Master Thesis

Advanced control methods for precise payload pointing of flexible spacecraft

János Bezsilla

Delft University of Technology

TU Delft

Master Thesis

Advanced control methods for precise payload pointing of flexible spacecraft

by

János Bezsilla

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday August 29, 2023 at 10:00 AM.

Student number:	5133394	
Thesis committee:	Dr. Jian Guo,	TU Delft, chair, supervisor
	Dr. ir. Jasper Bouwmeester,	TU Delft, examiner
	Dr. ir. Erik-Jan van Kampen,	TU Delft, examiner
	Dr. Béla Takarics,	SZTAKI SCL, supervisor

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

Pursuing my childhood dream of a career in space engineering was a long journey that has now reached fruition with this Master Thesis that concludes my studies in Aerospace Engineering at the Delft University of Technology. The project started during the second wave of the pandemic and took a great deal of research work at the Aerospace Guidance, Navigation and Control Group of SZTAKI to completely finish.

First of all, I would like to thank my supervisors. From TU Delft, Jian Guo who provided guidance at crucial moments along the way, and whose patience and continued support was instrumental in the success of this thesis. From SZTAKI, Béla Takarics and Bálint Vanek for their everyday assistance and constant feedback that kept me on track and helped me immensely in doing focused research. And a special thanks to Bálint Patartics, who familiarized me with his PhD work on verification systems, which proved to be a major contribution to the final chapter of the thesis. I appreciate all of you for encouraging me to showcase my research, an effort that led to the presentation of my conference article at the 10th Vienna International Conference on Mathematical Modelling, the highlight of this project.

I would also like to thank my colleagues at SZTAKI for all the cups of coffee in the break room, the technical chats over the foosball table, the video calls and game nights during the pandemic, as well as the after-work hangouts. Ábel, Ákos, Bence, Bence, Dani, Laci, Marci, Réka, Sári, Szabi, Suma, and Nathan, your presence filled the office with joy and kept me going through the hardest weeks.

Finally, I would like to thank my friends and family. First of all my parents, for being an example and for their unconditional support in this endeavour. My friends, for being by my side and filling these months with memories that I will cherish for years to come. And Dóri, thank you most of all for your love, understanding, and encouragement throughout this all.

János Bezsilla
Budapest, June 2023

The research reported in this paper is part of project no. SZTAKI-NVA-01, implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021 funding scheme.

Summary

This thesis focuses on the creation of a control design and verification framework to tackle the precise pointing problem of flexible spacecraft. Nowadays, many ongoing and proposed missions require extremely accurate pointing, which is usually achieved with the use of rigid vehicles. Otherwise, disturbances from the environment, the structural dynamics of the spacecraft, and the internal microvibration sources (such as reaction wheels) would severely impact image quality and line-of-sight stability. Additionally, these effects and their interactions are imprecisely known because the gravity environment on Earth does not allow for rigorous testing. The resulting uncertainty in the flexible structural parameters and vibrational dynamics of the spacecraft leads to major challenges in control.

Since payload mass restrictions can often only be met with low-stiffness structures, the control problem of flexible spacecraft has been a major theme in attitude determination and control systems research for several decades. The proposed work aims to tackle this challenge by achieving a dual goal of stabilizing the flexible system and increasing pointing performance. This is done by building a high-fidelity, nonlinear simulation environment of an Earth-observing Low Earth Orbit spacecraft in Simulink and modelling the vibration environment by taking into account both flexible appendages and reaction wheels. A sensor suite, state estimator, and reference angle generator are also implemented to better represent a realistic control system. Then, the main control loop is designed in a concurrent process with a proportional-integrative-derivative controller, an H_∞-based robust controller, and a model predictive controller that all have to pass a fixed ground-point tracking error requirement of 2 arcsecs.

Afterwards, pointing indicators standardized by the European Space Agency are used to evaluate the performance of closed-loop control, and it is found that to increase the accuracy and stability of pointing, an active-passive vibration isolation platform needs to be designed. By the implementation of a nonlinear model of a hexapod Stewart-Gough platform and the design of a secondary robust controller for payload pointing, the performance of the cascade control loop is evaluated in three pointing scenarios: single-point tracking, dual-point tracking, and fast multi-point tracking. It is concluded that the inclusion of an isolator platform is an absolute necessity due to the increased cutoff frequency for reference tracking, its ability to actively dampen at least the first flexible mode of structural dynamics, and to alleviate the heavy control effort of the outer loop.

Finally, the designed control algorithm is validated by the combined use of frequency-domain and time-domain linear models, as well as the nonlinear simulation environment. This fills the gap in research that was identified during the literature study, where it was determined that the main blocker in increasing pointing performance is a lack of synergy between modelling, control, and validation efforts. The validation campaign involved an analysis of parameter uncertainty, as well as a multi-frequency worst-case construction method which concluded that all controllers are robust enough to both parametric and dynamic uncertainty.

In conclusion, the main research question of improving pointing precision in flexible spacecraft can be answered in three parts. First, even if the structural dynamics and onboard microvibration sources are weak, single-loop control design has to take the flexible modes into account because system performance is greatly limited by the bandwidth and saturation limits of actuators. A cascade control loop with a vibration-isolated payload solves this issue by expanding the frequency range of the controller and offloading control requirements from the main loop to the isolator, leading to more relaxed control in the outer loop and higher pointing performance in the inner loop. Second, purely frequency-domain validation processes are not sufficient to prove robustness, as time-domain performance is vastly more important considering the standard precise pointing indices. Nonlinear high-fidelity modelling techniques offer significant improvements in the validation of control design by the reliable evaluation of system behaviour in identified worst cases. Third, there are plenty of articles focusing separately on high-fidelity modelling, advanced control, or verification&validation techniques, but individually these fields contain significant blind spots. Only their synergistic combination can provide a robust framework for the control engineering of flexible space structures.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviation	Definition
ADCS	Attitude Determination and Control System
ALOS	Advanced Land Observing Satellite
AOCS	Attitude and Orbit Control System
APE	Absolute Pointing Error
ARW	Allan Random Walk
COG	Center of Gravity
Diag	Diagonal
DOF	Degree of Freedom
ECEF	Earth-Centered Earth-Fixed (Frame)
ECI	Earth-Centered Inertial (Frame)
ECSS	European Cooperation for Space Standardization
ESA	European Space Agency
GPS	Global Positioning System
Hinf	H_∞-based Robust Controller
IMU	Inertial Measurement Unit
Isol	Isolator
JAXA	Japan Aerospace Exploration Agency
LEO	Low Earth Orbit
LFT	Linear Fractional Transformation
Lin	Linear or Linearized
LLA	Latitude, Longitude, Altitude
LOS	Line of Sight
LTI	Linear Time Invariant (System)
LVLH	Local Vertical Local Horizontal (Frame)
MC	Monte Carlo (Method)
MIMO	Multi-Input Multi-Output (System)
MPC	Model Predictive Control
MPE	Mean Pointing Error
NASA	National Aeronautics and Space Administration
NED	North-East-Down
Nom	Nominal
Par	Parameter Set
PID	Proportional-Integral-Derivative (Controller)
Ref	Reference
RMS	Root-Mean-Square
RPE	Relative Pointing Error
RW/RWA	Reaction Wheel (Assembly)
SC	Spacecraft
SDT	Satellite Dynamics Toolbox
SISO	Single-Input Single-Output (System)
STR	Star Tracker
SZTAKI	Institute for Computer Science and Control
TLE	Two-Line (Orbital) Element
V&V	Verification and Validation
WC	Worst Case
WGS84	World Geodetic System 1984

Common Latin Symbols

Symbol	Definition	Unit
$\mathbf{D}$	Damping matrix	
$\mathbf{e}$	Error signal	-
$\mathbf{f}_A^F$	Force acting in point A (or COG) and frame F.	N
$\mathbf{F}_{u,l}(\mathbf{M}, \mathbf{\Delta})$	Upper or lower LFT of uncertain system	-
$\mathbf{G}(s)$	System	-
$\mathbf{I}$	Identity matrix	-
j	Complex number whose square is -1	-
$\mathbf{J}$	Inertia matrix	kg m^2
$\mathbf{K}$	Stiffness matrix	N m^{-1}
$\mathbf{K}(s)$	Controller	-
l	Length	m
m	Mass	kg
$\mathbf{M}$	Mass matrix	kg
$\mathbf{P}(s)$	Plant model	-
$\mathbf{q}_{BA}$	Rotation quaternion from frame A to B	-
$\mathbf{r}_{BA}^F$	Position vector from point A to B in frame F	m
$\mathbf{R}_{BA}$	Rotation matrix from frame A to B	-
s	Laplace variable	-
T	Time	s
$\mathbf{u}$	Input signal	-
$\mathbf{v}_{BA}^F$	Velocity of B compared to A in frame F	m s^{-1}
$\mathbf{W}(s)$	Weighing function	-
$\mathbf{W}_P$	Wrench (i.e. force and torque) acting in point P	N, Nm
$\mathbf{y}$	Output signal	-

Common Greek Symbols

Symbol	Definition	Unit
γ	H_∞-norm of closed-loop system	-
$\mathbf{\Delta}(s)$	Uncertainty block	-
θ	Pitch angle	rad
$\bar{\sigma}$	Singular value of system	-
$\boldsymbol{\tau}_A^F$	Torque acting in point A (or COG) and frame F.	N m
ϕ	Roll angle	rad
ψ	Yaw angle	rad
ω_{BA}^F	Angular velocity of B compared to A in frame F	rad s^{-1}
ω	Angular frequency	rad s^{-1}

Common Indices

Symbol	Definition
B	Spacecraft Body frame
CP	Center of Pressure
F	Earth-Centered Earth-Fixed frame
I	Earth-Centered Inertial frame
L	Local frame
P	Connection point
P	Isolator payload frame
R	Reaction wheel assembly frame
S	Isolator support frame
$\oplus$	Earth or FI

1 Introduction

Nowadays, there are many ongoing and proposed missions involving spacecraft where extremely accurate pointing is required, either for cosmic vision or Earth observation purposes. These applications usually require rigid vehicles, otherwise a number of disturbing effects would severely impact image quality and line-of-sight stability, such as flexible appendages and internal microvibration sources [1].
Flexible appendages tend to be prevalent in complex missions, as payload mass restrictions can often only be met with low-stiffness structures. Their vibration characteristics are imprecisely known because the gravity environment on Earth does not allow for rigorous testing. This uncertainty of frequency and damping of the flexible modes and potential coupling between axes leads to major challenges in attitude determination and control system (ADCS) design [2].
Additionally, manufacturing imperfections in reaction wheels generate residual harmonic microvibrations which can be amplified on interaction with the structure of the spacecraft, perturbing the onboard instrumentation's line-of-sight stability or even leading to structural failure. As a result, accurately predicting spacecraft microvibrations due to onboard internal disturbance sources is a formidable multidisciplinary engineering challenge [3].
The proposed work aims to tackle the control problem of stabilizing the flexible system while also aiming to increase pointing performance. Based on the frequency ranges of disturbances and available control methods (fig. 1.1), this is best achieved in a cascade control loop with an active-passive vibration isolation system that is then validated against potential uncertainties in the spacecraft model.

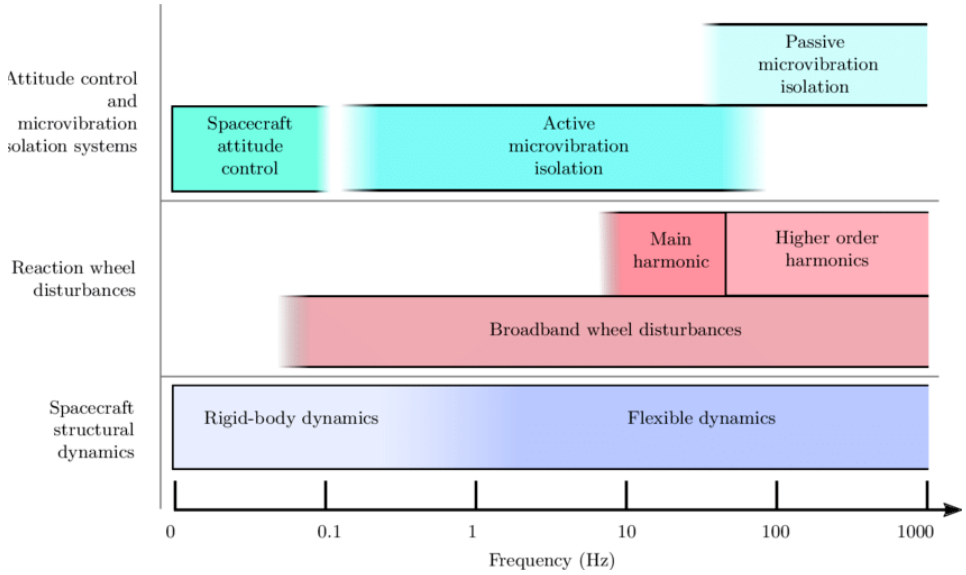

Figure 1.1: Typical frequency ranges for spacecraft dynamics and control systems. [1]

1.1. Literature review

In the literature review, the first task was to find relevant books on the topic to serve as a basis for further research. Eventually, the selection was restricted to these three, which have been prominently cited in relevant research and cover a significant portion of the topics in the thesis:

- "Fundamentals of Spacecraft Attitude Determination and Control" by F. Landis Markley and John L. Crassidis (2014) [4]
- "Vibration control of active structures" by Andre Preumont (1997) [5]
- "Multivariable feedback control: analysis and design" by Sigurd Skogestad and Ian Postlethwaite (2005) [6]

In terms of research papers on flexible spacecraft, it was noted that research output significantly changed after 2004, with several prominent researchers either retiring or starting to publish. In terms of pre-2004 research, Leonard Meirovitch published the earliest impactful articles on vibrations and flexible dynamics, while Stefano di Gennaro and Bong Wie expanded the scope of research and inspired hundreds of future articles. A summary of 50 years of research can be found in Singhose's 2009 review [7].
This historical context is instrumental to following the trail of research through the years following 2004, as research output steadily increased. In China, teams led by researchers such as Qinglei Hu, Gruangfu Ma, and Bing Xiao have published several papers on the sliding mode attitude tracking control of flexible satellites under various conditions (such as actuator saturation, disturbances, uncertainty, etc.), or focusing on the attitude stabilization control of such satellites with high accuracy [8]. In Europe, there has been a focus on integrated control/structure spacecraft design, and structured H_∞ control design [9]. Globally, there have also been efforts made on the predictive control of flexible spacecraft [10], with wide surveys being carried out on the entire aerospace industry, seeking opportunities for the application of model predictive control [11].
The next important step is assessing the state of the art of sensor characteristics. To this end, JAXA's Advanced Land Observing Satellite (ALOS) proved to be an important source of hardware data [12], as it has been one of the first high-precision flexible spacecraft in orbit. In our day, the most precise form of attitude determination seems to come from star sensors, so special attention has been given to assessing their performance and analyzing their theoretical limits and characteristics [13].
Then, the topic of vibration isolation and suppression was surveyed. After reading the thorough literature study of Liu et al [14], three approaches have been identified: passive, active, and hybrid isolation. A number of publications use struts (the most prevalent is a hexapod setup, although bipods also exist), while others use smart materials. Promising articles by Valentin Preda and his team at ESA combine robust microvibration mitigation with pointing performance analysis [15], which is greatly relevant to the project. For designing the cascade control structure, literature on precise payload pointing control has also been assessed. Several approaches have been identified, involving Stewart platforms [16], high precision tracking via iterative control [17], and model predictive control [18].
Finally, it has been noted that literature on the verification of controllers for flexible spacecraft is rather lacking, with very few published articles in the field. This is the gap that this project is trying to fill, by building on the body of research accumulated over the years on the topics that have previously been discussed. Important publications include the work of Wang et al in a joint project between the University of Exeter and ESA ESTEC on verification [19] and robustness analysis [20]. Also valuable to the project is the work of Gasbarri et al of Thales Alenia and Sapienza Universita di Roma [2].

1.2. Research objective

The main research question that was identified based on the gaps in the literature is: How can the pointing precision of flexible spacecraft be improved, compared to the current state of the art? It was determined at the start of the thesis that the best approach to achieve higher pointing precision in flexible spacecraft is by the analysis and validation of concurrent control approaches in a nonlinear simulation environment. First, an overview of the state of the art implementations was gained by reviewing available literature on the topic. Then, a reliable simulation environment is built in Simulink by analyzing available literature on the dynamics of flexible spacecraft in orbit. The modelling of the vibration environment is carried out by taking into account both flexible appendages and mechanical spinning devices such as reaction wheels. Finally, sensor dynamics are implemented in the model after a thorough analysis of available sensing methods for attitude determination.

Next, the coupling of the satellite and payload dynamics is modelled by reviewing previous work that has been done on the behaviour of vibration isolation systems. Indicators for the evaluation of pointing precision are identified to monitor the performance of control systems and the effect of disturbances. To increase pointing precision, attitude errors are minimized in a cascade control structure by designing an attitude control loop to stabilize the spacecraft in the correct orientation, and a payload pointing control loop to point the camera at the desired ground target.
Finally, the designed control algorithm is evaluated by carrying out a verification and validation campaign using the developed high-fidelity nonlinear simulation environment. Then, comparisons are made between various approaches by writing a report on implementation aspects. The validation of the work allows for making the scientific value of the results clear.

1.3. Outline

This thesis is structured as follows. Chapter 1 is the introduction, written to give a brief overview of the current practices in the precise pointing control of flexible spacecraft. It also identifies the gaps in the body of knowledge and proposes research objectives to investigate them.
Chapter 2 details the mathematical modelling of the flexible spacecraft in Low Earth Orbit. It describes the mass and inertia properties of the reference spacecraft based on Sentinel-2 and then presents the orbital parameters of the vehicle. Then, it defines four frames of reference to be used throughout the thesis: Earth-Centered Inertial, Earth-Centered Earth-Fixed, Local Vertical Local Horizontal, and Body-fixed. Using these frames, the equations of motions are derived from the properties of relative dynamics. They mainly rely on a combination of rotation matrices and quaternions, but the conversions to and from Euler angles are also described. Afterwards, the initial conditions of simulation are calculated from Kepler's equation and the orbit propagator is validated with measurement data. Then, the structural dynamics are described in terms of a flexible beam element that stands for the solar panel. There are also four environmental disturbances: gravity gradient torque, aerodynamic torque, magnetic torque, and non-spherical gravity. Finally, the model of the spacecraft is linearized and validated against the `Satellite Dynamics Toolbox` of Alazard et al [21].
Chapter 3 deals with the design of the attitude determination and control system of the spacecraft. It describes the modelling aspects of the microvibration environment of a reaction wheel assembly, as well as the noise characteristics of a sensor suite containing GPS, star trackers, gyros, and accelerometers. Then, a simulator-based state estimator algorithm is proposed whose output is used in the quaternion-based reference angle generator which is responsible for targeting the pointing system. For the main control loop of the spacecraft, a concurrent control design effort is carried out which resulted in a linear PID controller, a robust H_∞-based controller, and a predictive MPC. This process is described in detail to aid readers who might be unfamiliar with the controller tuning process, which can be time-intensive and highly iterative.
Chapter 4 deals with the evaluation of control performance in the outer loop, using standardized pointing performance indicators such as APE, MPE, and RPE. The ground track error is also calculated algorithmically from the line-of-sight vector and the WGS84 reference ellipsoid. Using these error indices, the effect of trajectory generation is evaluated in the tracking of a single ground target, as well as in a dual-tracking scenario. To improve control performance, a robust H_∞-based cascade control loop is proposed, involving a hexapod Stewart-Gough platform that serves a dual purpose as an active-passive vibration isolator and a pointing platform. With this platform, the viability of a multi-tracking algorithm is explored at the end of the chapter.
Chapter 5 ties the project together in a verification and validation campaign. There are two major components to this effort. First, the parameter uncertainty analysis process whose preliminary results were presented in 2022 at the 10th Vienna International Conference on Mathematical Modelling [22]. Second, a multi-frequency worst-case uncertainty construction method presented by Patartics et al [23], which combines frequency-domain and time-domain performance indicators to augment Monte Carlo methods, which were found to have significant shortcomings in literature. Chapter 6 concludes the project by explaining the most significant lessons learned and answering the initial research question. It also presents a critical review of the work done with some recommendations on potential future goals, and a short description of the ongoing research effort.

2

Mathematical Modelling

To describe and analyze the dynamics of the nonlinear system, a realistic mathematical model was built in Simulink. The scientific basis of this model was the research done throughout the literature study, which answered the following questions:

- How are spacecraft modelled in a simulation environment?
  - What are the differential equations governing the translational and rotational motion of a body in space?
  - What are the major disturbing effects in Low Earth Orbit and how are they modelled?
- What are the main sources of vibration on a flexible spacecraft and how are they affecting its dynamics?
  - What are the different approaches to modelling flexible appendages?

The resulting system achieves high fidelity in modelling the dynamics of an Earth-observing spacecraft while still retaining a high simulation speed due to the efforts made to increase efficiency. A simulation time of 6000 s was determined to model roughly a single orbit, with a fixed time step of 1 ms. To numerically solve the differential equations, the Bogacki-Shampine method is used (`ode3`), which was found sufficiently accurate compared to higher order solvers in `Simulink`.

2.1. Modelling the spacecraft

Throughout the thesis, Sentinel-2B, a European Earth observation satellite is used as a reference spacecraft for both orbit modelling and system characteristics. This features a spacecraft bus with a similar size and mass to the original, a rotating solar panel, and a variable-mass (currently half-full) propellant tank. The nominal values for these parameters are presented in table 2.1.

Part	Size (m)	Mass (kg)	Center of mass (m)
Bus	$[3.4; 1.8; 2.35]$	975	$[0; 0; 0]$
Panel	$[0.01; 3.1; 2.3]$	40	$[0; -2.45; 0]$
Tank	$[0.5; 0.5; 0.5]$	125/2	$[-1.4; 0; 0]$
Sum	-	1140	$[-0.0812; -0.0910; 0]$

Table 2.1: Reference spacecraft parameters.

From this table, the mass moment of inertia of the reference spacecraft can be calculated, which results in a rough approximation of the real value for Sentinel-2B. First, inertia matrices are defined for each part, approximating them as cuboids of mass m_i and dimensions $l_{x,y,z}$.

$$\mathbf{J}_i = \frac{m_i}{12} \begin{bmatrix} l_y^2 + l_z^2 & 0 & 0 \\ 0 & l_x^2 + l_z^2 & 0 \\ 0 & 0 & l_x^2 + l_y^2 \end{bmatrix} \tag{2.1}$$

Part	$\mathbf{J}_{i,diag}\ (kg\,m^2)$	$\mathbf{r}_i^T\ (m)$
Bus	$[711.95, 1387.95, 1202.50]$	$[-0.0812, -0.0910, 0]$
Panel	$[49.6667, 17.6337, 32.0337]$	$[-0.0812, 2.3590, 0]$
Tank	$[2.6042, 2.6042, 2.6042]$	$[1.3188, -0.0910, 0]$

Table 2.2: Reference spacecraft partial inertias.

Then, a generalization of the parallel axis theorem (also known as Huygens–Steiner theorem) is used for the summation of these three parts into the center of gravity (COG), which results in the complete inertia tensor of the spacecraft. Considering the vector $\mathbf{r}_i$ to be the displacement vector from the assembly COG to the element center of mass, the inertia matrix for the reference spacecraft works out to be:

$$\mathbf{J} = \sum_i \mathbf{J}_i + m_i \left(\mathbf{r}_i^T \mathbf{r}_i \mathbf{I}_3 - \mathbf{r}_i \mathbf{r}_i^T\right) = \begin{bmatrix} 995.41 & 7.9582 & 0 \\ 7.9582 & 1523.6 & 0 \\ 0 & 0 & 1583.7 \end{bmatrix} kg\,m^2 \tag{2.2}$$

The reference spacecraft (as well as Sentinel-2B) occupies a Sun-synchronous orbit, which is highly inclined and circular at 790 km altitude. For accurate orbit modelling, the orbital parameters are frequently updated from the spacecraft's two-line orbital elements (TLE). To give a (truncated) example for 1st October 2022:

```
1  42063U  17013A  22274.80190422  .00000096  00000-0  53102-4  0 9990
2  42063   98.5628  348.0182       0000835    85.9840  274.1436 14.30822891∼
```

The first line contains the epoch of the anomaly t_0 in terms of the year (20**22**) and decimal day with the fractional portion (273.61309745), while the second line has the classical orbital elements necessary for further modelling (shown in table 2.3),

Orbital Parameter	Symbol	Value	Unit
Inclination	i	98.5628	deg
Right ascension of the ascending node	Ω	348.0182	deg
Eccentricity	e	0.0000835	-
Argument of perigee	ω	85.9840	deg
Mean anomaly at epoch	M_0	274.1436	deg
Mean motion	n	14.30822891	day^{-1}

Table 2.3: Orbital parameters of Sentinel-2B.

Using the standard gravitational parameter of Earth ($\mu = 3.986 \cdot 10^{14}\,m^3\,s^{-2}$), the semi-major axis ($a$) of the orbit can be inferred from the mean motion of the spacecraft:

$$T_P = 2\pi\sqrt{\frac{a^3}{\mu}} = \frac{24 \cdot 3600}{n} = 6038.485\,s \longrightarrow a = 7.1671 \cdot 10^6\,m \tag{2.3}$$

In order to set up an overpass above Delft, the mean anomaly is propagated to 2022-10-01 19:14:44, with ΔT as the time passed between the epoch and the defined start time of the simulation:

$$M = M_0 + 2\pi\frac{\Delta T}{T_P} = 29.3637^\circ \tag{2.4}$$

2.2. Reference frames

The translational and rotational motion of a body in space is described by the use of reference frames. Generally, they are defined by their origin and the orientation of their coordinate axes. A huge variety of such coordinate systems exist, however, only a small subset are relevant to simulating spacecraft motion.

Inertial frame

As Newton's equations of motion are only valid in inertial reference frames, such systems are important when analyzing the dynamics of a body in orbit. They are reference frames that move at constant velocity and without rotation, relative to frames in which the universe appears spherically symmetric.

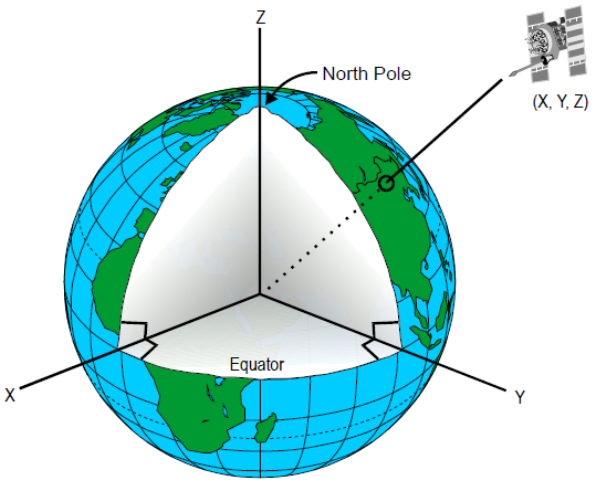

Figure 2.1: Position in ECI frame. (source: Wikimedia Commons)

The Earth-Centered Inertial (ECI, or "I") frame is a Cartesian coordinate system, whose origin is defined in the Earth's center of mass. The z-axis is aligned with the mean North Pole, which means it is normal to the mean equatorial plane of the planet. The x-axis is aligned with the mean vernal equinox at some epoch, here J2000.0 (12:00 Terrestrial Time on 1 January 2000). The y-axis completes the right-handed orthogonal system, meaning the x-y plane is coincident with the equatorial plane. This is the only inertial frame in the model, with the three other frames having their definitions tied to it [4].

Earth-fixed frame

For navigation and observation purposes, it is important to know the position of a spacecraft relative to Earth's surface. While the ECI frame allows for making these calculations, it is simpler to define a reference frame that takes into account the rotation of the Earth.

