate it on-orbit. Knowing an estimation for the dipole moment, one can compensate it when applying the control torques. Huang and Jing [HJ06] describe such a method. This section presents this method and proposes an adaptation which makes it more efficient for the ADCS of MOVE-II.
The authors show that the pointing accuracy can be improved by a factor of two, when subtracting the residual dipole moment $\mathbf{m}_{\text{res}}$ from the desired control moment $\mathbf{m}_{\text{c}}$ and applying the obtained result $\mathbf{m} = \mathbf{m}_{\text{c}} - \mathbf{m}_{\text{res}}$ by commanding the corresponding control currents $\mathbf{i}_{\text{c}}$ to the coils. Even with a model uncertainty of 20%, they show that the pointing accuracy improves significantly. However, satellites which do not actuate constantly over time, but rather apply the control torque in time intervals, cannot fully benefit from this method. In the non-actuation intervals, the residual magnetic dipole moment is not compensated and thus causes a change in rotational velocity of the satellite. To overcome this disadvantage, we propose the following adaptation.
Recalling the duty cycle $D = 50\%$ for MOVE-II presented in Section 2.5, it becomes clear that the correction term of the magnetic dipole applied by the coils acts only for 50% of the time, while the residual magnetic dipole itself acts 100% of the time due to its intrinsic nature. To revert the effect of the residual magnetic dipole moment completely, we must multiply it by the compensation factor

$$C = \frac{1}{D}, \tag{4.16}$$

which is defined by the inverse of the duty cycle D, before we subtract it from the control dipole and apply a resulting commanding dipole moment

$$\mathbf{m} = \mathbf{m}_{\text{c}} - \mathbf{m}_{\text{comp}} = \mathbf{m}_{\text{c}} - C \cdot \mathbf{m}_{\text{res}} \tag{4.17}$$

to the satellite, where $\mathbf{m}_{\text{comp}} = C \cdot \mathbf{m}_{\text{res}}$ is the applied magnetic dipole moment compensation. This means that for MOVE-II the optimal compensation factor is $C = 2$ and that the estimated residual dipole moment is multiplied by 2 before it is subtracted. With this formula it is possible to calculate the optimal correction term for any other duty cycle as well. However, this derivation is based on the assumption that the torque caused by the residual magnetic dipole moment is constant during the actuation-measurement period T_{am}. If the satellite changes its attitude significantly during one period, also the magnetic field in the BF changes significantly, and thus the torque generated by the residual dipole moment is not constant any

more. Therefore, this assumption is fulfilled in the first approximation, if the change of attitude is small during the period T_{am}. The nominal rotation rate of MOVE-II is $\omega_{\mathrm{s}} = 0.1\,\mathrm{rad/s}$, which means that the satellite turns 6° during one period. This is a small angle, thus the assumption is fulfilled in this case.

4.4 Actuation at Low Currents

In Section 2.5 we state that there is a minimum current I_{min} which the hardware can handle. The coils cannot actuate lower currents and will stop actuating, if the commanded current is lower than this value. The Default Controller actuates the control currents $\mathbf{i}_{\mathrm{c}}$ for a fixed period of time T_{a}. However, the firmware allows to actuate a constant current for a shorter period of time. In this section we propose a concept which makes it possible to actuate the equivalent of low currents.
From (2.29) and (2.30) in Section 2.4 we obtain the control torque

$$\boldsymbol{\tau}_{\mathrm{c}} = \mathbf{m}_{\mathrm{c}} \times \mathbf{b} = n \cdot A_{\mathrm{eff}} \cdot \mathbf{i}_{\mathrm{c}} \times \mathbf{b} \tag{4.18}$$

dependent on the control current $\mathbf{i}_{\mathrm{c}}$. From (2.21) in Section 2.3 we see that this control torque affects the derivative of the angular velocity. Let

$$\dot{\boldsymbol{\omega}}_{\mathrm{ib},\boldsymbol{\tau}_{\mathrm{c}}} = \mathbf{I}^{-1}\boldsymbol{\tau}_{\mathrm{c}} \tag{4.19}$$

be the change in angular velocity caused by the control torque only. Then for controlling the satellite we are interested in the difference of angular velocity

$$\Delta\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib},\boldsymbol{\tau}_{\mathrm{c}}} = \int_{T_{\mathrm{a}}} \dot{\boldsymbol{\omega}}^{\mathrm{b}}_{\mathrm{ib},\boldsymbol{\tau}_{\mathrm{c}}}\,\mathrm{d}t \;\approx\; (\mathbf{I}^{\mathrm{b}})^{-1} \cdot n \cdot A_{\mathrm{eff}} \cdot \int_{T_{\mathrm{a}}} \mathbf{i}^{\mathrm{b}}_{\mathrm{c}}\,\mathrm{d}t \times \mathbf{b}^{\mathrm{b}} \tag{4.20}$$

during one actuation period caused by the control torque, where the approximately equal sign holds for a slow spinning satellite and thus the magnetic field $\mathbf{b}^{\mathrm{b}}$ expressed in the BF is assumed to be constant during the time over the integral. We make this assumption already in Section 4.3.
Let $i_{\mathrm{c},j}$ be a component of the current vector $\mathbf{i}_{\mathrm{c}}$ which corresponds to a specific coil. Then for every $i_{\mathrm{c},j}$ of the three actuator coils it is checked, if the current is less than I_{min}. If so, the coil is actuated with the current I_{min} and a shorter period of time $\Delta t_{\mathrm{c},j}$ instead. To achieve the same effect in change of angular velocity as with the lower current, the integrals

$$\int_{T_{\mathrm{a}}} i_{\mathrm{c},j}\,\mathrm{d}t = i_{\mathrm{c},j} \cdot T_{\mathrm{a}} \overset{!}{=} \int_{\Delta t_{\mathrm{c},j}} I_{\mathrm{min}}\,\mathrm{d}t = I_{\mathrm{min}} \cdot \Delta t_{\mathrm{c},j} \quad \forall j = 1,2,3 \tag{4.21}$$

must be constant. The actuation time for currents below I_{min} therefore is

$$\Delta t_{\mathrm{c},j} = \frac{i_{\mathrm{c},j} \cdot T_{\mathrm{a}}}{I_{\mathrm{min}}}. \tag{4.22}$$

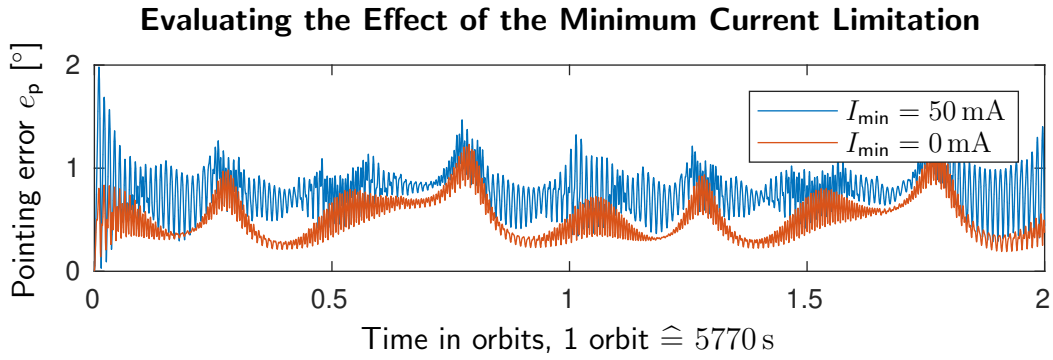

Figure 4.4: Two long-term simulations starting from optimal initial conditions evaluating the effect of the minimum current limitation.

To simulate this proposed extension, the simulation value for $I_{\min}$ is set to zero. Figure 4.4 shows the comparison between two simulation runs, one applying this proposed extension, and the other one without, starting from optimal initial conditions. One obtains that with $I_{\min} = 0$, there is less overshooting and less fluctuation of the pointing error. The mean pointing error with this extension is $\mu_{\text{p,on}} = 0.5°$ and without $\mu_{\text{p,off}} = 0.8°$, while the variance $\sigma_{\text{p}}^2 = 0.04\,\text{deg}^2$ is equal for both simulation runs. The mean pointing error decreases by 0.3°, which is greater in numbers than the standard deviation.

4.5 Utilizing the Extended Kalman Filter

The Default Controller relies solely on the instantaneous sun sensor and magnetometer measurements in order to obtain information about the sun vector and the Earth magnetic field. This means the Default Controller has to deal with the full unfiltered sensor noise. Also, it does not utilize sensor fusion. The most serious disadvantage however is, that during eclipse no sun vector information is available. The Extended LQR Controller shall utilize the Extended Kalman Filter (EKF) developed by the ADCS team to determine its attitude in order to solve these issues. Meßmann [Meß18] provides a thoroughly analysis of the performance of the EKF. According to Meßmann, the mean absolute difference of the attitude estimation error amounts to 0.16° for a realistic testrun. He also states the necessary conditions for the EKF to achieve a low estimation error.
For the EKF to function as expected it is important that the magnetometer and the sun sensors are calibrated properly. Uncalibrated sensors lead to a large error. Calibration is performed thoroughly on ground before the launch of MOVE-II by the ADCS team. Additionally there is the possibility to recalibrate the sensors on-orbit.
It is needless to say that the sensors also must be functional. In case of a faulty sun sensor, this sensor can be deactivated manually to not disturb the EKF with

faulty measurement data. If the faulty sun sensor is located at the Sidepanel and the sun faces this panel, no sun vector information is available. Since this is not an operation point, the absence of a sun vector measurement is only temporary and the EKF is able to interpolate the attitude during this time with sensor fusion from the magnetometer data. A faulty sun sensor located at the Toppanel however is critical for the mission. Since this is the operation point, it is not expected that the sun sensors located at the Sidepanels provide information about the sun position. They are overshadowed by the Flappanels in this case. If the magnetometer is faulty, another one of the five redundant magnetometers can be enabled manually during the next overpass of MOVE-II.

It is also important for the functionality of the EKF that the realtime clock of MOVE-II does not diverge too fast. The clock time can be updated on an overpass of the satellite, thus this will not be an issue if a stable communication link can be established.

Another important factor is the initialization of the EKF. If no sun vector is available during initialization and the initial estimated attitude is far from the real attitude, the EKF can become unstable.

If all these conditions are fulfilled, namely functional and calibrated sensors, an accurate on-board real-time clock and a proper initialization of the EKF, it provides an accurate attitude information through sensor fusion. More than that, the ADCS has access to the sun vector information during eclipse. In Section 3.3.1 we obtain an increasing pointing error during eclipse. Utilizing the EKF, we expect the pointing error not to increase during eclipse. The attitude information is provided by the EKF in terms of a unit quaternion calculated by sensor fusion from the magnetometer and sun sensor measurements as inputs. The state vector contains the sun vector, but not the unit quaternion. Therefore the sun vector has to be calculated from the unit quaternion provided by the EKF. This is performed by a sun position model, which is already part of the firmware implementation. The EKF also estimates the angular velocity of the satellite and updates the bias calibration of the gyroscopes.

4.6 Simulation Results of the Extended LQR Controller

This section evaluates the Extended LQR Controller with respect to orbit changes, random initial conditions and under the presence of different residual magnetic dipole moments. It is also evaluated to what extend the magnetic dipole compensation is suitable, and general limitations of the Extended LQR Controller are shown. Whenever appropriate, a comparison is made to the Default Controller.

4.6.1 Sensitivity to Orbit Changes

This Section evaluates the Extended LQR Controller with gain $\mathbf{K}_{\mathrm{LQR}}$ and all its proposed extensions in different orbits. The orbit parameters and other simulation settings are the same as in Section 3.3.1, which allows to compare the results from the Extended LQR Controller to the results of the Default Controller.

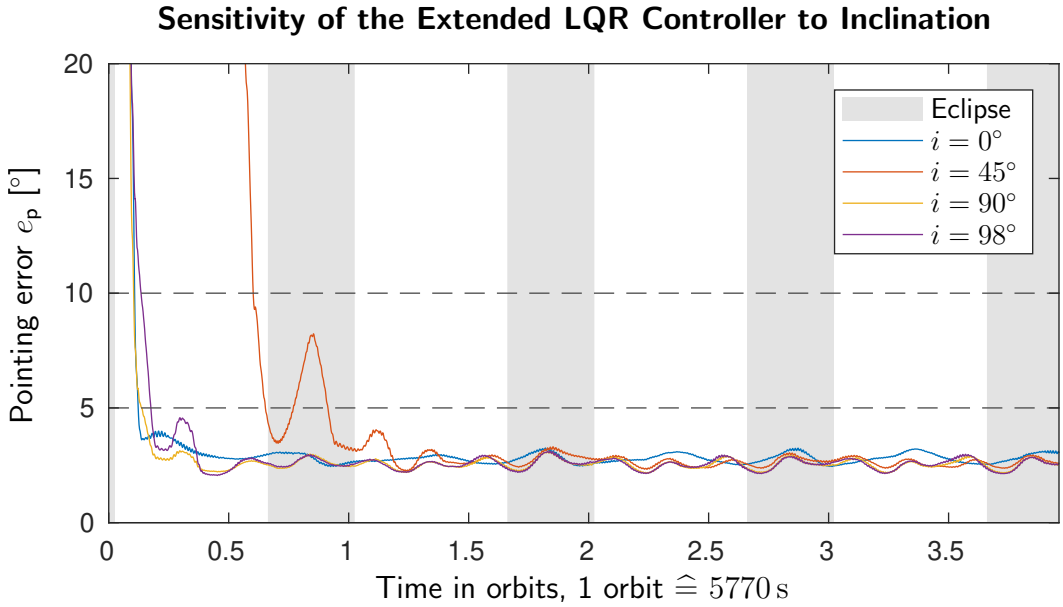

Figure 4.5: Pointing error achieved with the Extended LQR Controller for differently inclined orbits. The graphs are smoothed with a moving average filter.

Figure 4.5 shows the pointing error with the Extended LQR Controller in orbits of different inclinations. The eclipse time is again illustrated with gray areas. As for the results of the Default Controller, the results of this and the next figure are smoothed with a moving average filter of window size 5e3, while the data is sampled at 10 Hz. One obtains that the eclipse has no negative effect any more on the pointing error. The use of the EKF is mainly responsible for this by providing the controller with sun position information even during eclipse. The results also show that the pointing error converges fast for a broad spectrum of inclinations. Only the orbit with inclination of 45° needs more than four times as long as the others to converge, but it does converge eventually. The average pointing error after convergence for all four simulations is in the range of 2.6° to 2.8°, with variances in the range of 7e-3 deg^2 to 8e-3 deg^2.
In Figure 4.6 we see the results for simulations with different orbit heights and shapes. Again, the parameters chosen are the same than in the simulations for the Default Controller in Section 3.3.1. The results show that the Extended LQR Controller is not sensitive to large orbit size or shape changes. In every of these simulations the average pointing error converges to a value of less than 2.6° with a variance of about 6e-3 deg^2. Only the highly elliptic orbit with eccentricity $e = 2/3$

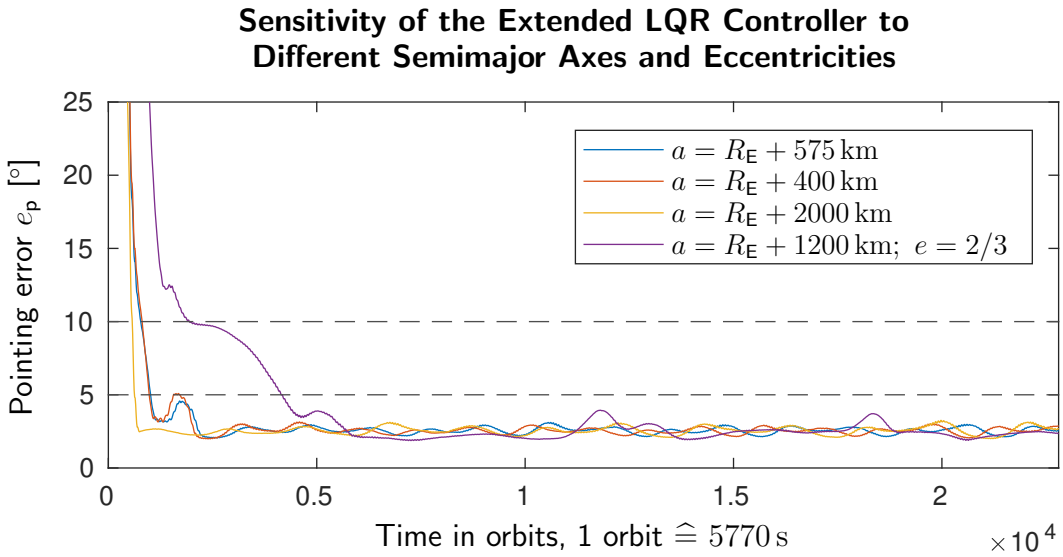

Figure 4.6: Pointing error achieved with the Extended LQR Controller for different orbit sizes and shapes. The graphs are smoothed with a moving average filter.

seems to have minor difficulties. The convergence time to a pointing error below 5° for this orbit is four times greater than for the other orbits, and there occur small spikes periodically. However, this is not reflected in the mean pointing error.

4.6.2 Sensitivity to Initial Values

In the following we compare the sensitivity of the Extended LQR Controller to the Default Controller with respect to initial values. The setup of the Monte Carlo simulation is the same as described in Section 3.3.2, including the range of the varied parameters and the number of runs, which is $n = 100$. The magnetic dipole moment, however, is overcompensated by 20% to simulate the inaccuracy of the dipole estimation on-orbit. Therefore, we have $\mathbf{m}_{\mathrm{comp}} = 1.2 \cdot \mathbf{m}_{\mathrm{res}}$. The simulation length is 2500 s, which is approximately 42 min or 0.43 orbits.
Figure 4.7 illustrates four different behaviors, which can be observed in the Monte Carlo simulation. Table 4.1 categorizes each observed behavior and states their frequentness.

Behavior	Description	Times Observed	In Percent
$b_{\mathrm{E},1}$	Nominal behavior	49	49%
$b_{\mathrm{E},2}$ & $b_{\mathrm{E},3}$	Slow convergence	27 & 4	31%
$b_{\mathrm{E},4}$	Unstable behavior	20	20%

Table 4.1: Frequentness analysis of the four different outcome categories in the Monte Carlo simulation with $n = 100$ runs for the Extended LQR Controller.

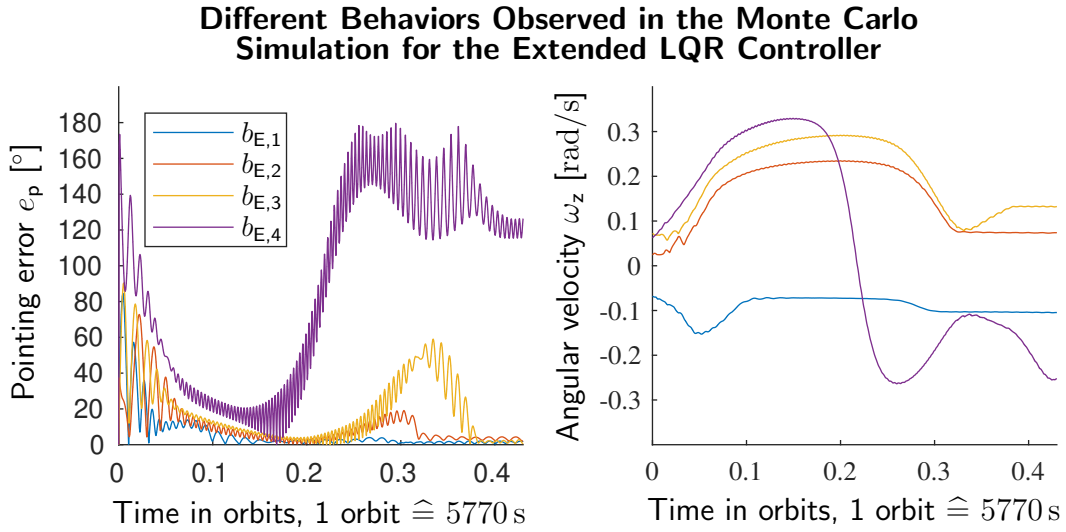

Figure 4.7: Results of the Monte Carlo simulation for the Extended LQR Controller show four different possible outcomes.

Behavior $b_{E,1}$ is the desired nominal behavior. The pointing error converges fast, stays low and the angular velocity in Z-direction converges to one of the two operating points. The categorization of this behavior is according to the definition used in Section 3.3.2. Therefore, a testrun is only categorized as nominal behavior, when the pointing error falls below 10° within a time of 1500 s, or 0.26 orbits, and does not exceed this boundary afterwards.

Thus, behavior $b_{E,2}$ is not categorized as nominal behavior, since it exceeds the 10° boundary after 0.3 orbits again. This observed behavior belongs to the settling time, because the state component ω_z reaches the operation point only afterwards. When comparing the pointing error of $b_{E,2}$ to the change in the Z component of the angular velocity, one obtains that this distortion appears when the controller minimizes the magnitude of the Z component of the angular velocity until it reaches its operation point. This spike in the pointing error usually is not large. In order to be classified as $b_{E,2}$, the spike must be below 20°.

Occasionally, however, the spike can become large and exceed this boundary as well. All those runs are categorized as behavior $b_{E,3}$. The illustration of behavior $b_{E,3}$ in Figure 4.7 is the largest distortion that is observed within these 100 simulation runs. Both observed behaviors $b_{E,2}$ and $b_{E,3}$ are categorized as slow convergence, since the settling time is long, but eventually they reach the desired operating point.

This is not the case for behavior $b_{E,4}$, at least not within half an orbit. One observes that the pointing error is converging for the first 750 s, but then the pointing error increases in a sudden movement up to nearly 180°. Simultaneously the spin rate of the satellite changes its sign. Figure 4.8 depicts all components of the angular velocity together with the control torque $\boldsymbol{\tau}_c$ and the nutation torque $\boldsymbol{\tau}_n$. One observes that the nutation torque grows until it exceeds the magnitude of the control torque.

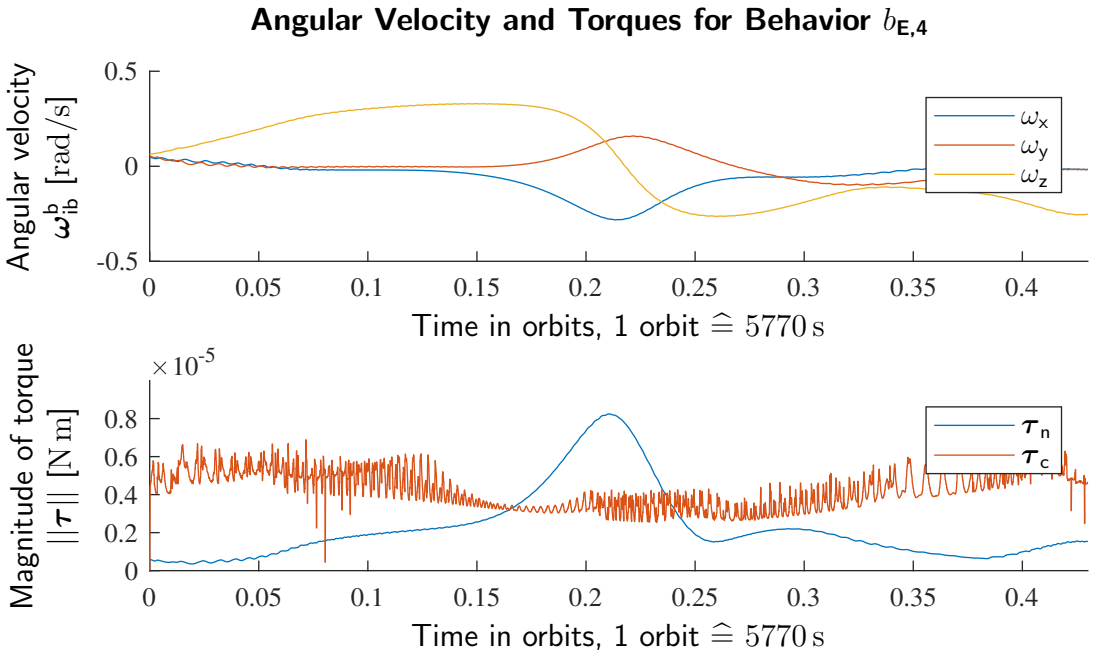

Figure 4.8: Comparing the components of the angular velocity, the control torque $\boldsymbol{\tau}_c$ and the nutation torque $\boldsymbol{\tau}_n$ of one of the outcomes of the unstable behavior category $b_{E,4}$.

The simulation data shows that this is the maximum possible control torque the satellite is able to generate at that moment, since at least one coil is limited by the maximum current. This means that the nutation torque becomes greater than the maximum possible control torque, caused by the high spin rate around the Z-axis. This effect only occurs, because MOVE-II spins around an unstable, intermediate principal axis. The sudden 180° flip of the angular velocity vector around the other two non-spinning axes of the satellite, like it is observed in Figure 4.8, is a telltale sign for unstable behavior caused by spinning around the intermediate principle axis of inertia.

Comparing the outcomes to the results of the Default Controller, one obtains that about the same amount of simulation runs show a nominal behavior, this is 49% in both cases. However, the sum of nominal behavior and slow convergence increased with the Extended LQR Controller from 69% to 80%.

In Figure 4.9 the performance of all nominal behavior runs of both controllers are compared. Part (a) illustrates the results for the pointing error with the Default Controller, part (b) with the Extended LQR Controller. Part (c) illustrates the Z component of the angular velocity for the corresponding simulation runs with the Extended LQR Controller.

One obtains that the Extended LQR Controller has a significant shorter convergence time than the Default Controller. After 12.5 min, the must requirement for the pointing error to be below 10° is fulfilled by all nominal behavior runs with the

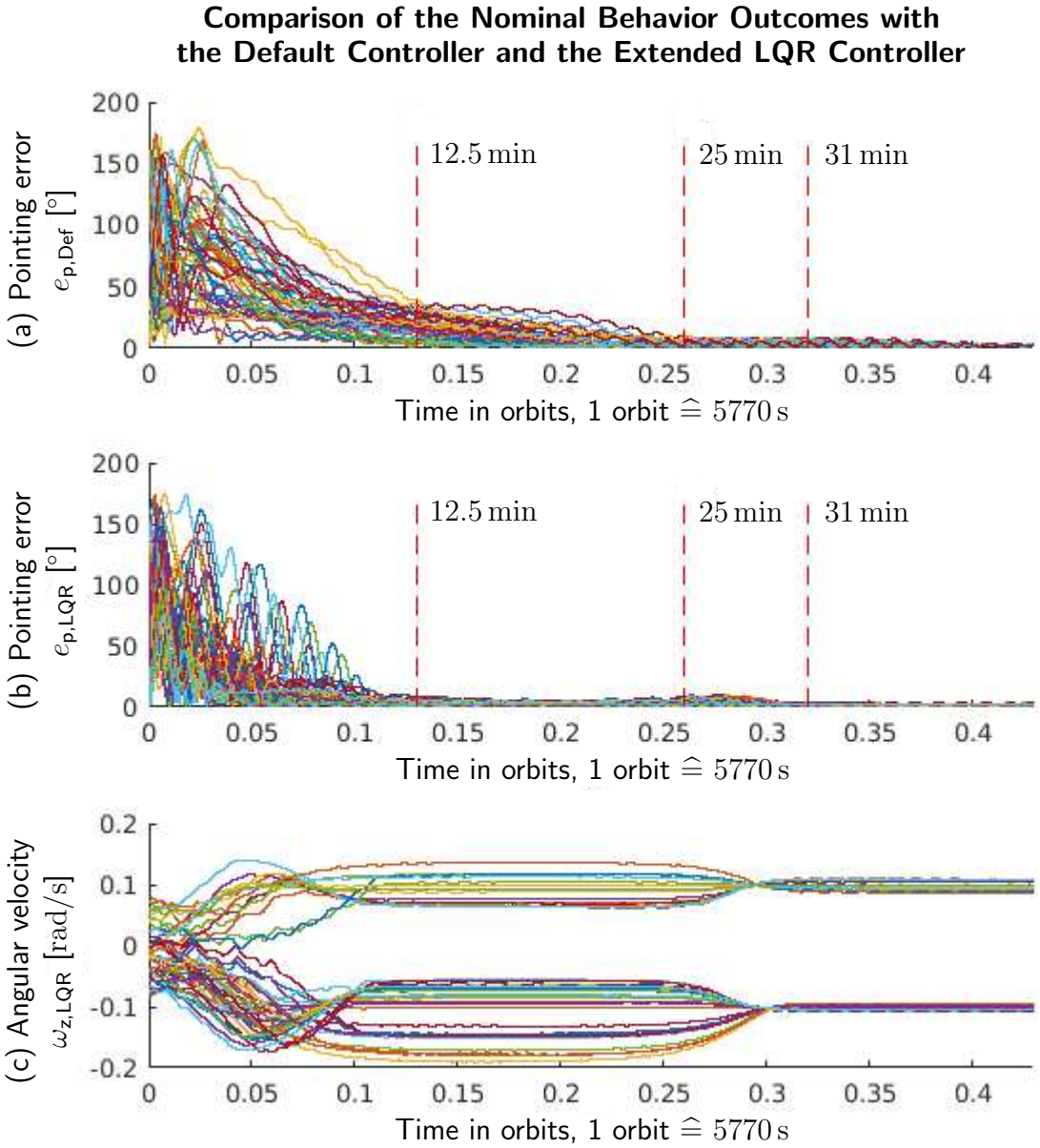

Figure 4.9: Comparison of the nominal behavior runs from the Monte Carlo simulations with the Default Controller and the Extended LQR Controller. Part (a) shows the pointing error with the Default Controller, part (b) with the Extended LQR Controller. Part (c) shows the Z component of the angular velocity with the Extended LQR Controller.

Extended LQR Controller, whereas for the Default Controller many runs have a pointing error between 20° and 50° at the same time. Only after 25 min, all simulation runs with the Default Controller fall below a pointing error of 10°. Therefore,

the Extended LQR Controller converges approximately twice as fast as the Default Controller in the nominal behavior category.

Another result obtained is that eventually after 31 min all of the simulation runs with the Extended LQR Controller fall below the 5° pointing error boundary for all nominal behavior runs, which is a should requirement for MOVE-II. It is observed that with the Default Controller the maximum pointing error among all simulation runs within the nominal behavior category falls below 10° after 31 min only.

To analyze this in more detail, statistics are calculated. For every individual run, the average pointing error μ_p and its variance σ^2_p are calculated within different time intervals. The mean pointing error $\mu_{\mathrm{p},n}$ is the mean over the number $n = 49$ of all those average pointing errors in the nominal behavior category, and $\sigma^2_{\mu_{\mathrm{p},n}}$ is its variance. The mean variance $\mu_{\sigma^2_\mathrm{p}}$ is the mean value of all the variances σ^2_p from the individual simulation runs. It is a measure of how much the instantaneous pointing error is fluctuating in average. Table 4.2 presents the results.

Controller	Statistic	$12.5 - 42$ min	$25 - 42$ min	$31 - 42$ min
Default Controller	$\mu_{\mathrm{p},n}$ [°]	5.5	3.3	3.0
	$\sigma^2_{\mu_{\mathrm{p},n}}$ [deg^2]	6.6	0.37	0.14
	$\mu_{\sigma^2_\mathrm{p}}$ [deg^2]	27	2.1	1.7
Extended LQR Controller	$\mu_{\mathrm{p},n}$ [°]	2.9	2.7	2.3
	$\sigma^2_{\mu_{\mathrm{p},n}}$ [deg^2]	0.32	0.14	0.11
	$\mu_{\sigma^2_\mathrm{p}}$ [deg^2]	1.5	1.5	0.075

Table 4.2: Statistics for the Default Controller and the Extended LQR Controller over multiple different time spans.

One obtains that the Default Controller settles after 25 min. The statistics do not change significantly afterwards. Before this time however, one obtains from the high values for the variance and mean variance that the mean pointing error and the instantaneous pointing errors respectively are varying strongly.

The Extended LQR Controller shows a convergence after 12.5 min already. The statistics do not change significantly in the intervals afterwards and the mean pointing error, its variance and the mean variance are low. The only exception is the mean variance in the time interval $31 - 42$ min. This value is reduced significantly after the 31 min mark. This means that the fluctuation of the pointing error is reduced significantly and the satellite does not wobble as much as with the Default Controller. The confidence interval $\mu_{\mathrm{p},n} \pm 2\sigma_{\mu_{\mathrm{p},n}}$ with confidence level 95% shows that the mean pointing error lies between 1.3° and 3.0° for the Extended LQR Controller. The mean variance shows that the instantanous pointing error fluctuates with ±0.55° for the same confidence level.

For the Default Controller, the confidence interval of the mean pointing error ranges from 2.3° to 3.7°, and the instantanous pointing error fluctuates with ±1.3°. We conclude that the Extended LQR Controller is able to reduce both the mean pointing error and the pointing error fluctuation significantly.
However, this performance improvement comes at a cost of increased power consumption. In the time interval $12.5 - 42$ min, the coils consume 27 mW in average, when using the Default Controller, whereas with the Extended LQR Controller, the coils need 32 mW. This is an increased power consumption of 19%. Also, the fast convergence time comes at a cost. In the first 12.5 min of the simulation, the power consumption of the coils with the Extended LQR Controller amounts to 248 mW in average, compared to 156 mW with the Default Controller. This is an increase of 59%.
Part (c) of Figure 4.9 shows that the operation point switching strategy is working as expected. When the initial Z component of the angular velocity lies below 0, the negative operation point is targeted, otherwise the positive. Both operating points are stable, since for each operation point a unique stabilizing gain is used.

4.6.3 Evaluating the Magnetic Dipole Compensation

This section evaluates the influence of the magnitude and direction of the residual magnetic dipole moment with and without the compensation technique presented in Section 4.3. This method is analyzed in detail using the Default Controller with gain $\mathbf{K}_{\text{def}}$. It is also verified in operation with the entire Extended LQR Controller, including all extensions proposed in Chapter 4. Furthermore, we state the effectiveness and investigate the limitations of this method.
The following simulations use the Default Controller, extended with the EKF and $I_{\min}$ set to zero. Figure 4.10 (a) shows the pointing error dependent on the magnitude of the residual dipole without compensation. The dipole takes magnitudes in the range of 0 to 10 times the residual dipole moment evaluated in Section 2.6.3, while retaining its direction. A moving average filter over 10e3 data points, which are sampled at 5 Hz, is applied to the results in order to smooth out the oscillations caused by the satellite spinning around its pointing axis. This filter is applied to all figures in this section. This helps in understanding the long-term behavior of the pointing error. In Figure 4.10 (a), the time points when the satellite passes by close to the Earth magnetic poles are marked with vertical gray lines.
One obtains a significant influence of the dipole magnitude onto the pointing error. All statistics in this section are calculated from the settling time $t_{\text{s}} = 0.5$ orbits until the end of simulation $t_{\text{E}} = 2.6$ orbits. While the pointing error without residual dipole converges to a mean error $\mu_{\text{p},0} = 2.3°$ with a variance of $\sigma^2_{\text{p},0} = 0.9\,\text{deg}^2$, with a dipole magnitude of $55\,\text{mA}\,\text{m}^2$ the mean pointing error increases to $\mu_{\text{p},55} = 22°$ with a variance of $\sigma^2_{\text{p},55} = 62\,\text{deg}^2$. With higher dipole magnitudes, the controller becomes unstable and is not able to align itself towards the sun. Another observation is the fact that the pointing error becomes smaller for dipole magni-

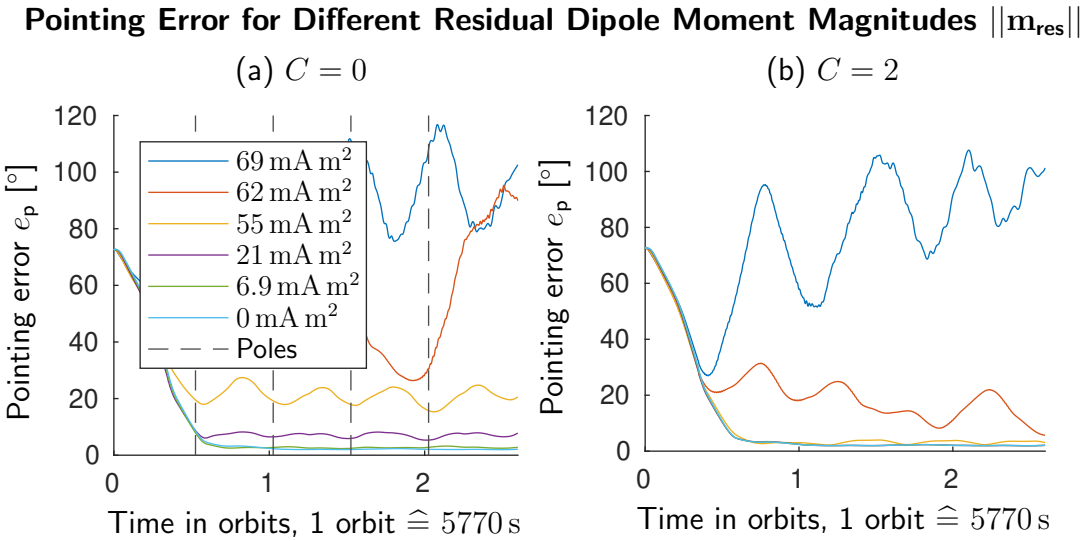

Figure 4.10: Pointing error with respect to different residual magnetic dipole moment magnitudes $||\mathbf{m}_{\text{res}}||$, (a) without and (b) with compensation. The graphs are smoothed using a moving average filter.

tudes $||\mathbf{m}_{\text{res}}|| \leq 55\,\text{mA}\,\text{m}^2$ when passing by close to the poles of Earth. The residual dipole moment of MOVE-II affects the pointing error stronger when close to the equator than close to the poles. We see furthermore that the mean pointing error for $||\mathbf{m}_{\text{res}}|| = 6.9\,\text{mA}\,\text{m}^2$ with $\mu_{\text{p},6.9} = 2.7°$ is similarly low compared to the case without a residual dipole, though the corresponding variance $\sigma^2_{\text{p},6.9} = 1.8\,\text{deg}^2$ is doubled.

Figure 4.10 (b) shows the pointing error dependent on the magnitudes of the residual dipole with the compensation activated and the for MOVE-II optimal compensation factor $C = 2$. The same values for the dipole magnitudes as in Figure 4.10 (a) and the same legend is used. Dipoles with magnitudes of up to $55\,\text{mA}\,\text{m}^2$ can be compensated. For low dipole magnitudes, the mean pointing error does not increase significantly. For example, with $||\mathbf{m}_{\text{res}}|| = 21\,\text{mA}\,\text{m}^2$ we obtain a mean pointing error of $\mu_{\text{p},21,\text{c}} = 2.6°$ with a variance of $\sigma^2_{\text{p},21,\text{c}} = 0.9\,\text{deg}^2$. The variance does not increase and the mean pointing error increases only slightly compared to the uncompensated case without a residual dipole. For $||\mathbf{m}_{\text{res}}|| = 55\,\text{mA}\,\text{m}^2$ we obtain a slightly higher mean pointing error $\mu_{\text{p},55,\text{c}} = 3.1°$ with a variance of $\sigma^2_{\text{p},55,\text{c}} = 3\,\text{deg}^2$. The compensation method reaches a limit for a dipole magnitude of $62\,\text{mA}\,\text{m}^2$. We obtain a mean pointing error of $\mu_{\text{p},62,\text{c}} = 16°$ with a variance of $\sigma^2_{\text{p},62,\text{c}} = 127\,\text{deg}^2$. However, the figure shows a converging trend in this case, whereas the same dipole magnitude causes the controller to become unstable in the uncompensated case. Increasing the dipole magnitude further, the controller becomes unstable even with the compensation method.

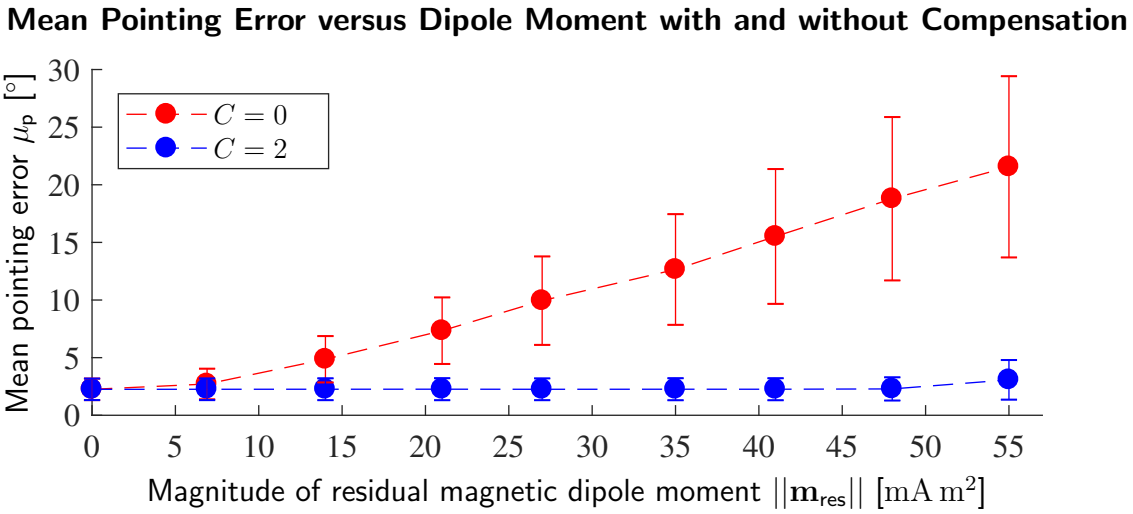

Figure 4.11: Comparing the mean pointing error and the corresponding standard deviations with the compensation method to the uncompensated case with respect to the magnitude of the residual magnetic dipole moment.

Figure 4.11 summarizes these results and adds more data points from simulation runs with dipole magnitudes in the range of 0 to 55 mA m^2. The mean pointing error is plotted against the dipole magnitude for both cases with and without compensation. The error bars visualize the standard deviation of the pointing error. One obtains that small dipole magnitudes have no significant impact on the mean pointing error, but starting from a dipole magnitude of 7 mA m^2, the mean pointing error increases linearly with the dipole magnitude, if not compensated. Using the proposed compensation technique however, the dipole moment can be compensated up to a value of 55 mA m^2, which means that both the mean pointing error and the standard deviation do not change significantly compared to the case without a residual dipole up to this value.

These results come at a cost however. The actuation coils of the ADCS need more current to compensate the dipole, which increases the power consumption of the satellite. Figure 4.12 shows the total power consumption P_c of all three actuator coils for varying dipole magnitudes for both cases, compensation deactivated and activated. These results are also filtered by a moving average filter over 3e3 data points which are sampled at a frequency of 10 Hz. One obtains that with an increased dipole magnitude also the power consumption increases, even when compensation is not active. With active compensation however, the power consumption increases significantly. The coil power is calculated assuming a coil resistance of $R = 13\,\Omega$ and a coil driver loss of 30%. The control currents are increased accordingly, to compensate for the loss of the coil driver. Taking this into account and the maximum control current $I_{max} = 0.3$ A for each coil as stated in Section 2.5, this yields a maximum possible effective power consumption of $P_{c,max} = 1.7$ W per coil.