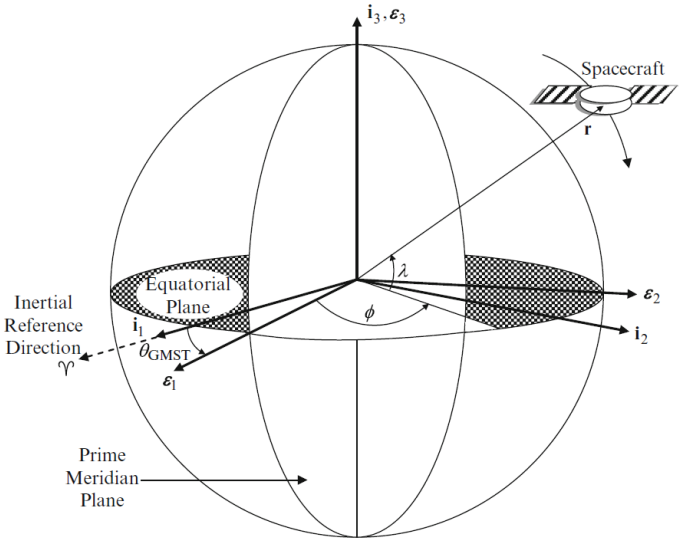

Figure 2.2: Definition of ECEF frame. [4]

The Earth-Centered Earth-Fixed (ECEF, or "F") frame shares an origin with ECI, as well as optionally the z-axis by disregarding precession, nutation, and other complex motions of Earth compared to the fixed stars. It is non-inertial, as the x-y axes are rotating with the planet, with x pointing towards the prime meridian (0 longitude, 0 latitude). Figure 2.2 compares the ECI frame base vectors $(\mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3)$ with those of the ECEF frame $(\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, \boldsymbol{\varepsilon}_3)$ [4].
The rotation matrix $\mathbf{R}_{FI}$ can be formulated as a simple rotation around the z-axis, with the angle θ_{GMST} as the product of Earth's rotation rate $(\omega^I_\oplus)$ and the time passed (ΔT) since the ECI epoch. One should note that the J2000.0 model starts at the Greenwich angle (280.46°). However, this simple model doesn't take into account precession and nutation, so the more complex model IAU-2000/2006 is used for deriving this rotation matrix in the thesis (in Matlab: `dcmeci2ecef`).

$$\mathbf{R}_{FI} = \begin{bmatrix} \cos\theta_{GMST} & \sin\theta_{GMST} & 0 \\ -\sin\theta_{GMST} & \cos\theta_{GMST} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

$$\theta_{GMST} = \Delta T \cdot \omega^I_\oplus + 280.46° \tag{2.6}$$

For modelling the planet and converting between ECEF and LLA (geodetic latitude, longitude, altitude), the thesis uses the WGS84 reference ellipsoid. Here, the geoid of Earth is approximated with a reference ellipsoid of semimajor axis $R_{eq} = 6378.1370$ km (equatorial radius) and semiminor axis $R_{pol} = 6356.7523$ km (polar radius). The rotation of the planet is defined to be $\omega^I_\oplus = 72.92115 \cdot 10^{-6}$ rad/s.

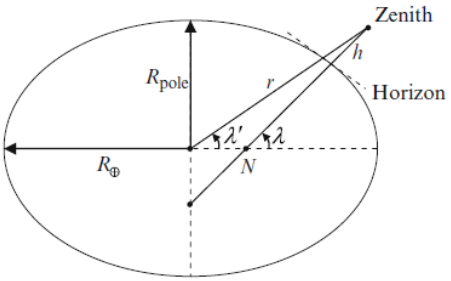

Figure 2.3: Geocentric and geodetic latitude. [4]

Using this model, ECEF position vectors can also be specified by their geocentric latitude λ', longitude ϕ, and magnitude r. This is done via a simple conversion from the Cartesian system to spherical coordinates. However, these coordinates are not the ones that are widely used in navigation. Instead of using the geocentric latitude, the geodetic latitude λ is defined, perpendicular to the reference ellipsoid (fig. 2.3). Another difference is that altitude h is used instead of magnitude r. A closed-form solution to converting between the geodetic coordinates (λ, ϕ, h) and ECEF (r^F_{BF}) position is given in Markley, while the thesis uses Matlab's `ecef2lla` and `lla2ecef` commands in case a conversion is needed.

Local frame

For Earth-pointing spacecraft, it is convenient to define a local frame, one that is referenced to the spacecraft's position in orbit.
A commonly used frame is Local Vertical Local Horizontal (LVLH, or "L"), which is a local frame with an origin that is fixed to the center of mass of the spacecraft. The z-axis points in the direction of the nadir (towards the center of mass of the Earth), the y-axis to the negated normal vector of the orbital plane, and the x-axis completes the right-handed orthogonal system. This is convenient for declaring the rotation matrix $\mathbf{R}_{LI}$ in simulations, as the nadir vector is the same as the negated unit ECI position vector, while the normal vector of the orbital plane is the same as the unit cross product of the ECI position $(\mathbf{r}^I_{BI})$ and velocity $(\mathbf{v}^I_{BI})$ vectors. The transformed rotation matrix $\mathbf{R}_{LF}$ can also be defined with ECEF vectors and the WGS84 Earth rotation rate $\boldsymbol{\omega}^F_\oplus$, using orthogonal basis vectors $\hat{\mathbf{o}}_i$ [4].

$$\begin{aligned} \mathbf{o}_1 &= \mathbf{o}_2 \times \mathbf{o}_3 \\ \mathbf{o}_2 &= -\mathbf{r}^F_{BF} \times \left(\mathbf{v}^F_{BF} + \boldsymbol{\omega}^F_\oplus \times \mathbf{r}^F_{BF}\right) \\ \mathbf{o}_3 &= -\mathbf{r}^F_{BF} \\ \mathbf{R}_{LF} &= [\hat{\mathbf{o}}_1 \quad \hat{\mathbf{o}}_2 \quad \hat{\mathbf{o}}_3] \end{aligned} \tag{2.7}$$

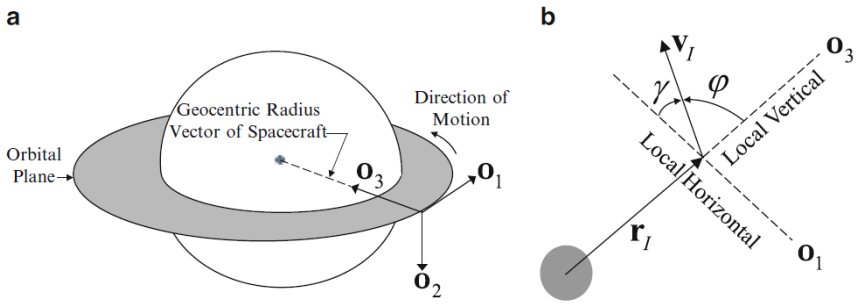

Figure 2.4: Local Vertical Local Horizontal frame definition. [4]

This is a geocentric system, meaning that the nadir vector is not normal to the surface of the reference ellipsoid. It is also important to mention that the x-axis is not coincident with the velocity vector, instead it is tangential to the trajectory of the spacecraft. The angle between the velocity vector and the local horizontal plane is called the flight path angle, denoted by γ in figure 2.4.
Another approach is the North-East-Down (NED) system. It is a similarly non-inertial frame of reference, that shares an origin with LVLH in the center of mass of the spacecraft. One of the differences is that the z-axis ("down") points to the negative normal of the surface ellipsoid, making the NED system geodetic (instead of geocentric, like LVLH). The x-axis points north parallel to the geoid surface in the polar direction, while the y-axis points east parallel to the geoid surface along a latitude curve. Because of these complexities, it is usually computationally inefficient compared to LVLH and suffers from bad numerical conditioning close to the poles. Despite NED's wide use in Matlab and `Simulink` toolboxes, this thesis exclusively uses LVLH as the local frame. [24]

Spacecraft body frame

The final reference system is the spacecraft body frame ("B"). Its origin is fixed to the COG of the body, and its orientation is fixed to the shape of the body. By using the orientation of the spacecraft, two final rotation matrices will be defined between the ECEF and Body frames ($\mathbf{R}_{BF}$), as well as the LVLH and Body frames ($\mathbf{R}_{BL}$) in the next section.

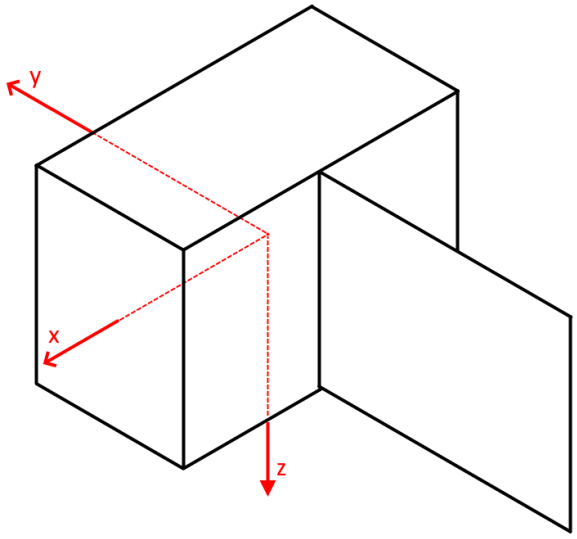

Figure 2.5: Model of the reference spacecraft with Body axes.

As shown in fig. 2.5, the x-axis is the surface normal of the "forward-facing" side of the spacecraft bus, and the z-axis points straight down in line with the optical axis. The y-axis completes the right-handed system, pointing opposite to the solar panel.

2.3. Equations of motion

In order to derive the equations of motion of the rigid body, four state vectors are selected to be integrated in the numerical solver. Two of them are used in the description of orbital dynamics: the ECEF-frame position vector ($\mathbf{r}_{BF}^{F}$) and the ECEF-frame velocity vector ($\mathbf{v}_{BF}^{F}$) of the spacecraft. Additionally, two more states are used for rotational motion: the ECEF-to-Body rotation quaternion ($\mathbf{q}_{BF}$) and the angular velocity of the spacecraft to the inertial frame, expressed in the Body frame ($\boldsymbol{\omega}_{BI}^{B}$).

Gravity modelling

Another important choice to make in the simulation is the fidelity of the gravity model. Since Earth is not a uniform sphere, but rather a geoid with a non-uniform mass distribution, the gravity field will reflect this nonuniformity. The most common approach to modelling the field uses a spherical harmonic expansion (fig. 2.6), usually in the form of zonal harmonics. Zonal harmonics provide a simple explanation of the nonuniform field, because they are symmetrical about the polar axis, meaning there is no longitude-dependence. The strongest of these effects arises from the J2 zonal coefficient, with all others being orders of magnitude weaker [4].
Over the years, a number of models have been developed based on measurements of the gravity field of Earth, such as WGS 84, EGM 96, EGM 2008, and EIGEN-GL04C. These models are readily available to a desired degree of accuracy, providing quick and reliable data for the gravity force vector for any point in Earth orbit.

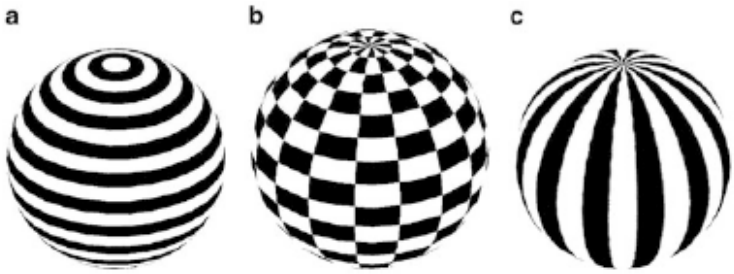

Figure 2.6: Zonal (a), tesseral (b), and sectoral (c) spherical harmonics. [4]

In this thesis, a zonal harmonic expansion to the fourth degree was used to account for the non-spherical property of the field (`Spherical Harmonic Gravity Model` in `Simulink`). This choice was the result of a trade-off between simulation speed and the accuracy of calculations over the simulated timespan.

Rigid body dynamics

The orbital dynamics of the satellite are described with two sets of differential equations, one for the ECEF position vector, and another for the ECEF velocity vector. These equations of motion are the result of Newton's laws of motion expressed in the rotating, non-accelerating frame ECEF frame. This rotation means there are also centrifugal and Coriolis components in the dynamics, while the Euler component can be safely neglected due to the slow rate of change in the angular velocity of the frame.

$$\dot{\mathbf{r}}_{BF}^{F} = \mathbf{v}_{BF}^{F} \tag{2.8}$$

$$\dot{\mathbf{v}}_{BF}^{F} = \mathbf{R}_{FB}\frac{\mathbf{f}^{B}}{m} - 2\boldsymbol{\omega}_{\oplus}^{F} \times \mathbf{v}_{BF}^{F} - \boldsymbol{\omega}_{\oplus}^{F} \times \left(\boldsymbol{\omega}_{\oplus}^{F} \times \mathbf{r}_{BF}^{F}\right) \tag{2.9}$$

Attitude representations

Up to this point in the thesis, rotation matrices were used as the only method of transforming vectors from one reference frame to another. If a vector $\mathbf{v}$ needs to be rotated from frame A to frame B, the task is performed by multiplication with the matrix $\mathbf{R}_{BA}$:

$$\mathbf{v}^{B} = \mathbf{R}_{BA} \cdot \mathbf{v}^{A} \tag{2.10}$$

Euler's theorem states that any rotation is a rotation about a fixed axis. As the unit vector $\mathbf{e}$, this is called the Euler axis, and it corresponds to the eigenvector of the matrix. The angle of rotation

about this axis is the Euler angle θ, and these two define one of the many rotation formalisms in three dimensions [4].

$$\mathbf{R}(\mathbf{e},\theta) = \begin{bmatrix} \cos\theta + (1-\cos\theta)\, e_1^2 & (1-\cos\theta)\, e_1 e_2 + \sin\theta e_3 & (1-\cos\theta)\, e_1 e_3 - \sin\theta e_2 \\ (1-\cos\theta)\, e_2 e_1 - \sin\theta e_3 & \cos\theta + (1-\cos\theta)\, e_2^2 & (1-\cos\theta)\, e_2 e_3 + \sin\theta e_1 \\ (1-\cos\theta)\, e_3 e_1 + \sin\theta e_2 & (1-\cos\theta)\, e_3 e_2 - \sin\theta e_1 & \cos\theta + (1-\cos\theta)\, e_3^2 \end{bmatrix} \tag{2.11}$$

In Aerospace applications, one of the most widely used formalisms is the Euler angle representation, which expresses a rotation from an initial frame I to a final frame F as the product of three rotations: a rotation first from I to an intermediate frame H, then to a second intermediate frame G, and finally to frame F. As such, the overall transformation is expressed as a series of three rotations [4].

$$\mathbf{R}_{FI}(\mathbf{e}_\phi, \mathbf{e}_\theta, \mathbf{e}_\psi; \phi, \theta, \psi) = \mathbf{R}_{FG}(\mathbf{e}_\psi, \psi)\mathbf{R}_{GH}(\mathbf{e}_\theta, \theta)\mathbf{R}_{HI}(\mathbf{e}_\phi, \phi) \tag{2.12}$$

The rotation axes of the classical Euler angle representation are selected from a set of three vectors, with a compacted notation [4].

$$\mathbf{e}_X = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{e}_Y = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \mathbf{e}_Z = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.13}$$

$$\mathbf{R}_{ijk}(\phi, \theta, \psi) = \mathbf{R}(\mathbf{e}_k, \psi)\mathbf{R}(\mathbf{e}_j, \theta)\mathbf{R}(\mathbf{e}_i, \phi) \tag{2.14}$$

There are 12 valid rotation sequences in total, Tait-Bryan angles being the most useful ones for the thesis, considering their property to represent horizontal attitude with zero degree elevation. The sequence used in the thesis is XYZ, which is generally useful for describing the attitude of an Earth-pointing spacecraft. The three angles in such a sequence are called roll (ϕ), pitch (θ), and yaw (ψ) [4]. An added benefit for the project is the sequence's property to render the last rotation (yaw) to a simple one around the optical axis of the spacecraft. The rotation matrix can be developed as follows.

$$\mathbf{R}_{XYZ}(\phi, \theta, \psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\phi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{2.15}$$

Euler angles have some major disadvantages. First off, using three rotation matrices that all involve trigonometrical functions is a huge strain on performance. Even worse is the existence of singularities. When $\cos\theta = 0$ (so the pitch angle is $\pm 90°$), gimbal lock occurs, which is the undesirable state of collinearity of the physical rotation axis vectors of the first and third rotations in the sequence, leading to the loss of one degree of freedom in rotation [4].

As such, Euler angles are only used for visualization and attitude commands in the thesis, and the dynamical model uses quaternion representation. At the cost of a somewhat increased degree of abstraction, the use of quaternions involves relatively simple multiplications in the dynamics and avoids gimbal lock through redundancy. Quaternions are defined by substituting $\sin(\theta) = 2\sin(\theta/2)\cos(\theta/2)$ and $\cos(\theta) = \cos^2(\theta/2) - \sin^2(\theta/2)$ into the rotation matrix $\mathbf{R}(\mathbf{e}, \theta)$[4]:

$$\mathbf{q}(\mathbf{e}, \theta) = \begin{bmatrix} \cos(\theta/2) \\ \mathbf{e}\sin(\theta/2) \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{2.16}$$

This means that the conversion from quaternions to rotation matrices and vice versa is a relatively simple operation:

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_0 q_2 + q_1 q_3) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_0 q_1 + q_2 q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{2.17}$$

For the inverse operation, four mathematically equivalent paths exist, and the selection depends on the numerical conditioning of the denominator q_0:

$$q_0 = \frac{1}{2}\sqrt{1 + \mathrm{tr}(\mathbf{R})} \qquad \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{4q_0} \begin{bmatrix} R_{12} + R_{21} \\ R_{13} + R_{31} \\ R_{32} - R_{23} \end{bmatrix} \tag{2.18}$$

Rotational kinematics with quaternions

The second set of differential equations was derived from the relative dynamics of rotating reference frames and the kinematics of quaternions. Including the body-frame torques $\boldsymbol{\tau}^B$ and reaction wheel angular momentum $\mathbf{h}_{rw}$ in the equation, the angular velocity of the body frame with respect to the inertial frame can be expressed as:

$$\dot{\boldsymbol{\omega}}_{BI}^B = \mathbf{J}^{-1}\left[\boldsymbol{\tau}^B - \boldsymbol{\omega}_{BI}^B \times \left(\mathbf{h}_{rw}^B + \mathbf{J}\boldsymbol{\omega}_{BI}^B\right)\right] \tag{2.19}$$

Using a scalar-first convention and defining $\mathring{\boldsymbol{\omega}}_{BF}^B$ as a quaternion with a zero scalar element, the kinematics of the quaternion between the fixed and body frames can be expressed as a single quaternion product. Note that during simulation, the inertial frame is not tracked with rotation matrices to reduce model complexity. As such, the $\mathbf{R}_{BF}$ matrix is used to rotate the angular velocity of the Earth, considering the rotation axis of $\mathbf{R}_{FI}$ is largely coincident with the planet's axis of rotation.

$$\dot{\mathbf{q}}_{BF} = \frac{1}{2}\mathbf{q}_{BF} \circ \mathring{\boldsymbol{\omega}}_{BF}^B \qquad \mathring{\boldsymbol{\omega}}_{BF}^B = \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{BI}^B - \mathbf{R}_{BF}\boldsymbol{\omega}_{\oplus}^F \end{bmatrix} \tag{2.20}$$

The outer product works out to the following matrix operation:

$$\dot{\mathbf{q}}_{BF} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix}\mathbf{q}_{BF} \tag{2.21}$$

Additionally, the orientation is defined to be the quaternion between the Body and the local frames. This quaternion also serves as the basis for the Local-to-Body rotation matrix, as well as the definition of Euler angles with an XYZ rotation order.

$$\mathbf{q}_{BL} = \mathbf{q}_{BF}\mathbf{q}_{LF}^{-1} \longrightarrow \mathbf{R}_{BL} \longrightarrow \phi, \theta, \psi \tag{2.22}$$

Initial conditions

Initial conditions for all the states have been calculated by extracting position and velocity data from the reference spacecraft's two-line orbital elements. The perifocal frame is used in this section: its origin lies at the focus of the orbit, with the x-axis directed towards the perigee, the y-axis having a true anomaly of 90 degrees past the perigee, and the z-axis being orthogonal to the orbital plane. [25]

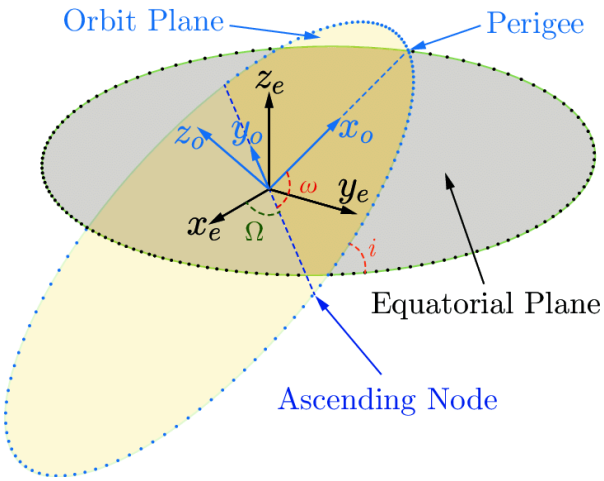

Figure 2.7: Perifocal frame of the satellite. [25]

First, Kepler's equation is solved numerically to determine the eccentric anomaly (E) from the mean anomaly (M) and eccentricity (e) of the orbit. Due to the circular orbit, this roughly matches the mean anomaly:

$$M = E - e\sin E \longrightarrow E = 29.3660° \tag{2.23}$$

From here, Kepler's first law of planetary motion is employed to calculate the position ($\mathbf{r}_p$) and velocity ($\mathbf{v}_p$) of the spacecraft in the perifocal frame, using the semi-minor axis (b), specific relative angular momentum (h), and distance from the Earth (r) [4].

$$b = a\sqrt{1-e^2} \tag{2.24}$$

$$\mathbf{r}_{BP}^{P} = \begin{bmatrix} a\left(\cos E - e\right) \\ b\sin E \\ 0 \end{bmatrix} \tag{2.25}$$

$$h = \sqrt{a\mu\left(1-e^2\right)} \qquad r = a\left(1 - e\cos(E)\right) \tag{2.26}$$

$$\mathbf{v}_{BP}^{P} = \frac{h}{r}\begin{bmatrix} -\frac{a}{b}\sin E \\ \cos E \\ 0 \end{bmatrix} \tag{2.27}$$

From there, the position and velocity vectors are rotated from the perifocal to the ECI frame, taking into account the angles of right ascension (Ω), inclination (i), and argument of perigee (ω) [4].

$$\mathbf{R}_{IP}^{T} = \begin{bmatrix} \cos\omega & \sin\omega & 0 \\ -\sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & \sin i \\ 0 & -\sin i & \cos i \end{bmatrix}\begin{bmatrix} \cos\Omega & \sin\Omega & 0 \\ -\sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.28}$$

$$\mathbf{r}_{BI}^{I} = \mathbf{R}_{IP}\mathbf{r}_{BP}^{P} \tag{2.29}$$

$$\mathbf{v}_{BI}^{I} = \mathbf{R}_{IP}\mathbf{v}_{BP}^{P} \tag{2.30}$$

Finally, the position and velocity of the spacecraft are expressed in the ECEF frame, giving the initial conditions for the first two state vectors:

$$\mathbf{r}_{BF}^{F} = \mathbf{R}_{FI}\mathbf{r}_{BI}^{I} \tag{2.31}$$

$$\mathbf{v}_{BF}^{F} = \mathbf{R}_{FI}\mathbf{v}_{BI}^{I} - \boldsymbol{\omega}_{\oplus}^{F} \times \mathbf{r}_{BF}^{F} \tag{2.32}$$

In order to determine the initial conditions of the rotation state vectors, first some assumptions need to be made. In this case, the spacecraft is in a nadir-pointing orientation (i.e. straight down) and has zero angular velocity with respect to the local frame (i.e. stays roughly nadir-pointing through the orbit).

$$\mathbf{q}_{BL} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T \longrightarrow \mathbf{R}_{BL} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.33}$$

$$\omega_{BL}^{B} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \tag{2.34}$$

The initial quaternion between the Body and the ECEF frame can be calculated with a simple multiplication:

$$\mathbf{R}_{BF} = \mathbf{R}_{BL} \cdot \mathbf{R}_{LF} \longrightarrow \mathbf{q}_{BF} \tag{2.35}$$

However, determining the initial angular velocity of the Body in the inertial frame is more complicated. The rotation rates of all rotating frames have to be rotated into the Body frame and summed up in order to keep the body nadir-pointing.

$$\boldsymbol{\omega}_{BI}^{B} = \boldsymbol{\omega}_{BL}^{B} + \boldsymbol{\omega}_{LF}^{B} + \boldsymbol{\omega}_{FI}^{B} = \boldsymbol{\omega}_{BL}^{B} + \mathbf{R}_{BF}\boldsymbol{\omega}_{LI}^{I} + \mathbf{R}_{BI}\boldsymbol{\omega}_{\oplus}^{I} \tag{2.36}$$

The angular velocity of the local frame can be calculated from the specific relative angular momentum divided by the square of the distance from Earth.

$$\boldsymbol{\omega}_{LI}^{I} = \frac{\mathbf{r}_{BI}^{I} \times \mathbf{v}_{BI}^{I}}{||\mathbf{r}_{BI}^{I}||^2} \tag{2.37}$$

First results

Once these equations are all implemented in `Simulink`, the first step was to verify the system behaviour against an independent reference. This is illustrated in figure 2.8.

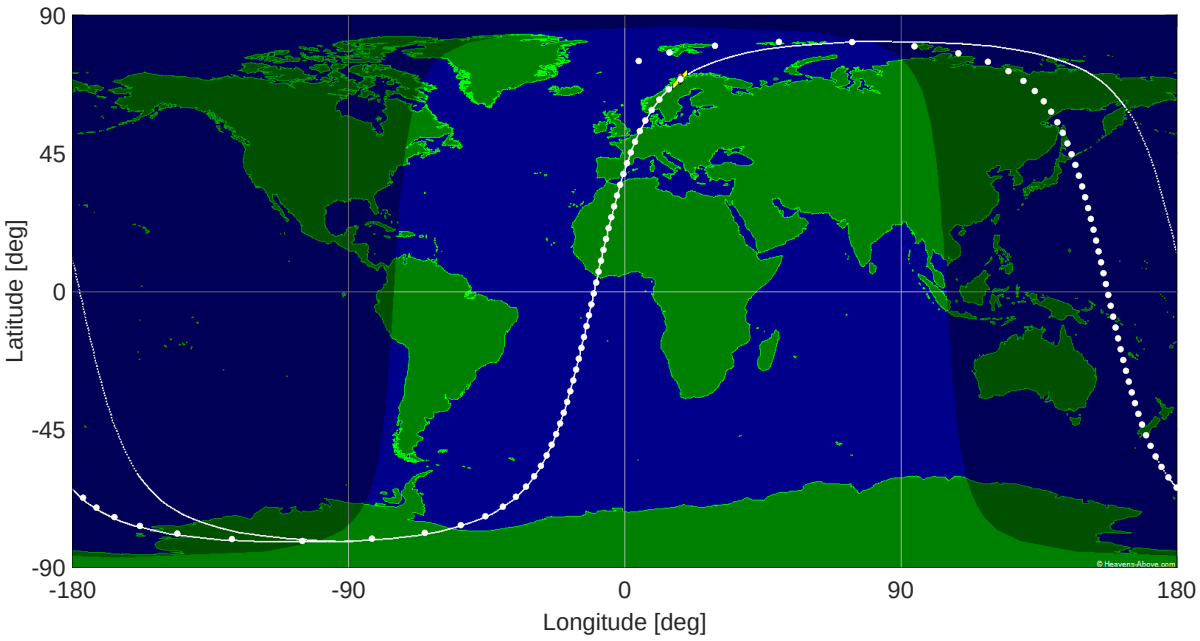

Figure 2.8: Ground track verification for orbit propagator.

The dots are placed one minute apart from each other, and show the ground track of the satellite as calculated by the orbit propagator. The solid line is the reference ground track from `Heavens Above`, half an orbital period into the past and future. Since there's a close visual match between the two lines, the longitude and latitude calculations can be considered valid.

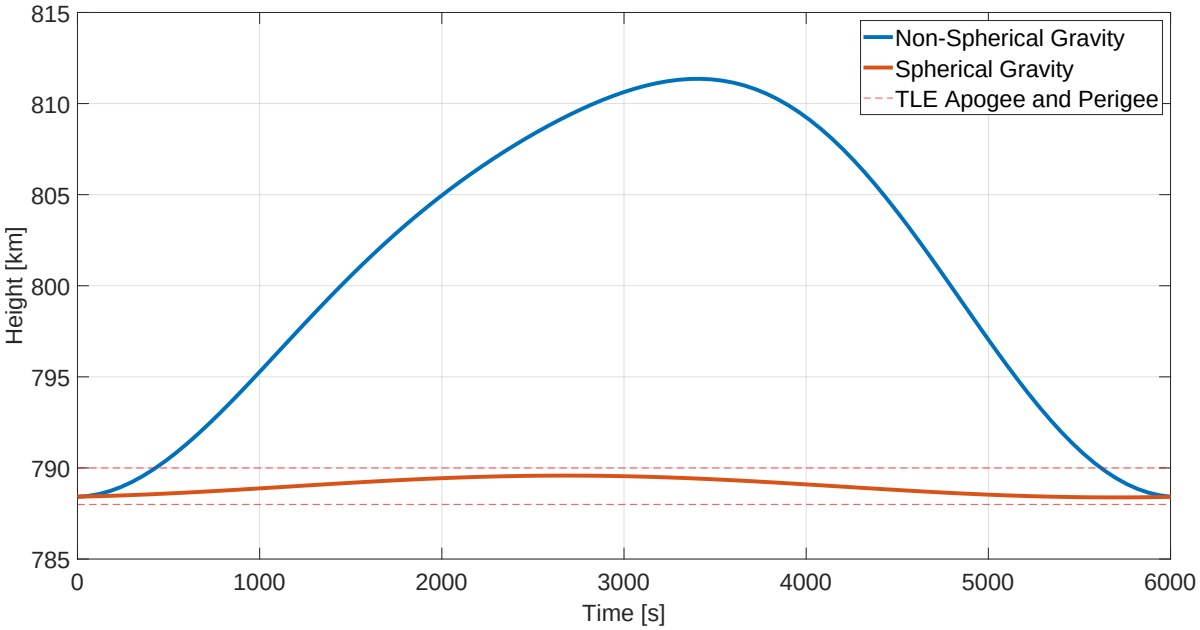

Figure 2.9: Desired apogee overshoot with non-spherical gravity.

It should be noted that while the nominal perigee was respected, there was an enormous overshoot of the nominal apogee height of the spacecraft (fig. 2.9). This discrepancy comes from the assumptions made during the calculation of initial conditions: classical orbital elements and orbital mechanics are based on

a spherical gravity model, so in order to accommodate non-spherical gravity models, the calculations would require some adjustments. However, an exact match with the real altitude of Sentinel is not necessary for the thesis, so the effect is accepted as an environmental disturbance.

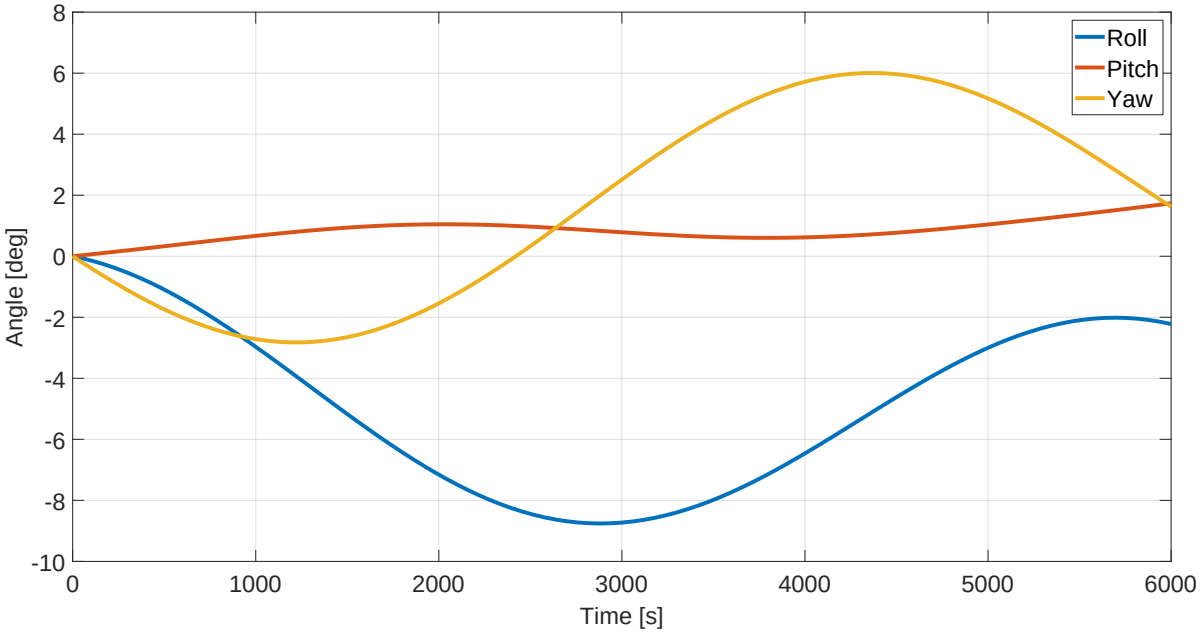

Figure 2.10: Euler angles with no control.

The Euler angles of the spacecraft (fig. 2.10) were also checked to see the magnitude of attitude drift. Ideally, all angles should remain close to zero, given the initial conditions for the angular velocity. This ensures that when the designed controllers are switched on, they do not have to rein in a rapidly rotating spacecraft and can expend more control effort on ground tracking. Since none of the angles drift outside of 10 degrees during a single orbit, this requirement is also fulfilled.

2.4. Modelling a flexible appendage

A major source of disturbance to the spacecraft comes from its structural dynamics. The solar panel attached to the side of the satellite acts as a flexible appendage, are there are numerous approaches to modelling its behaviour. In this project, the panel is considered to be a flexible beam element with a rigid single-point connection (P) to the spacecraft. This description of the flexible body results in a linear state-space model, following the derivation of pure flexion dynamics from Chebbi et al [26].

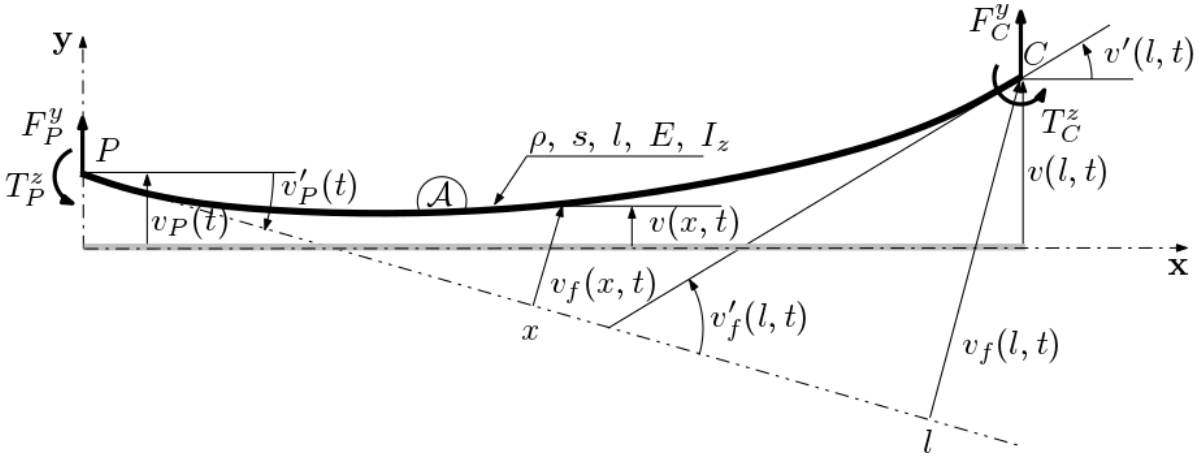

Figure 2.11: Parametrization of the flexion beam deflection. [26]

This approach is a Lagrange-based technique that involves a moving body frame attached to the beam at the connection point P. Considering the shape and material properties of the panel, torsion and traction modes can be safely neglected, as well as the x-z plane bending modes. As such, this section is mainly concerned with the derivation of a linear model for the x-y bending modes of the beam element representing the solar panel. The final model is verified against the `Satellite Dynamics Toolbox` of Alazard et al [21][27][28].
As presented in figure 2.11, in the x-y plane, the dynamics of the beam only consist of two components: translation along the y-axis and rotation around the z axis. The state of the beam ($\mathbf{q}$) is described using the beam deflection v and its spatial derivatives and then split into rigid and flexible components by the shape functions $\mathbf{\Phi}_r$ and $\mathbf{\Phi}_f$.

$$\mathbf{q}(x,t) = \begin{bmatrix} v(x,t) \\ v'(x,t) \\ v''(x,t) \end{bmatrix} = \mathbf{\Phi}_r(x)\mathbf{q}_r(t) + \mathbf{\Phi}_f(x)\mathbf{q}_f(t) \tag{2.38}$$

Our main concern in this description is determining the effects of flexible behaviour. As such, every step along the way the purely rigid components of the beam dynamics are slowly being worked away. This is done through an analysis of the elements of equation 2.38. The total deflection is also split into components, describing the purely rigid and purely flexible motion of the beam:

$$v(x,t) = v_P(t) + xv'_P(t) + v_f(x,t) \qquad v_P(t) = v(0,t) \tag{2.39}$$

Here, the flexible part $v_f(x,t)$ is approximated with a 5th order polynomial for x with time-dependent coefficients $\mathbf{a}_f(t)$ such that $v_f(0,t)$ and $v'_f(0,t)$ are both zero:

$$v_f(x,t) = \mathbf{p}(x)\mathbf{a}_f(t) \qquad \mathbf{p}(x) = \begin{bmatrix} x^2 & x^3 & x^4 & x^5 \end{bmatrix} \tag{2.40}$$

Considering the four-component vector of flexible coordinates, one can make a connection to the polynomial coefficients of the deflection:

$$\mathbf{q}_f(t) = \begin{bmatrix} v''_f(0,t) \\ v_f(l,t) \\ v'_f(l,t) \\ v''_f(l,t) \end{bmatrix} = \mathbf{P}\mathbf{a}_f(t) \quad \mathbf{P} = \begin{bmatrix} \mathbf{p}''(0) \\ \mathbf{p}(l) \\ \mathbf{p}'(l) \\ \mathbf{p}''(l) \end{bmatrix} \tag{2.41}$$

As a result of this parametrization, the shape functions take the following values:

$$\mathbf{\Phi}_r(x) = \begin{bmatrix} 1 & x \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad \mathbf{\Phi}_f(x) = \mathbf{P}_f(x)\mathbf{P}^{-1} \qquad \mathbf{P}_f(x) = \begin{bmatrix} \mathbf{p}(x) \\ \mathbf{p}'(x) \\ \mathbf{p}''(x) \end{bmatrix} \tag{2.42}$$

Now, they can be used to define the mass and stiffness matrices of the flexible system. The following formulas are derived from the Lagrange equations for the kinetic and elastic potential energies [26]. Note that in the mass matrix, the rotational contribution of the kinetic energy is neglected.

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{rr} & \mathbf{M}_{rf} \\ \mathbf{M}_{rf}^T & \mathbf{M}_{ff} \end{bmatrix} = \int_0^l \begin{bmatrix} \mathbf{\Phi}_r^T \\ \mathbf{\Phi}_f^T \end{bmatrix} \begin{bmatrix} \rho s & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_r & \mathbf{\Phi}_f \end{bmatrix} dx \tag{2.43}$$

$$\mathbf{K} = \int_0^l \mathbf{\Phi}_f^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & EJ_z \end{bmatrix} \mathbf{\Phi}_f dx \tag{2.44}$$

These matrices can then be used in the dynamic model obtained by Lagrange derivation, giving the 2 DOF wrench $\mathbf{W}_P$ (z-torque and y-force) applied by the beam and acting in point P as output.

$$\begin{bmatrix} \mathbf{M}_{rr} & \mathbf{M}_{rf} \\ \mathbf{M}_{rf}^T & \mathbf{M}_{ff} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{q}}_f \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{2x2} & \mathbf{0}_{2x4} \\ \mathbf{0}_{4x2} & \mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{q}_r \\ \mathbf{q}_f \end{bmatrix} = -\mathbf{W}_P \tag{2.45}$$

In order to define a damping matrix $\mathbf{D}$, the following eigenvector-eigenvalue problem is solved for the purely flexible mass matrix and the stiffness matrix:

$$\mathbf{M}_{ff}\mathbf{V}\boldsymbol{\Omega}^2 = \mathbf{K}\mathbf{V} \longrightarrow \mathbf{V}, \boldsymbol{\Omega}^2 = diag\left(\omega_i^2\right) \tag{2.46}$$

$$\mathbf{D} = \mathbf{V} diag\left(2\xi\omega_i\right)\mathbf{V}^{-1} \tag{2.47}$$

Considering the acceleration twist $\ddot{\mathbf{q}}_P$ at point P as the input and the wrench $\mathbf{w}_P$ as output, the linear dynamics of the solar panel in the local frame of P can be described with the following state-space matrix. Note that the purely rigid component of the mass matrix (in red) is set to zero in the final model, in order to not duplicate the inertial contribution of the solar panel in the full model of the spacecraft.

$$\begin{bmatrix} \dot{\mathbf{q}}_f \\ \ddot{\mathbf{q}}_f \\ \mathbf{w}_P \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{4x4} & \mathbf{I}_{4x4} & \mathbf{0}_{4x2} \\ -\mathbf{M}_{ff}^{-1}\mathbf{K} & -\mathbf{D} & -\mathbf{M}_{ff}^{-1}\mathbf{M}_{rf}^T \\ \mathbf{M}_{rf}\mathbf{M}_{ff}^{-1}\mathbf{K} & \mathbf{M}_{rf}\mathbf{D} & \mathbf{M}_{rf}\mathbf{M}_{ff}^{-1}\mathbf{M}_{rf}^T - \mathbf{M}_{rr} \end{bmatrix} \begin{bmatrix} \mathbf{q}_f \\ \dot{\mathbf{q}}_f \\ \ddot{\mathbf{q}}_P \end{bmatrix} \tag{2.48}$$

The calculation of this matrix requires the definition of five material properties for the beam (tab. 2.4). These can either be determined from the dimensions and mass of the solar panel. In the case of the Young modulus, an approximation for single-crystal silicon is used from catalogue data.

Symbol	Definition	Value	Unit
ρ	Average density	561	$kg\,m^{-3}$
s	Cross-sectional area	0.023	m^2
l	Beam length	3.1	m
E	Young modulus	150	GPa
J_z	Mass moment of inertia	$1.9167 \cdot 10^{-7}$	$kg\,m^2$

Table 2.4: Material properties of the solar panel.

In order to include the dynamical equations of the beam in the larger nonlinear model of the spacecraft, first it has to be expanded from a two-input-two-output system to one with six inputs and six outputs, then has to be transformed from the Panel frame to the Body frame. To this end, four transformation matrices are applied to the state-space model. The first matrix ($\mathbf{R}_{PB_1}$) expands the size of inputs and outputs from two to six, while the second matrix ($\mathbf{R}_{PB_2}$) flips the axes of the panel reference frame to is aligns with the body axes.

$$\mathbf{R}_{PB_1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{R}_{PB_2} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.49}$$

The third matrix ($\mathbf{R}_{PB_3}$) rotates the inputs and outputs according to the angle of the solar panel. The fourth one ($\mathbf{R}_{PB_4}$) shifts the origin of the frame from the panel connection point P to the body center of gravity B using a skew-symmetric transformation matrix.

$$\mathbf{R}_{PB_3} = \begin{bmatrix} \cos\alpha_P & 0 & -\sin\alpha_P \\ 0 & 1 & 0 \\ \sin\alpha_P & 0 & \cos\alpha_P \end{bmatrix} \qquad \mathbf{R}_{PB_4} = \left[\begin{bmatrix} -0.0812 \\ -0.0910 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ -0.9000 \\ 0 \end{bmatrix} \right]_\times \tag{2.50}$$

Finally, all of these matrices are combined into a single transformation:

$$\mathbf{R}_{PB} = \mathbf{R}_{PB_1} \begin{bmatrix} \mathbf{R}_{PB_2}\mathbf{R}_{PB_3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{R}_{PB_2}\mathbf{R}_{PB_3} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{R}_{PB_4} \\ \mathbf{0}_{3x3} & \mathbf{I}_{3x3} \end{bmatrix} \tag{2.51}$$

Which is then applied to the component matrices of the state-space model $\mathbf{G}_{flex}$:

$$\begin{bmatrix} \dot{\mathbf{q}}_f \\ \ddot{\mathbf{q}}_f \end{bmatrix} = \mathbf{A}_\mathbf{F} \begin{bmatrix} \mathbf{q}_f \\ \dot{\mathbf{q}}_f \end{bmatrix} + \mathbf{B}_\mathbf{F} \begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix}_{PB}^B \tag{2.52}$$

$$\begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{bmatrix}_P^B = \mathbf{C}_\mathbf{F} \begin{bmatrix} \mathbf{q}_f \\ \dot{\mathbf{q}}_f \end{bmatrix} + \mathbf{D}_\mathbf{F} \begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix}_{PB}^B \tag{2.53}$$

$$\mathbf{A}_F = \begin{bmatrix} \mathbf{0}_{4x4} & \mathbf{I}_{4x4} \\ -\mathbf{M}_{ff}^{-1}\mathbf{K} & -\mathbf{D} \end{bmatrix} \qquad \mathbf{B}_F = \begin{bmatrix} \mathbf{0}_{4x2} \\ -\mathbf{M}_{ff}^{-1}\mathbf{M}_{rf}^T \end{bmatrix} \mathbf{R}_{PB} \tag{2.54}$$

$$\mathbf{C}_F = \mathbf{R}_{PB}^T \begin{bmatrix} \mathbf{M}_{rf}\mathbf{M}_{ff}^{-1}\mathbf{K} & \mathbf{M}_{rf}\mathbf{D} \end{bmatrix} \qquad \mathbf{D}_F = \mathbf{R}_{PB}^T \begin{bmatrix} \mathbf{M}_{rf}\mathbf{M}_{ff}^{-1}\mathbf{M}_{rf}^T \end{bmatrix} \mathbf{R}_{PB} \tag{2.55}$$

The resulting linear model can be directly integrated into the larger nonlinear model of the spacecraft while preserving its modularity. For the nominal case of the solar panel, the Bode plots presented in figure 2.12 show the reaction torques and forces as a response to Body-frame acceleration.

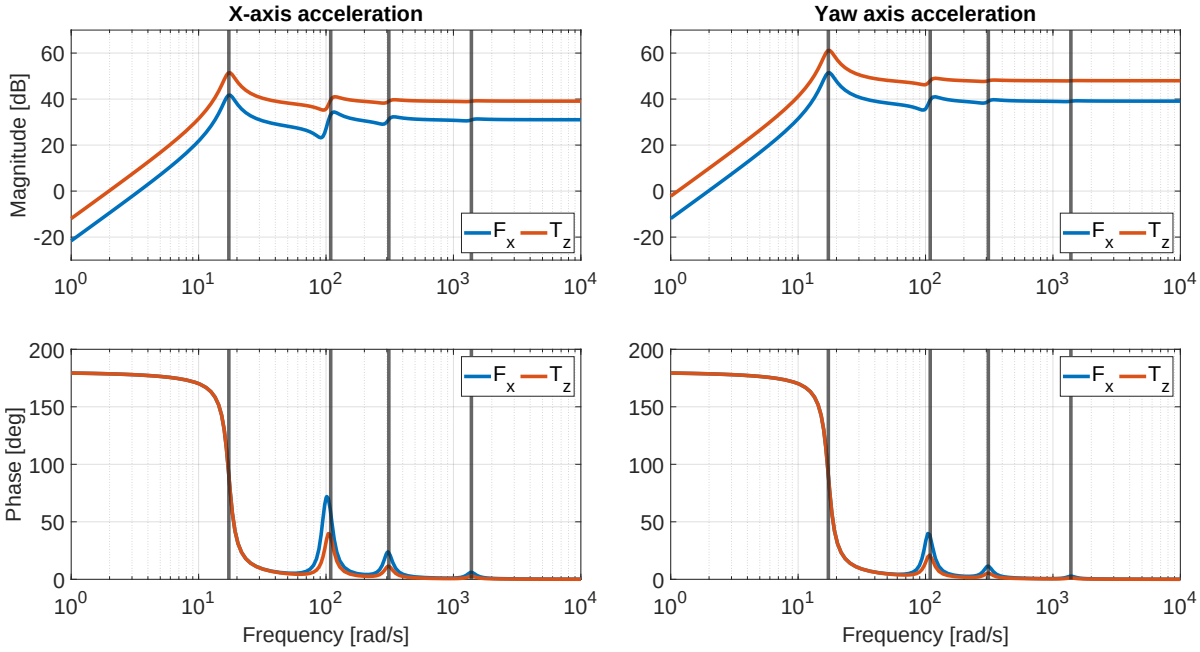

Figure 2.12: X-axis force and z-axis torque response of the panel to acceleration input.

The frequency response of the flexible system resembles a high-pass filter with a cutoff frequency of roughly 17 rad/s. This is ideal for control purposes as the frequency is well within the bounds of available spacecraft actuators. There are three more resonant frequencies for the appendage (table 2.5), two of which are still within reasonable bounds for control purposes. The largest frequency bending mode is likely uncontrollable, but fortunately has a low magnitude and can still be simulated with the current time step, so its effect on control performance can be evaluated.

Frequency (rad/s)	Damping coeff.
17.2702	0.1
108.836	0.1
311.149	0.1
1383.16	0.1

Table 2.5: Flexible characteristics.

2.5. Modelling external disturbances

There are a number of environmental factors that have undesirable effects on the dynamics of the spacecraft by changing its overall momentum (linear or angular). This section will mainly focus on external torques, which can vary greatly in magnitude, depending on the characteristics of the spacecraft, the altitude of the orbit, or even time, among other factors. Note that for these disturbances, the main goal is not to model them as numerically accurately as possible, which would be hard enough even with unlimited computational resources. Rather, the goal is to build optimized representations of them that match their real-world frequency and magnitude characteristics.

A plot of the most significant external disturbing torques is shown in fig. 2.13. Due to the reference spacecraft's position in Low Earth Orbit, the sources of the main disturbing torques are aerodynamics, magnetism, and gravity. Solar radiation pressure is excluded from modelling to increase simulation speed, as it is a relatively minor effect in the selected altitude range.

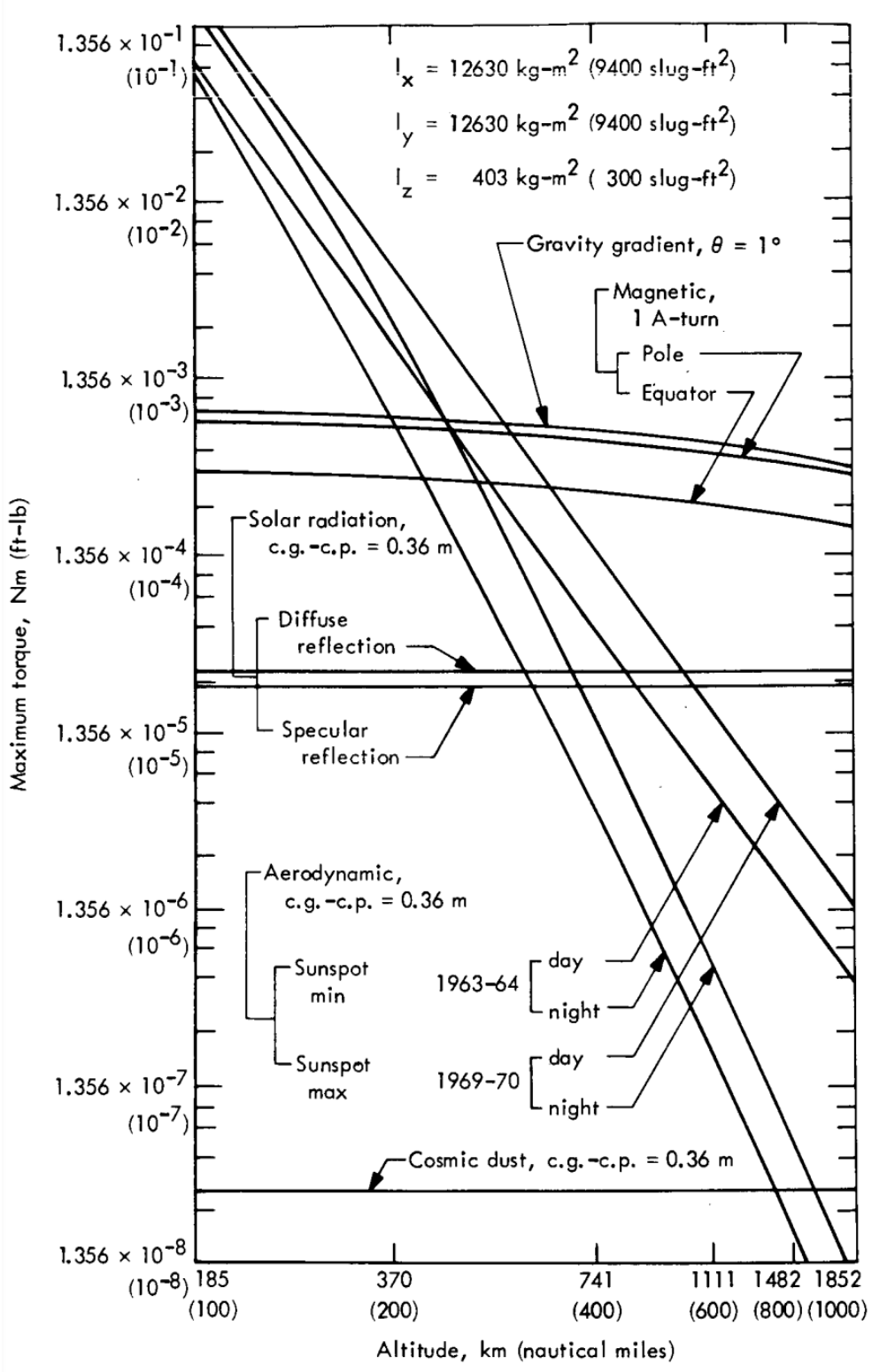

Figure 2.13: Magnitudes of environmental torques on an Earth-orbiting satellite. [29]

Gravity gradient torque

Any non-symmetrical rigid body in a gravity field is subject to a gravity-gradient torque. This is the result of a drop in the magnitude of the field as various parts of the body are located farther away from the source (i.e. the center of the planet). This torque is computed by the summation of gravitational forces acting on the point masses constituting the rigid body. Assuming that the gravity field is spherically symmetric, the torque can be calculated as [4]:

$$\boldsymbol{\tau}_{gg}^{B} = \frac{3\mu}{||r_{BF}^{F}||^{3}} \mathbf{n}^{B} \times \mathbf{J}\mathbf{n}^{B} \tag{2.56}$$

Where μ is the gravitational parameter of the Earth, r_{BF}^{F} is the distance from the center of the Earth, $\mathbf{n}^{B}$ is the geocentric nadir-pointing unit vector in the body frame (same as $\hat{\mathbf{r}}_{BF}^{B}$), and $\mathbf{J}$ is the moment of inertia matrix of the body. Essentially, the gravity gradient torque will tend to align a spacecraft with its principal axis of minimum inertia aligned with the nadir vector. This means that for specific designs (not Sentinel-2B or the reference spacecraft), the disturbing torque can be used to provide some coarse, passive stabilization to such spacecraft around rotation angles perpendicular to the nadir.