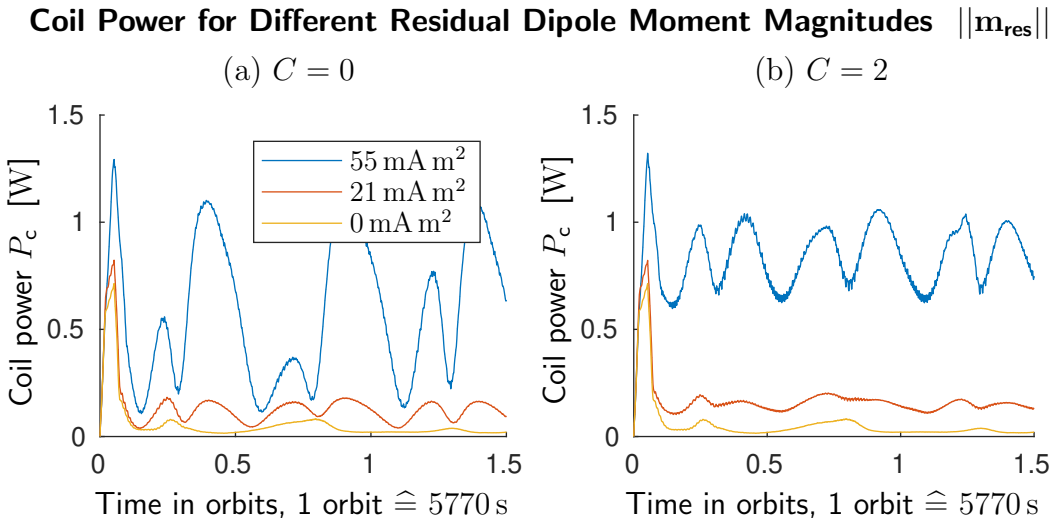

Figure 4.12: Power consumption of the actuator coils with respect to different residual magnetic dipole moment magnitudes $||\mathbf{m}_{\text{res}}||$, (a) without and (b) with compensation. The graphs are smoothed with a moving average filter.

C	Stat.	$0\,\text{mA}\,\text{m}^2$	$6.9\,\text{mA}\,\text{m}^2$	$14\,\text{mA}\,\text{m}^2$	$21\,\text{mA}\,\text{m}^2$	$27\,\text{mA}\,\text{m}^2$
0	μ_{P_c} [mW]	32.7	41.4	68.9	115	178
	Increase wrt. $\|\|\mathbf{m}_{\text{res}}\|\| = 0$	0%	26.6%	111%	251%	444%
2	μ_{P_c} [mW]	32.7	45.9	85.1	150	241
	Increase wrt. $C = 0$	0%	8.70%	23.5%	31.0%	35.6%

Table 4.3: Mean power consumption of the coils without and with the compensation method for different magnitudes of the residual dipole moment $\mathbf{m}_{\text{res}}$.

Table 4.3 gives an overview over the mean power consumption μ_{P_c} measured for different dipole magnitudes. The first line gives the mean power consumption in mW without compensation, dependent on the dipole magnitude. The second line compares each measurement to the value without residual dipole and without compensation in the first row and first column. This provides information about how strong an increase in the dipole magnitude affects the mean power consumption. While the dipole magnitude increases linearly, the mean power consumption increases stronger. The third row repeats the measurements with dipole compensation activated. The fourth row compares each measurement with the corresponding value in row one. This provides information about the additional cost of the compensation method, when the dipole magnitude is fixed. For the estimated dipole of MOVE-II with magnitude $6.9\,\text{mA}\,\text{m}^2$, this means the dipole causes the actuator coils to increase

their mean power consumption by 26.6%. Activating the compensation method, the mean power consumption is additionally increased by 8.7%.

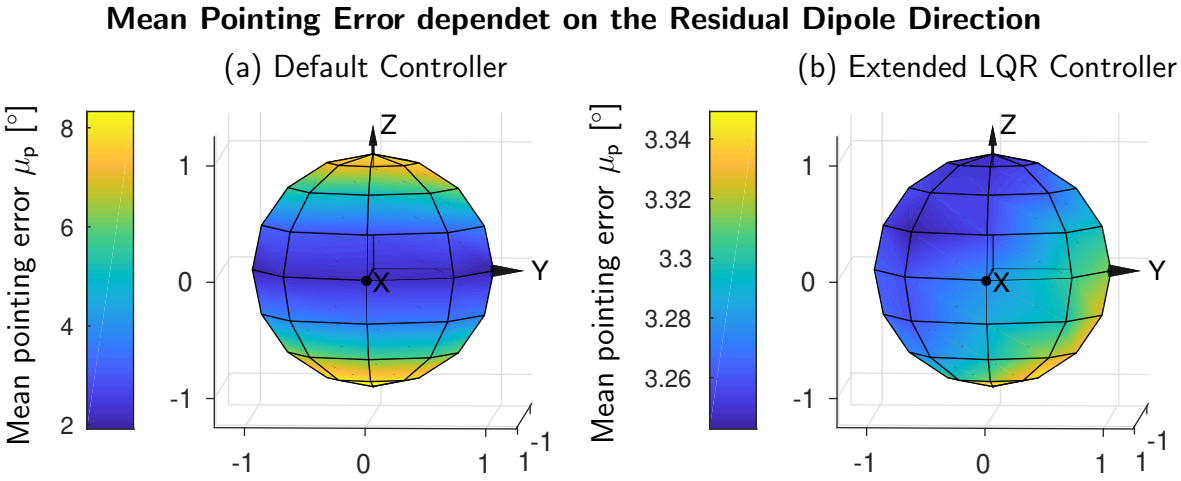

Figure 4.13: The mean pointing error is evaluated for different directions of the residual magnetic dipole for both (a) the Default Controller and (b) the Extended LQR Controller. The result is plotted on the sphere at the direction of the dipole in BF coordinates.

Figure 4.13 compares the impact of different directions of the residual dipole onto the mean pointing error. Both controllers, the Default Controller and the Extended LQR Controller, are evaluated. In total, 58 different simulations are performed per controller, with the dipole direction varying around the unit sphere. The magnitude of the dipole is set to $||\mathbf{m}_{\text{res}}|| = 6.9\,\text{mA}\,\text{m}^2$, as evaluated for MOVE-II. The simulation lengths is 2500 s, and the pointing error is averaged over the complete simulation time. Optimal initial conditions are applied, meaning that the initial pointing error is zero and the spin rate is $\omega_z = \omega_s$. Thus, the settling time is zero in these cases. The results show that the mean pointing error with the Default Controller is strongly dependent on the direction of the dipole. With a stronger Z component of the dipole, the pointing error increases. It ranges from 2° in the X-Y-plane to 8° along the Z-axis. In contrast, with the Extended LQR Controller the mean pointing error does not significantly vary with the direction of the dipole.

In the following paragraphs we show that the in Section 4.3 proposed compensation factor of $C = 2$ is indeed optimal for MOVE-II. Figure 4.14 compares the pointing error and the coil power to different compensation factors, simulating with a residual dipole of magnitude $||\mathbf{m}_{\text{res}}|| = 35\,\text{mA}\,\text{m}^2$. The simulation, which converges towards the lowest pointing error, is in fact the simulation with $C = 2$. The exact values of the mean pointing error and its variance can be found in Table 4.4. Figure 4.14 (b) shows that a higher compensation factor results in a higher power consumption for the simulated setting. One obtains that there periodically appear points, where the curves overlap. This means that at these points, the dipole has no effect on the

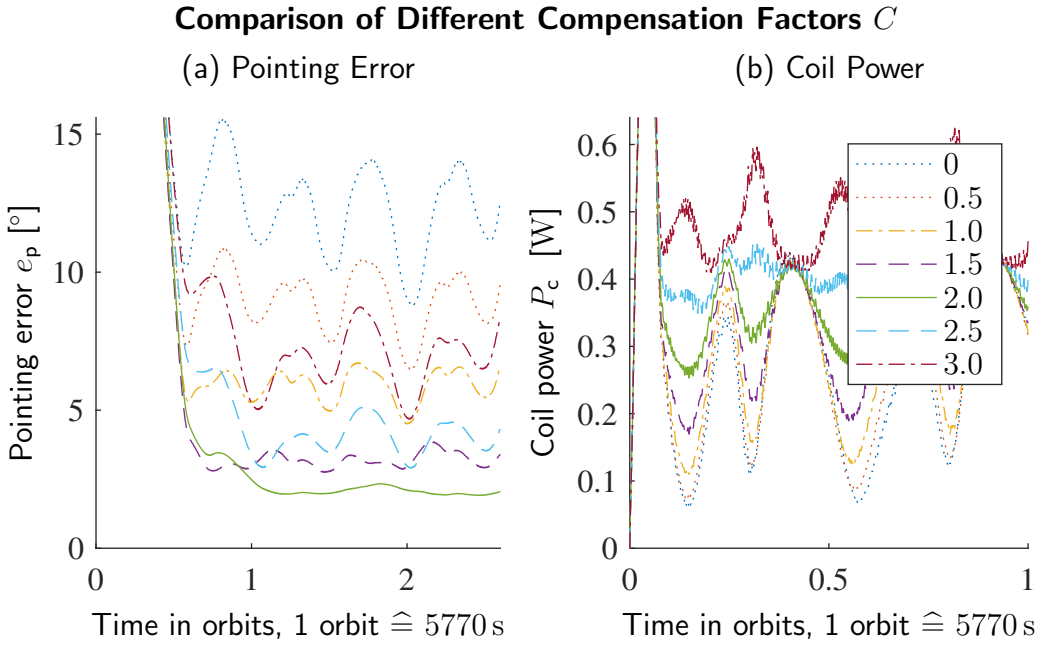

Figure 4.14: Pointing error and coil power consumption with respect to different compensation factors.

satellite and the controller does not need to compensate it. This happens when the Earth magnetic field aligns with the residual dipole. Due to the calculation formula for the magnetic control moment (2.32), currents which have no actuation effect are suppressed automatically.

The same simulation is also performed with the measured dipole magnitude of MOVE-II, namely $||\mathbf{m}_{\text{res}}|| = 6.9\,\text{mA}\,\text{m}^2$. The results can also be found in Table 4.4. When looking for an optimum compensation factor with regards to the pointing error, one obtains $C = 1.5$ as an optimum and not $C = 2.0$. However, the pointing

$\|\|\mathbf{m}_{\text{res}}\|\|$	Statistic	$C =$ 0	0.5	1.0	1.5	2.0	2.5	3.0
$6.9\,\text{mA}\,\text{m}^2$	$\mu_{\text{p},6.9,\text{c}}$	2.71°	2.36°	2.19°	2.15°	2.26°	2.49°	2.86°
	$\sigma^2_{\text{p},6.9,\text{c}}$ [deg^2]	1.8	1.3	0.82	0.62	0.89	1.5	2.2
$35\,\text{mA}\,\text{m}^2$	$\mu_{\text{p},35,\text{c}}$	12.6°	9.29°	6.07°	3.22°	2.26°	4.35°	7.37°
	$\sigma^2_{\text{p},35,\text{c}}$ [deg^2]	23	13	6.0	2.3	0.91	5.1	12
$0\,\text{mA}\,\text{m}^2$	$\mu_{\text{p},0}$	2.25°						
	$\sigma^2_{\text{p},0}$ [deg^2]	0.87						

Table 4.4: Mean pointing errors μ_{p} and corresponding variances σ^2_{p} in dependency of different compensation factors C and the magnetic dipole moment.

error for $C = 2$ is very close to this result and the deviation is not statistically significant. For a more convincing result, more simulation with different initial conditions need to be performed. In conclusion we see that with a compensation factor of $C = 2$ the mean pointing error and its variance do not change, regardless of the dipole moments considered.

4.7 Discussion of the Results for the Extended LQR Controller

The simulation results show that the Extended LQR Controller is an enhancement to the ADCS. It shows a satisfying performance and outperforms the Default Controller in every simulated aspect.
The way we design the gain $\mathbf{K}_{\mathrm{LQR}}$ forces the controller to prioritize for the pointing error. In this sense, the design choice for the Default Controller in having the weights related to the angular velocity five orders of magnitude greater than the weights related to the sun vector, is a suboptimal choice. With design choices closely oriented to the system requirements, the mean pointing error is decreased by 0.7°. This is an improvement of 23% with respect to the Default Controller. Additionally, the convergence time of the pointing error is decreased significantly from 25 min to 12 min with the Extended LQR Controller.
However, paying less attention to the angular velocity leads to the problem that the satellite spins up around the intermediate principal axis for certain initial conditions, such that the nutation torque becomes greater than the maximal possible control torque. This results in an unstable motion and the satellite turns around 180° to anti-sun-pointing. While it can be observed from selected simulation runs that the satellite is able to reach a sun pointing state eventually, a lot of energy is wasted during this process, and the solar panels are rotated away from the sun. Energy however is a critical and limited resource in space. The probability of 20% for this behavior to occur is high.
Considering this, the results of the Monte Carlo simulation are not satisfying. But they also show that the Extended LQR Controller is an improvement to the Default Controller nonetheless. Apart from introducing this new problematic behavior, the total number of converging and slow converging runs in the Monte Carlo simulation with 100 testruns increases from 69 to 80 with the Extended LQR Controller. In addition, the slow converging runs with the Extended LQR Controller are less critical than with the Default Controller. While the slow converging runs of the Extended LQR Controller achieve a low pointing error within 10 min and only show a pointing error peak with a maximum observed pointing error of 60° for about 10 min, the slow converging runs with the Default Controller do not reach a satisfying pointing error for the first 40 min.
It is also advantageous to use the true inertia tensor for the system matrices, instead of the diagonal approximation. With this step, the nutation torque is included into

the model and the LQR algorithm pays attention to this torque when optimizing for a minimal cost functional. It is assumed that this measure reduces the pointing error fluctuation of $\pm 1.3°$ in average with the Default Controller to only $\pm 0.55°$ in average with the Extended LQR Controller.
With regards to the angular velocity it is obtained that the Extended LQR Controller drives the ω_x and ω_y components fast towards zero, as does the Default Controller. But the Extended LQR Controller utilizes its freedom with respect to ω_z to achieve a low mean pointing error and to converge fast with respect to the pointing error. The Default Controller on the other hand never deviates much from the target spin rate ω_s. Thus the Extended LQR Controller fulfills all design goals. It minimizes the pointing error fast while obtaining a spin stabilization. The spin rate is corrected only when the magnetic field of Earth allows for corrections along the Z-axis. The spin rate $\omega_s = 0.1\,\mathrm{rad/s}$ is found to be reasonable for MOVE-II. It provides enough safety margin to the unstable gain matrices with characteristic multipliers greater than 1, and it is still small enough such that the nutation torque is low around the operation point.
The Monte Carlo simulation also shows that the hysteresis of $\pm 20\%$ for the operation point switching strategy is sufficient. No simulation runs with an oscillation around $\omega_z = 0$ is observed, and the oscillating behavior for initial conditions with $\omega_z < 0$ is eliminated.
Furthermore, simulations show that the residual magnetic dipole has a significant impact onto the Default Controller. Especially a dipole moment along the Z-axis increases the pointing error. With the dipole compensation it is possible to compensate the effect of the dipole for magnitudes of up to $55\,\mathrm{mA\,m^2}$. This is a strong dipole moment, considering that the Default Controller is designed to achieve a low pointing error in the presence of a dipole moment of about $20\,\mathrm{mA\,m^2}$. This is only achieved by tuning it heuristically.
The improved sun pointing with the dipole compensation method comes at the cost of an increases the power consumption. However, the main cause for the power consumption is the presence of the dipole itself, and the way the controller has to deal with it. For low dipole magnitudes, the additionally power consumption is in the range of a few mW.

Chapter 5

Evaluating Nonlinear Control Strategies

In Sections 3.3.2 and 4.6.2 it becomes clear that linearized control laws are not able to converge from any initial condition to the desired attitude in an appropriate amount of time. Much research therefore deals with nonlinear controllers for the satellite attitude to ensure a successful attitude maneuver from an arbitrary initial condition.
This chapter is organized as follows: First, in Section 5.1 related work about nonlinear controllers for the attitude control problem of CubeSats is presented. The section concludes that controllers based on Lyapunov functions are advantageous for the specific problem of this thesis. Three controller for further analysis are selected, namely the Delta-H, the JC2Sat and the Modular Controller. Then, the theoretical concept of Lyapunov stability and the theorem of Lyapunov are presented in Section 5.2. Sections 5.3 to 5.5 present the Delta-H Controller, the JC2Sat Controller and the Modular Controller in more detail. Finally, Section 5.6 presents the simulation results of the nonlinear controllers and evaluates their performance. We analyze their convergence behavior and compare their pointing accuracy after convergence with those of the linearized controllers. Furthermore, a Software-in-the-Loop (SiL) and a Hardware-in-the-Loop (HiL) simulation are performed for the Delta-H Controller in order to show that this control law performs as expected on the flight hardware and therefore can be implemented on MOVE-II.

5.1 Related Work

Shen and Tsiotras [ST99] propose an algorithm for numerically solving the time optimal control problem for a maneuver from an arbitrary attitude and an arbitrary angular velocity around a spin axis to a target attitude and a target angular velocity. This algorithm can be applied to spin stabilized satellites. An optimal trajectory is computed by solvers for the described boundary value problem. The drawback of this method is the high computational complexity, which makes it hard to calculate it on

board. Also no disturbances nor sensor feedback during the maneuver is considered, which makes the controller sensitive to disturbances. This is critical especially for CubeSats, where the disturbances are high compared to passive stabilizing effects like spin stabilization or gravity gradient stabilization.
Biggs and Horri [BH12] propose an algorithm with similar goals, which features an analytical solution. This decreases the computational complexity and they propose their algorithm suitable for on-board calculation. They also show how to track motions using an augmented quaternion feedback law. They assume, however, that the satellite features reaction wheels, which allow to generate torques in any direction, as opposed to magnetic actuation where torques can be generated only perpendicular to the magnetic field.
Louembet et al. [LCZ+09] take a different approach. They transform the nonlinear optimal control problem into a geometric programming problem. One advantage of their approach is that they allow for an arbitrary inertial tensor with diagonal entries, like the inertia tensor of MOVE-II. However, they also assume that the satellite is able to use reaction wheels, which is not the case for MOVE-II.
One method focusing specifically on satellites depending on magnetic actuation implements a neural network in order to deal with the nonlinearities. Sivaprakash and Shanmugam [SS05] present a two layer neuronal network, which receives the system state, consisting of the attitude quaternion, the angular velocities and their derivatives, as input and calculates the control currents for the coils as output. They train the weights of the neural network with backpropagation, using data generated from conventional proportional-derivative controllers for different initial conditions. While the results look promising, there is no thorough analysis of convergence for arbitrary initial conditions and no proof of stability.
There are other researches, however, which focus on or include magnetic actuation and which have a sound theoretical foundation [RC01, dR11, OB12]. This chapter focuses on these concepts specifically. The Global Spin Rate Controller proposed by Reynolds and Creamer [RC01] and the JC2Sat Controller described by De Ruiter are based on Lyapunov functions. Ousaloo and Badpa [OB12] do not state a Lyapunov function for their modularly constructed control law, which we call Modular Controller. However, a corresponding Lyapunov function for the derived controller can be constructed, and therefore stability can be proven. While the Global Spin Rate Controller is able to spin the satellite around its major principle axis only, more recent research by Shah [Sha11] extends this controller with the possibility to define a specific rotation axis for the satellite. Shah calls his algorithm Delta-H Controller. Therefore, the Delta-H Controller and the Global Spin Rate Controller are treated together in this chapter. All these controllers have one common drawback, however, which is the assumption that either the inertia tensor is diagonal or the spin axis is identical to a principal axis of inertia.

5.2 Lyapunov Stability

Stability always refers to an equilibrium point $\mathbf{x}^*$, which is any point in the system domain where the time derivative of the state vector $\dot{\mathbf{x}}$ equals zero. It is easy to see that the desired operation point (2.25) is an equilibrium point for the nonlinear system with system equations (2.11) and (2.21), when the nutation torque $\boldsymbol{\tau}_\mathrm{n}$ is considered as an additional disturbance torque instead of considering it as a part of the system equations. This assumption can be safely made since all control laws evaluated in this chapter assume an operation point where $\boldsymbol{\tau}_\mathrm{n} = 0$.