Magnetic torque

It is often the case for spacecraft that they have some magnetic dipole moment m_M, roughly estimated as a multiple (here: $10^{-3}\, A\, m^2\, kg^{-1}$) of the spacecraft mass in a random direction. This could be a result of residual magnetism from the onboard equipment, or electric current passing through a loop. In the magnetic field ($\mathbf{B}^{F}$) of Earth this generates a torque [4]:

$$\boldsymbol{\tau}_{m}^{B} = \mathbf{m}_{M} \times \mathbf{R}_{BF}\mathbf{B}^{F} \tag{2.57}$$

Modelling the magnetic field of the Earth is done using spherical harmonics, similar to modelling the gravity field. The major difference is the need to take into account the dipole nature of magnetism. To this end, a simple approach would be to take the International Geomagnetic Reference Field (IGRF, maintained by the International Association of Geomagnetism and Aeronomy) or the World Magnetic Model (WMM, maintained by the US National Oceanic and Atmospheric Administration - NOAA). Both are updated in regular, 5-year intervals and provide a precise approximation of Earth's magnetic field for given time periods [30].
However, the implementation of these models in `Simulink` is a major drain on computational resources, so in order to speed up the calculations, a streamlined reimplementation is required. Using a third-order approximation, most of the large model coefficients (table 2.6) can be captured in the calculation and simulation time is barely affected. The magnetic flux density in the ECEF frame $\mathbf{B}^{F}$ can be expressed as the negative spatial gradient of the scalar V:

$$\mathbf{B}^{F} = -\nabla V \tag{2.58}$$

Where the potential field is a spherical harmonic expansion of the field in terms of time, radius, and geocentric longitude and latitude:

$$V(t, r, \lambda, \phi') = a \sum_{n=1}^{N=3} \left(\frac{a}{r}\right)^{n+1} \sum_{m=0}^{n} \left(g_{n}^{m}(t) \cos(m\lambda) + h_{n}^{m}(t) \sin(m\lambda)\right) P_{n}^{m}(\sin\phi') \tag{2.59}$$

This formula described a third-degree expansion of the WMM and uses r geomagnetic reference radius of the Earth (6371200 m), $g_{n}^{m}(t)$ and $h_{n}^{m}(t)$ time-dependent Gauss coefficients of degree n and order m, and $\hat{P}_{n}^{m}$ Schmidt semi-normalized associated Legendre polynomials.

$$g_{n}^{m}(t) = g_{n}^{m}(t_0) + \Delta t \cdot \dot{g}_{n}^{m}(t_0) \qquad h_{n}^{m}(t) = h_{n}^{m}(t_0) + \Delta t \cdot \dot{h}_{n}^{m}(t_0) \tag{2.60}$$

As the exact algorithm is too unwieldy to present in this section, the code to calculate the flux density is attached to the appendix.

n	m	$g_n^m(t_0)$	$h_n^m(t_0)$	$\dot{g}_n^m(t_0)$	$\dot{h}_n^m(t_0)$
1	0	-29404.5	-	6.7	-
1	1	-1450.7	4652.9	7.7	-25.1
2	0	-2500.0	-	-11.5	-
2	1	2982.0	-2991.6	-7.1	-30.2
2	2	1676.8	-734.8	-2.2	-23.9
3	0	1363.9	-	2.8	-
3	1	-2381.0	-82.2	-6.2	5.7
3	2	1236.2	241.8	3.4	-1.0
3	3	525.7	-542.9	-12.2	1.1

Table 2.6: WMM2020 coefficients for degree n and order m in nT and nT/yr.

Aerodynamic torque

One way to approximate drag is to consider the spacecraft to be a collection of flat plates, each with surface area s_i and surface normal vector $\mathbf{n}_i$. Since the atmosphere is rotating with the Earth, an "airspeed" $\mathbf{v}_{rel}$ is also defined. Then, incident angles can be calculated for each of the plates, which allows for cancelling out the non-contributing ones. The total aerodynamic torque can then be calculated as the sum of individual forces and moment arms ($\mathbf{r}_{CP,i}$) on each of the plates making up the spacecraft:

$$\begin{gathered}\boldsymbol{\tau}_{ae,i}^B = \mathbf{r}_{CP,i}^B \times \left(-\frac{1}{2}\rho c_D s_i v_{rel}^2 \max\left\{\cos\theta_i, 0\right\} \cdot \hat{\mathbf{v}}_{rel}\right) \\ \mathbf{v}_{rel} = \mathbf{v}_{BF}^B \qquad \cos\theta_i = \mathbf{n}_i \cdot \hat{\mathbf{v}}_{rel}\end{gathered} \tag{2.61}$$

In the equation, there are two constants to be determined. First, the drag coefficient, which is assumed to be that of a flat plate ($c_D = 1.28$). Second, the atmospheric density (ρ), for which various models exist to specific degrees of precision. In this thesis, NRLMSISE-00 was used, which is one of the few models to also take into account the ionization of the upper atmosphere to give valid densities and temperatures up to 1000 km in altitude.

In order to significantly speed up calculation speed, an altitude-dependent exponential approximation (eq. 2.62) of the `atmosnrlmsise00` model between 700 and 900 km (with 1km steps) is developed by using averaged values for parameters that are relatively constant over the simulation timescale (minutes). Latitude and longitude are assumed to be the average of ground targets, or, if none exist, zero. The time measurement is assumed to be the same as the simulation start date at noon. Averaged values for 2022 are used for the magnetic indices F10.7 and APH. For the F10.7 index, 150, which is the level of noise generated by the Sun at wavelength 10.7 cm. For the APH index, 4, which indicates the general level of geomagnetic activity.

$$\rho(h) = Ae^{Bh} + Ce^{Dh} \tag{2.62}$$

Overview

Figures 2.14 and 2.15 present the effect of the disturbance models. The first observation is that as a result of these disturbances, the spacecraft is steadily becoming unstable in its orientation, and comes close to tipping over after just one orbit. Overall, the Euler angle response to the simplified and "real" models is the exact same, leading to the conclusion that the models derived in the thesis are high-fidelity enough for the purposes of the project.

The magnitude of various torques is also analyzed for both the early seconds of the simulation to capture any transient behaviour, as well as for a longer run of 2000 s to assess the overall contribution of each effect. Here, it can be concluded that the aerodynamic torque has the lowest contribution from the environment, at roughly two orders of magnitude less the other two effects. Gravity gradient starts out low but then increases to be the biggest component as the spacecraft starts tumbling. The magnetic component stays within one order of magnitude of its initial value throughout the entire run.

Due to their large effect on attitude dynamics, it is reassuring to see that the approximated flux density and gravity values match the ones calculated from higher fidelity models (WMM2020 and 120-degree EGM2008). For gravity, the quantity in the comparison is the "gravity anomaly": the difference in acceleration from the simple, spherical inverse-square model. The only noticeable divergence appears

in the air density calculations, which might stem from the strong time-varying behaviour of the models (air density drops at night and rises again during the day). However, due to the small scale of the aerodynamic disturbing torque, the current density model is sufficient enough for simulation purposes as it matches the order of magnitude and dynamics of NRLMSISE-00 well enough.
Another point of interest is the transient behaviour of flexible dynamics at the start of the simulation. Counting the ”bumps” (half-periods) on the magnitude plot, the first vibrational mode of the solar panel is clearly visible on the graph. The damping characteristics of the panel can also be observed, which eventually brings the mode shapes to a steady state as the panel bends against the body-frame accelerations acting in the connection point.

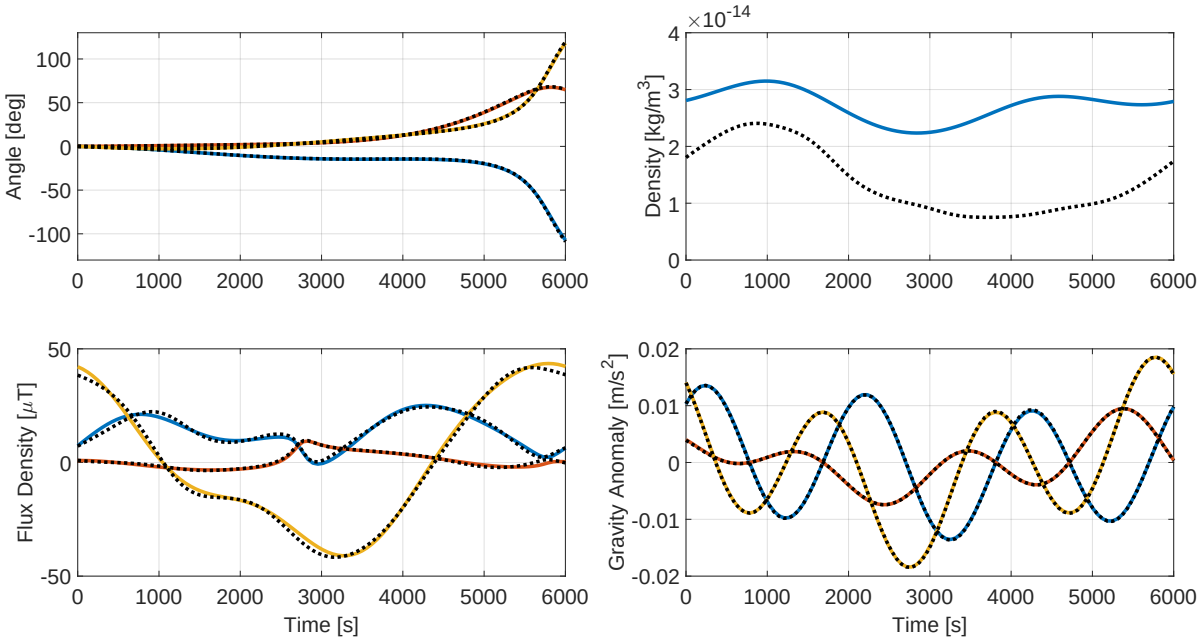

Figure 2.14: The effect of environmental models, vector components in blue-yellow-red, high-fidelity simulated values dotted.

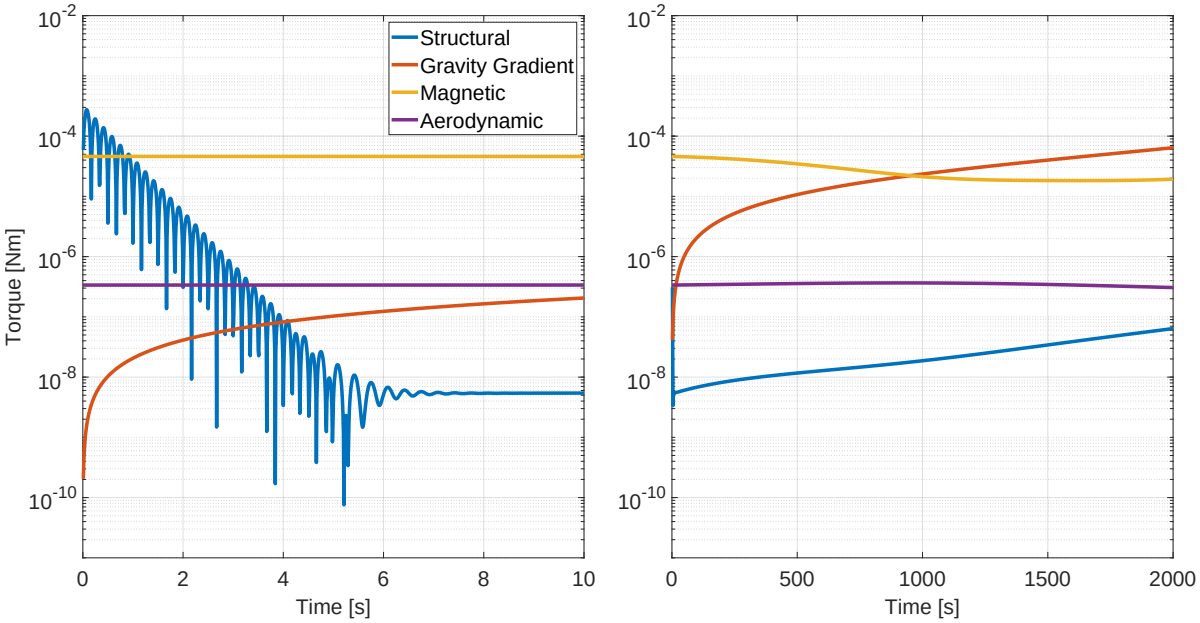

Figure 2.15: Disturbing torque magnitudes on initialization and in steady-state.

2.6. Linearization

Linearizing the spacecraft model is an important task, as the success of control design is heavily dependent on an accurate, linear model. For this chapter, consider that the inputs to the spacecraft are body-frame forces and torques in all three directions, and the outputs are the accelerations and the angular accelerations of the spacecraft body along the same axes. In the simplest, rigid case for a spacecraft of mass m and moment of inertia $\mathbf{J}$ at rest in inertial space, the following model holds:

$$\begin{bmatrix} \ddot{\mathbf{p}} \\ \ddot{\boldsymbol{\theta}} \end{bmatrix} = \underbrace{\begin{bmatrix} m\mathbf{I}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{J} \end{bmatrix}^{-1}}_{\mathbf{G}_{rigid}} \begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{bmatrix} \tag{2.63}$$

While a linear model for the environmental effects is hard to come by, one for the flexible appendage was just developed in the last section. The derived linear model can be included in a feedback loop (fig. 2.16) with the rigid system like so:

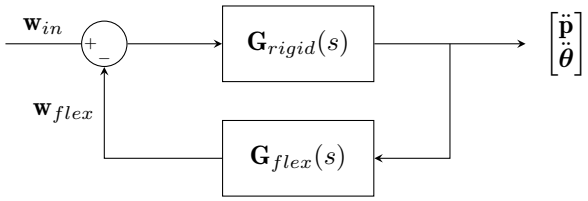

Figure 2.16: Block diagram of linearized spacecraft model $\mathbf{G}_{SC}(s)$.

Considering the model is essentially a streamlined and transformed version of the one found in `Satellite Dynamics Toolbox`, the toolbox is a good candidate to validate the final linear system. In the toolbox, it's possible to build a simple model of the spacecraft body in inertial space and add a flexible beam element with vibrational modes in all 6 degrees of freedom (fig. 2.17).

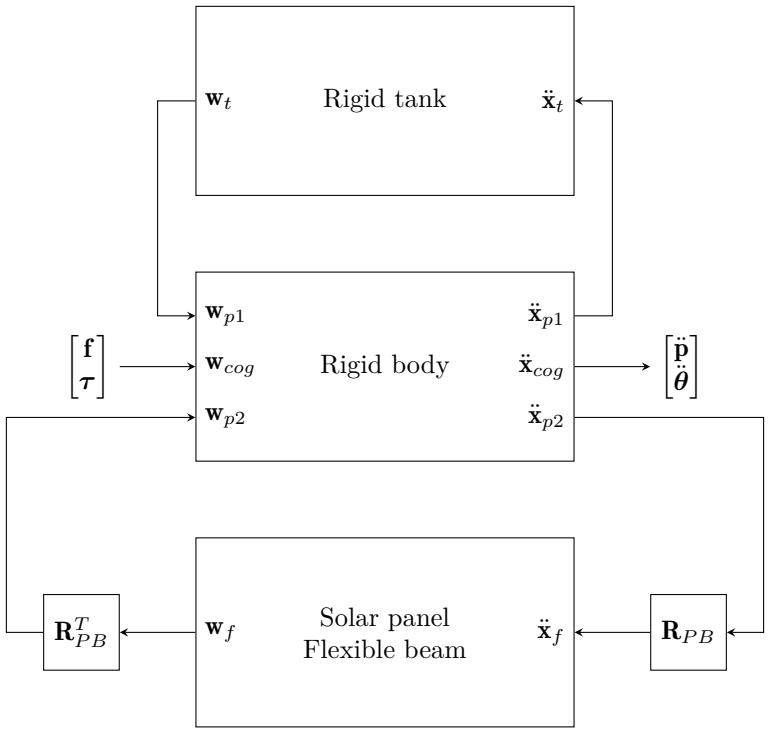

Figure 2.17: Flexible spacecraft representation in `SDT`.

In this simulation, an eighth-order Dormand-Prince solver (`ode8`) is used with a 10000 Hz simulation rate of the `SDT` model to accurately capture high-frequency behaviour. Both models are subjected to a step input of 1 N and 1 Nm in all directions (fig. 2.18). It's clear that there is little difference in the overall behaviour of the two systems, with the exception of the short-term high-frequency transients and some residual torque around the x-axis (likely as a reaction from the bending of the panel).

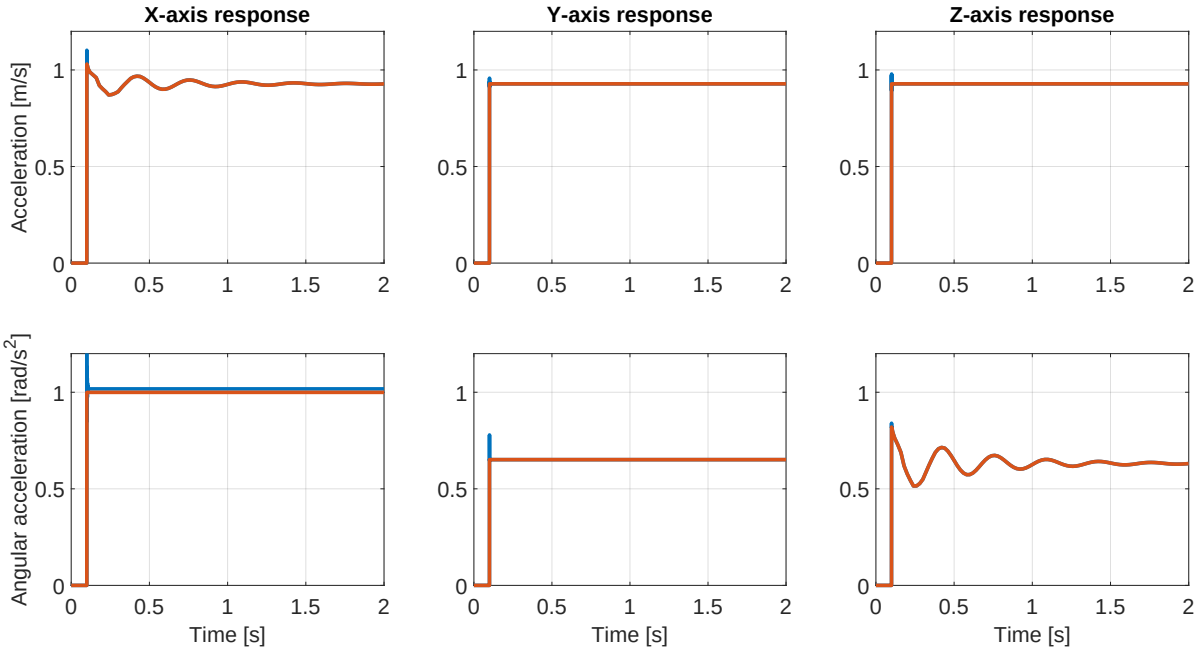

Figure 2.18: Time domain step-response of `SDT` model (blue) and simplified linear model (red).

3

Attitude Determination and Control

The attitude determination and control system (ADCS) is a critical component of any spacecraft that ensures the system stays in the correct orientation at all times. This chapter documents the modelling and design aspects of ADCS for the flexible spacecraft model that was derived in previous sections of the thesis. The scientific basis of this process was the research done throughout the literature study, which answered the following questions:

- What are the main sources of vibration on a flexible spacecraft and how are they affecting its dynamics?
  - What are the different approaches to modelling internal microvibration sources?
- How are flexible spacecraft controlled with high pointing precision?
  - What kind of control structure is needed for the pointing control of the satellite?
  - What are the state-of-the-art algorithms for pointing spacecraft at a ground target?
- What are the sensing and actuation characteristics of modern flexible spacecraft?
  - What is the selection of sensors and actuators to allow for redundancy and extreme precision?
  - How are sensor and actuator dynamics taken into account in modelling?
  - How are flexible characteristics measured?

The main goal of the resulting systems is to accurately simulate the sensing and actuation errors that might be present on a real spacecraft while providing a control solution that is both robust enough to overcome these disturbances and has enough authority to precisely guide the line-of-sight vector along the desired path for high-accuracy, stable imaging. To achieve this, a control-oriented linearized model is built based on the high-fidelity nonlinear dynamics and the reduced models of ADCS components.

3.1. Actuator modelling

Reaction wheels are a type of torque motor with high-inertia rotors. Each wheel can spin in either direction and can provide one axis of control. These rotating devices have two important parameters that define their performance: angular momentum capacity, and torque capability [31]. Due to the momentum storage of the wheels, they modify the total energy of rotational motion of the system, leading to a change in the overall dynamics of the larger nonlinear model through the reaction wheel angular momentum vector $\mathbf{h}_{rw}$ [4]:

$$\mathbf{h}_{rw,i} = \mathbf{J}_{rw}\boldsymbol{\omega}_{rw,i} \tag{3.1}$$

On most modern spacecraft, the primary instruments for attitude control actuation are reaction wheels. This is due to their high precision and wide performance range. As such, they are a natural choice for actuation in flexible spacecraft. There is one drawback: despite many of these wheels having operated flawlessly for decades, there is a growing number of missions facing problems due to wheel failures. Protection can usually be achieved through the use of redundancy in the wheel assembly [31].

Spacecraft that are equipped with more than three reaction wheels can use a range of different configurations. Redundant wheels are commonly placed in a hot backup configuration, and used in parallel with the others. One such setup is the NASA Standard Configuration (fig. 3.1), which is also used in this project due to its relatively simple design. In the model, one reaction wheel is used for each of the spacecraft's principal axes. For redundancy, an additional fourth wheel is added diagonally, at 45 degrees rotation to the three orthogonal ones.

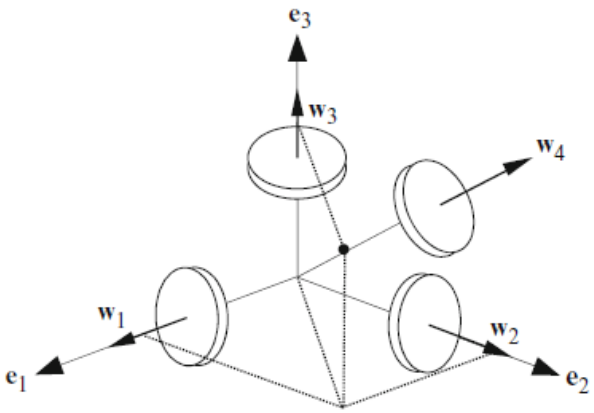

Figure 3.1: NASA Standard Configuration. [4]

In order to distribute torques between the four wheels, a pseudoinverse distribution law was applied that produces the following transformation matrices from the rotating frame to the body frame and vice versa [4]:

$$\mathbf{R}_{BR}^{rw} = \begin{bmatrix} 1 & 0 & 0 & \frac{1}{\sqrt{3}} \\ 0 & 1 & 0 & \frac{1}{\sqrt{3}} \\ 0 & 0 & 1 & \frac{1}{\sqrt{3}} \end{bmatrix} \qquad \mathbf{R}_{RB}^{rw} = \begin{bmatrix} \frac{5}{6} & -\frac{1}{6} & -\frac{1}{6} \\ -\frac{1}{6} & \frac{5}{6} & -\frac{1}{6} \\ -\frac{1}{6} & -\frac{1}{6} & \frac{5}{6} \\ \frac{1}{2\sqrt{(3)}} & \frac{1}{2\sqrt{(3)}} & \frac{1}{2\sqrt{(3)}} \end{bmatrix} \qquad \mathbf{R}_{BR}^{rw}\mathbf{R}_{RB}^{rw} = \mathbf{I}_3 \tag{3.2}$$

Because of the many publications available on its experimental behaviour, the Ithaco TW-50E300 reaction wheel was selected as the actuator of the system. Reaction wheel assemblies have various components: a rotating flywheel surrounded by ball bearings, an internal brushless DC electric motor, and electronics to control the system. The driving motor of a single wheel is modelled as a low-pass filter with a cutoff frequency of 100 rad/s, based on the experiences of the institute with BLDC motors. Wheel friction, or drag, is usually modelled as a sum of viscous and Coulomb components that are based on the velocity of the wheel (ω^w). The coefficients k_v and k_c are empirically determined and can be temperature-dependent. The model is not adequate for low speeds, where a dynamic model incorporating hysteresis is required. To avoid this, the motors are usually operated away from the motionless position. For the Ithaco wheel, these components are measured as $k_v = 10^{-6}$ Nms and $k_c = 10^{-3}$ Nm. The rotation rate is directly related to the nominal torque, which can be expressed as the difference between drag torques and the motor torque. The axial moment of inertia of the wheel $J_w = 0.2$ kgm^2 is taken from catalogue data for a wheel with 0.5 Nm of maximum torque, 50 Nms capacity, and 3850 rpm maximum rotation speed. The wheel's behaviour follows the differential equation presented below [32]:

$$\dot{\omega}^w = J_w^{-1}\tau_{nom}^w \qquad \tau_{nom}^w = \tau_{mot} - k_v\omega^w - k_c\text{sign}\left(\omega^w\right) \tag{3.3}$$

Microvibrations

Reaction wheels are also a source of attitude disturbances, which come from forces and torques in both axial and radial directions. Since disturbing forces only generate torques that are proportional to the force's distance from the center of mass, their effect can be minimized by placing the wheel close to that point.

Deviations of the wheels from perfect balance are the main contributor to disturbances. This could either be a static imbalance or a dynamic imbalance. In the case of static imbalance, the wheel's center of mass is not on the axis of rotation. In this case, the spacecraft provides a centripetal force through the bearings to accelerate the center of mass about the axis of rotation. Dynamic imbalance, on the other hand, is the condition when the axis of rotation of the wheel is not a principal axis. In this case, to keep the wheel rotating at a constant rate about the spin axis, a counter-torque is required that presents a disturbing effect. In addition to static and dynamic imbalance, a reaction wheel also exhibits structural dynamic modes (fig. 3.2) resulting from compliance in the bearings [4].

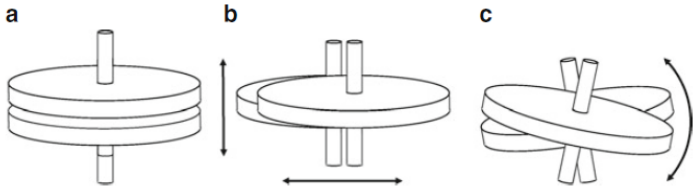

Figure 3.2: Dynamical modes of a reaction wheel: (a) axial, (b) radial, (c) rocking. [4]

The rocking modes also interact with the rotation of the wheel, resulting in whirl modes. All of these vibrational modes can be visualized on a Campbell diagram of the wheel (fig. 3.3). The fundamental mode ω^w arises from dynamics imbalance, as does the first harmonic $2\omega^w$. Other non-integer modes are the result of bearing disturbances. Intersections of the frequencies of different disturbances (marked by circles) indicate resonance conditions. If the bearing restoring forces are perfectly isotropic, this is not the case for intersections with ω^- [4].

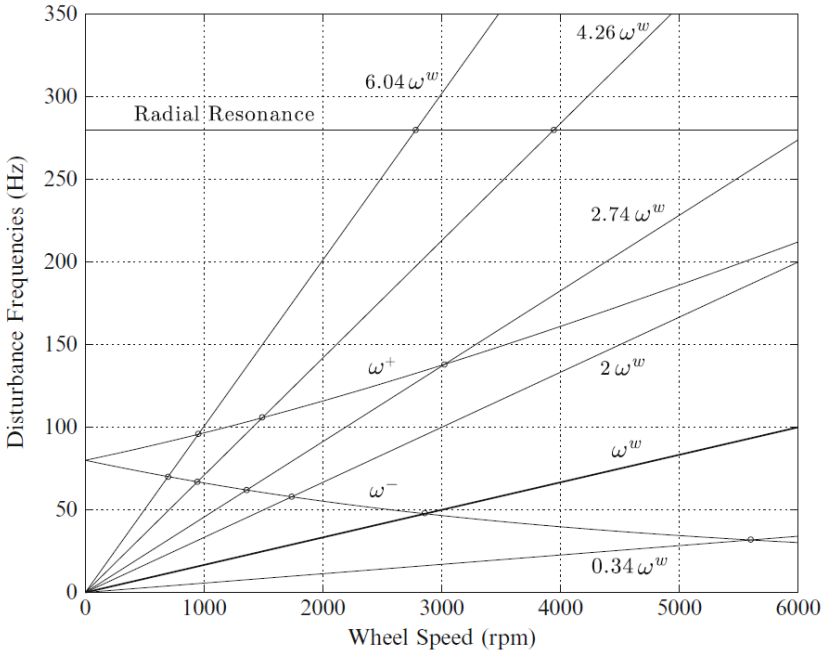

Figure 3.3: Campbell diagram of radial moment disturbances. [4]

Simulation results

Since the dynamics of reaction wheels generally have a much higher frequency than orbital or even attitude dynamics, modelling them to such high fidelity would be severely detrimental to simulation speed. However, some empirical models are able to approximate these high-frequency microvibrations with a harmonic series expansion of reaction wheel dynamics [15][32][33]. In the wheel's rocking frame, N harmonic disturbances affect the wheel, which present as forces and torques in the x-y plane and then are added to the nominal values. Additionally, there is an axial force acting along the axis of rotation. The parameters $a_i^{f,t,a}$ and $h_i^{f,t,a}$ are the same as the ones used by Masterson et al (which are estimated from experimental data - table 3.1), while $\phi_i^{f,t,a}$ are random phase angles defined during initialization.

i	h_i^f	$a_i^f\ [10^{-6}]$	h_i^a	$a_i^a\ [10^{-6}]$	h_i^t	$a_i^t\ [10^{-6}]$
1	1	3.7889	1	2.7703	1	2.0107
2	2	0.7587	1.98	2.5697	2	0.5553
3	3	0.4952	2.96	0.6556	3	0.2207
4	4	0.5663	4	0.6246	4	0.2125
5	4.42	1.0003	4.33	0.9219	4.42	0.4423
6	5.37	0.4942			5.58	0.4541
7	5.57	0.6292				

Table 3.1: Experimental parameters of the simplified reaction wheel model in SI.

As such, the full 6-DOF force and torque dynamics of the reaction wheel work out to the model presented below which is mainly dependent on the angular velocity of a given wheel:

$$\begin{bmatrix} f_i^x \\ f_i^y \\ f_i^z \\ \tau_i^x \\ \tau_i^y \\ \tau_i^z \end{bmatrix} = \begin{bmatrix} -a_i^f \omega^2 \sin\left(h_i^f \omega t + \phi_i^f\right) \\ a_i^f \omega^2 \cos\left(h_i^f \omega t + \phi_i^f\right) \\ a_i^a \omega^2 \sin\left(h_i^a \omega t + \phi_i^a\right) \\ a_i^t \omega^2 \cos\left(h_i^m \omega t + \phi_i^t\right) \\ a_i^t \omega^2 \sin\left(h_i^m \omega t + \phi_i^t\right) \\ -\tau_{nom}^w \end{bmatrix} \tag{3.4}$$

Then, the forces and torques for a single wheel are the sum of contributions from the individual harmonic components:

$$\begin{bmatrix} \mathbf{f}^{x,y,z} \\ \boldsymbol{\tau}^{x,y,z} \end{bmatrix} = \sum_i \begin{bmatrix} f_i^x & f_i^y & f_i^z & \tau_i^x & \tau_i^y & \tau_i^z \end{bmatrix}^T \tag{3.5}$$

The resulting model is presented in block diagram form in figure 3.4. Note that the transformation matrices and electric motor are not included for clarity, and equation 3.4 is represented with the parameter-varying subsystem $MV(\omega_{rw})$

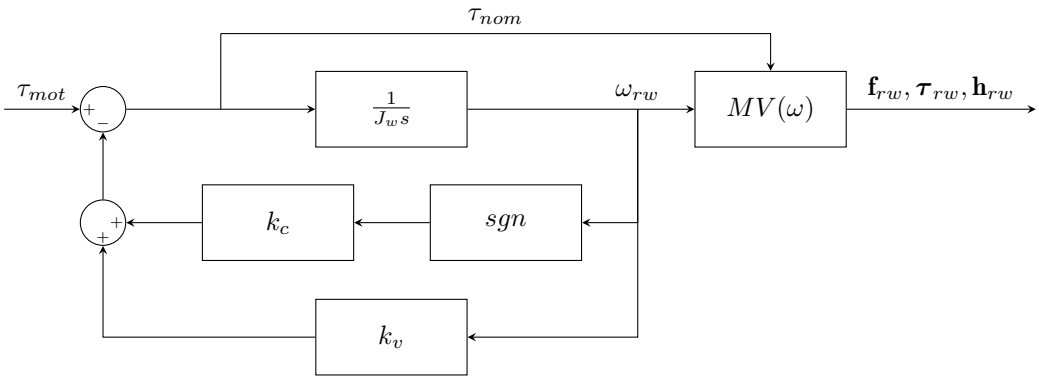

Figure 3.4: Block diagram of a single reaction wheel's mechanical model.

The model was tested in Simulink with a 10^{-4} s time step and a constant 1 Nm torque on the wheel. This leads to a linear increase in wheel speed up to 6000 rpm, as well as the gradual deterioration of its torque capabilities as disturbances rise in magnitude (fig. 3.5). The Ithaco wheel is rated up to 3850 rpm, at which point there is a roughly 2:1 ratio between nominal and disturbance torques. The disturbing torque's quadratic relation to wheel speed is also clearly visible.

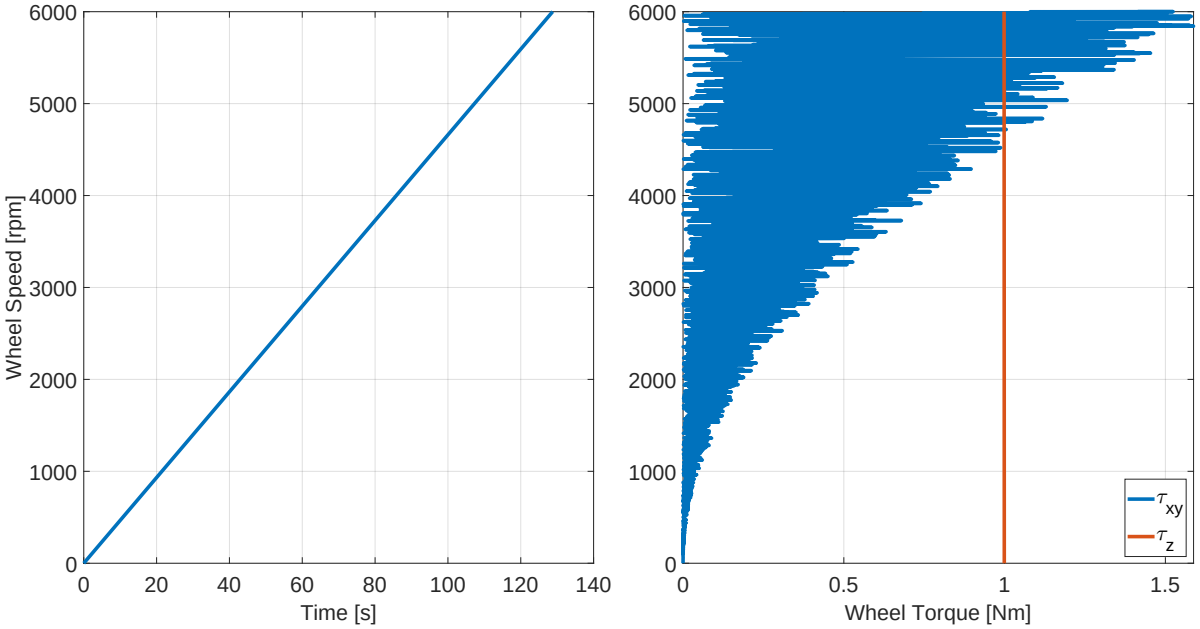

Figure 3.5: Simulated torque disturbances as a result of increasing wheel speed.

The disturbances can be better visualized using a spectrogram (fig. 3.6), in order to find similarities between the simulation results and the Campbell diagram of a generic wheel. For disturbance torques in the x-y plane, the linear relationship between wheel speeds and excited frequencies is very prominent on the chart, as the gradual rise in magnitude can also be observed under increasing wheel speeds.

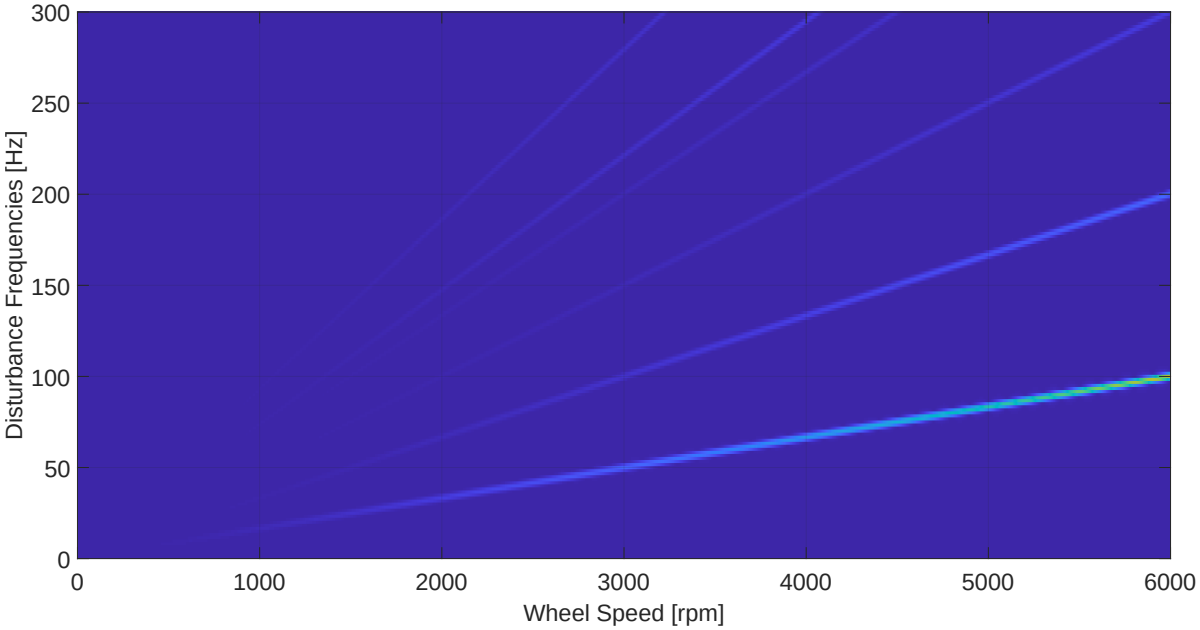

Figure 3.6: Spectrogram of simulated reaction wheel torque disturbances.

Linearization

A linearized model of the reaction wheel assembly (RWA) is constructed for the purpose of control design. It includes the transformation matrices from the body frame to the wheel frames (and back) and the four electric motor transfer functions. Microvibrations are not included in the controller tuning process due to their parameter-varying nature, but they are considered in the evaluation of control performance. The result is a 3-input 3-output LTI MIMO system from the desired torques to the reaction wheel output torques.

$$\mathbf{G}^*_{RWA}(s) = \mathbf{R}^{rw}_{BR} \underbrace{\begin{bmatrix} \frac{100}{s+100} & 0 & 0 & 0 \\ 0 & \frac{100}{s+100} & 0 & 0 \\ 0 & 0 & \frac{100}{s+100} & 0 \\ 0 & 0 & 0 & \frac{100}{s+100} \end{bmatrix}}_{\mathbf{G}_{Mot}(s)} \mathbf{R}^{rw}_{RB} \tag{3.6}$$

Note that wheel friction is excluded from this linearized model. This is done due to the low bandwidth of the process which degrades the performance of many automatic controller tuning algorithms. In the validation process, an adjusted model is used that also takes into account the linear components of wheel friction.

$$\mathbf{G}_{RWA}(s) = \mathbf{R}^{rw}_{BR} \underbrace{\begin{bmatrix} \frac{s}{s+k_v/J_w} & 0 & 0 & 0 \\ 0 & \frac{s}{s+k_v/J_w} & 0 & 0 \\ 0 & 0 & \frac{s}{s+k_v/J_w} & 0 \\ 0 & 0 & 0 & \frac{s}{s+k_v/J_w} \end{bmatrix}}_{\mathbf{G}_{Fric}(s)} \mathbf{G}_{Mot}(s)\mathbf{R}^{rw}_{RB} \tag{3.7}$$

3.2. Sensor modelling

Due to the use of 3-axis stabilization in precise pointing missions to meet the required attitude accuracy, a higher class of sensors is needed for state estimation. This usually means a top-shelf star sensor for orientation and precision GPS for position determination. However, the structural modes of flexible spacecraft have a higher bandwidth than what these sensors can cover. To expand the capabilities of the control algorithm, gyros and accelerometers are also used in an IMU (inertial measurement unit) package [31].

This selection of sensors is supported by many of the relevant missions examined in the literature review process, the principal one being JAXA's Advanced Land Observing Satellite (ALOS), a flagship mission for high-resolution Earth observation in the 2000s and one of the first high precision flexible spacecraft in orbit [12]. As a more recent example, the sensor suite of ESA's Sentinel-2 also proved to be a great source of inspiration for the design of this project, whose normal mode of operations also relies on a fusion of IMU, GPS, and star tracker measurement data. The exact hardware selection and characteristics have a decent record in publications and allowed this project to find modern products that match or exceed their performance [34].

In conclusion, to estimate the states, four physical quantities are measured and then mixed with normally distributed random noise in the sensor models. ECEF position is measured with high-precision GPS, Body-frame velocity with an integrated accelerometer, body-frame angular velocity with a drifting gyro, and Euler angles with a star tracker. This section presents all of these models in detail, which are largely based on the examples in Bayard's *High Precision Three-Axis Pointing and Control* [35].

GPS characteristics

GPS (Global Positioning System) sensors are frequently used in space applications for position and velocity sensing, for example in the Sentinel program. The current state-of-the-art of precise real-time orbit determination in LEO is described in detail by Montenbruck et al, supported by six months of Sentinel-6 measurement data [36]. Following their findings, the sensor is modelled as a white noise ($\mathbf{n}_{GPS}$) block being added to the ECEF position of the spacecraft [4].

$$\mathbf{r}_{GPS} = \mathbf{r}^F_{BF} + \mathbf{n}_{GPS} \tag{3.8}$$

This noise block has no bias and 10^{-2} variance at 1 Hz, matching the GPS performance of a Sentinel-class spacecraft at 0.2 m (fig. 3.7).

Star tracker characteristics

Star trackers (STR) are among the most precise sensors used in the three-axis stabilization of spacecraft, providing attitude measurements with errors on the scale of a few arcseconds. They are essentially optical cameras with sophisticated algorithms matching visible stars to a known database of celestial coordinates. The performance of contemporary star trackers was reviewed in 2019 by Fialho et al in a rigorous analysis [13]. In this project, their sensing characteristics are modelled with additive white noise ($\mathbf{n}_{SS}$) to the Euler angles of the spacecraft.

$$\mathbf{q}_{SS} \sim \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}_{SS} = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}_{BF} + \mathbf{n}_{SS} \tag{3.9}$$

This noise block has no bias and 10^{-11} variance at 10 Hz, to reach the current state-of-the-art performance given by Fialho et al at 2 arcsecs (fig. 3.7).

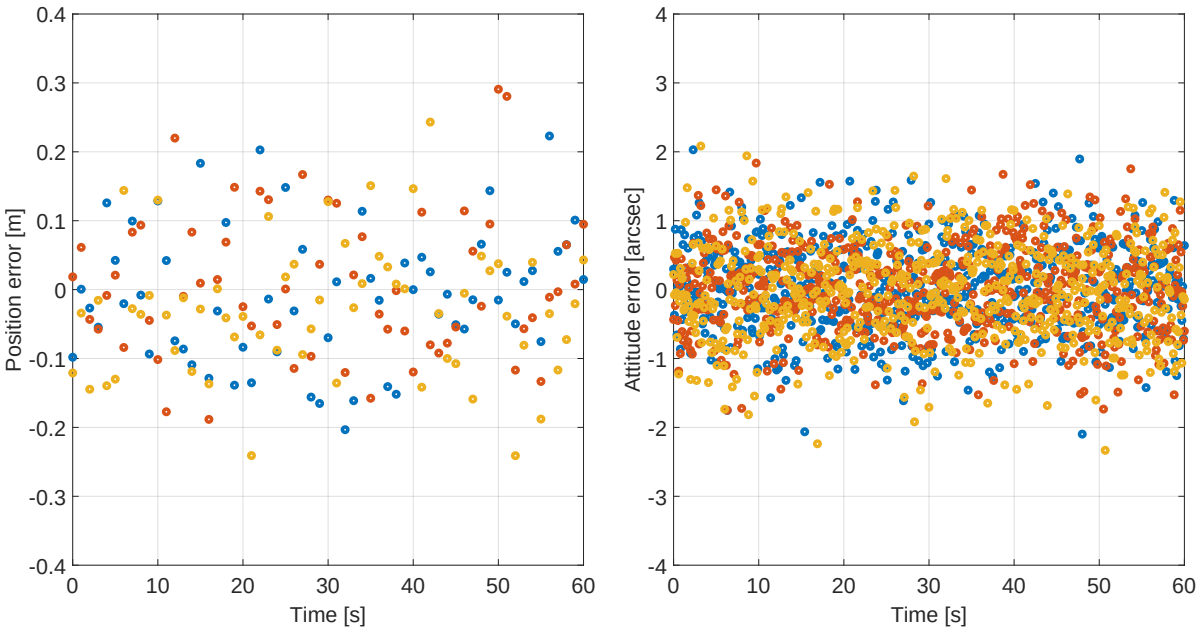

Figure 3.7: Sensor noise on 3-axis GPS and STR measurements (each color an axis).

IMU characteristics

An inertial measurement unit is a device that measures the acceleration and angular rate of a body relative to a known starting point with a combination of accelerometers and gyroscopes in an orthogonal configuration. They have decent accuracy and high bandwidth but are very susceptible to sensor drift due to the integrative nature of measurements. As a result, they need to be regularly calibrated to clear any accumulated errors [37].

The sensor characteristics of the project were selected from the Astrix line of gyros from Airbus, and the accelerometers found in Honeywell's IMU suites [38][39]. The aerospace-grade products of these manufacturers are featured in a wide variety of missions and were found to be performant enough to be included in a precise pointing application.

The accelerometer has a single noise component ($\mathbf{d}_{ACC}$) that is integrated and added to the measured body-frame velocity.

$$\mathbf{v}_{IMU} = \mathbf{v}_{BF}^{B} + \frac{\mathbf{d}_{ACC}}{s} \tag{3.10}$$

This noise block has no bias and 10^{-9} variance at 100 Hz, which matches the desired 0.01 mg precision of Honeywell sensors. These characteristics also result in a position drift on the scale of 0.1-1 m over an hour of runtime (fig. 3.8).

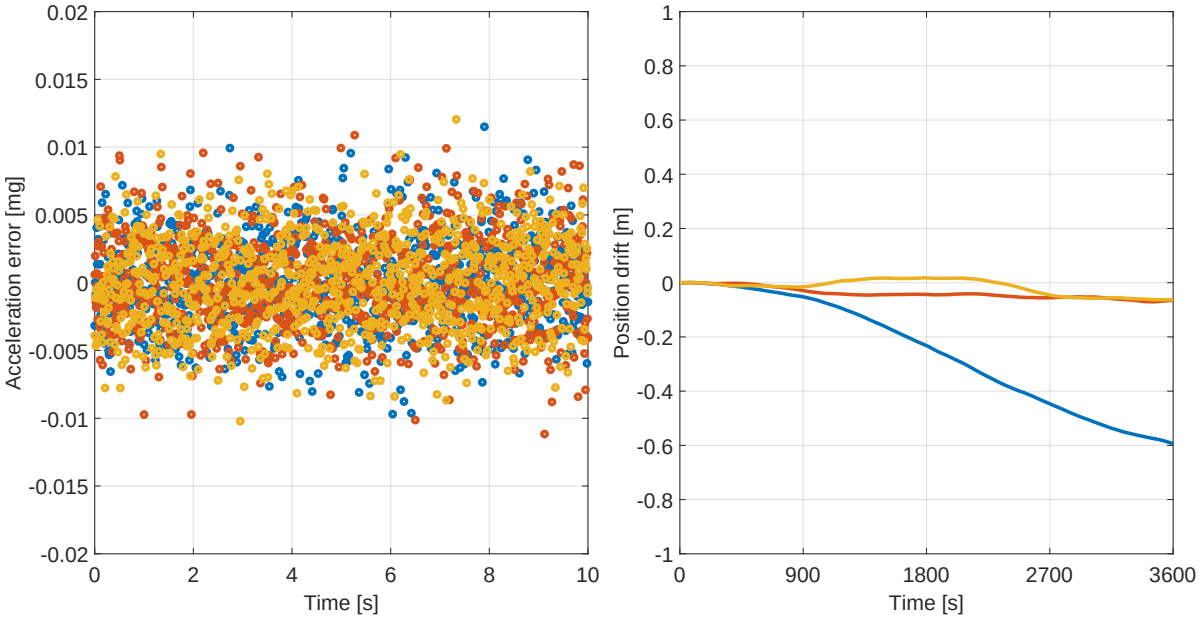

Figure 3.8: Sensor noise and drift on 3-axis accelerometer measurements (each color an axis).

The gyroscope has two noise components that are added to the body-frame angular velocity vector: an integrated noise source to model sensor drift ($\mathbf{d}_{GYR}$), and another white noise source to model angle random walk ($\mathbf{n}_{GYR}$).

$$\boldsymbol{\omega}_{IMU} = \boldsymbol{\omega}_{BI}^{B} + \mathbf{n}_{GYR} + \frac{\mathbf{d}_{GYR}}{s} \tag{3.11}$$

The noise source for sensor drift has no bias and $7 \cdot 10^{-18}$ variance at 100 Hz. The source for angle random walk has also no bias and 10^{-18} variance at 100 Hz. This ensures the matching of Astrix gyros which have a 0.0006 deg/hr precision in angular rate, and 0.002 deg/$\sqrt{hr}$ Allan random walk (ARW), meaning after 1 hour of integration the standard deviation is 0.002 deg [37] (fig. 3.9).

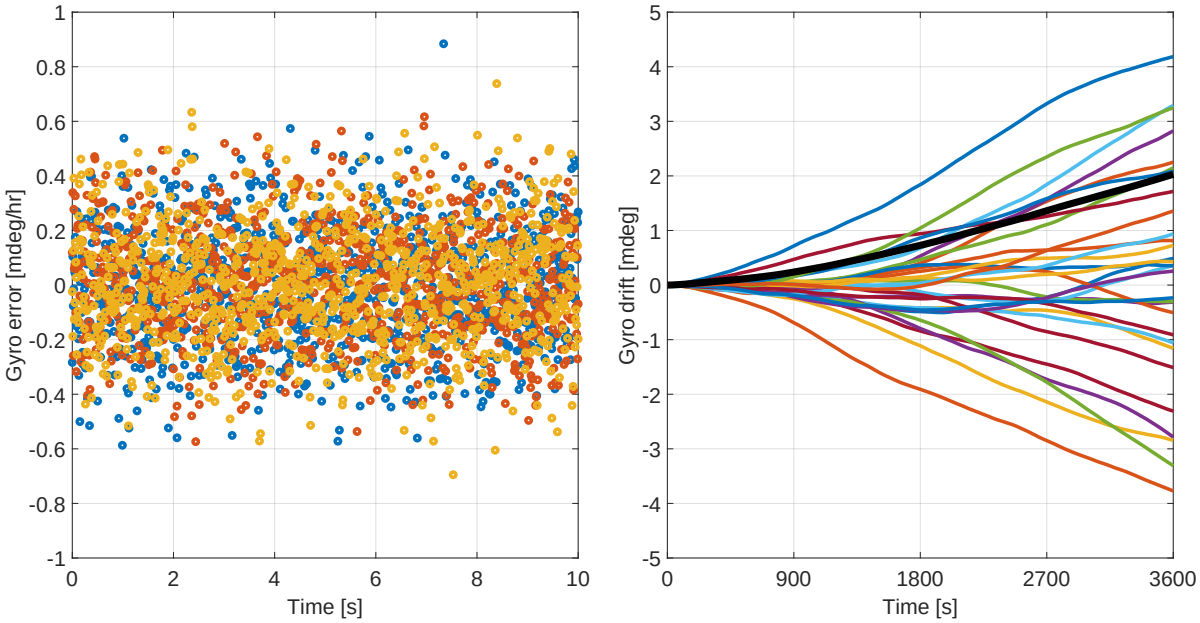

Figure 3.9: Left: Sensor noise on 3-axis gyro measurements (each color an axis). Right: Standard deviation of gyro drift in black compared to random samples of ARW.

3.3. State estimation

Raw sensor measurements are often too noisy to define an attitude reference and control the spacecraft. Additionally, the outputs of the sensors are not the same physical quantities as the ones required by these algorithms. Therefore, a state estimator block is constructed that takes care of filtering and processing the measured signals. Usually, this process would be handled by Kalman filtering, however in order to limit the scope of the thesis, a "simulator-based" approach is taken. At the cost of some performance loss in sensor fusion, this removes all tuning and linearization requirements in designing the estimator.

The raw sensor output signals are filtered with a first-order holder and then fed into a system of equations based on the previously derived equations of motion. The outputs of state estimation to be used in reference generation and control are ECEF position $\mathbf{r}_B^F$, ECEF velocity $\mathbf{v}_B^F$, body-frame angular velocity $\boldsymbol{\omega}_{BI}^B$, and the Euler angles $\boldsymbol{\theta}$. To handle sensor drift, the integrators driven by the IMU's gyro and the accelerometer are periodically reset with more accurate lower frequency measurements from the GPS and star tracker. The state estimator also has some time delay (0.01 s) to model the duration of electrical processes, which is linearized as a 5th order Padé approximant in the control-oriented model with a safety factor of 10 [6].

$$e^{-10\cdot 0.01s} \approx G^*_{Pade}(s) = -\frac{s-72.93}{s+72.93}\frac{s^2-134.1s+5709}{s^2+134.1s+5709}\frac{s^2-92.99s+7263}{s^2+92.99s+7263} \tag{3.12}$$

Of course for analytical purposes, the exact length of the time delay should be used in modelling:

$$e^{-0.01s} \approx G_{Pade}(s) = -\frac{s-729.3}{s+729.3}\frac{s^2-1341s+57090}{s^2+1341s+57090}\frac{s^2-929.9s+72630}{s^2+929.9s+72630} \tag{3.13}$$

The 5th-order approximation allows the control tuning algorithms to take into account the dead time characteristics of the embedded hardware to a degree that was deemed acceptable for this project (fig. 3.10). Further increasing the accuracy of the approximant would have led to an increase in the states of the system, which is undesirable in control design.

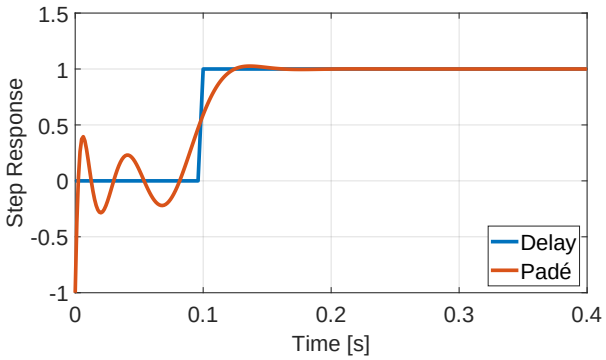

Figure 3.10: Padé approximation of time delay function.

In the estimation algorithm, the first step is determining the body-frame angular velocity $\boldsymbol{\omega}_{sen} = \boldsymbol{\omega}_{BI}^B$, which is the exact same as the output of the gyro:

$$\boldsymbol{\omega}_{sen}[T] = \boldsymbol{\omega}_{IMU}[T] \tag{3.14}$$

Then, the attitude quaternion $\mathbf{q}_{sen} = \mathbf{q}_{BL}$ and ECEF-to-Body rotation matrix $\mathbf{R}_{sen} = \mathbf{R}_{BF}$ are calculated from a discretized version of the kinematic model that was detailed in the previous chapter.

$$\dot{\mathbf{q}}_{sen}[T] = \frac{1}{2}\mathbf{q}_{sen}[T-1] \circ \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{sen}[T-1] - \mathbf{R}_{sen}[T-1]\cdot\boldsymbol{\omega}_{\oplus}^F \end{bmatrix} \tag{3.15}$$

Here, the resulting derivative of the quaternion is numerically integrated using the previously mentioned method of resetting the integrator block (fig. 3.11) with incoming star sensor measurements.

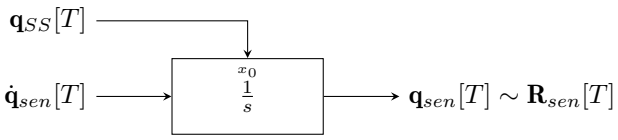

Figure 3.11: Block diagram of quaternion integrator.

Afterwards, the equations of translational motion are used to calculate the ECEF position $\mathbf{r}_{sen} = \mathbf{r}_{BF}^{F}$ and ECEF velocity $\mathbf{v}_{sen} = \mathbf{v}_{BF}^{F}$ vectors.

$$\mathbf{v}_{sen}[T] = \mathbf{R}_{sen}^{-1}[T] \cdot \mathbf{v}_{IMU}[T] \tag{3.16}$$

To obtain the position vector, a similar method is used for integration (fig. 3.12) as previously shown for the quaternion derivative.

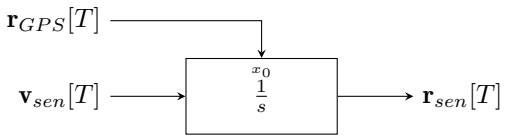

Figure 3.12: Block diagram of position integrator.