An equilibrium point $\mathbf{x}^*$ of a system is stable according to Lyapunov [Kha02], if a trajectory starting close to $\mathbf{x}^*$ stays close to $\mathbf{x}^*$ for all $t > t_0$. This is the case, if for each $\epsilon > 0$ there is a $\delta = \delta(\epsilon) > 0$ such that

$$||\mathbf{x}(0) - \mathbf{x}^*|| < \delta \implies ||\mathbf{x}(t) - \mathbf{x}^*|| < \epsilon, \ \forall t \geq 0. \tag{5.1}$$

The equilibrium point is asymptotically stable, if a trajectory starting close to $\mathbf{x}^*$ converges to $\mathbf{x}^*$ eventually. This is the case if $\mathbf{x}^*$ is stable and there is a δ such that

$$||\mathbf{x}(0) - \mathbf{x}^*|| < \delta \implies \lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^*. \tag{5.2}$$

Lyapunov proposes a method to test stability for an arbitrary equilibrium point, which we call the theorem of Lyapunov [Kha02, p. 114]. This theorem is based on the consideration of the potential energy in the system. When the potential energy is zero, the system must be at rest. This means that the equilibrium point must be asymptotically stable, if the potential energy is strictly monotonically decreasing and becomes zero at the equilibrium point. It is not always simple or even possible to construct such an energy function however. Lyapunov shows that it is sufficient to find a pseudo energy function $V(\mathbf{x})$ which replaces the potential energy function. The theorem of Lyapunov states that the equilibrium point $\mathbf{x}^*$ is asymptotically stable, if there is a positive definite pseudo energy function $V(\mathbf{x} - \mathbf{x}^*)$ such that the time derivative $\dot{V}(\mathbf{x})$ is negative definite. Any pseudo energy function $V(\mathbf{x})$ fulfilling these criteria is called Lyapunov function.

5.3 Delta-H Controller

This section presents the Delta-H Controller introduced by Shah [Sha11]. The foundation for this controller is the Global Spin Rate Controller proposed by Reynolds and Creamer [RC01]. They introduce an approach which provides nutation, precession and spin rate control within a single control law, which only depends on the angular velocity rate error. This is convenient to implement, since the ADCS can

measure this value directly. The control law is based on a Lyapunov function and its time derivative

$$V_{\mathrm{G}} = \frac{1}{2} \cdot (\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{s}^{\mathrm{b}})^{\mathrm{T}} (\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{s}^{\mathrm{b}}) + \frac{1}{2} \cdot (\mathbf{h}^{\mathrm{b}})^{\mathrm{T}} (\lambda_{\max} \cdot (\mathbf{I}^{\mathrm{b}})^{-1} - \mathbf{1}_{3\times 3}) \mathbf{h}^{\mathrm{b}} \tag{5.3}$$

$$\dot{V}_{\mathrm{G}} = \lambda_{\max} \cdot (\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} - \omega_{\mathrm{d}} \cdot \mathbf{s}^{\mathrm{b}})^{\mathrm{T}} \boldsymbol{\tau}_{\mathrm{c}}^{\mathrm{b}}, \tag{5.4}$$

which guarantees the convergence to a unique global minimum in the specified cost, where $\lambda_{\max}$ is the maximum eigenvalue of the inertia tensor $\mathbf{I}^{\mathrm{b}}$ and h_{d} is the desired magnitude of the angular momentum $\mathbf{h}^{\mathrm{b}}$ in the direction of the sun vector $\mathbf{s}^{\mathrm{b}}$. The first term of V_{G} implies that the spin axis of the satellite aligns with the sun vector after settling, and the second term implies that the satellite spins around its major principle axis after settling. When these two criteria are fulfilled, the Lyapunov function becomes zero.

The time derivative is calculated under the assumption that the desired spin axis of the spacecraft is the major principle axis of inertia, therefore $\omega_{\mathrm{d}} = h_{\mathrm{d}}/\lambda_{\max}$ is the desired magnitude of the angular velocity in the direction of the sun vector. For MOVE-II we chose the value evaluated in Section 4.1 as desired spin rate $\omega_{\mathrm{d}} = \omega_{\mathrm{s}}$. It is assumed that the control torque $\boldsymbol{\tau}_{\mathrm{c}}$ is the only torque acting on the satellite, disturbances are not considered for stability analysis.

Reynolds and Creamer show that V_{G} is indeed a Lyapunov function, when choosing a control law such that the control torque always points in the opposite direction of the rate error vector $(\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} - \omega_{\mathrm{d}} \cdot \mathbf{s}^{\mathrm{b}})$. For a magnetic actuated satellite, the authors state that this is the fact when choosing the magnetic control dipole moment such that

$$\mathbf{m}_{\mathrm{c}}^{\mathrm{b}} = k \cdot \mathrm{sign}((\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} - \omega_{\mathrm{d}} \cdot \mathbf{s}^{\mathrm{b}}) \times \mathbf{b}^{\mathrm{b}}), \tag{5.5}$$

where k is an arbitrary positive constant. The function $\mathrm{sign}(x)$ is defined as

$$\mathrm{sign}(x) = \begin{cases} -1 \text{ if } x < 0 \\ +1 \text{ else} \end{cases} \tag{5.6}$$

and is applied component wise for vectorial input. Due to this sign function, the control law is a discrete control law with respect to the direction of $\mathbf{m}_{\mathrm{c}}^{\mathrm{b}}$. The control dipole moment can only take $2^3 = 8$ different directions.

Shah [Sha11] however shows that this control law has an ambiguity in attitude. Instead of spinning around an axis with angular velocity ω_{d} and pointing this axis towards the sun vector, it is also possible that the satellite spins around an axis with angular velocity $-\omega_{\mathrm{d}}$ and points away from the sun vector. Both constellations make V_{G} zero. Therefore Shah introduces another term, appending it to the Lyapunov function, in order to solve this ambiguity. He specifies a spin axis $\mathbf{p}^{\mathrm{b}}$, around which the satellite has to rotate and which is constant in the BF. Therefore

the rotation axis of the satellite is well defined and the Lyapunov function of the Delta-H Controller becomes

$$V_{\mathrm{D}} = \frac{k_{\mathrm{D},1}}{2} \cdot (\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{s}^{\mathrm{b}})^{\mathrm{T}} (\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{s}^{\mathrm{b}}) +$$

$$\frac{k_{\mathrm{D},2}}{2} \cdot (\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{p}^{\mathrm{b}})^{\mathrm{T}} (\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{p}^{\mathrm{b}}) +$$

$$\frac{k_{\mathrm{D},1} + k_{\mathrm{D},2}}{2} \cdot (\mathbf{h}^{\mathrm{b}})^{\mathrm{T}} (\lambda_{\max} \cdot (\mathbf{I}^{\mathrm{b}})^{-1} - \mathbf{1}_{3\times 3}) \mathbf{h}^{\mathrm{b}} \tag{5.7}$$

$$\dot{V}_{\mathrm{D}} = \lambda_{\max} \cdot (k_{\mathrm{D},1} + k_{\mathrm{D},2}) \cdot \left(\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} - \omega_{\mathrm{d}} \cdot \frac{k_{\mathrm{D},1} \cdot \mathbf{s}^{\mathrm{b}} + k_{\mathrm{D},2} \cdot \mathbf{p}^{\mathrm{b}}}{k_{\mathrm{D},1} + k_{\mathrm{D},2}} \right)^{\mathrm{T}} \boldsymbol{\tau}_{\mathrm{c}}^{\mathrm{b}}. \tag{5.8}$$

As proposed by Shah, the tuneable gains $k_{\mathrm{D},1}$ and $k_{\mathrm{D},2}$ are chosen such that $k_{\mathrm{D},1} + k_{\mathrm{D},2} = 1$, where $k_{\mathrm{D},1} \in [0; 1]$ is the free variable and $k_{\mathrm{D},2}$ is defined as $k_{\mathrm{D},2} = 1 - k_{\mathrm{D},1}$. The time derivatives of V_{D} and V_{G} are obtained in a similar manner and under the assumption that the desired spin axis of the of the spacecraft is the major principle axis. A step by step derivation is found in [Sha11].
Shah provides no control law for magnetically actuated satellites. He assumes that the spacecraft features reaction wheels. However, similar to the considerations of Reynolds and Creamer, a control law for magnetically actuated spacecrafts can be derived from the Lyapunov function. To stabilize the operation point, a control dipole moment

$$\mathbf{m}_{\mathrm{c}}^{\mathrm{b}} = g_{\mathrm{D}} \cdot \mathrm{sign}(\mathbf{e}_{\mathrm{c,D}}^{\mathrm{b}} \times \mathbf{b}^{\mathrm{b}}) \tag{5.9}$$

with control error vector

$$\mathbf{e}_{\mathrm{c,D}}^{\mathrm{b}} = \boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} - \omega_{\mathrm{d}} \cdot (k_{\mathrm{D},1} \cdot \mathbf{s}^{\mathrm{b}} + k_{\mathrm{D},2} \cdot \mathbf{p}^{\mathrm{b}}) \tag{5.10}$$

is selected, which ensures that the satellite points the angular momentum axis towards the sun and rotates around the specified body axis $\mathbf{p}^{\mathrm{b}}$. With this new variable $\mathbf{p}^{\mathrm{b}}$ it is possible to select a different axis than the major principle axis as spin axis. For MOVE-II we chose the spin axis $\mathbf{p}^{\mathrm{b}} = (0, 0, 1)^{\mathrm{T}}$ to be the positive Z-axis. This means, however, that V_{D} becomes not zero at the operation point due to the nutation reduction term $(\mathbf{h}^{\mathrm{b}})^{\mathrm{T}}(\lambda_{\max} \cdot (\mathbf{I}^{\mathrm{b}})^{-1} - \mathbf{1}_{3\times 3})\mathbf{h}^{\mathrm{b}}$ and strictly speaking V_{D} is not a Lyapunov function any more. Since the principal moments of inertia of MOVE-II are about the same size in numbers due to the cubic shape of the satellite and the desired spin axis is close to a principal axis of inertia, it is assumed that this control law will work good enough in practice.
The gain factor g_{D} in (5.9) extends the original Delta-H controller. It must always be positive, in order for the resulting control dipole moment to be stabilizing. If g_{D} is a constant, due to the sign function returning only the values 1 or -1, the controller generates a constant control dipole moment at all times. This resulting bang-bang

control is not energy efficient. In an example for the Global Spin Rate Controller, Reynolds and Creamer propose a g_D which becomes smaller with a smaller pointing error. In the scope of this thesis, however, a different approach is taken. Assuming that the control error $\mathbf{e}_{\mathrm{c,D}}^{\mathrm{b}}$ aligns with the magnetic field $\mathbf{b}^{\mathrm{b}}$, then the cross product becomes zero and small random fluctuations in the measurements cause the control dipole moment to flip the sign. The closer the angle between $\mathbf{e}_{\mathrm{c,D}}^{\mathrm{b}}$ and $\mathbf{b}^{\mathrm{b}}$, the less effective is the resulting $\mathbf{m}_{\mathrm{c}}^{\mathrm{b}}$ and the greater is the influence of sensor noise. The gain factor

$$g_\mathrm{D} = (||\hat{\mathbf{b}}^{\mathrm{b}} \times \hat{\mathbf{e}}_{\mathrm{c,D}}|| \cdot s_\mathrm{D} + b_\mathrm{D}) \cdot m_{\mathrm{c,max}} \cdot c \tag{5.11}$$

accounts for this by scaling down the control dipole moment $\mathbf{m}_{\mathrm{c}}^{\mathrm{b}}$ depending on the angle between $\mathbf{e}_{\mathrm{c,D}}^{\mathrm{b}}$ and $\mathbf{b}^{\mathrm{b}}$. The constant $m_{\mathrm{c,max}} = I_{\mathrm{max}} \cdot n_{\mathrm{c}} \cdot A_{\mathrm{eff}}$ is the maximum possible control moment for every coil. Control moments exceeding this value are limited to $m_{\mathrm{c,max}}$ automatically by the MOVE-II firmware. The values $\hat{\mathbf{b}}^{\mathrm{b}}$ and $\hat{\mathbf{e}}_{\mathrm{c,D}}^{\mathrm{b}}$ describe the normalized vectors of $\mathbf{b}^{\mathrm{b}}$ and $\mathbf{e}_{\mathrm{c,D}}^{\mathrm{b}}$ respectively. The expression $||\hat{\mathbf{b}}^{\mathrm{b}} \times \hat{\mathbf{e}}_{\mathrm{c,D}}^{\mathrm{b}}||$ is identical to $\sin(\alpha)$, where $\alpha \in [0°; 180°]$ is the angle between $\mathbf{e}_{\mathrm{c,D}}^{\mathrm{b}}$ and $\mathbf{b}^{\mathrm{b}}$. The scaling factor $s_\mathrm{D} = 1.6$ and the bias $b_\mathrm{D} = 0.2$ are chosen such that

$$g_\mathrm{D} \geq m_{\mathrm{c,max}} \quad \text{for } 30° \leq \alpha \leq 150°, \tag{5.12}$$

$$g_\mathrm{D} = 0.2 \cdot m_{\mathrm{c,max}} \quad \text{if } \alpha = 0° \text{ or } \alpha = 180°. \tag{5.13}$$

Therefore, if the error vector direction is close to the magnetic field direction and thus the direction of $\mathbf{m}_{\mathrm{c}}^{\mathrm{b}}$ is prone to errors due to noisy measurements, the control dipole moment is reduced. The factor $c = 1 \cdot \frac{\mathrm{A\,m^2\,s}}{\mathrm{rad\,T}}$ in (5.11) corrects the unit mismatch in (5.9) between $\mathbf{m}_{\mathrm{c}}^{\mathrm{b}}$ and the result of the sign function.
The tuning value $k_{\mathrm{D},1} = 0.015$ is found heuristically with the help of several simulations. Increasing the value of $k_{\mathrm{D},1}$ puts more weight on the pointing error while at the same time decreasing the weight of the angular velocity error. For values above 0.75, the characteristic of the Delta-H control law approaches the characteristic of the Global Spin Rate Controller. The influence of the term defining the spin axis is low, which causes the satellite to spin around a different axis more easily. For values of $k_{\mathrm{D},1}$ below 0.001, the influence of the term driving the spin axis towards the sun vector is too low, which causes the pointing error to increase after convergence. Varying $k_{\mathrm{D},1}$ between these two boundary values has no significant impact

Parameter	Value
$k_{\mathrm{D},1}$	0.015
s_D	1.6
b_D	0.2

Table 5.1: Default tuning parameters for the Delta-H Controller.

onto the controller performance. However, it is obtained that low values tend to result in a decreased settling time, therefore a value from the lower range is selected. Appendix B.1 provides the results of the simulations leading to this decision.
The default gains and tuning parameters for this controller as used for simulations, if not stated otherwise, are summarized by Table 5.1.

5.4 JC2Sat Controller

This section introduces the controller used by the Japan Canada Joint Collaboration Satellite-Formation Flying (JC2Sat-FF) mission as presented by De Ruiter [dR11]. This mission launches two nano satellites, each with a mass of 15 kg. These satellites rely solely on magnetic actuation like MOVE-II and the mission goals are similar. Spin stabilization is required as well as pointing the spin axis towards an inertially fixed point. The control law is derived from a Lyapunov function and its time derivative

$$V_{\mathrm{J}} = \frac{1}{2} \cdot [(\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{s}^{\mathrm{b}})^{\mathrm{T}}(\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{s}^{\mathrm{b}}) + k_{\mathrm{J},1} \cdot (h_{\mathrm{z}} - h_{\mathrm{d}})^2 + k_{\mathrm{J},2} \cdot (\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}})^{\mathrm{T}} \mathbf{P} \mathbf{I}^{\mathrm{b}} \boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}] \tag{5.14}$$

$$\dot{V}_{\mathrm{J}} = [(\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{s}^{\mathrm{b}}) + k_{\mathrm{J},1} \cdot (h_{\mathrm{z}} - h_{\mathrm{d}}) \cdot (0,0,1)^{\mathrm{T}} + k_{\mathrm{J},2} \cdot \mathbf{P} \boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}]^{\mathrm{T}} \boldsymbol{\tau}_{\mathrm{c}}^{\mathrm{b}} \tag{5.15}$$

$$\text{with } \mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and where h_{z} describes the Z component of the angular momentum vector $\mathbf{h}^{\mathrm{b}}$. V_{J} is adapted for a spin around the Z-axis of the satellite, De Ruiter proposes originally a controller designed for a spin around the Y-axis. De Ruiter shows that the resulting control dipole moment

$$\mathbf{m}_{\mathrm{c}}^{\mathrm{b}} = \frac{k_{\mathrm{J},0}}{||\mathbf{b}^{\mathrm{b}}||^2} \cdot \mathbf{e}_{\mathrm{c,J}}^{\mathrm{b}} \times \mathbf{b}^{\mathrm{b}} \tag{5.16}$$

calculated from the control error

$$\mathbf{e}_{\mathrm{c,J}}^{\mathrm{b}} = (\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{s}^{\mathrm{b}}) + k_{\mathrm{J},1} \cdot (h_{\mathrm{z}} - h_{\mathrm{d}}) \cdot (0,0,1)^{\mathrm{T}} + k_{\mathrm{J},2} \cdot \mathbf{P} \boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}} \tag{5.17}$$

is asymptotically stabilizing the operation point, even under the failure of up to two coils and in presence of coil saturation. To show this he assumes an orbit not lying in the equatorial plane of Earth. This assumption is justified for MOVE-II.
The tuneable gains $k_{\mathrm{J},0}$, $k_{\mathrm{J},1}$ and $k_{\mathrm{J},2}$ characterize the behavior of the controller. Increasing $k_{\mathrm{J},0}$ decreases the convergence time. A high value of $k_{\mathrm{J},0}$ however results in a bang-bang controller, which increases the convergence time. In simulations, a value of $k_{\mathrm{J},0} = 0.008 \cdot 1/\mathrm{s}$ is found to provide reasonable magnitudes of the control

current. In order to converge to a spin around the desired body axis, the term amplified by $k_{\mathrm{J},1}$ is needed. Simulations show that changing this value does not change the behavior of the controller significantly, as long as the value is greater than 1. De Ruiter states that a low value for $k_{\mathrm{J},1}$ is sufficient, therefore $k_{\mathrm{J},1} = 5$ is chosen. The term amplified by $k_{\mathrm{J},2}$ provides nutation damping. De Ruiter states that some nutation damping is needed, but a high value of $k_{\mathrm{J},2}$ is counterproductive. This observation is confirmed by simulations with MOVE-II. For $k_{\mathrm{J},2} < 0.01\,\mathrm{kg\,m^2}$ the angular velocity components ω_x and ω_y are not damped effectively which results in an increased pointing error. Values of $k_{\mathrm{J},2} > 0.2\,\mathrm{kg\,m^2}$ slow down the pointing error convergence, since the controller does not allow for high angular velocities in the X- and Y-direction, which are needed to adjust the Z-axis correctly. A value of $k_{\mathrm{J},2} = 0.1\,\mathrm{kg\,m^2}$ shows a satisfying performance for MOVE-II. Appendix B.2 provides the simulation results on which these decisions are based. The value for h_d, which, unlike with the Delta-H Controller, appears in the control law, is obtained from the operation point as $h_\mathrm{d} = ||\mathbf{I}^\mathrm{b} \cdot \omega_\mathrm{s}(0,0,1)^\mathrm{T}||$.
De Ruiter also shows that the satellite converges to the specified attitude, when the tuneable gains fulfill the criteria

$$k_{\mathrm{J},1} > 1 \tag{5.18}$$

$$k_{\mathrm{J},2} > 0 \tag{5.19}$$

$$V_\mathrm{J} < V_{\mathrm{J.max}} = 2 \cdot \frac{k_{\mathrm{J},1}}{k_{\mathrm{J},1}+1} \cdot h_\mathrm{d}^2. \tag{5.20}$$

Criteria (5.18) and (5.19) are fulfilled. Criterion (5.20) is also fulfilled for MOVE-II in all relevant cases. This is because MOVE-II features a detumbling controller, which reduces its angular velocity to a magnitude $||\boldsymbol{\omega}_\mathrm{ib}^\mathrm{b}||$ below $0.15\,\mathrm{rad/s}$, before the attitude controller is activated. Due to the low weight of MOVE-II, the inertia tensor features low values and the initial values of V_J are approximately of magnitude 10^{-6} $\mathrm{kg^2\,m^4\,s^{-2}}$ for the worst case scenario. In contrast to $V_{\mathrm{J,max}} = 0.017\,\mathrm{kg^2\,m^4\,s^{-2}}$ the typical initial value of V_J is several magnitudes lower and criterion (5.20) is fulfilled. A more than worst case approximation with the maximum magnitude of the angular velocity as angular velocity value in every axis $\boldsymbol{\omega}_\mathrm{ib}^\mathrm{b} = -(0.15, 0.15, 0.15)^\mathrm{T}\,\mathrm{rad/s}$ and $\mathbf{s}^\mathrm{b} = (0,0,1)^\mathrm{T}$ resulting in $V_\mathrm{J} = 7.7\text{e-}6\,\mathrm{kg^2\,m^4\,s^{-2}}$ underlines this observation. This means, MOVE-II always ends up in the desired configuration utilizing this control law.
Table 5.2 summarizes the tuning gains which are used in simulations with the JC2Sat Controller per default.

Parameter	Value
$k_{\mathrm{J},0}$	$0.008\,\mathrm{s^{-1}}$
$k_{\mathrm{J},1}$	5
$k_{\mathrm{J},2}$	$0.1\,\mathrm{kg\,m^2}$

Table 5.2: Default tuning parameters for the JC2Sat Controller.