These equations already allow for the estimation of orbital dynamics. In order to visualize the data better, only the first 60 seconds of simulation runtime are shown with the number of data points greatly reduced in frequency (fig. 3.13).

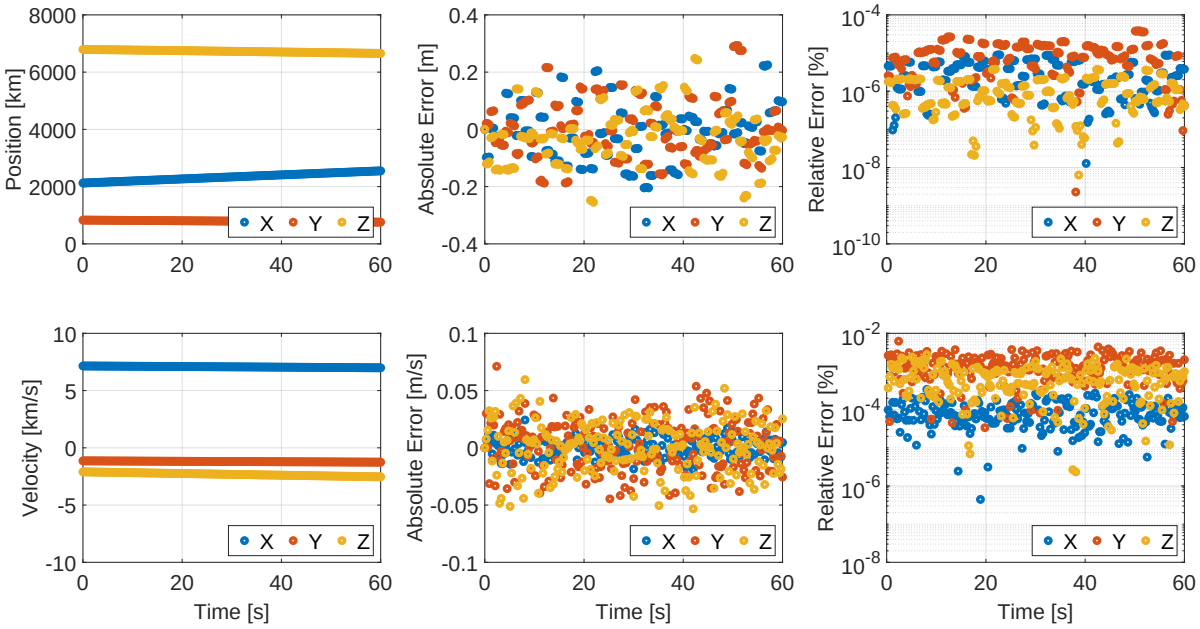

Figure 3.13: Estimated states of orbital dynamics.

The absolute errors are no surprise after the performance of sensor models developed in the previous section. The accelerometer aided by the high-precision GPS sensor provides reliable position data at 0.2 m accuracy which translates to a relative error between $10^{-7} - 10^{-5}$. Some "choppiness" is visible on the absolute error plot, which can be attributed to the regular resets of the position integrator.
This jagged data stream gives way to slightly drifting white noise when looking at the absolute error of velocity measurements. The relative error is slightly lower at $10^{-5} - 10^{-3}$, which is explained in part by the difference in numeric scale between spacecraft positions and velocities. Overall, it can be noted that there are no significant differences between the measurements made on the three different axes, and they are in line with the expected values as seen in the AOCS datasheets of the analyzed missions.

Finally, the Euler angles are calculated from the processed position, velocity, and quaternion signals using the previously presented method for constructing the LVLH-to-ECEF rotation matrix:

$$\begin{aligned} \mathbf{o}_1[T] &= \mathbf{o}_2[T] \times \mathbf{o}_3[T] \\ \mathbf{o}_2[T] &= -\mathbf{r}_{sen}[T] \times \left(\mathbf{v}_{sen}[T] + \boldsymbol{\omega}_{\oplus}^{F} \times \mathbf{r}_{sen}[T]\right) \\ \mathbf{o}_3[T] &= -\mathbf{r}_{sen}[T] \\ \mathbf{R}_{LF}[T] &= \begin{bmatrix} \hat{\mathbf{o}}_1[T] & \hat{\mathbf{o}}_2[T] & \hat{\mathbf{o}}_3[T] \end{bmatrix} \end{aligned} \tag{3.17}$$

The angles follow an XYZ rotation order, which makes the line-of-sight vector (body frame "Z") independent of the yaw angle. This is beneficial in two ways: first, the complexity of trajectory generation is vastly reduced as pointing precision only requires two Euler angles. Second, a potential operations team would gain more options as the yaw angle becomes a free parameter.

$$\mathbf{R}_{sen}[T] \cdot \mathbf{R}_{LF}[T] = \mathbf{R}_{LB}[T] \longrightarrow \phi[T], \theta[T], \psi[T] \tag{3.18}$$

Similarly to the orbital dynamics, only the first 60 seconds of simulation runtime are shown with the number of data points greatly reduced in frequency to aid in the visualization of estimator performance (fig. 3.14).

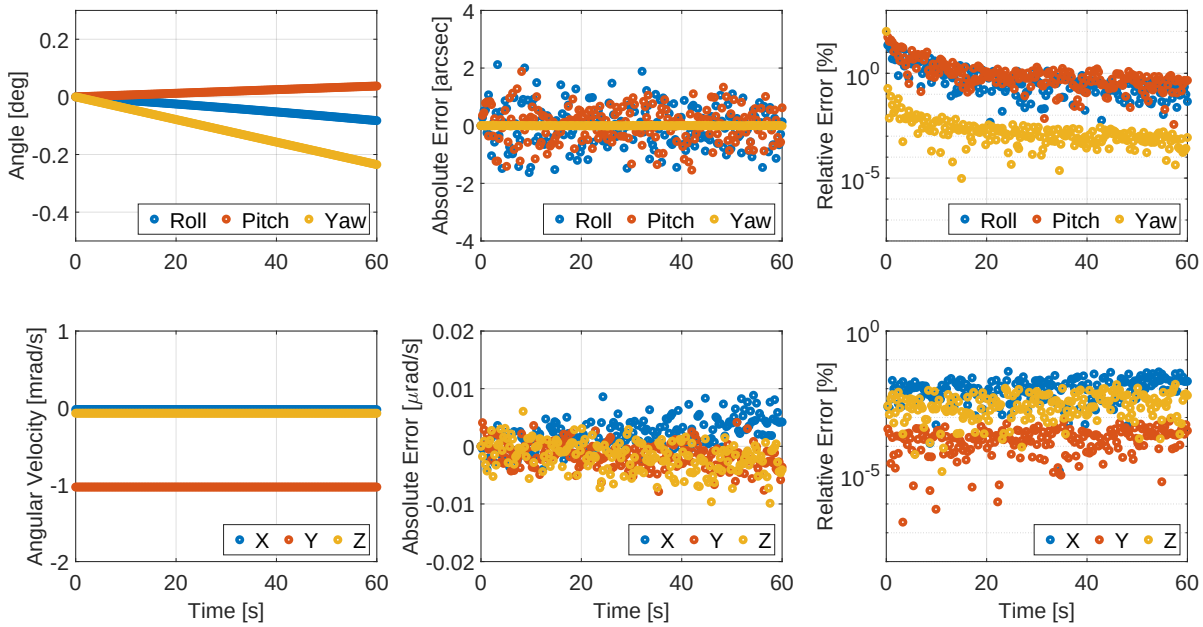

Figure 3.14: Estimated states of rotational dynamics.

Unlike the estimation of orbital states, the estimated rotational states uncover some interesting dynamical properties of the algorithms. First off, due way sensor data is combined in the calculation of the Euler angles, yaw errors are much lower than roll and pitch, by about 2 orders of magnitude. This could have some uses in a potential future system identification experiment, as the main flexible modes largely excite the yaw dynamics of the system. Relative error on the Euler angles starts off quite bad around the start time of the simulation, largely due to the initial conditions being set around zero and some transient properties of the Simulink solver. This indicates the need for a waiting period after startup time in order to properly assess control performance.
In the angular velocity error charts, a drifting white noise behaviour is visible, which is the main drawback of the "simulation-based estimator". While orientation data is periodically reset with precise, non-drifting measurements from the star trackers, the same can not be done again with angular rates. As a result, rate control on the spacecraft will gradually decrease over the simulation time span. While this is still acceptable for the relatively short maneuvers detailed in this thesis, building a Kalman-based estimator algorithm is necessary to solve this issue and future-proof the project.

3.4. Attitude reference definition

The geometry of pointing control for an Earth-observing spacecraft is considered in the general case, such as the one presented by Lian et al [40]. The nadir vector of the local frame aligns with the optical axis of the spacecraft, which needs to be pointed at some ground point on the surface of the planet. This is essentially a rotation between two unit vectors, which is accomplished in the paper by building a rotation matrix from the nadir vector to the desired pointing vector and then converting it to Euler angles. However, this approach heavily relies on the use of inverse trigonometric functions which have undesirable qualities in terms of computational performance and the existence of singularities.
For this reason, a quaternion-based algorithm was developed for trajectory generation, which solves the following question: what is the quaternion $\mathbf{q}_{BL}$ that rotates the nadir vector $\mathbf{i}_3$ ("Z" base of LVLH) into the vector $\mathbf{r}_{TB}$ that connects the position of the spacecraft $\mathbf{r}_{BF}$ with the ground target position $\mathbf{r}_{TF}$? The first vector corresponds to the line-of-sight vector where all Euler angles are zero, while the second vector defines the desired orientation of the spacecraft's optical axis. Since the coordinates of both the ground target and the spacecraft are given in the ECEF frame, first a rotation matrix from the ECEF to LVLH frames needs to be constructed from available sensor data:

$$\begin{aligned} \mathbf{o}_1[T] &= \mathbf{o}_2[T] \times \mathbf{o}_3[T] \\ \mathbf{o}_2[T] &= -\mathbf{r}_{sen}[T] \times \left(\mathbf{v}_{sen}[T] + \boldsymbol{\omega}^F_{\oplus} \times \mathbf{r}_{sen}[T]\right) \\ \mathbf{o}_3[T] &= -\mathbf{r}_{sen}[T] \\ \mathbf{R}_{LF}[T] &= \begin{bmatrix} \hat{\mathbf{o}}_1[T] & \hat{\mathbf{o}}_2[T] & \hat{\mathbf{o}}_3[T] \end{bmatrix} \end{aligned} \tag{3.19}$$

Then, both the targeting and nadir vectors are calculated for the current time step:

$$\mathbf{r}^L_{TB}[T] = \mathbf{R}_{LF}[T]\left(\mathbf{r}^F_{TF}[T] - \mathbf{r}^F_{BF}[T]\right) \qquad \mathbf{i}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{3.20}$$

The desired rotation quaternion can then be found with a simple combination of scalar and vector products of these two vectors. The targeting vector is normalized to improve numerical performance, then the constructed 4-length vector is normalized to unit length in order to convert it to quaternion form.

$$\mathbf{q}_{BL}[T] \sim \begin{bmatrix} 1 + \mathbf{i}_3 \cdot \hat{\mathbf{r}}^L_{TB}[T] \\ \mathbf{i}_3 \times \hat{\mathbf{r}}^L_{TB}[T] \end{bmatrix} \tag{3.21}$$

Reaching the desired targeting orientation usually requires a large slew angle which most control algorithms would fail to complete on their own without significant overshoot or the excitation of flexible modes in the system. In order to reduce the required control effort, some smoothing is required [17]. In this thesis, a polynomial smoothing factor is applied to the attitude reference calculations. The value of the smoothing factor starts at zero and ends at one when reaching the end of the defined smoothing time. The derivatives of the polynomial are all required to be zeroes up to the third order on both ends, which ensures continuity in the reference signal and further reduces the demanded control effort.

$$\begin{aligned} y(x) &= P_1 x^7 + P_2 x^6 + P_3 x^5 + P_4 x^4 + P_5 x^3 + P_6 x^2 + P_7 x^1 + P_8 \\ y(0) &= y'(0) = y''(0) = y'''(0) = 0 \\ y'(T_{max}) &= y''(T_{max}) = y'''(T_{max}) = 0 \qquad y(T_{max}) = 1 \end{aligned} \tag{3.22}$$

As it turns out, the solution to this problem can be parametrized by the smoothing length T_{max} to reduce all coefficients to integers (eq. 3.23). This is greatly beneficial to calculation speed as it reduces this section of an algorithm to a simple multiplication of integers with some power of a double-precision floating-point number between 0 and 1.

$$P_{1..8} = \begin{bmatrix} -\frac{20}{T^7_{max}} & \frac{70}{T^6_{max}} & -\frac{84}{T^5_{max}} & \frac{35}{T^4_{max}} & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.23}$$

For the current case of single-point tracking, a smoothing time of 150 s was selected to ensure that the slew maneuver can be completed within the torque limits of the reaction wheel assembly. Throughout the thesis, various other smoothing times are used which are derived from the same parametrization and presented in fig. 3.15.

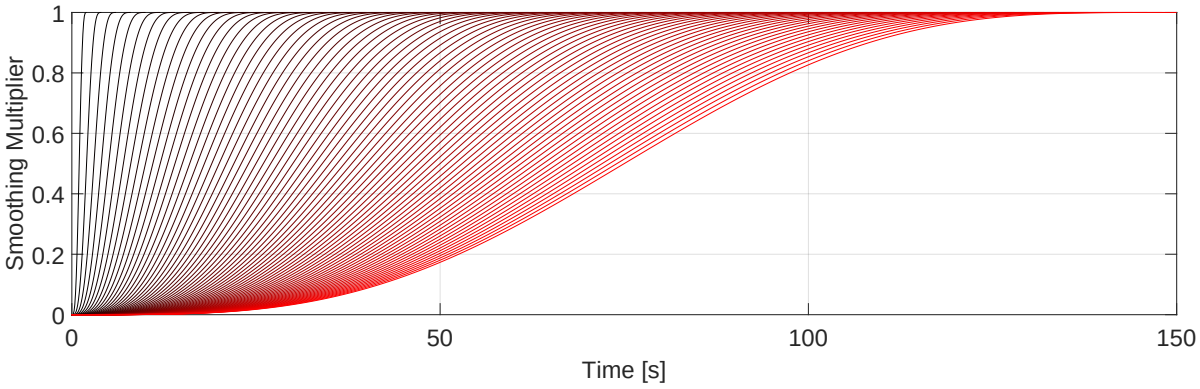

Figure 3.15: Generated smoothing multiplier curves for various time intervals.

In order to calculate the reference signal for the angular velocities, the measured position and velocity of the spacecraft are used to propagate its orbit 0.01 s ahead in time using the Bogacki–Shampine method on the orbital equations of motion [41].

$$\mathbf{y}'[T] = f(T, \mathbf{y}[T]) = \begin{bmatrix} \mathbf{r}'[T] \\ \mathbf{v}'[T] \end{bmatrix} = \begin{bmatrix} \mathbf{v}[T] \\ -\mu_{\oplus} \frac{\hat{\mathbf{r}}[T]}{r[T]^2} - \boldsymbol{\omega}_{\oplus} \times (\boldsymbol{\omega}_{\oplus} \times \mathbf{r}[T]) - 2\boldsymbol{\omega}_{\oplus} \times \mathbf{v}[T] \end{bmatrix} \tag{3.24}$$

$$\begin{aligned} \mathbf{k}_1[T] &= f\left(T, \mathbf{y}[T]\right) \\ \mathbf{k}_2[T] &= f\left(T + \frac{1}{2}T_s, \mathbf{y}[T] + \frac{1}{2}T_s\mathbf{k}_1[T]\right) \\ \mathbf{k}_3[T] &= f\left(T + \frac{3}{4}T_s, \mathbf{y}[T] + \frac{3}{4}T_s\mathbf{k}_2[T]\right) \\ \mathbf{y}[T+1] &= \mathbf{y}[T] + \frac{2}{9}T_s\mathbf{k}_1[T] + \frac{3}{9}T_s\mathbf{k}_2[T] + \frac{4}{9}T_s\mathbf{k}_3[T] \end{aligned} \tag{3.25}$$

Then, the desired rotation quaternion is also calculated, normalized, and smoothed for the propagated states. Finally, both are rotated to represent ECEF-to-Body quaternions ($\mathbf{q}_{BF}$). Then, first-order numerical differentiation is applied to get $\dot{\mathbf{q}}_{BF}$, which was plugged into the inverted rotation kinematics to define the desired $\boldsymbol{\omega}^B_{BI}$ [42].

$$\mathring{\boldsymbol{\omega}}^B_{BI,T}[T] = \frac{\frac{1}{T_s}\left(\mathbf{q}_{BF}[T+1] - \mathbf{q}_{BF}[T-1]\right)}{\mathbf{q}_{BF}[T]} + \begin{bmatrix} 0 \\ \mathbf{R}_{BF}[T]\boldsymbol{\omega}_{\oplus} \end{bmatrix} \tag{3.26}$$

Since the control algorithms use Euler angles, a single conversion from quaternions is still needed. However, since the Yaw angle represents a rotation around the optical axis in the XYZ rotation order, it can be set to zero for the entire length of the observation period, which slightly decreases the computational effort required to determine the angles, reduced momentum buildup on the reaction wheel, and limits the excitation of the flexible structure.

3.5. Outer loop control design

For the main loop that will control the orientation and rotation rate of the spacecraft with reaction wheels, three controllers are designed. First, a linear PID as a baseline. Second, a robust H_∞-based controller that would improve performance by taking into account the noise and uncertainty characteristics of the system. And third, a model predictive controller (MPC) whose main advantage is the online optimization effort that seeks the best path to matching future samples of the reference signal.

On control synthesis

First, a closed-loop is constructed that consists of a state estimator with sensor models, a trajectory generator using sensor measurements, and a controller that calculates desired torques to be delivered by the reaction wheels. The control-oriented linear model ($\mathbf{G}_{SC}(s)$) serves as the basis of control design.

There, the linearized reaction wheel torques ($\mathbf{G}^*_{RWA}(s)$) are free of vibrations and friction but still have transient behaviour related to motor dynamics. The reduced flexible dynamics only respond to torques, as force control is not part of the design process. To prepare the controller for real-world systems, a Padé-approximant of time delay is added with a safety factor of 10 ($\mathbf{G}^*_{Pade}(s)$). A block diagram of the plant structure with the double integrators is presented in figure 3.16.

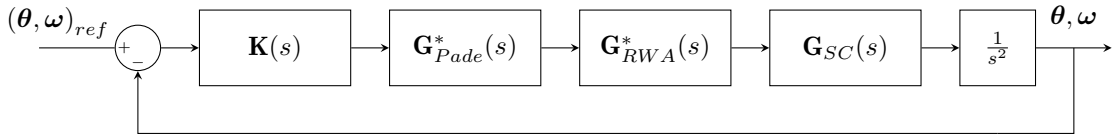

Figure 3.16: Linearized system model with time delay and actuator dynamics.

This closed-loop is then tuned by adjusting the parameters of the controller, either using the automatic methods of Matlab toolboxes or manually changing the numerical values. Performance is iteratively evaluated by the step response and frequency-domain characteristics of the linearized closed-loop, as well as the pointing performance in the high-fidelity nonlinear environment for a simple scenario.
This scenario is the single-point tracking case, whose reference angles and angular rates are presented in fig. 3.17. There, the satellite starts from the previously defined initial conditions and is pointed at the tip of the TU Delft Library, at latitude 52.00265 deg and longitude 4.3756 deg. The precise pointing segment of this trajectory is used in the temporal design of all controllers in this chapter, which completes the otherwise suboptimal frequency-domain design in an iterative manner.

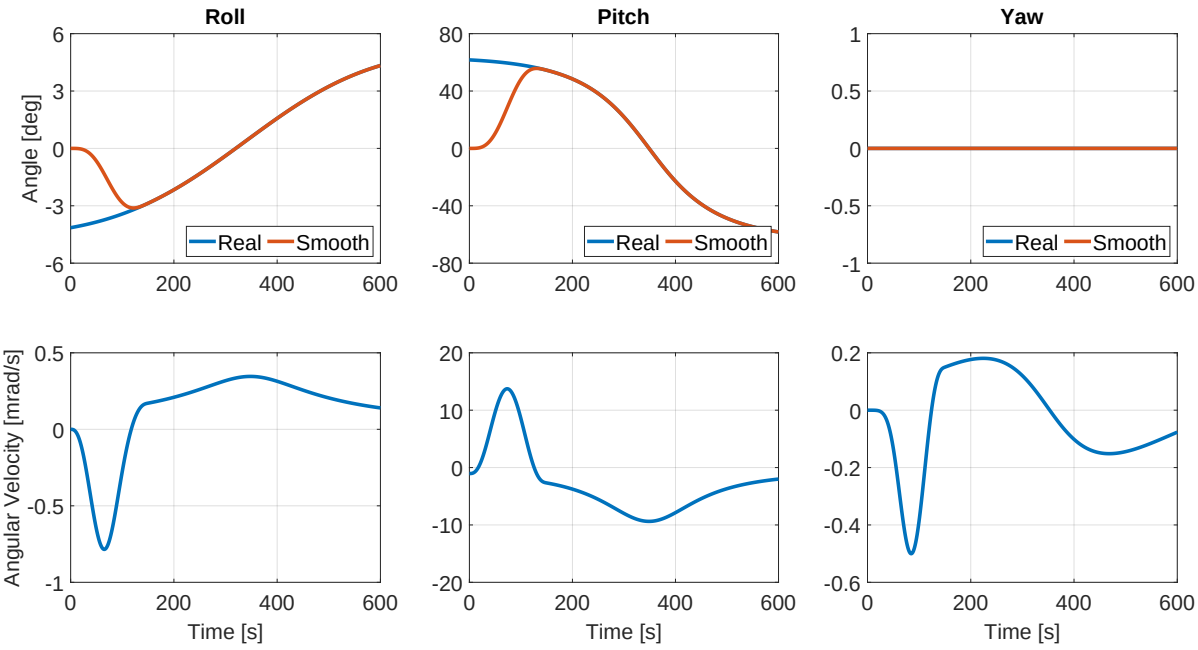

Figure 3.17: Comparison of ideal and smooth reference attitude angles and angular rates around Euler angles.

On control analysis

For the purposes of linear control analysis, the closed-loop system is a simple negative feedback loop of the controller and the linearized system dynamics (fig. 3.18). The plant, $\mathbf{G}_{Lin}(s)$ is slightly different from the one used for control design, as the reaction wheel model ($\mathbf{G}_{RWA}(s)$) is not subjected to friction, and time delay is not approximated with a safety factor anymore ($\mathbf{G}_{Pade}(s)$).
The analysis of a controller starts with a pole-zero map. For a given closed-loop system, **poles** (marked x) are eigenvalues of the matrix A in a state-space representation, and roots of the denominator polynomial in a transfer function representation. They indicate system stability, in that a linear dynamic system is only stable if all the poles are in the left half-plane, i.e. they have a real component smaller than zero. **Zeroes** (marked o) arise from competing effects within the system, and they cause zero outputs even when the inputs and states are not zero. From the state-space matrix, zeroes are usually found as

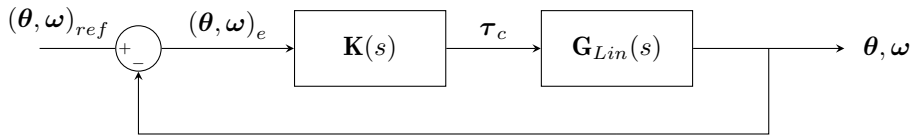

Figure 3.18: Block diagram of closed-loop control model.

non-trivial solutions to the eigenvalue problem in which case the system loses rank. They can also be found as the roots of the numerator polynomial in a transfer function representation [6].
The pole-zero map of the closed-loop system was generated after some reduction steps took place. Most notably, numerical inaccuracies in Matlab result in some poles and zeros very close to the origin (and in some cases in the right half-plane) that can be cancelled out in pairs with some tolerance using the `minreal` function. Another useful function is `balred`, which reduces the order of the system using a balanced realization in order to remove the lowest singular values that have little effect on system dynamics. These two functions are used at several points in the thesis to improve numerical consistency, the performance of automated control design algorithms, and visualization.
Another tool for frequency-domain analysis is a Bode plot, which presents the magnitude and phase of the system response as a function of sinusoidal input frequencies. The open-loop Bode diagrams (left half of figure 3.21) are important for validating stability criteria. The greater the phase and gain margins of the open-loop system, the better the stability of the closed-loop system will be. **Gain margin** is the amount of gain by which the curve of the Bode magnitude diagram is below 0 dB at the frequency where the phase is equal to -180 degrees. Similarly, **phase margin** is the amount of phase by which the curve of the Bode phase diagram is above -180 degrees at the frequency where the magnitude is equal to 0 dB.

Controller 1: PID

To begin the control design process, first a PID (proportional-integral-derivative) controller is developed and tested. A PID is a relatively simple controller with a several decade-long research background and solid theoretical foundations. The demanded control torque is calculated by first subtracting measured Euler angles ($\boldsymbol{\theta}_{sen}$) and angular velocities ($\boldsymbol{\omega}_{sen}$) from the desired reference signals ($\boldsymbol{\theta}_{ref}$ and $\boldsymbol{\omega}_{ref}$) to get error signals $\boldsymbol{\theta}_e(t)$ and $\boldsymbol{\omega}_e(t)$. Then, these error signals are multiplied by the individual components of the PID.

$$\boldsymbol{\tau}_c(t) = \mathbf{k}_P \cdot \boldsymbol{\theta}_e(t) + \mathbf{k}_I \int \boldsymbol{\theta}_e(t)dt + \mathbf{k}_D \cdot \boldsymbol{\omega}_e(t) \tag{3.27}$$

The **term P** is proportional to the current error of the system and is usually the biggest component of PID control in bringing the output closer to the reference. Raising the gain k_P decreases settling time at the cost of a larger overshoot and more violent residual vibrations. Setting it too high may cause the system to become unstable, setting it too low results in insufficient control authority and a decreased ability to reject disturbances.
The **term I** is proportional to the accumulated error in the outputs and is related to managing residual error. Increasing the gain k_I eliminates steady-state errors and decreases the convergence time of the output at the cost of overshooting the reference signal.
The **term D** is proportional to the rate of change in the error signal and is related to damping. It does not consider the current error of the system but rather seeks to flatten out the trajectory of the output. It reduces overshoot and vibrations at the cost of slowing down control response and introducing noise sensitivity. For this reason, either a filter is applied to the term to reduce the effect of sensor noise, or the D term is replaced with a P term on the angular velocity error. This thesis takes the latter approach.

Controller	k_P	k_I	k_D
Roll	500	400	2000
Pitch	500	400	2000
Yaw	1e-1	1e-4	100

Table 3.2: List of PID controller coefficients.

There are many available options for tuning the controller coefficients, this thesis employs a combination of automated software tools and manual tuning. First, the time-delayed linearized equations of rotational dynamics are decoupled into three separate models, one for each Euler angle. Then, the proprietary `PID Tuner` tool of `Simulink` is used on each SISO model to reach a stable system response. The resulting controllers are further tuned manually to improve stability on the linearized system and pointing performance on the nonlinear system. Roll and pitch are more strictly controlled as they have the largest impact on pointing precision, while Yaw is tuned for disturbance rejection as it is the main channel for the effects of structural dynamics. The final design is presented in table 3.2 [22].