5.5 Modular Controller

Ousaloo and Badpa [OB12] present an algorithm to control the spin rate, the nutation and the spin axis individually. To every coil located on each axis of the satellite an individual control law is applied with different goals. They propose two different control laws and both have to be applied to a number of coils simultaneously, since the spin rate and the spin axis control influence each other through side effects and vice versa. Due to this modular character, we name this controller the Modular Controller.
The authors build this controller with fail safety in mind. It is most efficient to control the spin axis with the coil placed along the spin axis. The other two transverse coils are controlling the spin rate. When the axial coil fails however, one of the transverse coils takes over to control the spin axis instead. Therefore, they show that full attitude and spin rate control can be obtained even when the axial coil is shut off.
They also propose a mode to speed up the attitude acquiring process. When additionally to the axial coil one transverse coil is used for spin axis control, the convergence time can be reduced.
They propose the control laws

$$m_{\mathrm{R},i} = m_{\mathrm{c,max}} \cdot \mathbf{a}_i \circ \mathrm{sign}((\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{p}^{\mathrm{b}}) \times \mathbf{b}^{\mathrm{b}}) \tag{5.21}$$

$$m_{\mathrm{A},i} = m_{\mathrm{c,max}} \cdot \mathbf{a}_i \circ \mathrm{sign}((\hat{\mathbf{h}}^{\mathrm{b}} - \mathbf{s}^{\mathrm{b}}) \times \mathbf{b}^{\mathrm{b}}), \tag{5.22}$$

where for every coil i with alignment axis $\mathbf{a}_i$ one of the control moments $m_{\mathrm{R},i}$ or $m_{\mathrm{A},i}$ is selected. The vector $\hat{\mathbf{h}}^{\mathrm{b}}$ describes the normalized angular momentum $\mathbf{h}^{\mathrm{b}}$, the vector $\mathbf{p}^{\mathrm{b}} = (0, 0, 1)^{\mathrm{T}}$ is the desired spin axis fixed in the BF and $m_{\mathrm{c,max}}$ is the maximum momentum which each coil can generate. The spin rate control moment $m_{\mathrm{R},i}$ acquires the desired spin rate around the body axis $\mathbf{p}^{\mathrm{b}}$, while the attitude control moment $m_{\mathrm{A},i}$ points the spin axis towards sun. From these two laws a third control law, the combined control moment

$$m_{\mathrm{C},i} = m_{\mathrm{c,max}} \cdot \mathbf{a}_i \circ \mathrm{sign}((\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{p}^{\mathrm{b}} + \hat{\mathbf{h}}^{\mathrm{b}} - \mathbf{s}^{\mathrm{b}}) \times \mathbf{b}^{\mathrm{b}}) \tag{5.23}$$

can be constructed, which combines both goals within one control law. Due to the sign functions, these control laws are also discrete with respect to direction, like the control law of the Delta-H Controller. Since the control moments always take the magnitude $m_{\mathrm{c,max}}$, this control law implements a bang-bang controller.
To show that these control laws are stabilizing the operation point, consider the Lyapunov functions

$$V_{\mathrm{R}} = 1/2 \cdot (\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{p}^{\mathrm{b}})^{\mathrm{T}} (\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{p}^{\mathrm{b}}) \tag{5.24}$$

$$V_{\mathrm{A}} = 1/2 \cdot (\hat{\mathbf{h}}^{\mathrm{b}} - \mathbf{s}^{\mathrm{b}})^{\mathrm{T}} (\hat{\mathbf{h}}^{\mathrm{b}} - \mathbf{s}^{\mathrm{b}}) \tag{5.25}$$

$$V_{\mathrm{C}} = V_{\mathrm{R}} + V_{\mathrm{A}} \tag{5.26}$$

and their derivatives

$$\dot{V}_{\mathrm{R}} = (\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{p}^{\mathrm{b}}) \circ \boldsymbol{\tau}_{\mathrm{c}}^{\mathrm{b}} \tag{5.27}$$

$$\dot{V}_{\mathrm{A}} = (\hat{\mathbf{h}^{\mathrm{b}}} - \mathbf{s}^{\mathrm{b}}) \circ \boldsymbol{\tau}_{\mathrm{c}}^{\mathrm{b}} \tag{5.28}$$

$$\dot{V}_{\mathrm{C}} = (\mathbf{h}^{\mathrm{b}} - h_{\mathrm{d}} \cdot \mathbf{p}^{\mathrm{b}} + \hat{\mathbf{h}}^{\mathrm{b}} - \mathbf{s}^{\mathrm{b}}) \circ \boldsymbol{\tau}_{\mathrm{c}}^{\mathrm{b}}. \tag{5.29}$$

The derivative of V_{C} follows from linearity of the derivative and the dot product. The derivative of V_{R} is shown by Ousaloo and Badpa under the assumption that the inertia tensor is diagonal, and the derivative of V_{A} follows similarly. With $\boldsymbol{\tau}_{\mathrm{c}}^{\mathrm{b}} = \mathbf{m}_{\mathrm{c}}^{\mathrm{b}} \times \mathbf{b}^{\mathrm{b}}$ it follows that the control moments $m_{\mathrm{R},i}$, $m_{\mathrm{A},i}$ and $m_{\mathrm{C},i}$ make the corresponding derivatives negative and therefore stabilize the operation point.
Depending on the selection of the control law for every coil, different modes can be defined. Table 5.3 gives an overview of the considered modes. To every coil located at the X-, Y- and Z-axis, one control law is assigned.

Mode	Control Moment $\mathbf{m}_{\mathrm{c}}^{\mathrm{b}}$
Mode 1	$(m_{\mathrm{R,x}}, m_{\mathrm{R,y}}, m_{\mathrm{A,z}})^{\mathrm{T}}$
Mode 2	$(m_{\mathrm{R,x}}, m_{\mathrm{A,y}}, m_{\mathrm{A,z}})^{\mathrm{T}}$
Mode 3	$(m_{\mathrm{R,x}}, m_{\mathrm{R,y}}, m_{\mathrm{C,z}})^{\mathrm{T}}$
Mode 4	$(m_{\mathrm{C,x}}, m_{\mathrm{C,y}}, m_{\mathrm{C,z}})^{\mathrm{T}}$

Table 5.3: Assigment of control laws to every coil located on the axes X, Y and Z for the different modes.

Mode 1 is proposed by Ousaloo and Badpa for nominal operation. The axial coil, which is the coil aligned with the Z-axis for MOVE-II, is used for attitude control, while the transverse coils are used for spin rate control. Mode 2 is the proposed acceleration mode, which speeds up the convergence of the spin axis towards the sun. Therefore, the transverse coil located on the Y-axis is assigned the attitude control moment $m_{\mathrm{A,y}}$. Mode 3 and 4 utilize the combined control moment $m_{\mathrm{C},i}$, proposed by the author of this thesis. Mode 3 enhances the focus on angular velocity control by replacing the attitude control moment for the axial coil with the combined control moment. This increases the capability of the controller for nutation damping. Mode 4 assigns to every coil the combined control moment and therefore accounts for spin rate and attitude control equally.
Simulations show, that Mode 3 is advantageous over the other modes, thus this mode is used for further simulations. Appendix B.3 contains the results of the simulations which leads to this decision.

5.6 Simulation Results

This section evaluates the performance of the nonlinear controllers and compares them to the Default Controller and the Extended LQR Controller. Section 5.6.1

evaluates the convergence of the nonlinear controllers with respect to initial conditions in a Monte Carlo simulation. Section 5.6.2 analyzes the long-term performance of linear and nonlinear controllers. It is also evaluated, how the sensor noise and the estimation error affects the performance of the controllers. Finally, Section 5.6.3 concludes with the SiL and HiL simulation results of the Delta-H Controller.
The residual magnetic dipole compensation, the actuation at low currents and the use of the EKF introduced in Chapter 4 for the Extended LQR Controller are also utilized for the nonlinear controllers.

5.6.1 Sensitivity to Initial Conditions

In this section the convergence behavior of the nonlinear controllers is evaluated. A Monte Carlo simulation with $n = 100$ testruns and a simulation time of 2500 s is performed for the randomly initialized values $\mathbf{q}_{\mathrm{ib},0}$ and $\boldsymbol{\omega}^{\mathrm{b}}_{\mathrm{ib},0}$. Like in previous Monte Carlo simulations, the initial attitude takes any possible values of a unit quaternion while the initial angular velocity takes values with magnitudes of up to 0.085 rad/s in each axis.
Figure 5.1 presents the results. Part (a) shows the pointing error e_{p} and the angular velocity in Z-direction ω_{z} for runs with the Delta-H Controller, part (b) and (c) with the JC2Sat and the Modular Controller respectively. The graphs for the pointing error are smoothed with an moving average filter of window size 500 for data points sampled at a frequency of 10 Hz.
One obtains that the Delta-H Controller is the only one for which all 100 simulation runs converge below a pointing error of 15°. The convergence time is strongly dependent on the initial conditions. Most runs show a convergence time of less than 25 min ($\widehat{=}$ 0.26 orbits), equal to the convergence time for the Default Controller. However, after only 12.5 min ($\widehat{=}$ 0.13 orbits), several runs are not converged, which equals the convergence time of the Extended LQR Controller.
With the JC2Sat Controller, two things are observed. Firstly, the runs which are converging show the lowest pointing error among all three controllers. And secondly, there is a significant amount of runs, which do not converge within the simulation time. These runs show also a negative Z component of the angular velocity ω_{z}. This behavior looks similar to the behaviors b_2 and b_3 observed with the Default Controller in Section 3.3.2.
Figure 5.2 analyzes three testruns with the JC2Sat Controller during an extended simulation time of 1.5 orbits with initial conditions that lead to a not converging behavior in the Monte Carlo simulation. The reason, why these runs do not converge at first can be seen by the control error angle β_{e}, which is the angle between the control error (5.17) and the magnetic field vector $\mathbf{b}^{\mathrm{b}}$. When these two vectors align, the angle between them is either 0° or 180° and the cross product in (5.16) becomes zero. From the graph of β_{e} it becomes clear that the controller maneuvers itself into a deadlock which it is not able to leave within a reasonable time span. While the magnetic field vector is rotating along the orbit, the control error is rotating in the

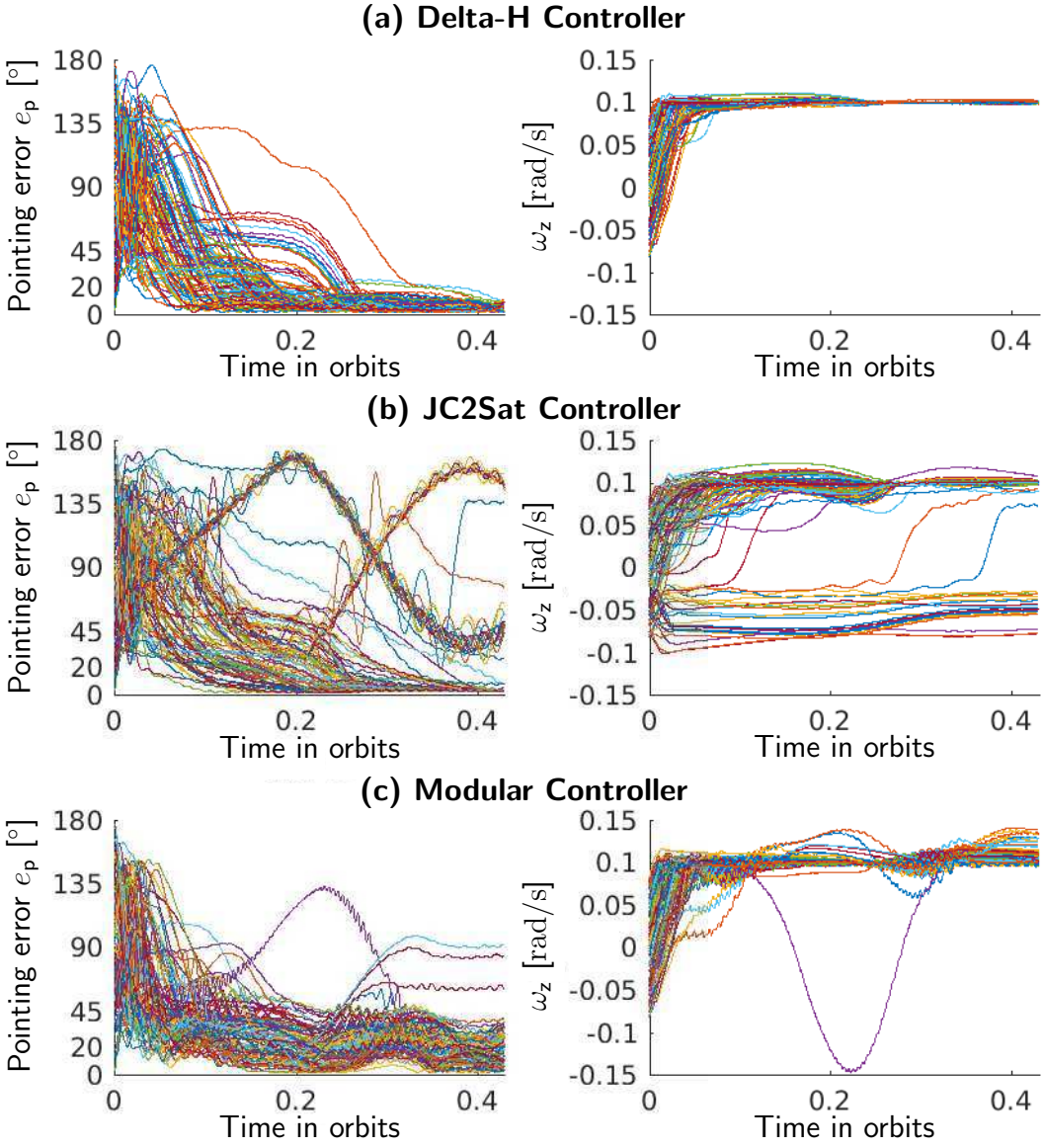

Figure 5.1: Results of the Monte Carlo Simulation with $n = 100$ testruns for the Delta-H (a), the JC2Sat (b) and the Modular Controller (c) with orbit period $T = 5770\,\mathrm{s}$. The pointing error is smoothed with a moving average filter.

same manner, which keeps the control error angle constant. The two runs r_2 and r_3 however converge eventually. Figure 5.2 shows that the control error angle leaves the deadlock after approximately 0.8 orbits and 1.4 orbits respectively for those two runs. This means that the controller is able to produce a control moment again,

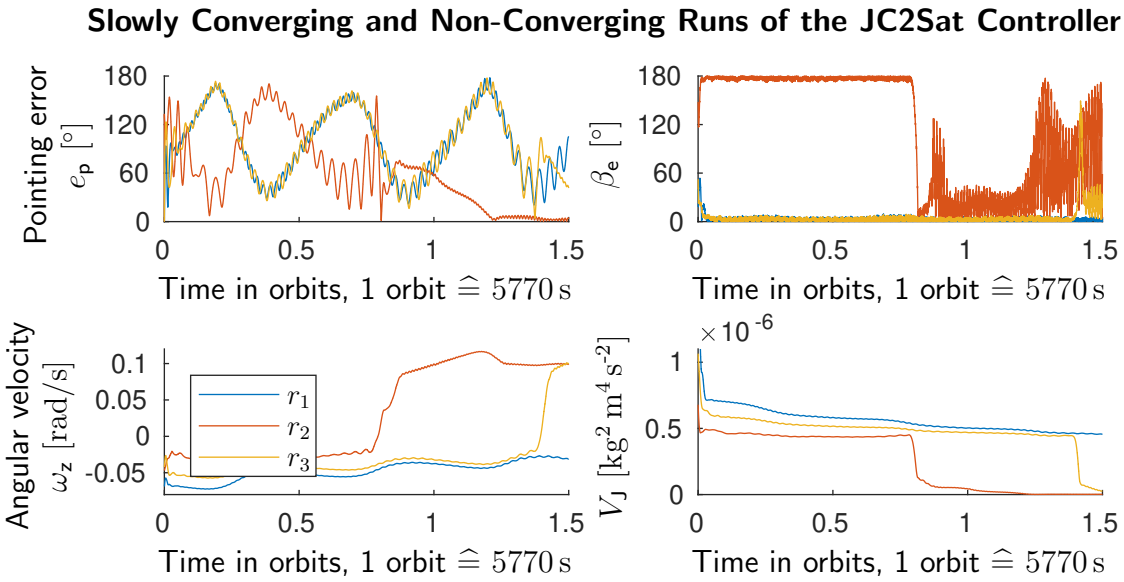

Figure 5.2: Analysis of three testruns with the JC2Sat Controller with initial conditions such that the pointing error e_p does not converge within half an orbit. The value β_e describes the angle between the magnetic field vector $\mathbf{b}^b$ and the control error vector $\mathbf{e}^b_{c,J}$, and V_J is the Lyapunov function.

which decreases the pseudo energy function V_J immediately by a large factor. The operation point of ω_z is obtained eventually and the pointing error starts to decrease. It is not for certain however, that the controller leaves the deadlock eventually. The two runs r_1 and r_3 have similar initial conditions and show similar trajectories of ω_z and e_p. However, r_1 does not converge and remains in the deadlock scenario for the complete simulation time, while r_3 is able to converge eventually, at which point the trajectories of r_1 and r_3 start to diverge.

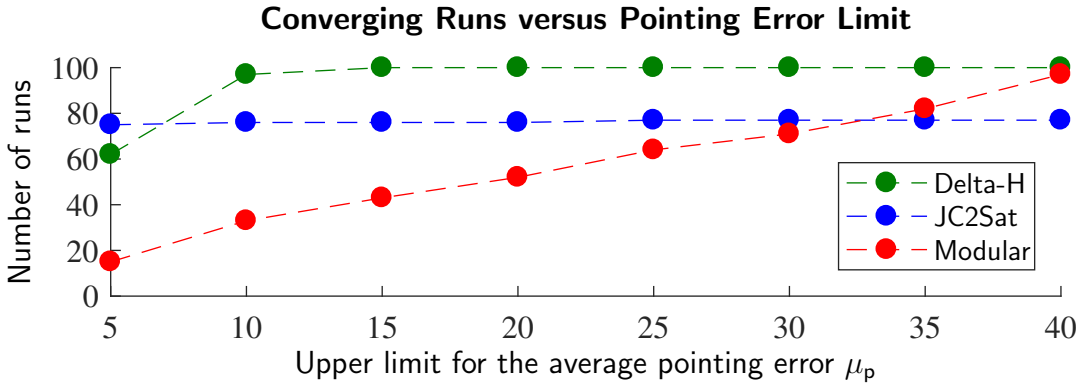

Figure 5.3: The number of testruns for which the average pointing error μ_p over the last 500 s lies below a certain upper limit.

The Modular Controller shows convergence for all 100 simulation runs. The pointing error after convergence varies greatly. The majority of runs converge to a pointing error of less than 45°, and the operation point of the angular velocity is obtained.
Table 5.4 shows a summary for the results of the Monte Carlo simulation. For each controller an average pointing error μ_p over the last 500 s in the Monte Carlo simulation is calculated. The value $\mu_{\mathrm{p},n}$ shows the mean value of these average pointing errors over all $n = 100$ simulation runs, and $\sigma^2_{\mu_{\mathrm{p},n}}$ is the corresponding variance. Furthermore, the number of runs for which the average pointing error μ_p converges to a value lower than 5° and 10° respectively is given, which are the pointing errors limits specified by the MOVE-II requirements, as well as the total number of the converging testruns. Figure 5.3 illustrates this data. The graphs show for each of the nonlinear controllers how many runs converge to a pointing error of less than a specific value μ_p.

Controller	$\mu_{\mathrm{p},n}\,[^\circ]$	$\sigma^2_{\mu_{\mathrm{p},n}}\,[\mathrm{deg}^2]$	Runs with μ_p below 5°	Runs with μ_p below 10°	Converging runs
Delta-H	4.46	4.98	62	97	100
JC2Sat-FF	4.63	235	75	76	78
Modular	20.6	253	15	33	100

Table 5.4: Statistics of the Monte Carlo simulation with $n = 100$ testruns for the nonlinear controllers.