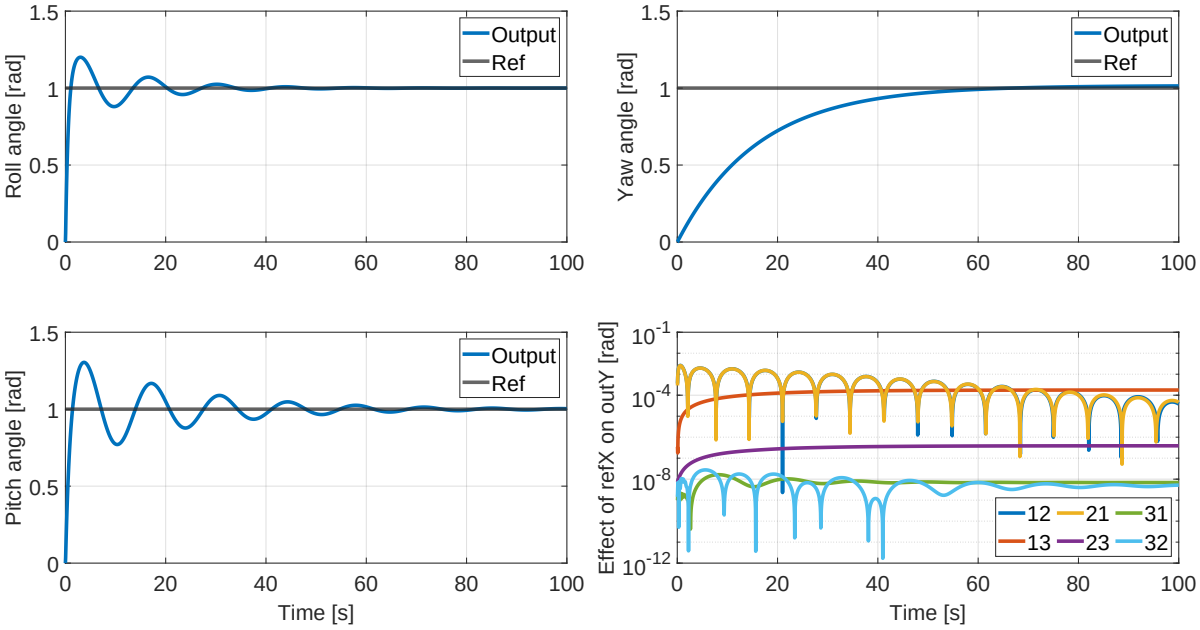

Figure 3.19: Step response of PID loop. Bottom right: output Y when a reference is applied to input X.

As shown in fig. 3.19, the closed-loop system has a promising step response with a roughly 60 sec settling time around all angles and about 25% overshoot in roll and pitch. Cross-coupling effects are negligible with sufficiently dampened vibrations at $10^{-8} - 10^{-3}$ rad magnitudes in all channels.

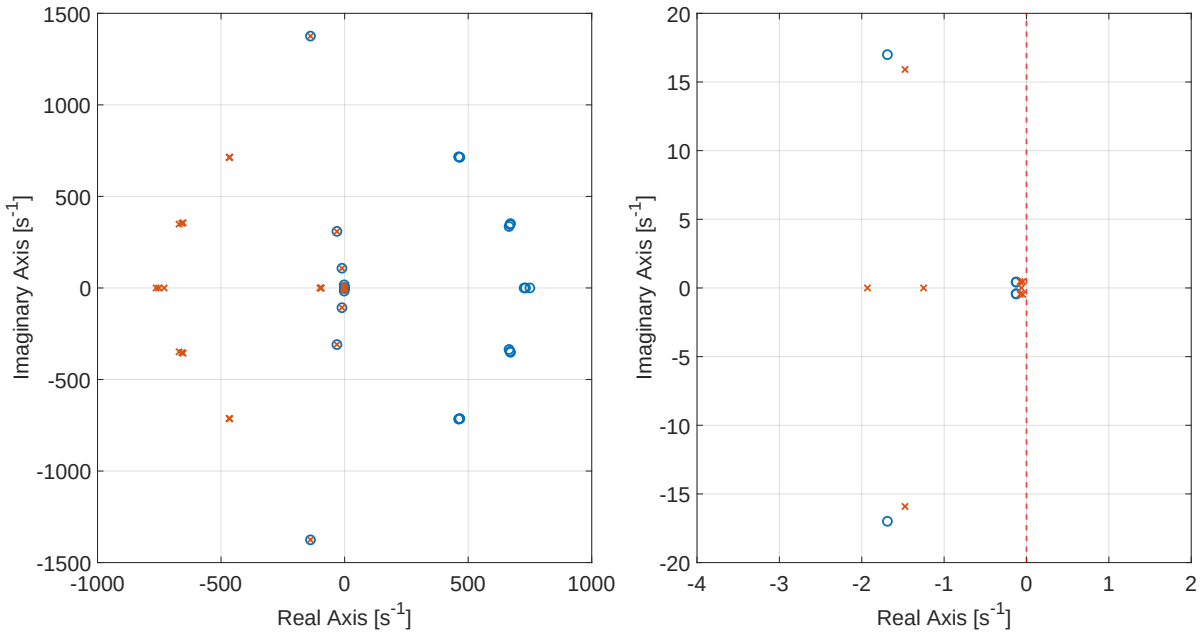

Figure 3.20: Closed-loop pole-zero map of designed PID controller. Right side: zoom on origin.

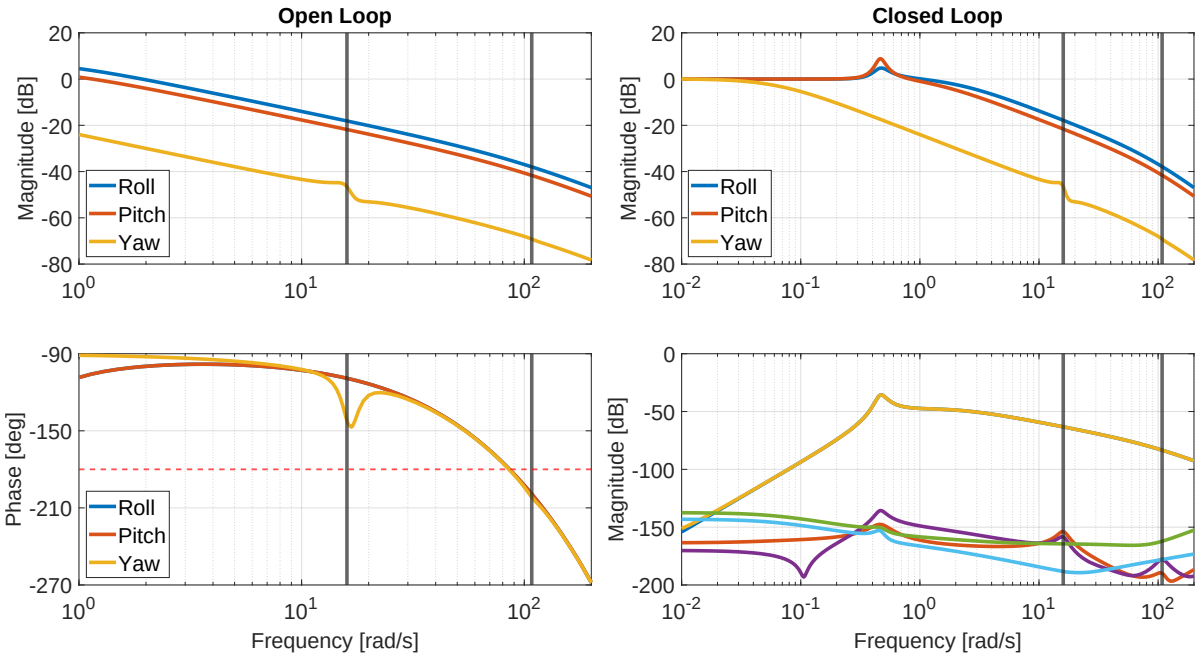

Figure 3.21: Bode plots of designed PID controller, first two flexible modes in grey. Right side: reference tracking and disturbance rejection.

On the resulting pole-zero map (fig. 3.20), several interesting features can be observed. First, note that all poles of the system are in the left half-plane, meaning the system is stable. The flexible modes of the structure are also clearly visible, placed along two radial lines leaving the origin of the plot. Additionally, the 5th order Padé approximant forms a circle of poles and zeros a great distance away from the origin.
The phase of all three angles in the open-loop bode plot drops off dramatically around the 60 rad/s mark, with a large notch also visible on the yaw dynamics around the location of the largest flexible mode. The gain and phase margins indicate stable system behaviour.
The closed-loop bode magnitude plot shows decent reference tracking (top plot) up to 2 rad/s in roll and pitch, and 0.05 rad/s in yaw. The roll and pitch response has a maximum gain of 5-10 dB around 0.5 rad/s. In terms of disturbance rejection (bottom plot), all responses are placed well below the nominal roll, pitch, and yaw curves. Two exceptions are the cross-coupling responses between roll and pitch, which reach a maximum gain of -40 dB at 0.5 rad/s, which is still much lower than the gains of the three other Euler angle dynamics.
It can be concluded that the controller does not attempt to control, and barely excites the flexible modes of the system, opting to let them dissipate on their own. However, it shows acceptable reference tracking characteristics and decent reference tracking, and can now be tested in a nonlinear environment.

Smooth tracking performance: PID

Throughout the control design process, linear and nonlinear testing forms an iterative process in which results from both simulation environments are taken into account for adjusting the parameters of designed controllers. Reaction wheel torques are also monitored in the nonlinear model, as they need to fit within the torque limits imposed by the manufacturer and perform without producing unreasonably high noise levels on the actuators (beyond microvibrations). Additionally, as reaction wheels are not suited for operation around their resting points, their starting speeds are randomized on each start, hence the small ”wiggles” of the motor torque at the beginning of the simulation (fig. 3.22).
There are two clearly demarcated stages in the scenario. First, the large-scale slew maneuver rotates the spacecraft away from nadir orientation, and into targeting mode. This is accompanied by a generous torque around the pitch axis, which is supplied by all wheels to fit below the limits. Then, the much gentler pointing phase begins as the smoothing algorithm concludes its work at 150 s. The precise pointing section lasts 450 s, with most of the control effort still going towards the pitch axis to follow the target as the spacecraft speeds past its location.

The tracking errors of each Euler angle are shown in fig. 3.23. Roll and pitch stay well below 2 arcsecs for most of the precise pointing segment, with an increase in pointing error around the time of closest approach (where relative velocity between the spacecraft and the target is highest). Yaw is kept at a low angle by its slow PID loop tuned for disturbance rejection, at roughly a sixth of a degree.

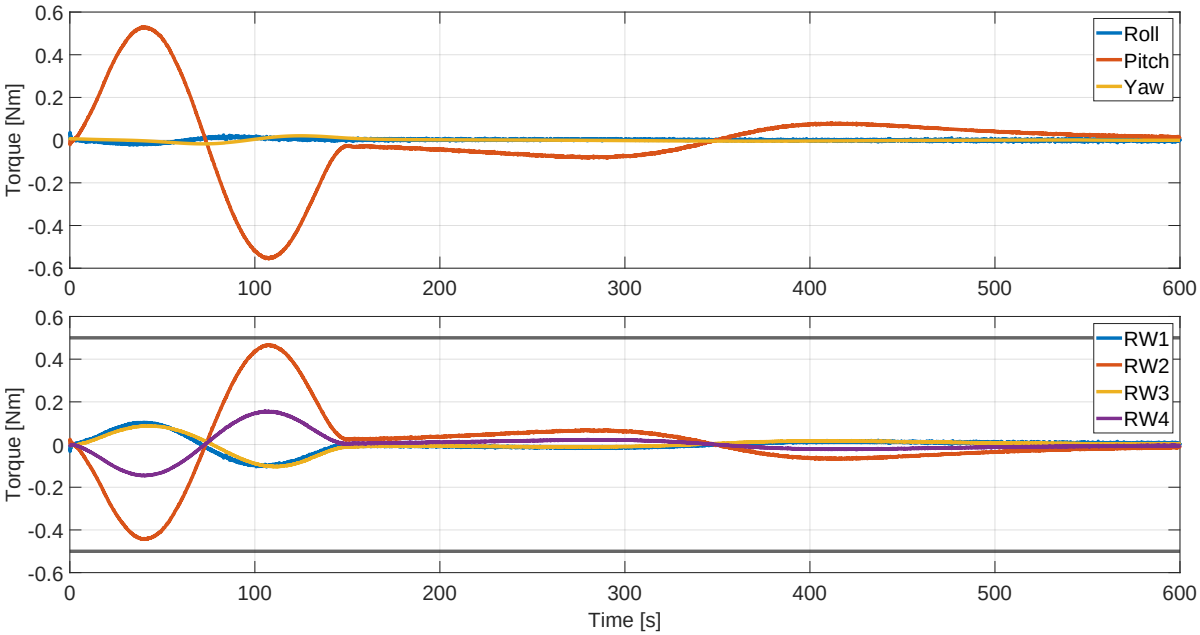

Figure 3.22: Reaction wheel outputs of designed PID controller, torque limits in grey.

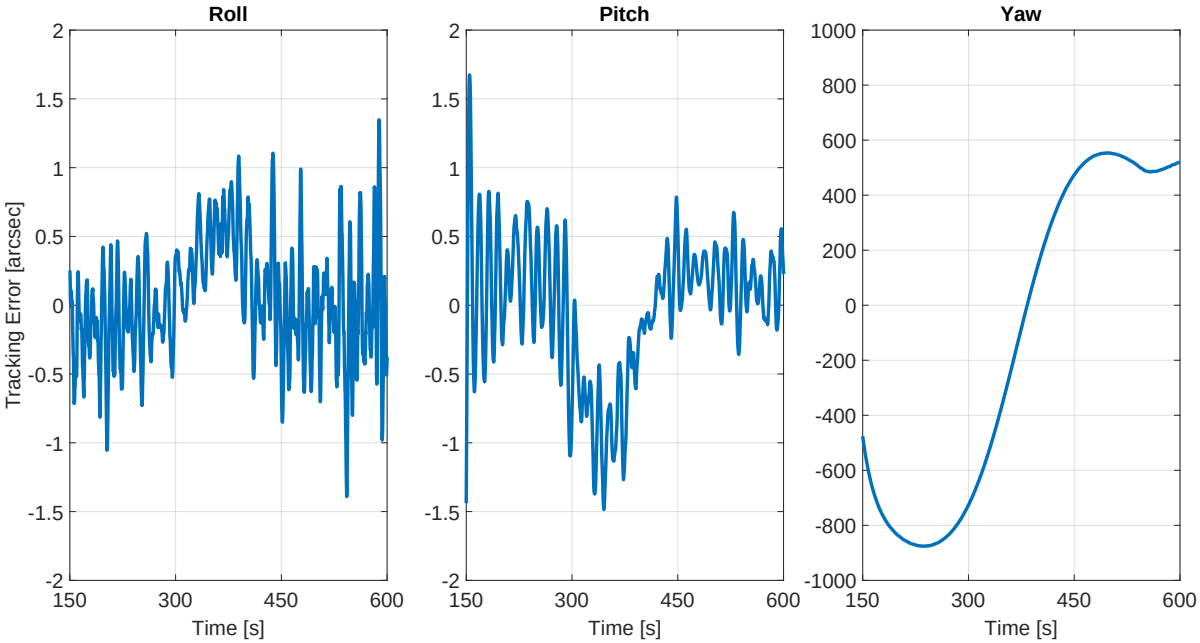

Figure 3.23: Euler angle tracking error of designed PID controller.

Controller 2: Robust H_∞-based

Generally speaking, in closed-loop control design there is a trade-off between performance and robustness. The closer a controller gets to zero in reducing errors in the loop, the more sensitive it becomes to disturbances, noise, delay, and uncertainty. In recent years, there has been an increase in robust control publications for flexible spacecraft across Europe, a technique that offers to find a balance in this trade-off and increase reliability for future missions [9][43].
In the field of robust control, one of the most important applied methods is H_∞-based loop shaping. For a given SISO transfer function G, the H_∞ norm is the worst-case magnitude of the system over the entire frequency domain and is defined as:

$$||G||_\infty = \sup_\omega |G(j\omega)| = \lim_{p\to\infty} \left(\int_{-\infty}^{\infty} |G(j\omega)|^p d\omega \right)^{\frac{1}{p}} \tag{3.28}$$

Now, consider a system **P** with an exogenous input **w** (e.g. disturbances and commands) and **u** control variables. There are also measured variables **v** and **z** error signals to be minimized. The controller **K** is used to transform the measurement signals into control signals in a feedback loop (fig. 3.24).

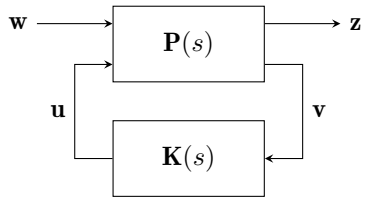

Figure 3.24: General configuration for H_∞-based control.

This system can be described with the following model:

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{v} \end{bmatrix} = \mathbf{P}(s) \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{11}(s) & \mathbf{P}_{12}(s) \\ \mathbf{P}_{21}(s) & \mathbf{P}_{22}(s) \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} \tag{3.29}$$

$$\mathbf{u} = \mathbf{K}(s)\mathbf{v} \tag{3.30}$$

From these, the closed-loop MIMO transfer function from the disturbances to error signals can be determined by linear fractional transformation:

$$\mathbf{z} = \mathbf{F}_l(\mathbf{P}, \mathbf{K})\mathbf{w} \tag{3.31}$$

where

$$\mathbf{F}_l(\mathbf{P}, \mathbf{K}) = \mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(\mathbf{I} - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21} \tag{3.32}$$

The main design goal is the minimization of the H_∞ norm of this system in order to attenuate the effect of disturbances on the system's performance. This also provides a measure of system robustness with γ, which is the H_∞ norm of the closed-loop system. Note that the definition of the norm has been adjusted to account for the MIMO nature of the system, with $\bar{\sigma}()$ representing the largest singular value of the matrix:

$$\gamma = ||\mathbf{F}_l(\mathbf{P}, \mathbf{K})||_\infty = \sup_\omega \bar{\sigma}(\mathbf{F}_l(\mathbf{P}, \mathbf{K})(j\omega)) \tag{3.33}$$

In practice, there is no need for an optimal controller to the H_∞-problem, and it is usually computationally and theoretically simpler to design a sub-optimal one. Because of this, the value of γ is iteratively reduced until a sufficient solution is found [6].
In the design of the controller, various weights are added to the input and output signals of the plant model. The result is called an augmented plant and is the major component of the loop-shaping process. These weights are used to prescribe desired characteristics in noise suppression, disturbance rejection, control effort, and control performance. Any transfer function can be used as a filter, even zero gain (for off-diagonal elements) or constant gain (to start with). Then, the resulting plant undergoes robust stabilization, which has no effect on gains in the lowest and highest frequencies but maximizes the system's stability margins by shifting the unity-gain crossover of the augmented plant. [6]

In the case of the flexible spacecraft model (fig. 3.25), there are four filters placed on the signals. The exogenous input signal is split between two components: disturbances and sensor noises, to be weighed by $\mathbf{W}_d$ and $\mathbf{W}_n$, respectively. Instead of the error output $\mathbf{z}$, a performance output $\mathbf{e}$ is used, split between system outputs (to be used as performance indicators) and control torques, to be weighed by $\mathbf{W}_e$ and $\mathbf{W}_u$. These are all diagonal matrices to speed up the design process, with their shape, cutoff frequency, and gain determined manually in an iterative process that involved the testing of each iteration in the nonlinear environment to reach the performance of the PID.
Initially, these transfer functions are all defined as constant gains which is sufficient for Matlab's algorithms to stabilize the system. Then, individually, each transfer function is exchanged with either a low-pass or a high-pass filter with a cutoff frequency of 1 rad/s, depending on the expected physical characteristics of the given signal (see: next page). Once an ideal combination is found, the cutoff frequencies and gains are shifted by small steps until a desirable solution is found for pointing performance. As a last step, additional integrators or derivators might be added, for example on the noise weights.

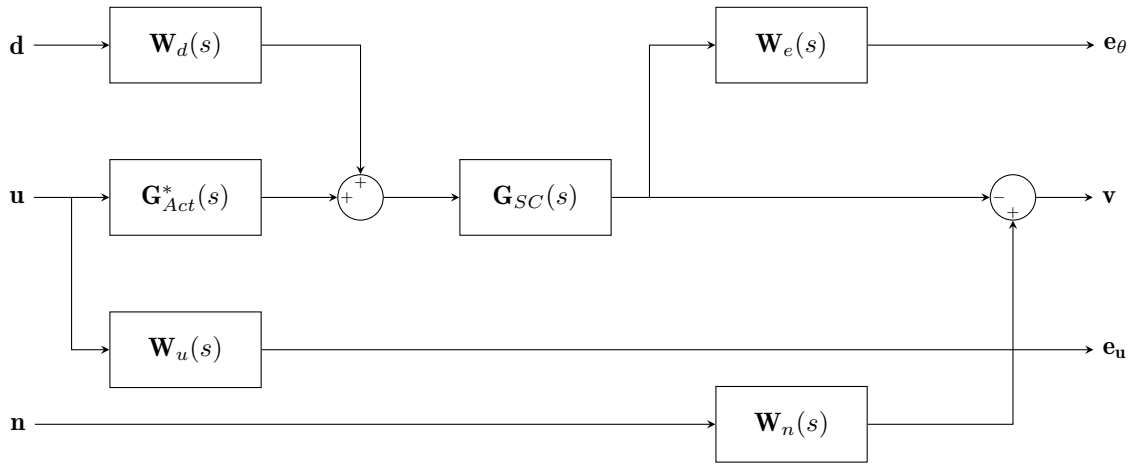

Figure 3.25: Augmented plant for robust control design.

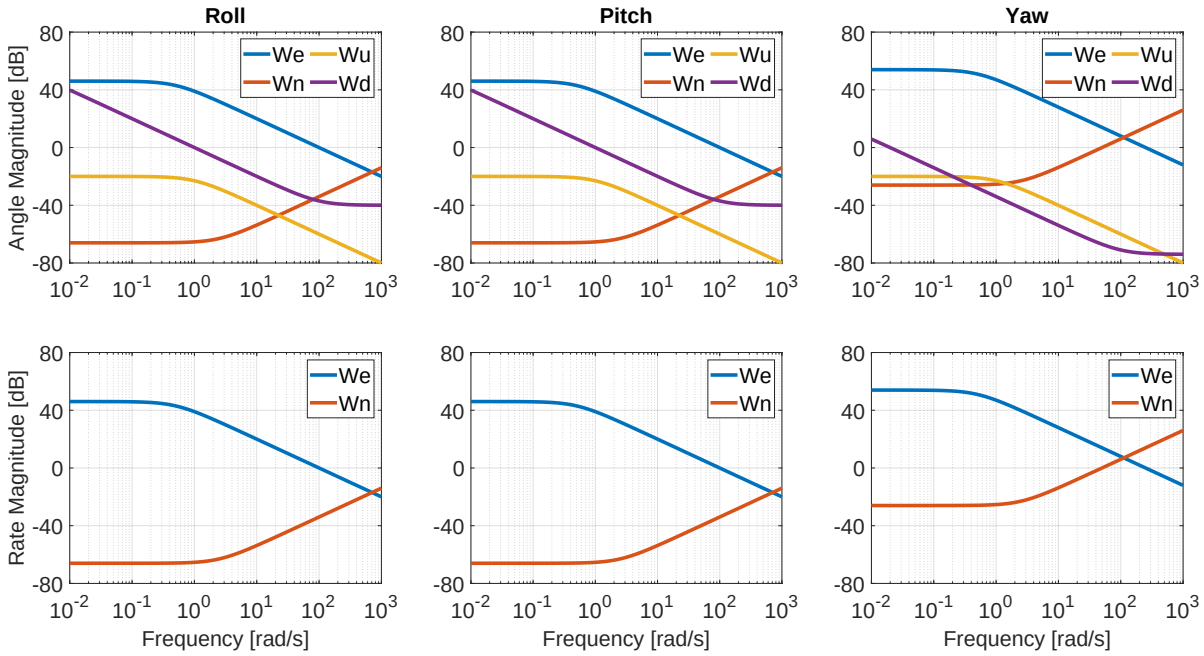

Figure 3.26: Weights used in the design of H_∞-based controller.

While the final weights themselves (fig. 3.26) are slightly arbitrary, they do hold some relevance to the real-world frequency response of the system. For example, the input weights bar the controller from demanding high-frequency torques from the reaction wheel assembly:

$$\mathbf{W}_u(s) = diag\left(\frac{0.1}{s+1}, \frac{0.1}{s+1}, \frac{0.1}{s+1}\right) \tag{3.34}$$

The disturbance filter defines high-frequency torques as the main detrimental effect to be rejected:

$$\mathbf{W}_d(s) = diag\left(\frac{0.01s+1}{s+0.002}, \frac{0.01s+1}{s+0.002}, \frac{0.0002s+0.02}{s+0.002}\right) \tag{3.35}$$

The performance filters give increased weight to the elimination of steady-state errors with the increased gain on lower frequencies:

$$\mathbf{W}_{e,\theta}(s) = \mathbf{W}_{e,\omega}(s) = diag\left(\frac{100}{s+0.5}, \frac{100}{s+0.5}, \frac{250}{s+0.5}\right) \tag{3.36}$$

Finally, the noise filter defines the range in which sensor noise sources are active:

$$\mathbf{W}_{n,\theta}(s) = \mathbf{W}_{n,\omega}(s) = diag\left(\frac{10s+25}{s+50000}, \frac{10s+25}{s+50000}, \frac{1000s+2500}{s+50000}\right) \tag{3.37}$$

The closed-loop system that `hinfsyn` produced has a γ of 50, as well as a controller with 51 states. To reduce numerical complexity and improve computational performance, this must be drastically cut back. It was found that control performance is not decreased if 32 states with the lowest Hankel singular values are eliminated from the controller block. The final, 19-state H_∞-based controller can then be tested in a closed-loop to evaluate time-domain performance.

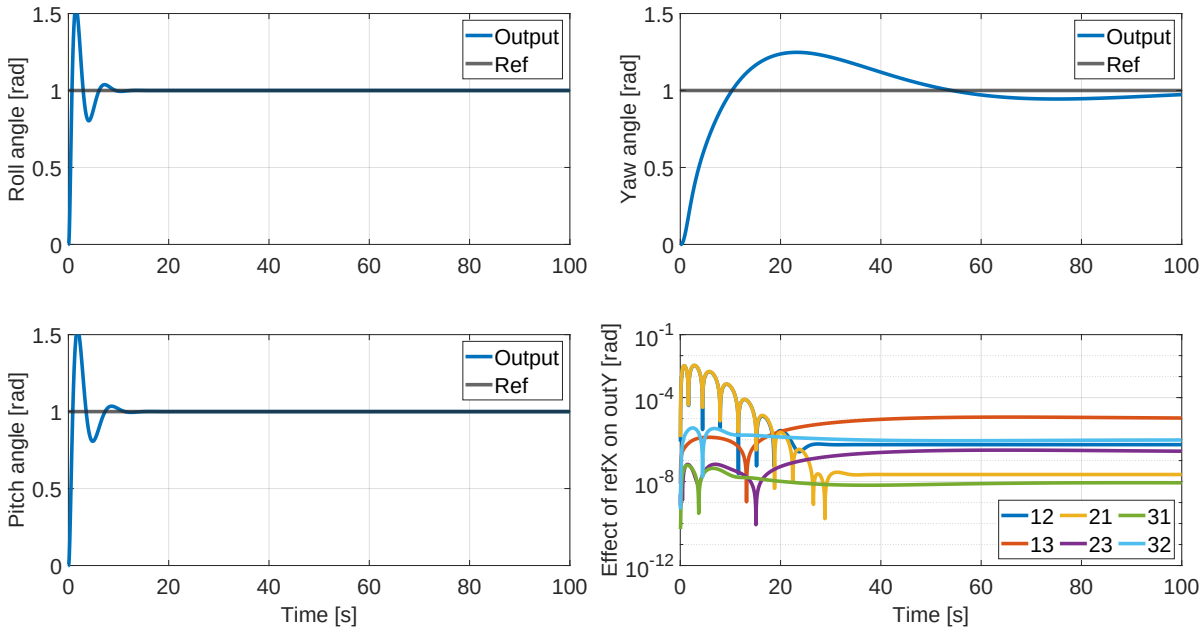

Figure 3.27: Step response of H_∞-based loop. Bottom right: output Y when a reference is applied to input X.

Regarding the step response of the closed-loop linear system (fig. 3.27), performance is markedly improved compared to the PID. Settling time is significantly reduced to 15s around the roll and pitch axes, and slightly increased to 100s around the yaw axis. Overshoot is doubled to 50% for roll and pitch, and there is a 25% overshoot added to the yaw axis where there was not any before. In terms of cross-coupling effects, the oscillations are dampened much faster, and they take a smaller magnitude at $10^{-8} - 10^{-5}$ rad on all channels.

On the pole-zero map (fig. 3.28), all closed-loop poles are on the right half-plane, meaning the system is stable. There are several new zeroes added on the left half-plane, largely to account for time delay (which was not possible in the PID design). There are also a number of new poles and zeros slightly right of the origin as well. These account for the majority of the difference in control action between the H_∞-based controller and the PID. Stability can also be verified on the Bode plots (fig. 3.29), which present greater phase and gain margins, as well as better disturbance rejection than the PID loop. In reference tracking, the roll and pitch dynamics are similar to the PID, but the frequency range of yaw tracking is slightly increased.