5.6.2 Long-Term Behavior and Noise Sensitivity

This section analyzes the long-term behavior of the nonlinear controllers and compares them to the behavior of the Default Controller and the Extended LQR Controller.
A simulation with standard initial conditions is performed for the time of four orbits. Figure 5.4 shows the results. Part (a) shows the pointing error e_p for a simulation with standard settings. Also the estimation error e_e is presented. This is the angle between the true sun vector $\mathbf{s}^\mathrm{b}$ and the sun vector $\mathbf{s}^\mathrm{b}_\mathrm{meas}$ that is obtained by the satellite through sensor measurements and, if activated, processed by the EKF.
Part (b) repeats the simulation from (a) but with deactivated sensor models. This implies $e_\mathrm{e} = 0$ and the satellite obtains the environment through ideal sensors, where the sun vector is available even during eclipse. All other disturbances discussed in Section 2.6 are still simulated. This setup is not realistic, but a comparison to the scenario with activated sensor models reveal how sensor distortions affect the control algorithm. Furthermore a best case approximation can be obtained for each controller in this idealized scenario.
In Figure 5.4 (a) in the estimation error diagram, the time when the satellite is in eclipse is visualized by gray areas during this time span. It can be seen that the

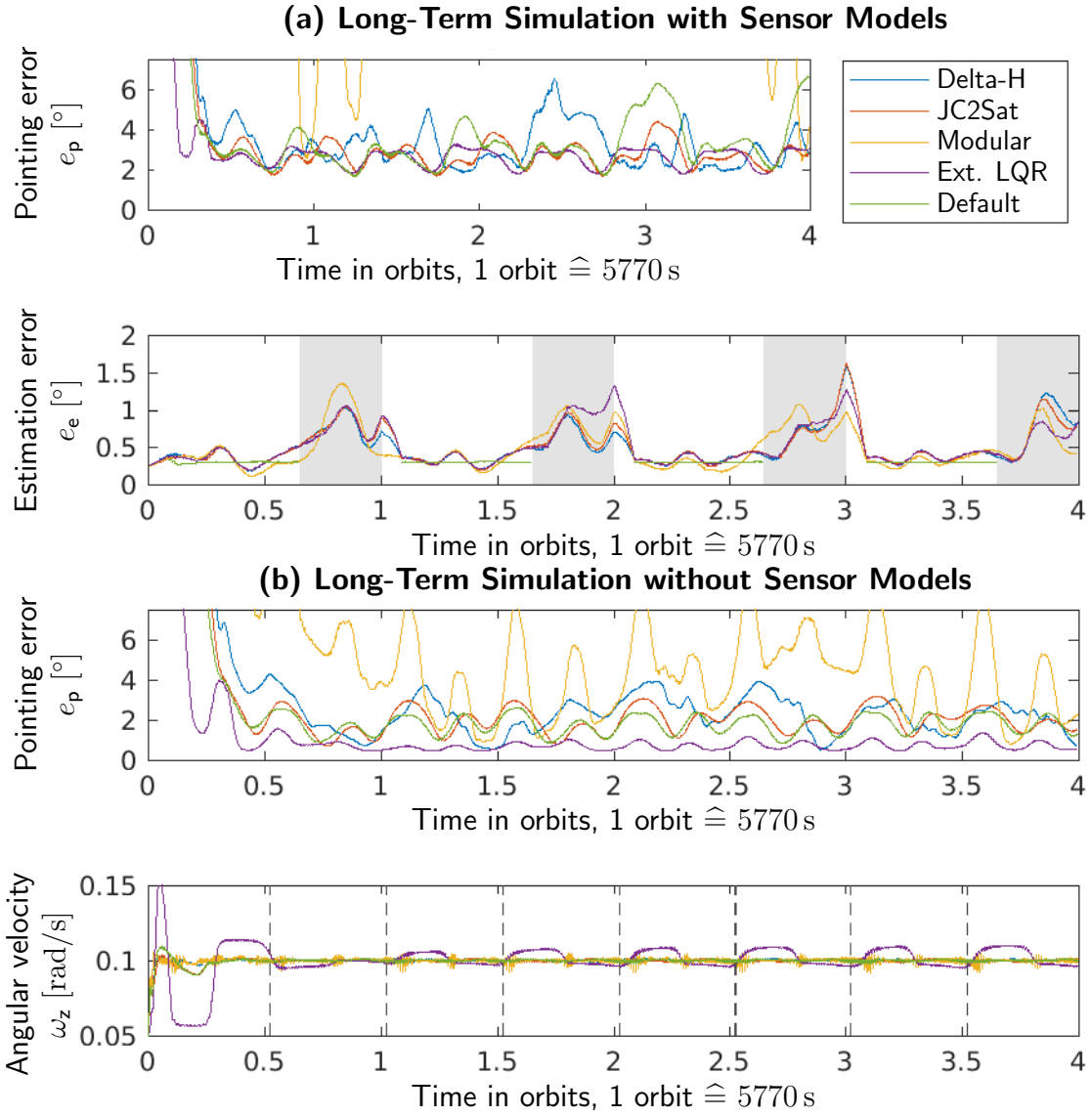

Figure 5.4: Results of the long-term simulation (a) with and (b) without sensor models. The grayed out area visualizes the time during eclipse, and the vertical dashed lines visualize the time points when the satellite passes close by to the poles of Earth. The legend is valid for both parts of the figure. The graphs for the pointing error e_p and the estimation error e_e are smoothed out using an moving average filter.

estimation error centers around $e_e \approx 0.25°$ when the sun is visible and peaks at about $e_e \approx 1.5°$ in eclipse.

	With sensor models		Without sensor models	
Controller	μ_p [°]	σ_p^2 [deg^2]	μ_p [°]	σ_p^2 [deg^2]
Delta-H	3.3	2.3	1.4	0.60
JC2Sat	3.0	2.0	1.8	0.29
Modular	10	99	3.1	5.6
Extended LQR	3.0	0.38	0.60	0.073
Default	5.8	4.3	1.8	0.73

Table 5.5: Statistics of the long-term simulation. For both a simulation with and without sensor models the average pointing error μ_p and the corresponding varricance σ_p^2 is given.

The graphs for the pointing error, which is sampled at 10 Hz, in Figure 5.4 is smoothed with a moving average filter of window size 5000, while the estimation error is sampled at a frequency of 100 Hz and a moving average filter with window size 5e4 is applied.

Table 5.5 accompanies Figure 5.4 and provides the average pointing error μ_p and the corresponding variance σ_p^2 over the last two orbits. The variance of the pointing error is a measure of how much the pointing error is fluctuating, caused through nutation and other disturbances.

From the results it can be seen that the Extended LQR Controller shows the lowest average pointing error and the lowest variance. All nonlinear controllers show a higher variance and two of them show a higher pointing error. The Extended LQR Controller is sensitive to sensor noise however. The simulation without sensor models leads to a significantly reduced average pointing error and variance. From Figure 5.4 (b) it can be seen that the Extended LQR Controller obtains such a low pointing error by a relaxed handling of the angular velocity. This results in an observable fluctuation of ω_z. This is exactly what is intended by designing this controller. It is also obtained that a change in ω_z occurs four times per orbit, where the changing points coincide with the points where the satellite passes close by the poles and the equator.

The Modular Controller has the highest average pointing error and the highest variance among the analyzed controllers. In the presence of sensor noise the pointing error is too high to be visible in Figure 5.4 (a). The behavior of this controller in presence of sensor noise can be seen in Appendix B.3. In absence of sensor noise however, the controller fulfills the must requirement of MOVE-II. The pointing error stays below 10° for the time of three orbits.

The JC2Sat Controller shows a similar behavior to the Extended LQR Controller in the presence of sensor noise. It shows a reduced average pointing error and variance compared to the Default Controller. It also fulfills the should requirement of MOVE-II, since the pointing error stays below 5° after convergence.

The Delta-H Controller shows a slightly increased pointing error and variance compared to the JC2Sat Controller. It fulfills the must requirement, but not the should requirement for MOVE-II. Between orbits 2 and 3 it shows a peak in the pointing error of about 6°. Using the Delta-H Controller instead of the Default Controller results in a decreased average pointing error and variance.

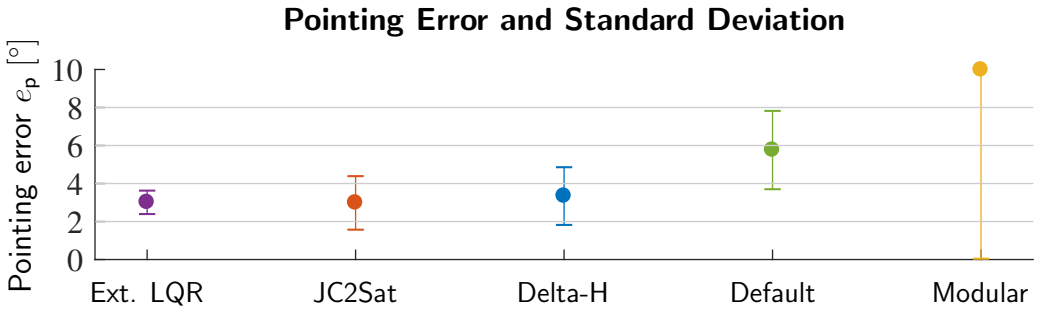

Figure 5.5: Visualizing the pointing error and standard deviation of the previously analyzed controllers.

Figure 5.5 wraps up the results. The controllers are sorted from left to right with ascending average pointing error. The error bars visualize the standard deviation of the average pointing error. Due to the big value of the standard deviation for the Modular Controller, the corresponding error bar is cut off at the upper part.

5.6.3 HiL Verification of the Delta-H Controller

In the previous section we stated that the Delta-H Controller has a smaller average pointing error and a lower pointing error fluctuation than the Default Controller. Furthermore, the Delta-H Controller solves the problem of not converging runs with arbitrary initial conditions. It is therefore an appropriate candidate to be implemented on the MOVE-II satellite. In order to verify if the control law is able to run on the flight hardware with limited calculation capacity, less numeric precision and in interaction with the other firmware parts, a HiL simulation is performed. A detailed description of the HiL setup is provided by Kiesbye [Kie17].
The control law of the Delta-H Controller is first implemented in C++. To check the correctness of the implementation, this C++ function is embedded into the Matlab simulation, where it replaces the Matlab implementation of the Delta-H Controller within a SiL simulation. Then the C++ implementation is embedded into the existing firmware of MOVE-II, which is flashed onto a hardware setup which equals the flight hardware. Via a network connection, the Matlab simulation sends the virtual sensor measurements to the HiL hardware, which calculates the control currents and sends them back to the Matlab simulation.
Figure 5.6 illustrates the results, comparing the pointing error and the Z component of the angular velocity for the Matlab implementation and for the C++ implementation in a SiL and HiL setup. For all simulations the residual magnetic dipole

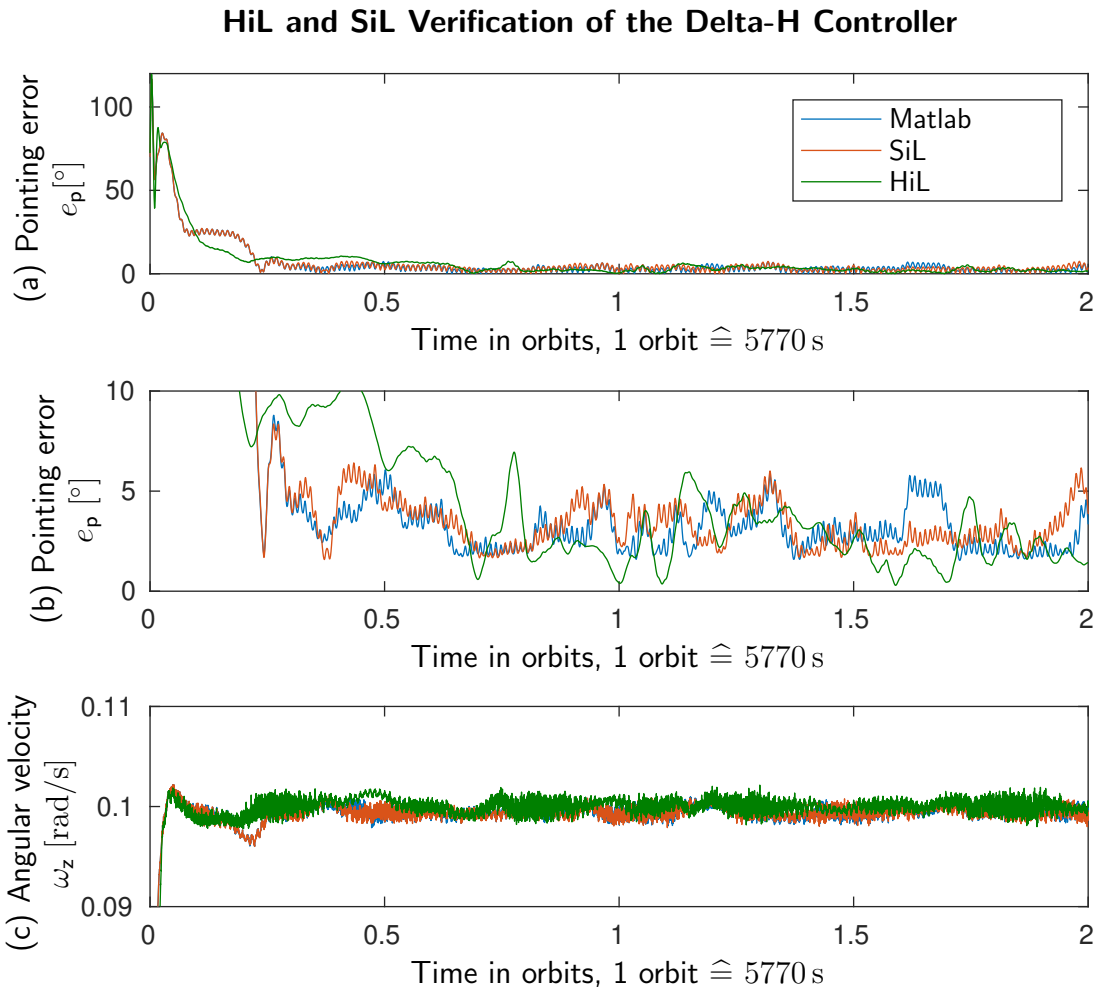

Figure 5.6: Comparing the Matlab implementation of the Delta-H Controller with the C++ implementation in a SiL and HiL setup.

compensation is used, with an overcompensation of the residual magnetic dipole by 20% in order to simulate the estimation uncertainty of the dipole.
Part (a) of Figure 5.6 shows the pointing error over the time of 2 orbits. It becomes clear that all implementations show a similar behavior. Part (b) shows a magnified segment of part (a), where the pointing error is smoothed with a moving average filter of window size 500 for data points sampled at 10 Hz. The HiL simulation shows a smoother pointing error than the other simulations. This difference is caused by the network connection, which delays the transmission of the measurements to and the transmission of the control dipole moment from the HiL hardware significantly. The delays are in the range of several hundred milliseconds, whereas the update cycle of the controller is 5 Hz. Apart from that one obtains that the Delta-H Controller performs as expected being a part of the firmware on the flight hardware.

No significant differences between the Matlab implementation and the C++ implementation in the SiL and HiL setup can be obtained. However, it is remarkable that the pointing error trajectories start to diverge after initial convergence for the Matlab implementation and the SiL setup. This difference is not significant but observable. The only reason to explain this is the sensitivity of the Delta-H controller to numerical errors close to the operation point. Other reasons cannot be identified. Usual causes for randomness can be excluded. The seeds for the random generators of the sensor models are equal in both setups and no additional delays are introduced in the SiL setup.
Another aspect can be tested concerning the HiL setup. The MOVE-II firmware features a daemon which sets control and sensor parameters for the ADCS at runtime. In the implementation for the HiL setup of the Delta-H Controller, the parameters ω_s, $k_{\mathrm{D},1}$, s_D and b_D, as well as the value for the maximum control moment $m_{\mathrm{c,max}}$ are realized as variables, which can be dynamically updated by the daemon. Simulations which vary these parameters during a simulation run show to be successful. It is therefore possible during one simulation to switch from the Delta-H Controller with a strong focus on the angular momentum error in the BF to the Global Spin Rate Controller, which solely controls the sun pointing and nutation damping. Also, the spin rate for spin stabilization can be increased during simulation, and the value for which the controller becomes instable due to strong nutation torques can be determined experimentally within the HiL setup. It is obtained that values of $\omega_\mathrm{s} > 0.25\,\mathrm{rad/s}$ start to become critically and should be avoided, due to the growing nutation torque. At this limit, the satellite starts to flip the pointing axis away from sun.
The capability of updating control parameters on-orbit via the daemon is valuable for the satellite operators. In case of unforeseen circumstances, the operators are able to adjust the control law for the current needs. Furthermore, if the satellite behaves on-orbit differently than in the simulations, it is easy to adjust for this with a simple update command of the daemon. The implementation proves to be compatible with the other parts of the MOVE-II firmware and also is stable in the presence of reduced numeric precision and increased delays. It is therefore suitable to be implemented on the actual flight hardware.

Chapter 6

Discussion

This section discusses the obtained results and provides an interpretation to the observed phenomena. Major insights are presented and elaborated.

One statistic we state throughout the thesis for every simulation is the variance of the average pointing error. This variance describes how the instantaneous pointing error fluctuates. This fluctuation is an oscillation of the pointing error around its average value and can only be observed in graphs which are not smoothed with a moving average filter, due to the high frequency of oscillation. Figure 4.1 shows the fluctuation of the pointing error for the Default Controller and the Extended LQR Controller. We observe that the amplitude of this oscillation is lower for the Extended LQR Controller than for the Default Controller. The reason is that the Extended LQR Controller utilizes the real inertia tensor with the non-zero diagonal entries for linearization of the model, while the Default Controller assumes a diagonal inertia tensor. Thus the Extended LQR Controller includes the nutation disturbance in its model and is able to counteract it to some extent. However, it cannot completely counteract it because of the actuation limitation inherent to the magnetic actuation. The nonlinear controllers, as well as the Default Controller, show an increased pointing error variance, since they do not include the non-diagonal inertia tensor in their control models.

The Lyapunov functions of the nonlinear controllers Delta-H and JC2Sat feature both, a term for nutation compensation which penalizes a deviation from zero for the the angular velocity components perpendicular to the desired spin axis, and a term which penalizes the deviation from the desired angular momentum. In order to rotate about a fixed spin axis which is not identical with a principle axis of inertia, the angular momentum has to rotate around this spin axis. A constant angular momentum is therefore not desired for the controller of MOVE-II. This is another reason, why the nonlinear controllers show an increased pointing error fluctuation as well, with a variance of $\sigma_{\mathrm{p}}^2 \approx 2\,\mathrm{deg}^2$. This value is significantly higher than the variance for the Extended LQR Controller ($\sigma_{\mathrm{p}}^2 = 0.38\,\mathrm{deg}^2$), which accounts for the nutation torque, but it is lower compared to the Default Controller ($\sigma_{\mathrm{p}}^2 = 4.2\,\mathrm{deg}^2$).

Constructing a Lyapunov function based on a fixed spin axis instead of a fixed angular momentum could potentially decrease the pointing error fluctuation.

In Section 5.2 we state that we consider the nutation torque as a disturbance and not as part of the system equations. If we consider the nutation torque as part of the system, however, the operation point is not an equilibrium point any more. Furthermore, since the desired spin axis of the Delta-H Controller is the major principle axis, its Lyapunov function is strictly speaking not a Lyapunov function for the use case of MOVE-II with the spin axis close to the intermediate principal axis. However, the results show that the pointing accuracy and the convergence rate are increased for those two controllers compared to the Default Controller nonetheless, and therefore the assumption of the nutation torque being a disturbance is justified. This also justifies the assumption from Section 5.3 that choosing a spin axis which is not the major principle axis has no big impact in the behavior of the controller due to the similar values of the principle moments of inertia.

Another observation is addressing the convergence behavior of the JC2Sat Controller. We observe that this controller ends up in a deadlock for some initial conditions, where the angle between the magnetic field and the control error vector is zero and thus the controller cannot generate a control moment. In Figure 5.2 the simulation runs r_1 and r_3 proceed on a similar trajectory for nearly 1.5 orbits. With proceeding time, the trajectories become even more similar, meaning that the distance between the Z component of the angular velocities ω_z shrinks and the pointing error trajectories start to overlap. However, at one point the trajectories start to diverge suddenly with great steps. This leads to the insight that this controller shows a chaotic behavior, where similar trajectories lead to completely different results. This property and the fact that this controller ends up in 22% of the cases in a deadlock leads to the understanding that the JC2Sat Controller is not reliable in terms of global convergence and therefore is not suited for acquiring the desired attitude from an arbitrary initial attitude.

For this task, the Delta-H Controller seems suitable with a convergence rate of 100%. This controller features an increased convergence rate, an increased pointing accuracy and a reduced pointing error fluctuation compared to the Default Controller. A HiL simulation shows that it also works well on the flight hardware and in interaction with the ADCS firmware. Therefore, it is suitable as a replacement for the Default Controller, although the Extended LQR Controller and the JC2Sat Controller have a slightly lower average pointing error after convergence.

The comparison simulations with and without activated sensor models reveal which controllers are prone to sensor noise. Table 5.5 provides the average pointing errors for both cases. One obtains that the JC2Sat Controller is least susceptible to noise, since the ratio between the average pointing error for the simulation with and without the sensor models is low. The noise sensitivity of the Delta-H controller is higher compared to the JC2Sat Controller, but it is still in the lower range. In contrast to this, the two linearized controllers, the Extended LQR Controller and the

Default Controller, show a high sensitivity to noise compared to those two nonlinear controllers.
Chapter 4 presents several extensions to the state feedback controller and develops a new LQR gain. Due to available interfaces on the current ADCS firmware, the extensions for correcting the dipole moment and using the EKF can be activated via uplink commands. The new gain can also be uploaded as a parameter to the state feedback controller. The other extensions need a firmware update in order to be used on the actual flight hardware. It is possible to upload an updated firmware onto the satellite, this requires a stable and continuous communication link, however. This cannot always be achieved, thus updating parameters like the gain matrix or activating the EKF with a short command is the preferred way of operating the satellite, rendering these extensions more valuable to the satellite operators. The satellite can benefit directly from uploading the new LQR gain since this shows to reduce the pointing error fluctuation. It also fulfills the design goal to achieve a decreased pointing error at the cost of a less accurate spin rate. However, as long as the satellite spins around the desired axis, the exact spin rate is not important for the mission of MOVE-II.

Chapter 7

Conclusion

The goal of this thesis is to find the weaknesses of the existing Default Controller and to develop new control strategies, which fix these weaknesses and which improve the overall pointing accuracy.

It is found that the Default Controller fulfills the must requirement for the pointing accuracy in simulations with the target orbit, but for orbits deviating from the target inclination $i \approx 98°$, the must requirement is violated by a pointing error becoming greater than 10°. Another weakness of the Default Controller is the unreliability in terms of convergence starting from arbitrary initial conditions. A Monte Carlo simulation reveals that the controller converges only in 49% of the cases within 25 min, and shows a converging trend only in 69% of the cases. The other cases show an oscillating behavior or converge to an attitude pointing the solar panels away from the sun.

To solve the shortcomings of the Default Controller, a new LQR gain for the state feedback controller is developed and the following extensions to the control algorithm are proposed. An operation point switching strategy is proposed, which allows to spin around the Z-axis with a positive or negative spin value, depending on which state is obtained more easily from the initial condition. The residual magnetic dipole moment compensation is presented and extended to compensate for the duty cycle of MOVE-II. Furthermore, a strategy to actuate at low currents and the use of the recently developed EKF are suggested. The new gain together with these extensions for the state feedback controller are summarized under the term Extended LQR Controller.

With the Extended LQR Controller, the should requirement for the pointing error is fulfilled. The pointing accuracy is increased by 2.8° in average and the instantaneous pointing error fluctuation is more than halved. This fluctuation is mainly caused by nutation torques due to the spin axis not being a principle axis of inertia. A Monte Carlo simulation shows that the Extended LQR Controller converges twice as fast as the Default Controller, considering the fast converging cases only. However, the probability of convergence within 25 min cannot be increased with the Extended LQR Controller and stays at 49%. The rate of runs which show a con-

verging trend increases from 69% to 80%, compared to the Default Controller. This means however, there is still a chance of 20% that the Extended LQR Controller does not converge after launch or after a system blackout.
The main reason to evaluate nonlinear control laws is to find a controller which is able to converge to the desired operation point starting from an arbitrary initial condition. This is an important property of the controller for a CubeSat, since the initial condition of the satellite after launch cannot be influenced. From the three analyzed controllers in Chapter 5, only the Delta-H Controller shows this property. It converges from an arbitrary initial condition in 100% of the cases within half an orbit. The average pointing error of the Delta-H Controller is increased by 0.3° with respect to the Extended LQR Controller, while the instantaneous pointing error fluctuation is increased by a factor of 2.5. The nonlinear controller JC2Sat shows slightly less fluctuation and a slightly increased pointing accuracy compared to the Delta-H Controller, but shows convergence only in 78% of the runs in the Monte Carlo simulation.
Furthermore, a Software-in-the-Loop and a Hardware-in-the-Loop simulation prove the Delta-H Controller to be suitable for implementation on the flight hardware. No significant differences are obtained between the Matlab and the C++ implementation. The controller also shows to run stable on the flight hardware in the presence of large delays and decreased numerical precision. Control parameters configurable at run-time during on-orbit maneuvers make this controller implementation interesting for satellite operators.
From these results we draw the conclusion that for converging towards the operating point, the Delta-H Controller is suited best. This controller also works as expected in a HiL simulation. In order to stabilize the operating point, the Extended LQR Controller is advantageous, however. Chapter 8 states the limitations to this conclusion and suggests further work to be done.

Chapter 8

Future Work

This thesis does not investigate failure scenarios, when one or more coils or sensors are not working properly. The results of this thesis are based on the assumption that the coils and sensors work as expected within their accuracy. However, fail safety is of interest for a spacecraft which can only be controlled remotely. If a coil or sensor is not working any more, it cannot be fixed and therefore it is important to know what the effect of a faulty coil or sensor is and if the impact onto the control behavior can be reduced. This issue needs to be addressed in future controller analyses.

Similar concerns are true for unexpected disturbances. For the linearized controllers, the impact of a residual magnetic dipole moment is analyzed. The results yield a good approximation about what impact to the controller behavior a certain disturbance magnitude causes, and how the pointing accuracy decreases with increasing disturbance magnitude. For the nonlinear controllers however, no such analysis is performed within this thesis. Therefore, no statement can be made about the sensitivity to disturbances for the nonlinear controllers. However, this is an important decision criterion, when choosing a controller for stabilizing the operation point. A controller with less disturbance sensitivity is preferred. Future work needs to analyze selected controllers with respect to different disturbances and to figure out, up to which disturbance magnitude the must requirement for the pointing error still can be fulfilled.

The Conclusion states that the Delta-H Controller is suited best for global convergence and the Extended LQR Controller is suited best for stabilizing the operation point. If these two control laws are implemented on the flight hardware, this needs still the manual decision about when to activate which controller. Monitoring a satellite continuously and being able to send commands to the satellite at any time is difficult, for many CubeSat teams this is even impossible. It is therefore suggested to combine those two controllers in a single hybrid controller. This can be implemented similar to the operation point switching strategy used for the Extended LQR Controller. An algorithm measures the distance of the current state from the state of the operation point. When the satellite is far from the operation point, the Delta-H Controller is activated. When the satellite state is converged close enough

towards the operation point, the controller is switched in favor of the Extended LQR Controller. What exactly far and close enough means, as well as an appropriate margin for a hysteresis between the switching points, need to be investigated by further simulations and observations.
Inspired by the neuronal network controller proposed by Sivaprakash and Shanmugam [SS05], another control concept can be investigated as well. This thesis shows that the linearized LQR controller has the lowest pointing error and fluctuation when close to the operation point, and also has the lowest convergence time, considering successfully converging runs only. Therefore, the state space can be divided into multiple partitions, for which each a linearized LQR controller is developed. These linearized controllers generate data to train the neuronal network, which then is able to interpolate between those linearized control laws and to mimic the appropriate linearized controller within each partition of the state space. It is suggested to perform an analysis in order to investigate, if this control concept decreases the time for global convergence. Also, some kind of stability analysis for this kind of control needs to be developed.

Appendix A

Matlab Simulation

In order to analyze the system and to verify the controller, a Matlab Simulink simulation is used. The the core part of this simulation was created by the ADCS team of MOVE-II. This simulation consists of a Simulink model which can be seen in Figure A.1, and of a Matlab script which allows automated testruns and variation of several parameters. Parameters which can be modified include orbit properties (see Chapter 2.2 for more details), control gains, controller flags, mechanic properties of the satellite and initial conditions like initial attitude and initial angular velocity.
In the following, all top level blocks and its functionality in the simulation are be explained. For better legibility, block and signal names are printed in italics.

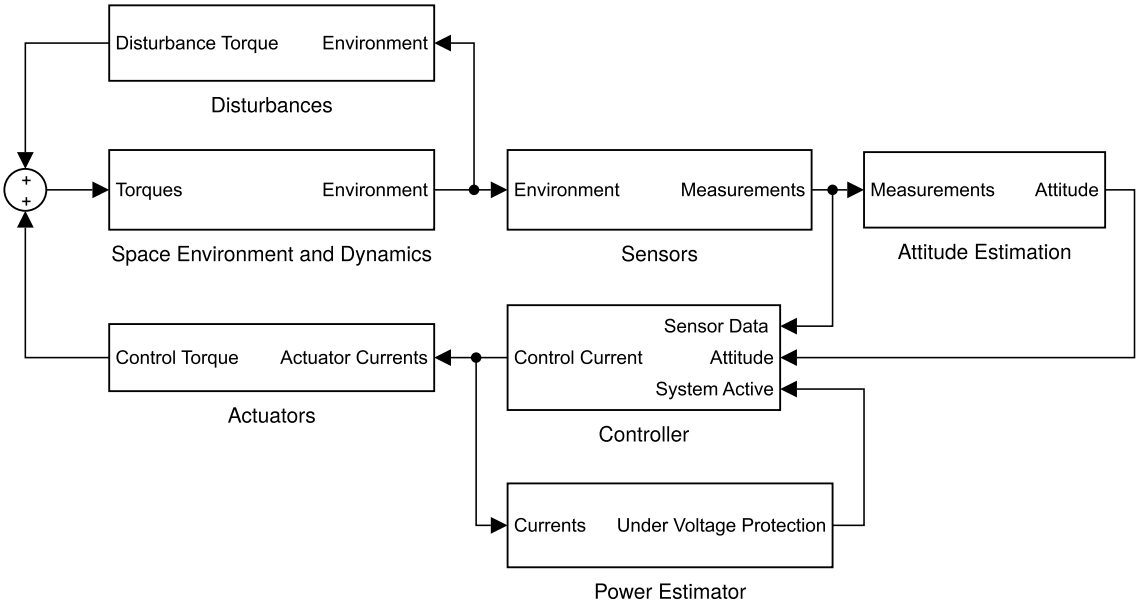

Figure A.1: Simulink Model used for analyzing the system and verifying the control strategies.

Space Environment and Dynamics

In this block, the nonlinear dynamic and kinematic system equations are implemented. It receives the torque vector acting on the satellite and calculates the current attitude and angular velocity of the satellite.

This block also includes an orbit propagator, which calculates the track of the satellite around the earth, given the orbit characteristics and an initial position. The commonly used SGP4 (Simplified General Perturbations) model is used for this purpose.

The position above the Earth affects the magnetic field to which the satellite is exposed. To determine the magnetic field, the IGRF-12 (International Geomagnetic Reference Field, 12th generation) model is used.

Furthermore there is a sun model which calculates the true sun vector in the BF for the satellite.

All this data is bundled together in the *Environment* bus and is passed forward to the *Disturbances* and *Sensors* blocks.

Sensors

The *Sensors* block receives the environmental data and applies the specific sensor characteristics of the sensors of MOVE-II to it.

First, the data is sampled with a sample rate of 0.2 s. Then the noise and bias components for every sensor is added. The bias and noise characteristics of the sensors are determined from the flight model of MOVE-II, so that this simulation accurately imitates the real sensors. The sensor characteristics are discussed in more detail in [Meß18].

MOVE-II has three different types of sensors available: A sun sensor which measures the sun vector in body coordinates, a magnetic field sensor, and a gyroscope, which measures the angular velocity.

This sensor data data is bundled in the *Measurement* bus, which is used by the *Attitude Estimation* and the *Controller* block.

Attitude Estimation

The *Attitude Estimation* block implements the EKF (Extended Kalman Filter) described by Messmann [Meß18] and uses the *Measurements* of the *Sensors* block to calculate the estimated attitude. The attitude is represented as a unit quaternion. This allows to rely on a more accurate information about the current attitude than if only the noisy and biased sensor measurements were available.

This information then is forwarded to the *Controller* block.

Controller

The *Controller* block can rely on both, the sensor *Measurements* and the estimated *Attitude*. Depending on the control strategy, it calculates and outputs the *Control Currents*.
In this block the various different control strategies are implemented. It features the Default Controller as well as the other controllers evaluated within this thesis. They can be activated, deactivated and configured with the simulation script via the parameter settings. For example, the Default Controller does not use the EKF and relies only on sensor measurements to determine the satellite's attitude. However, by setting the corresponding flag, the controller is made to use the estimated *Attitude* instead of the noisy sensor *Measurements*.
This block also offers an interface to which the flight controller hardware can be connected. With this possibility, a HiL (Hardware in the Loop) simulation can be performed. Before new software is permitted to be uploaded onto the satellite, it must prove to run stable on the flight controller hardware. Thus, for promising control strategies, it is mandatory to implement these strategies in C++ and to verify them in a HiL environment.

Power Estimator

The *Power Estimator* keeps track of the consumed energy of the satellite. The major components are the *Control Currents*. It also keeps track of the power generated by the solar panels. A proper sun pointing ensures more power generation due to the good alignment of the solar panels to the sun and thus allows a higher current consumption of the satellite.
With this block it can be verified, if a controller is power positive. If the simulated battery becomes empty, the *Power Estimator* deactivates the *Controller* block to simulate a blackout. The satellite then starts to tumble freely and to recharge its battery. When the battery reaches a threshold level, the *Controller* block gets activated again and it takes over the control of the satellite.
Like the Controller block it is possible to simulate this block also in a HiL simulation. A solar array simulator developed by Nagy [Nag18] calculates the information about the sun intensities of the solar panels and charges a real battery. The systems of the satellite including the control hardware is connected to this battery. If this battery is empty, a real blackout is emulated.

Actuators

This block models the actuator characteristics of the actuation coils. The *Control Currents* flow through the coils subject to physical properties and produce a magnetic field. This magnetic field then interacts with the Earth magnetic field and produces a *Control Torque* acting on the satellite. This torque is forwarded to the *Space Environment and Dynamics* block to close the loop.

Disturbances

Several *Disturbance Torques* affect the motion of the satellite in space. They depend strongly on the current position and attitude of the satellite. These disturbance torques are modeled in this block and are fed back to the *Space Environment and Dynamics* block.
Chapter 2.6 discusses the disturbance models in detail.

Appendix B

Gain Selection of the Nonlinear Controllers

This appendix presents the rationals behind the selection of the gains and tuning parameters for the nonlinear controllers. Section B.1 deals with the Delta-H Controller, Section B.2 presents results for different gains of the JC2Sat Controller and Section B.3 shows the results for different modes of the Modular Controller.

B.1 Parameter for the Delta-H Controller

The Delta-H controller features only one relevant parameter which can be tuned. This is the value $k_{\mathrm{D},1} \in [0; 1]$, which defines the relation of how much the sun pointing error is weighted versus the weighting of the angular velocity errors. A value of $k_{\mathrm{D},1} = 1$ means, only the sun pointing error is considered, and a value of $k_{\mathrm{D},1} = 0$ means, only the angular velocity errors are considered in the control law. Figure B.1 shows the pointing error for different values of $k_{\mathrm{D},1}$. The pointing error is sampled at 10 Hz and smoothed with a moving average filter of window size 5000. One obtains that for values $k_{\mathrm{D},1} \geq 0.8$, the controller does not converge to a small pointing error. Values of $k_{\mathrm{D},1} \leq 0.001$ result in a high pointing error, since the weighting of the pointing error gets too low and the controller focuses more on the angular velocity errors. Values of $k_{\mathrm{D},1}$ somewhere in between show a converging behavior with a low final pointing error. However, one obtains that low values, for example $k_{\mathrm{D},1} = 0.01$, tend to converge faster to a low pointing error.

B.2 Gains for the JC2Sat Controller

The JC2Sat Controller has three different gains which can be tuned, within the limitations of (5.18), (5.19) and (5.20). The gain $k_{\mathrm{J},0}$ controls the magnitude of the control dipole moment. It is selected to ensure reasonable magnitudes of the control

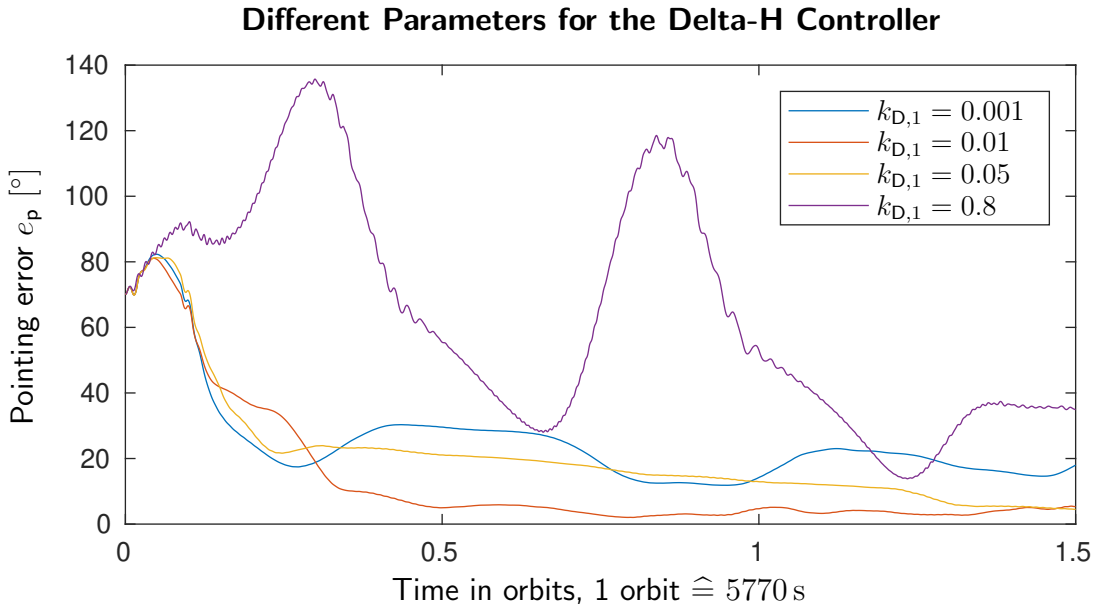

Figure B.1: Simulating the Delta-H Controller with different values for $k_{D,1}$. The graphs are smoothed with a moving average filter.

currents during and after convergence. Currents between 100 mA and 300 mA are considered to be reasonable for MOVE-II.

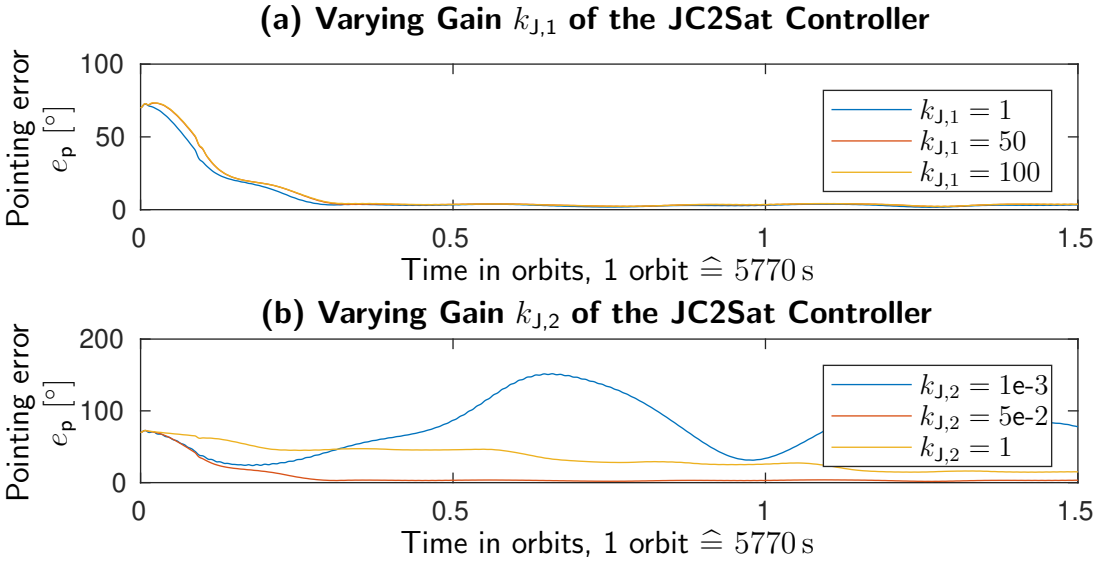

Figure B.2: Simulating the JC2Sat Controller with different gains. Part (a) varies $k_{J,1}$ while keeping the other parameter constant at $k_{J,2} = 0.05\,\mathrm{kg\,m^2}$. Part (b) varies $k_{J,2}$, expressed in the unit $\mathrm{kg\,m^2}$, while having $k_{J,1} = 2$. The graphs are smoothed with a moving average filter.

In order to select the gains $k_{J,1}$ and $k_{J,2}$, several simulation series are performed. Figure B.2 shows the result of two series. The graphs are smoothed with a moving average filter of window size 5000, while the pointing error is sampled at a frequency of 10 Hz. Part (a) of the figure illustrates that the choice of $k_{J,1} > 1$ has no impact on the general behavior of the controller. However, values of $k_{J,1}$ greater than 1 are required in order to converge to the desired spin axis, as shown by De Ruiter [dR11]. Part (b) of the figure evaluates the impact of different values for $k_{J,2}$. This gain influences the nutation damping. It is obtained that low values of $k_{J,2}$ are counterproductive for convergence and cause a high pointing error. On the other hand, high values increase the convergence time. The lowest pointing error and the fastest convergence is obtained with values in the magnitude of 10^{-2} kg m^2 for $k_{J,2}$.

B.3 Modes of the Modular Controller

This section evaluates the modes of the Modular Controller, which are described in Section 5.5 and defined by Table 5.3. Figure B.3 presents the pointing error, using different modes. The graphs are smoothed using a moving average filter of window size 5000, and the pointing error is sampled at 10 Hz.

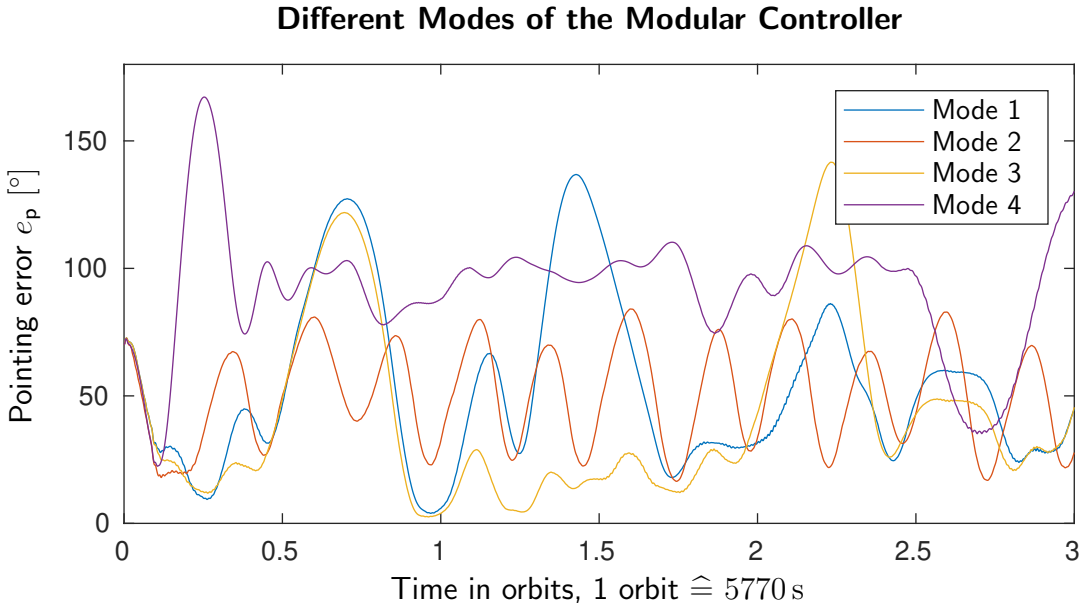

Figure B.3: Simulating different modes of the Modular Controller. The graphs are smoothed with a moving average filter.

It can be obtained that all modes start converging equally fast. However, after the pointing error graphs diverge, each mode starts to show a different characteristic behavior. Mode 4, which utilizes the combined control moment only, is not able to attain a low pointing error. With Mode 2, the pointing error is enclosed within the

range of 15° to 90°, but oscillating with a high amplitude and consistent frequency. This behavior increases the average pointing error.
Mode 1 and 3 are able to reach low pointing errors, however they show high peaks up to 140° occasionally. It is found that Mode 3 less frequently shows peaks and reaches a lower pointing error between the peaks.

List of Figures

List of Tables

Acronyms

ADCS	Attitude Determination and Control System.
BF	Body Frame.
CDR	Critical Design Review.
COM	(Satellite) Communication System.
ECI	Earth Centered Inertial Frame.
EKF	Extended Kalman Filter.
HiL	Hardware in the Loop.
IGRF	International Geomagnetic Reference Field.
LQR	Linear Quadratic Regulator.
MOVE	Munich Orbital Verification Experiment.
RAAN	Right Ascension of the Ascending Node.
SGP	Simplified General Perturbations Model.
SiL	Software in the Loop.
UTC	Coordinated Universal Time.
WARR	Wissenschaftliche Arbeitsgemeinschaft für Raketentechnik und Raumfahrt; EN: Scientific Workgroup for Rocketry and Spaceflight.

List of Symbols

$\mathbf{A}$	System matrix.
a	Semimajor axis of the orbit.
$\mathbf{A}_{\mathrm{c}}$	Closed loop system matrix.
A_{eff}	Effective area of one coil winding.
$\mathbf{A}_{\mathrm{f,d}}$	Full system matrix with diagonal inertia tensor.
$\mathbf{A}_{\mathrm{r}}$	Reduced system matrix with true inertia tensor.
$\mathbf{A}_{\mathrm{r,d}}$	Reduced system matrix with diagonal inertia tensor.
α	Tuning parameter for the LQR gain, concerning pointing.
$\mathbf{B}$	Input matrix.
$\mathbf{B}^*(t)$	Time dependent input matrix.
$\mathbf{B}_{\mathrm{r}}^*(t)$	Time dependent, reduced input matrix.
β	Tuning parameter for the LQR gain, concerning spin rate.
C	Magnetic dipole compensation factor.
D	Duty cycle of the actuation system.
$\mathbf{d}(\boldsymbol{\omega}_{\mathrm{bi}}, \boldsymbol{\tau}_{\mathrm{c}})$	Dynamic equations.
e	Orbit eccentricity.
E_0	Orbit epoch.
$\mathbf{e}_{\mathrm{c}}$	Control error vector.
e_{e}	Estimation error.
e_{p}	Pointing error.
γ	Tuning parameter for the LQR gain, concerning control input.
$\mathbf{h}$	Angular momentum.
$\mathbf{I}$	Inertia tensor of the satellite.
i	Orbit inclination.
$\mathbf{i}_{\mathrm{c}}$	Control currents.

$I_{\max}$	Maximum control current.
$I_{\min}$	Minimum control current.
$\mathbf{i}, \mathbf{j}, \mathbf{k}$	Fundamental quaternion units.
J	Cost functional.
$\mathbf{K}$	State feedback gain matrix.
$\mathbf{K}_{\text{def}}$	Gain matrix of Default Controller.
$\mathbf{K}_{\text{LQR}}$	Gain matrix of Extended LQR Controller.
λ	Eigenvalue of a matrix.
M	Orbit mean anomaly.
M_0	Initial mean anomaly at epoch E_0.
m_1, m_2, m_3	Principal moments of inertia.
$\mathbf{m}_{\text{c}}$	Magnetic control dipole moment.
$\mathbf{m}_{\text{comp}}$	Magnetic dipole moment for compensation.
$\mathbf{m}_{\text{E}}$	Magnetic dipole moment of the Earth.
$\mathbf{m}_{\text{res}}$	Residual magnetic dipole moment.
$\mathbf{m}_{\text{s}}(\mathbf{s}, \boldsymbol{\omega}_{\text{bi}})$	Equations of motion.
μ_0	Vacuum permeability.
μ_{B}	Gravitational constant of a celestial body.
μ_{p}	Average pointing error.
$\mu_{\text{p},n}$	Mean pointing error over n simulation runs.
$\mu_{\sigma_{\text{p}}^2}$	Mean value of σ_{p}^2 over several simulation runs.
n_{c}	Number of coil windings per coil.
Ω	Right Ascension of the Ascending Node (RAAN) of the orbit.
ω	Argument of perigee of the orbit.
$\boldsymbol{\omega}_{\text{bi}}$	Angular velocity of the BF with respect to the ECI.
$\boldsymbol{\omega}_{\text{bi,0}}$	Initial angular velocity.
$\boldsymbol{\omega}_{\text{bi,OP}}$	Operation point of $\boldsymbol{\omega}_{\text{bi}}$.
ω_{o}	Orbital rate.
ω_{s}	Spin stabilization rate.
$\omega_{\text{x}}, \omega_{\text{y}}, \omega_{\text{z}}$	Components of $\boldsymbol{\omega}_{\text{bi}}$ expressed in the BF.
$\mathbf{P}$	Solution to the Riccati equation.
$\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$	Principal axes of inertia.
P_{c}	Coil power consumption.

$\mathbf{\Phi}$	State transition matrix.
$\mathbf{\Psi}$	Monodromy matrix.
$\mathbf{Q}$	State weighting matrix for the LQR algorithm.
q_0	Scalar part of the unit quaternion $\mathbf{q}_{\mathrm{ib}}$.
$\mathbf{q}_{\mathrm{ib}}$	Unit quaternion rotating the BF with respect to the ECI.
$\mathbf{q}_{\mathrm{ib},0}$	Initial unit quaternion.
$\mathbf{R}$	Input weighting matrix for the LQR algorithm.
$\mathbf{r}$	Position vector of the satellite.
r_{a}	Radius of apogee.
R_{E}	Radius of the Earth.
$\hat{\mathbf{r}}$	Normalized position vector of the satellite.
r_{p}	Radius of perigee.
$\boldsymbol{\varrho} = (q_1, q_2, q_3)^{\mathrm{T}}$	Vector part of the unit quaternion $\mathbf{q}_{\mathrm{ib}}$.
$\mathbf{S}$	Controllability matrix.
$\mathbf{s}$	Normalized sun vector.
$\mathbf{s}_{\mathrm{OP}}$	Operation point of $\mathbf{s}$.
$s_{\mathrm{x}}, s_{\mathrm{y}}, s_{\mathrm{z}}$	Components of $\mathbf{s}$ expressed in the BF.
$\sigma^2_{\mu_{\mathrm{p},n}}$	Variance of the mean pointing errors over n simulation runs.
σ^2_{p}	Variance of the pointing error.
T	Orbit period.
T_{a}	Coil actuation time within one period T_{am}.
T_{am}	Actuation-measurement period.
T_{s}	System period.
$\boldsymbol{\tau}$	Torque acting on the satellite.
$\boldsymbol{\tau}_{\mathrm{c}}$	Control torque.
$\boldsymbol{\tau}_{\mathrm{dist}}$	Disturbance torque.
$\boldsymbol{\tau}_{\mathrm{d}}$	Drag torque.
$\boldsymbol{\tau}_{\mathrm{g}}$	Gravity gradient torque.
$\boldsymbol{\tau}_{\mathrm{m}}$	Residual magnetic dipole torque.
$\boldsymbol{\tau}_{\mathrm{n}}$	Nutation torque.
$\mathbf{u}$	Control input.
V	Lyapunov function.
$\mathbf{x}$	System state.

$\mathbf{x}_\mathrm{f}$	Full system state.
$\mathbf{x}_\mathrm{f,OP}$	Operation point of $\mathbf{x}_\mathrm{f}$.
$\mathbf{x}_\mathrm{r}$	Reduced system state.
$\mathbf{x}_\mathrm{r,OP}$	Operation point of $\mathbf{x}_\mathrm{r}$.

Notation

xey	Scientific e-notation for $x \cdot 10^y$, where x and y are numbers.
$\mathbf{a} \times \mathbf{b}$	Cross product of two 3-dimensional vectors $\mathbf{a}$ and $\mathbf{b}$.
$\mathbf{a} \circ \mathbf{b} \equiv \mathbf{a}\mathbf{b}^{\mathrm{T}}$	Dot product of two vectors $\mathbf{a}$ and $\mathbf{b}$.
$\|\|\mathbf{v}\|\|$	Euclidean norm of vector $\mathbf{v}$.
$\dot{\mathbf{v}}$	Time derivative of $\mathbf{v}$, expressed with dot on top.
$\mathbf{v}^{\mathrm{T}}$	Transposed vector of $\mathbf{v}$, expressed with superscript T.
$\mathbf{v}^{\mathrm{b}}$	A vector represented in the BF, marked with superscript b.
$\mathbf{v}^{\mathrm{i}}$	A vector represented in the ECI, marked with superscript i.
$[\mathbf{v}\times]$	Cross product operator for a 3-dimensional vector $\mathbf{v}$.
$\mathbf{1}_{x\times x}$	Identity matrix of size $x \times x$.
$\mathbf{0}_{x\times y}$	Matrix of size $x \times y$, filled with zeros.

Bibliography

[ABV+03] L. Alminde, M. Bisgaard, D. Vinther, T. Viscor, and K. Ostergard. Educational Value and Lessons Learned from the AAU-CubeSat Project. In *International Conference on Recent Advances in Space Technologies, 2003. RAST '03. Proceedings of*, pages 57–62, Istanbul, Turkey, 2003. IEEE. doi:10.1109/RAST.2003.1303391.

[ARL16] N. Appel, S. Rückerl, and M. Langer. Nanolink: A Robust and Efficient Protocol for Small Satellite Radio Links. In *Proceedings of the Small Satellites Systems and Services – The 4S Symposium 2016*, Valletta, Malta, May 2016.

[BBDS15] S. Busch, P. Bangert, S. Dombrovski, and K. Schilling. UWE-3, In-Orbit Performance and Lessons Learned of a Modular and Flexible Satellite Bus for Future Pico-Satellite Formations. *Acta Astronautica*, 117:73–89, December 2015. doi:10.1016/j.actaastro.2015.08.002.

[BC09] S. Bittanti and P. Colaneri. *Periodic Systems: Filtering and Control*. Communications and Control Engineering. Springer, London, 2009.

[BD03] S. P. Bhat and A. S. Dham. Controllability of Spacecraft Attitude under Magnetic Actuation. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, volume 3, pages 2383–2388, Maui, Hawaii, USA, 2003. IEEE. doi:10.1109/CDC.2003.1272976.

[BG10] J. Bouwmeester and J. Guo. Survey of Worldwide Pico- and Nanosatellite Missions, Distributions and Subsystem Technology. *Acta Astronautica*, 67(7-8):854–862, October 2010. doi:10.1016/j.actaastro.2010.06.004.

[BH12] J. D. Biggs and N. Horri. Optimal Geometric Motion Planning for Spin-Stabilized Spacecraft. *Systems & Control Letters*, 61(4):609–616, April 2012. doi:doi.org/10.1016/j.sysconle.2012.02.002.

[Cho02] V. A. Chobotov. *Orbital Mechanics, Third Edition*. AIAA Education Series. American Institute of Aeronautics and Astronautics, Reston, VA, USA, January 2002. doi:10.2514/4.862250.

[Cur14] H. D. Curtis. *Orbital Mechanics for Engineering Students*. Aerospace Engineering. Butterworth-Heinemann, Amsterdam; Boston, 3rd edition, 2014.

[dR11] A. de Ruiter. A Fault-Tolerant Magnetic Spin Stabilizing Controller for the JC2sat-FF Mission. *Acta Astronautica*, 68(1-2):160–171, January 2011. doi:10.1016/j.actaastro.2010.07.012.

[ESI+16] H. Ehrpais, I. Sünter, E. Ilbis, J. Dalbins, I. Iakubivskyi, E. Kulu, I. Ploom, P. Janhunen, J. Kuusk, J. Šate, R. Trops, and A. Slavinskis. ESTCube-2 Mission and Satellite Design. In *Proceedings of the Small Satellites Systems and Services – The 4S Symposium 2016*, Valletta, Malta, May 2016.

[GAC+15] S. Gaglione, A. Angrisano, G. Castaldo, P. Freda, C. Gioia, A. Innac, S. Troisi, and G. Del Core. The First Galileo FOC Satellites: From Useless to Essential. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3667–3670, Milan, Italy, July 2015. IEEE. doi:10.1109/IGARSS.2015.7326618.

[GFV02] T. Graversen, M. K. Frederiksen, and S. V. Vedstesen. Attitude Control System for AAU CubeSat, 2002. Master Thesis, Department of Control Engineering, Aalborg University, Aalborg, June 2002.

[HJ06] L. Huang and W. Jing. Correction of Remanent Disturbance Torque. In *2006 1st International Symposium on Systems and Control in Aerospace and Astronautics*, pages 164–168, Harbin, China, January 2006. IEEE. doi:10.1109/ISSCAA.2006.1627603.

[IAG14] IAGA. International Geomagnetic Reference Field, December 2014. Retrived 08.03.2019. URL: https://www.ngdc.noaa.gov/IAGA/vmod/igrf.html.

[Kha02] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle River, New Jeresey, USA, 3rd edition, 2002.

[Kie17] J. Kiesbye. Hardware-in-the-Loop Verification of the Distributed, Magnetorquer-Based Attitude Determination & Control System of the CubeSat MOVE-II, 2017. Master Thesis, Chair of Astronautics, Technische Universität München, Garching, 2017.

[KS72] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. Wiley-Interscience, New York, 1972.

[LAD+15] M. Langer, N. Appel, M. Dziura, C. Fuchs, P Günzel, J. Gutsmiedl, M. Losekamm, D. Meßmann, T. Pöschl, and C. Trinitis. MOVE-II - der

zweite Kleinsatellit der Technischen Universität München. In *Deutscher Luft- und Raumfahrtkongress (DLRK)*, Rostock, Germany, September 2015.

[LCZ+09] C. Louembet, F. Cazaurang, A. Zolghadri, C. Charbonnel, and C. Pittet. Path Planning for Satellite Slew Manoeuvres: A Combined Flatness and Collocation-Based Approach. *IET Control Theory & Applications*, 3(4):481–491, April 2009. doi:10.1049/iet-cta.2008.0054.

[LOH+15] M. Langer, C. Olthoff, J. Harder, C. Fuchs, M. Dziura, A. Hoehn, and U. Walter. Results and Lessons Learned from the CubeSat Mission First-MOVE. In *10th IAA Symposium on Small Satellites for Earth Observation*, Berlin, April 2015.

[MBC+16] D. Meßmann, J. Brügge, M. Cordovi, M. Dötterl, T. Grübler, F. Mauracher, A. Meraner, T. Ohlenforst, S. Plamauer, L. Schmidt, P. Schnierle, M. Seifert, and N. Wetter. MOVE-II Critical Design Review (CDR), ADCS, WARR, Garching, November 2016.

[MC14] F. L. Markley and J. L. Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control*. Number 33 in Space Technology Library. Springer, New York, 2014.

[MCN+17] D. Meßmann, F. Coelho, P. Niermeyer, M. Langer, H. Huang, and U. Walter. Magnetic Attitude Control for the MOVE-II Mission. In *7th European Conference for Aeronautics and Aerospace Sciences (EUCASS)*, Milan, Italy, July 2017.

[Meh78] K. Mehlem. Multiple Magnetic Dipole Modeling and Prediction of Satellites. *IEEE Transactions on Magnetics*, 14(5):8, September 1978. doi:10.1109/TMAG.1978.1059983.

[Meß18] D. Meßmann. Attitude Estimation for the MOVE-II CubeSat. May 2018. Master Thesis, Chair of Astronautics, Technische Universität München, Garching, 2018.

[MGC+17] D. Meßmann, T. Gruebler, F. Coelho, T. Ohlenforst, J. van Bruegge, F. Mauracher, M. Doetterl, S. Plamauer, P. Schnierle, T. Kale, M. Seifert, A. Fuhrmann, E. Karagiannis, A. Ulanowski, T. Lausenhammer, A. Meraner, and M. Langer. Advances in the Development of the Attitude Determination and Control System of the CubeSat MOVE-II. In *7th European Conference for Aeronautics and Aerospace Sciences (EUCASS)*, Milan, Italy, July 2017. doi:0.13009/EUCASS2017-660.

[Nag18] D. Nagy. 3-Channel Solar Array Simulator for CubeSat Power Budget Verification. 2018. Master Thesis, Institute for Measurement Systems and Sensor Technology, Technische Universität München, Munich, September 2018.

[NLG+09] J. F. D. Nielsen, J. A. Larsen, J. D. Grunnet, M. N. Kragelund, A. Michelsen, and K. Kjaer Sørensen. AAUSAT-II, A Danish Student Satellite. *I S A S Nyusu*, 2009.

[OB12] H. S. Ousaloo and A. Badpa. Magnetic Attitude Control System for Spin Stabilized Satellite. In *20th Iranian Conference on Electrical Engineering (ICEE2012)*, pages 1029–1034, Tehran, Iran, May 2012. IEEE. doi:10.1109/IranianCEE.2012.6292504.

[Pro14] The CubeSat Program. CubeSat Design Specification Rev. 13, February 2014. Retrieved 05.03.2019. URL: http://www.cubesat.org/resources.

[RC01] R. Reynolds and G. Creamer. Global Lyapunov Control of Spin Stabilized Spacecraft. In *Flight Mechanics Symposium*. NASA Conference Publication, June 2001.

[RKS+16] M. Rutzinger, L. Krempel, M. Salzberger, M. Buchner, A. Hohn, M. Kellner, K. Janzer, C. G. Zimmermann, and M. Langer. On-Orbit Verification of Space Solar Cells on the CubeSat MOVE-II. In *IEEE 43rd Photovoltaic Specialists Conference (PVSC)*, Portland, OR, USA, June 2016. doi:10.1109/PVSC.2016.7750120.

[SAS16] A. Saeed, S. U. Ali, and M. Z. Shah. Linear control techniques application and comparison for a research UAV altitude control. In *2016 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 126–133, Islamabad, Pakistan, January 2016. IEEE. doi:10.1109/IBCAST.2016.7429866.

[Sha11] N. Shah. A Globally Stable Lyapunov Pointing and Rate Controller for the Magnetospheric MultiScale Mission (MMS). In *IAA Guidance, Navigation, and Control Conference*, Portland, OR, USA, August 2011.

[SKK+14] A. Slavinskis, U. Kvell, E. Kulu, I. Sünter, H. Kuuste, S. Lätt, K. Voormansik, and M. Noorma. High Spin Rate Magnetic Controller for Nanosatellites. *Acta Astronautica*, 95:218–226, February 2014. doi:10.1016/j.actaastro.2013.11.014.

[SL05] E. Silani and M. Lovera. Magnetic Spacecraft Attitude Control: A Survey and Some New Results. *Control Engineering Practice*, 13(3):357–371, March 2005. doi:10.1016/j.conengprac.2003.12.017.

[SMA+11] W. A. Shiroma, L. K. Martin, J. M. Akagi, J. T. Akagi, B. L. Wolfe, B. A. Fewell, and A. T. Ohta. CubeSats: A Bright Future for Nanosatellites. *Open Engineering*, Volume 1, March 2011. doi:10.2478/s13531-011-0007-8.

[SS05] N. Sivaprakash and J. Shanmugam. Neural Network Based Three Axis Satellite Attitude Control Using Only Magnetic Torquers. In *24th Digital Avionics Systems Conference*, volume 2, Washington, DC, USA, 2005. IEEE. doi:10.1109/DASC.2005.1563440.

[ST99] H. Shen and P. Tsiotras. Time-Optimal Control of Axi-Symmetric Rigid Spacecraft Using Two Controls. *Journal of Guidance, Control, and Dynamics*, 22(5):682–694, 1999. doi:10.2514/2.4436.

[TMP13] M. Tassano, P. Monzon, and J. Pechiar. Attitude Determination and Control System of the Uruguayan CubeSat, AntelSat. In *2013 16th International Conference on Advanced Robotics (ICAR)*, Montevideo, Uruguay, November 2013. IEEE. doi:10.1109/ICAR.2013.6766523.

[VDMS17] L. Van Damme, P. Mardesic, and D. Sugny. The Tennis Racket Effect in a Three-Dimensional Rigid Body. *Physica D: Nonlinear Phenomena*, 338:17–25, January 2017. arXiv: 1606.08237. doi:10.1016/j.physd.2016.07.010.

[Wer02] J. R. Wertz. *Spacecraft Attitude Determination and Control*, volume 73 of *Astrophysics and Space Science Library*. D. Reidel Publishing Company, Dordrecht; Boston; London, 2002.

[Wis97] R. Wisniewski. *Satellite Attitude Control Using Only Electromagnetic Actuation*. PhD thesis, Department of Control Engineering, Aalborg University, Aalborg, 1997.

[XSZ+17] X. Xia, G. Sun, K. Zhang, S. Wu, T. Wang, L. Xia, and S. Liu. NanoSats/CubeSats ADCS Survey. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 5151–5158, Chongqing, China, May 2017. IEEE. doi:10.1109/CCDC.2017.7979410.

[Yan12] Y. Yang. Spacecraft Attitude Determination and Control: Quaternion Based Method. *Annual Reviews in Control*, 36(2):198–219, December 2012. doi:10.1016/j.arcontrol.2012.09.003.

[Yan17] Y. Yang. Spacecraft Attitude and Reaction Wheel Desaturation Control. *IEEE Transactions on Aerospace and Electronic Systems*, 53(1):286 – 295, February 2017. doi:10.1109/TAES.2017.2650158.

License