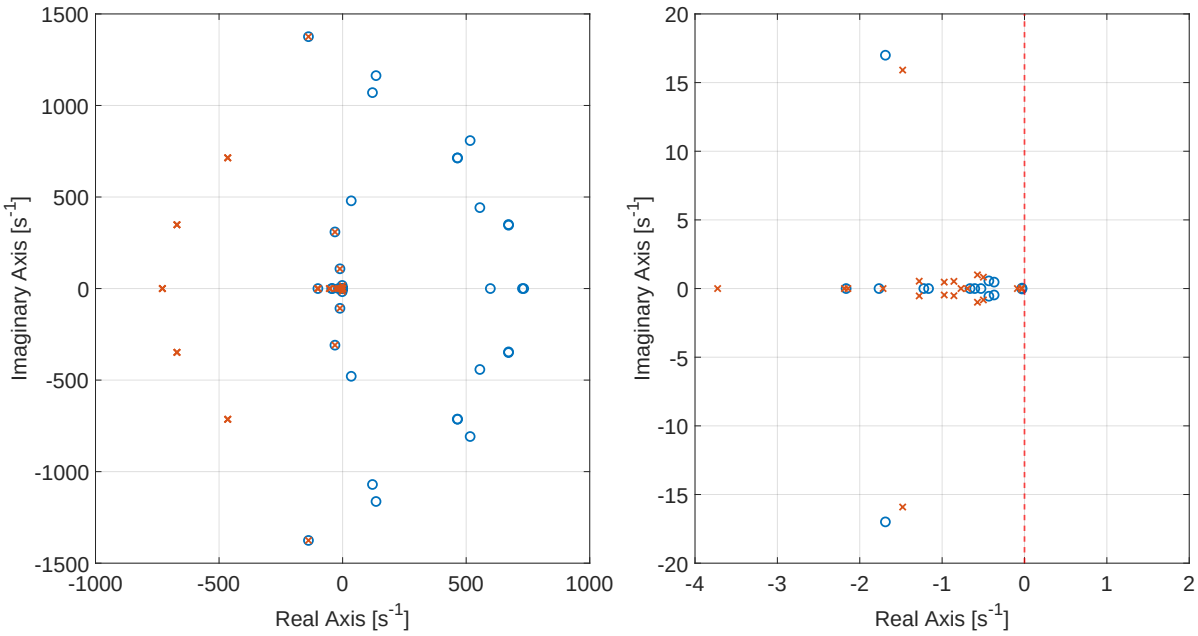

Figure 3.28: Closed-loop pole-zero map of designed H_∞-based controller. Right side: zoom on origin.

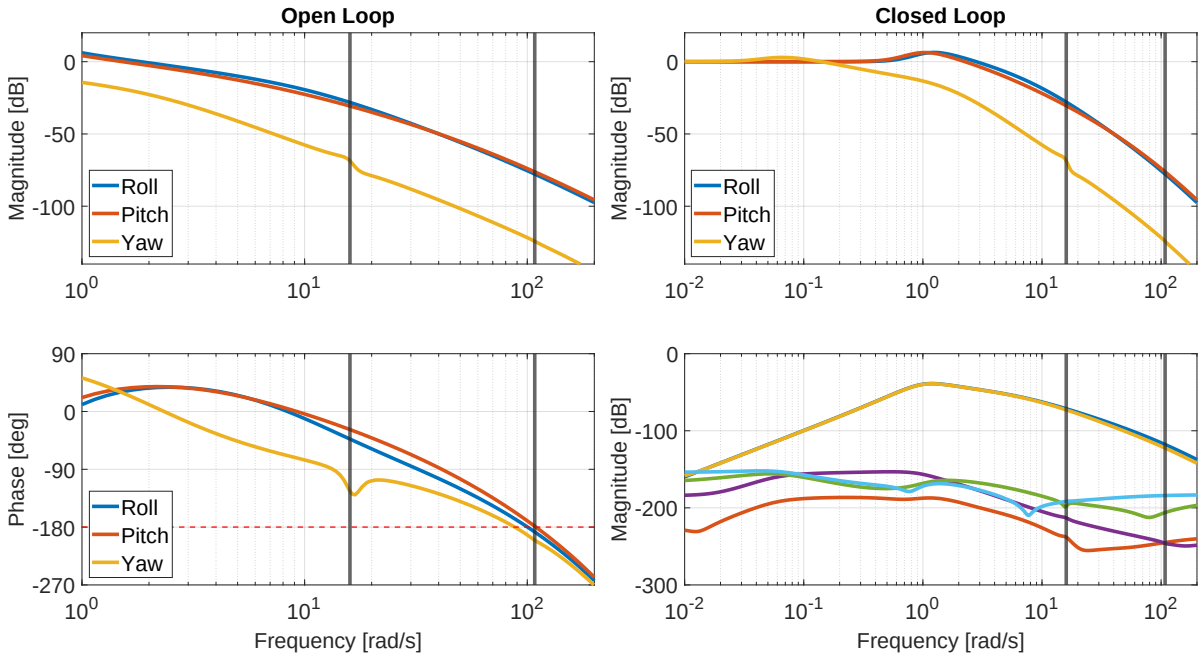

Figure 3.29: Bode plots of designed H_∞-based controller. Flexible modes in grey.

Smooth tracking performance: H_∞

Similarly to the PID loop, the tracking errors of each Euler angle are shown in figure 3.31. Roll and pitch stay well below 2 arcsecs for most of the precise pointing segment, with an increase in pointing error slightly before and after the time of the closest approach in the pitch axis. At the moment of closest approach, both roll and pitch errors approach zero. Yaw is kept at a much lower angle than what was achieved with the PID loop, staying within a few dozen arcsecs of zero at all times except for the first minute. Overall, pointing stability is greatly improved in H_∞ control, with the cost of slightly noisier reaction wheels (fig. 3.30 - increased noise levels below the threshold of printing resolution).

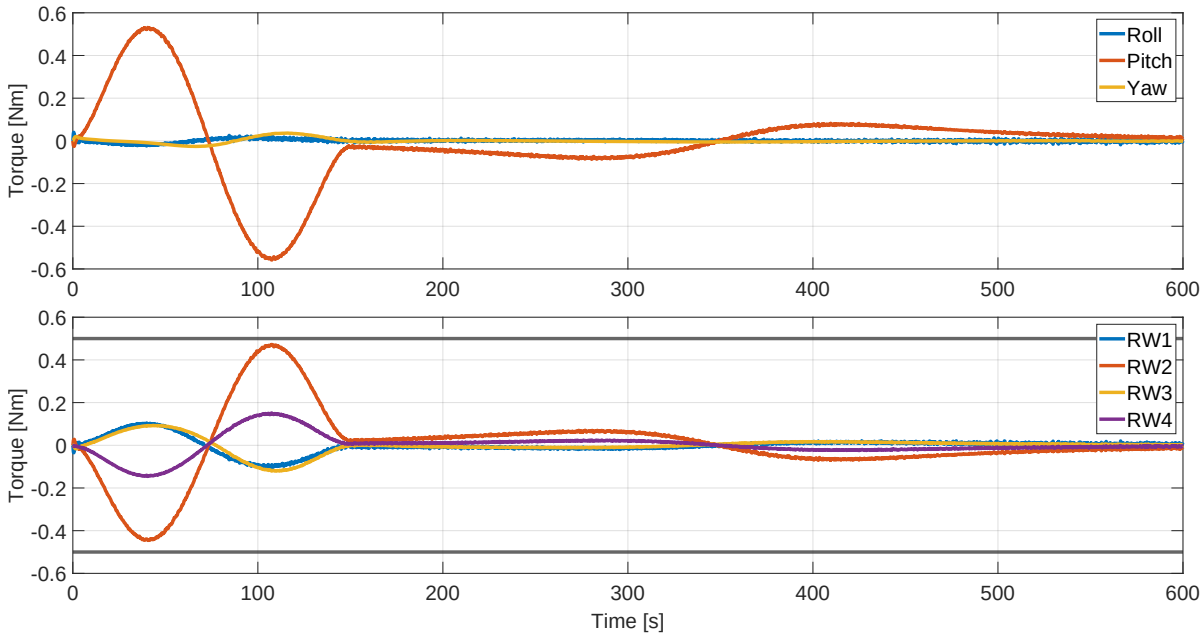

Figure 3.30: Reaction wheel outputs of designed H_∞-based controller.

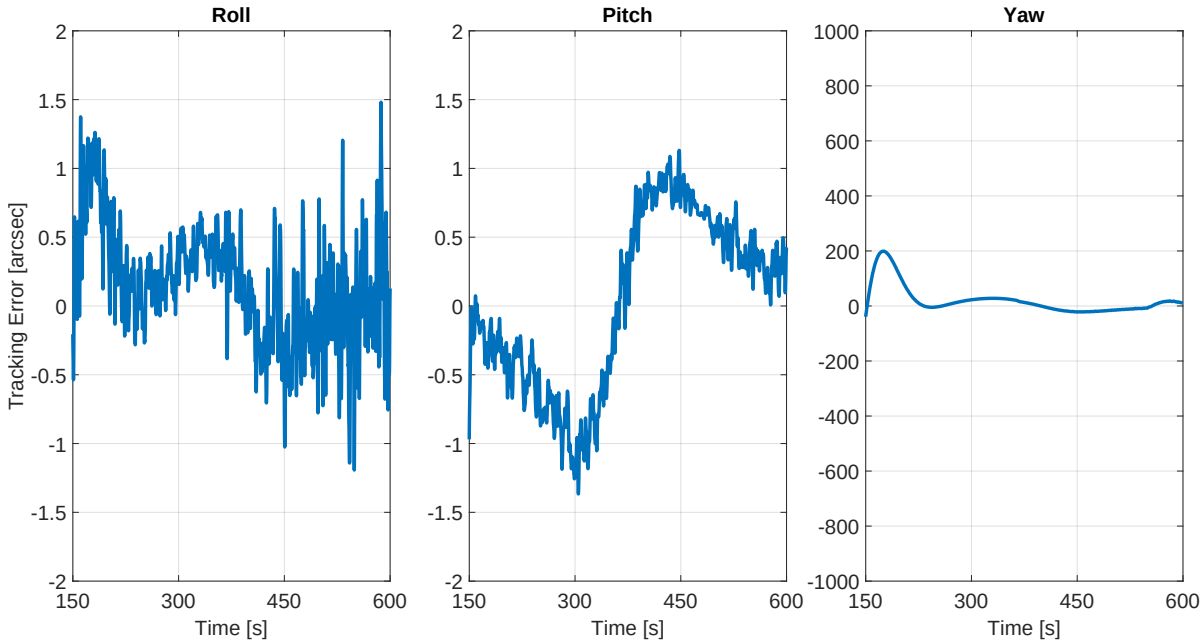

Figure 3.31: Euler angle tracking error of designed H_∞-based controller.

Controller 3: MPC

A significant drawback of the PID and H_∞-based controllers is that they only account for the present state of the system. Model Predictive Control accounts for the future states of the system within a given horizon while optimizing the system dynamics in real-time. Another feature of the MPC is constraint handling, which is the main way of preserving the stability of the system due to the lack of mathematical tools for the analysis of the closed-loop.
While predictive control has enormous utility in numerous engineering fields, MPCs have seen relatively few applications in the space sector despite their potential according to a thorough literature review carried out in 2017 by Eren et al According to their assessment, a model predictive control system has four constituents [11]:

1. The model, which is a discrete-time system. It can be described as either LTI (linear time-invariant), LTV (linear time-varying), LPV (linear parameter-varying), piecewise-affine, or non-linear - based on available computational capacity
2. The available information, which means that at any time, the system states are known when the control value is determined.
3. Stage constraints, which means that the system variables (states and controls) are subject to hard constraints. This can represent either actuator limitations or algorithmic concerns about feasibility.
4. A stage cost function, which associates the quality of performance with the pair of the state and control variables, and penalizes deviation from the desired values. A terminal cost function is also defined to help ensure suitable stabilizing properties.

The basic idea behind model predictive control is that this discrete-time system is used to propagate the currently measured states some time steps into the future to optimize the control signal on each time step. There are two horizons: first, the prediction horizon, which defines the number of predicted samples for the future state vector; second, the control horizon, which determines how many samples are predicted and optimized for the control input.

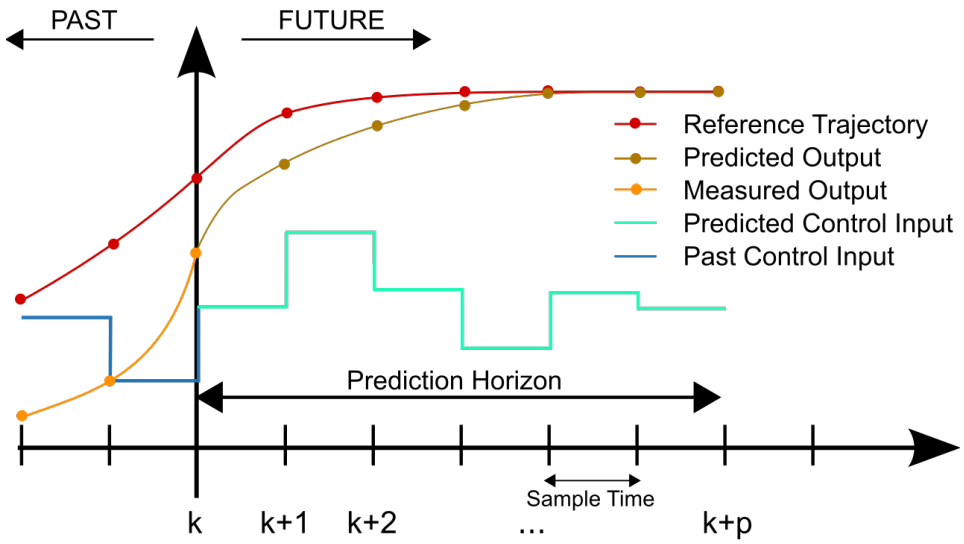

Figure 3.32: Basic operation of an MPC control loop. (source: Wikimedia Commons)

On every single time step, this control input is re-optimized and the first value is taken as the control command. As such, optimization is the core component of MPC, and its performance greatly depends on the available plant model. Constraints are also a major advantage of the technique, as an ordinary controller would often make unreasonable demands of a physical system or drive it far away from the operating point, hence the need for smoothing algorithms on the reference signal generator. MPC on the other hand is able to place bounds on its own control demands and handle possible overshoots of the system states [44].

Besides the handful of publications on the control of flexible spacecraft, there is very little research done on the impact of precise pointing control [10][18]. As such, this section is going to be slightly experimental and aims to match the performance of the previously designed controllers as closely as possible.
The design of the controller began with the definition of the plant, which is the exact same as the one used in both the design of the PID and the H_∞-based controllers. Another important difference between the MPC and the two other controllers is that the MPC block does not require an error signal for calculating the control torque: instead, it takes two signals, one for the ideal predicted states, and another for sensor data (fig. 3.33). Because of this, the operation of the trajectory generator is adjusted so it produces reference signals for the entire prediction horizon, instead of just the current time step. Then, the time step of the controller is set at 0.1s, due to the fact that an MPC has an enormous computational demand compared to PID or H_∞-based control, which are running at 100 Hz.

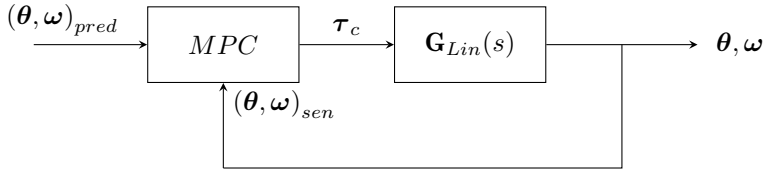

Figure 3.33: Block diagram of closed-loop model predictive controller.

With this time step defined, an iterative process began in which various prediction and control horizons were tested in the nonlinear simulation to find a combination that yields decent performance in both the large-scale slew maneuver as well as the precise pointing segment of the reference signal. It was found that the most promising candidates for horizon lengths are 15 steps for the prediction process, and 3 steps for the control signal. For the prediction horizon, going beyond this number gradually decreases pointing performance at high reference signal velocities because of inaccuracies in the prediction process. For the control horizon, any additional step drastically increased the length of optimization, and it was determined that going much beyond these two numbers would be unacceptable for the performance of space embedded hardware.
To improve numerical accuracy, scaling factors are used in the design process. They convert the relationship between outputs and control torques to dimensionless form and make it easier to specify tuning weight magnitudes. If a model is scaled well, the maximum absolute scaled response on the main input-output channels should be around 1. On Simulink's suggestion, the scaling factors were set as $5 \cdot 10^{-4}$ for all channels. To make use of the constraints provided by predictive control, the minimum and maximum values for the control signal were set as ± 0.6 Nm to fit the torque limits of the reaction wheel assembly. Additionally, weights were added to the outputs (OV), control signals (MV), and the rate of control signals (dMV) to slow down yaw dynamics and not disturb the flexible modes of the system:

$$\mathbf{W}_{OV} = [1 \quad 1 \quad 0.1 \quad 1 \quad 1 \quad 10] \tag{3.38}$$

$$\mathbf{W}_{MV} = [0 \quad 0 \quad 0] \qquad \mathbf{W}_{dMV} = [1 \quad 1 \quad 0.1] \tag{3.39}$$

Smooth tracking performance: MPC

Due to the lack of linear analysis tools for model predictive control loops, only the controller's nonlinear performance is analyzed in this section. There are two notable differences in the reaction wheel torques (fig. 3.34). First, there is a large transient at the start of the simulation which is mainly caused by the MPC being unable to optimize the system fast enough. However, this undesirable error gives way to a much smoother torque curve than what was determined in PID and H_∞-based control.
In terms of angle tracking error (fig. 3.35), the MPC also stays below ± 2 arcsec for the entire run, although there is a noticeable constant error in the pitch dynamics. Pointing stability is improved compared to both other controllers, and the yaw dynamics do not show any transient effects anymore, staying within a few dozen arcsecs of zero error.

This residual error might be caused by the internal state estimation functionality of Simulink's MPC block that uses the linearized system equations as a basis. Unfortunately, the exact settings of this block can be very hard to access and tune, and completely turning it off results in drastically reduced performance. As a result, the estimation of pitch dynamics is slightly delayed compared to the real system. To solve this problem, building a complete state estimator with Kalman filtering will be necessary, which will be carried out at a future stage of the project.

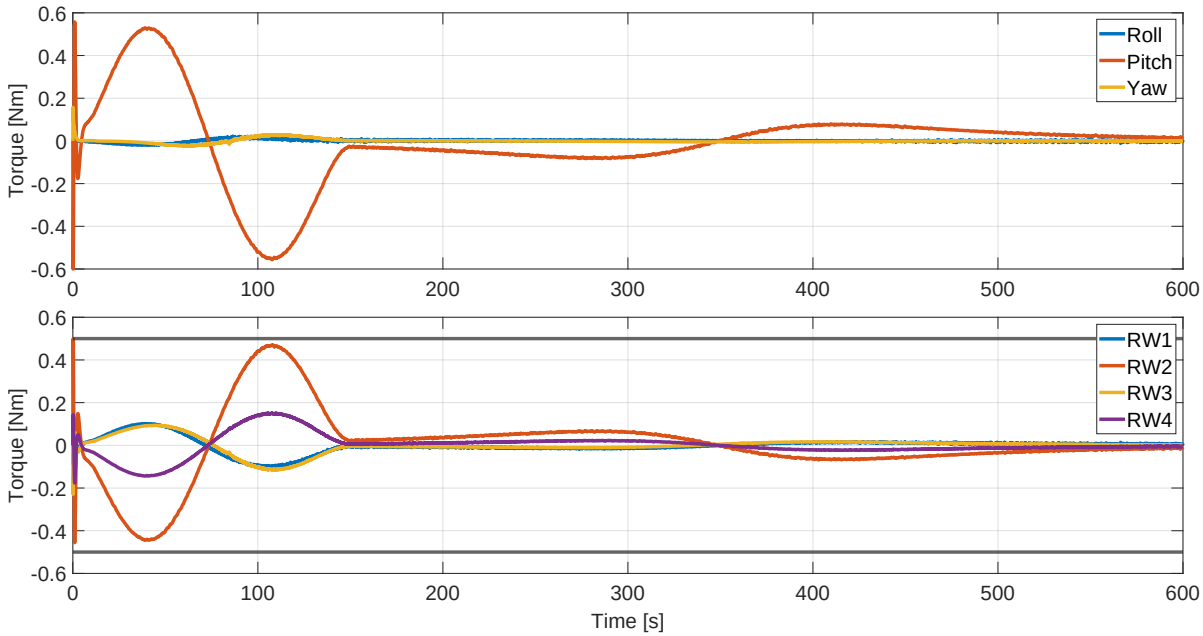

Figure 3.34: Reaction wheel outputs of designed model predictive controller.

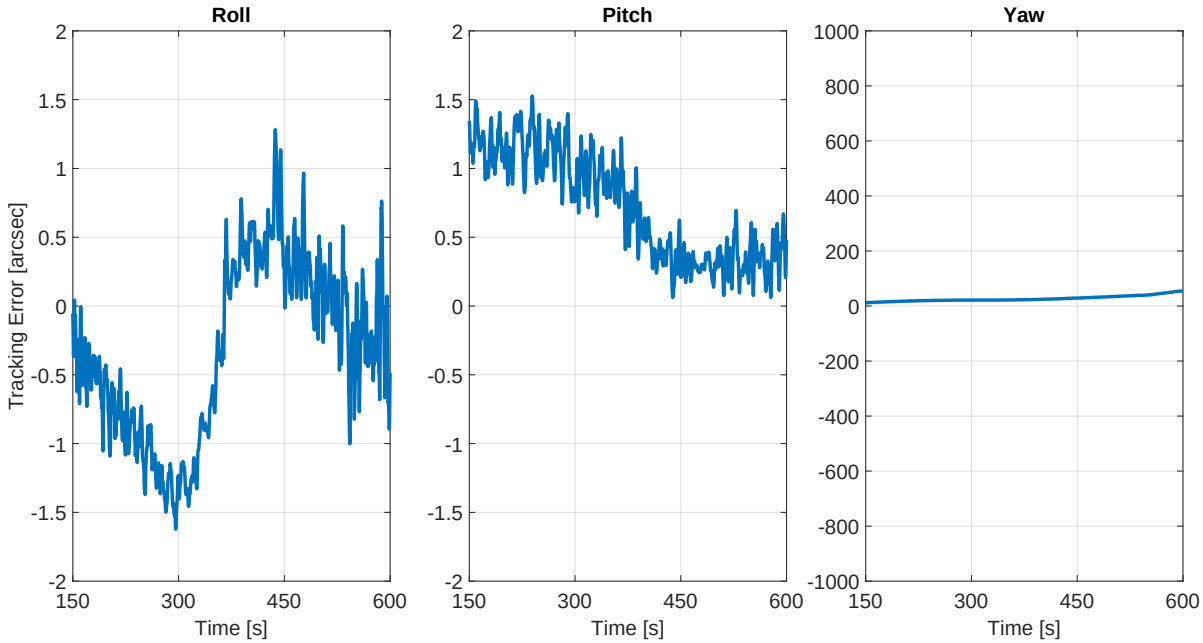

Figure 3.35: Euler angle tracking error of designed model predictive controller.

4

Pointing Performance

The main goal of the thesis is to raise the pointing performance of flexible spacecraft, which has two components: accuracy and stability. This chapter presents a rigorous performance assessment of the controllers designed in previous sections of the thesis and presents a cascade control loop design with an active-passive isolator platform to tackle the microvibration environment of the spacecraft, counteract residual errors from the outer control loop, and improve overall pointing stability. The scientific basis of this process was the research done throughout the literature study, which answered the following questions:

- How are flexible spacecraft controlled with high pointing precision?
  - What kind of control structure is needed for the pointing control of the satellite?
  - What are the state-of-the-art algorithms for pointing spacecraft at a ground target?
- How is the coupling between the spacecraft and the payload modelled?
  - What is the state of the art for vibration isolation of precision payloads?
  - What are the advantages of passive, active, and hybrid isolators?
  - What are the characteristics of a hexapod strut construction?

The framework of pointing performance indicators makes it possible to numerically compare the operations of different control loops, and the flexibility of the attitude generator block provides a way to push both the single-loop and cascade-loop systems to the edge of their capabilities. These developments are crucial in ADCS evaluation, going further beyond the more heuristic analysis of simple trajectories presented in the last chapter.

4.1. Pointing performance indicators

The performance of the payload can be assessed in terms of the imaging characteristics of the optical system. This can be affected by three main disturbances. First, displacement is the average pointing offset over the exposure interval, and it contributes to the focus of the image shifting off-center. Second, smear is a linear motion related to smear rate, the average rate of change in pointing error. It contributes to the smearing of the image as the rolling shutter fails to line up the samples of the camera. Third, jitter is the noise component of pointing error and causes a blurring of the image. These three components all add up to the absolute pointing error of the control system [45].
In terms of pointing performance requirements, the European Cooperation for Space Standardization (ECSS) categorizes the causes of the effects into pointing performance indices. Since these indices were identified as playing an important role in the future of cosmic vision and Earth observation missions, a great deal of effort went into the precise definition and efficient usage of them. In essence, there are three groups of errors: absolute, windowed-time, and stability-time. All three can be evaluated in the statistical analysis of time-domain simulation data, and some frequency-domain behaviours can also be described with the use of Image Motion Modulation Transfer Functions (MTF) and FFT. However, there are no straightforward ways to directly include the indices in the control design process [46].

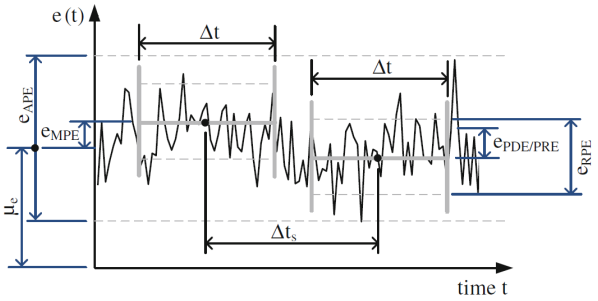

Figure 4.1: Pointing performance error indices. [46]

The ECSS defines five pointing performance error indices (fig. 4.1). Absolute performance error (APE) is the first and simplest one, and it is generally used in the control design of all pointing systems. It is the absolute difference of the current line-of-sight vector from the desired one in terms of pointing angles.
For the definition of the two windowed-time errors, a time-averaged error signal is defined:

$$\bar{e}(t, \Delta t) = \langle e(t) \rangle_{\Delta t} = \frac{1}{\Delta t} \int_{t-\Delta t/2}^{t+\Delta t/2} e(t) dt \tag{4.1}$$

Where $e(t)$ is the time-dependent error signal and Δt is the window-time. For the project, it was found that 10 sec produces a sufficiently smooth windowed-time error signal. Using this metric, the mean performance error (MPE) and relative performance error (RPE) can be defined, with MPE being equivalent to the windowed-time error signal, and RPE arising as the difference between APE and MPE. In terms of stability-time errors, the performance drift error (PDE) and performance reproducibility errors (PRE) can be defined with Δt_s as the stability-time. Despite them being mathematically identical (see table 4.1), a PDE requirement specifies the MPE difference among two points in time, but within one observation period. The PRE specifies an MPE difference between two points in time, which must be within different observation periods.

Index	Name	Metric	Formula
APE	Absolute Performance Error	Absolute	$e(t)$
MPE	Mean Performance Error	Windowed mean	$\bar{e}(t, \Delta t)$
RPE	Relative Performance Error	Windowed variance	$e(t) - \bar{e}(t, \Delta t)$
PDE	Performance Drift Error	Windowed mean stability	$\bar{e}(t, \Delta t) - \bar{e}(t + \Delta t_s, \Delta t)$
PRE	Performance Reproducibility Error	Windowed mean stability	$\bar{e}(t, \Delta t) - \bar{e}(t + \Delta t_s, \Delta t)$

Table 4.1: Description of pointing performance error indices.

In their 2011 paper, Ott et al describe a frequency-domain framework which can be used to approximate these error indices with specific filters placed on the APE channel of a pointing system. This can then be used in solving a multi-objective H_2/H_∞ control problem and designing a robust controller that optimizes the indices to the desired degree [46].
In this thesis, only the APE is used in the design of the outer loop controllers, with the rest of the parameters being only used in the analysis of time-domain error signals. This is done for two reasons: first, the goal of this research is not merely to design an acceptable controller for a given mission, but to develop a suitable framework for the integration of control design, nonlinear simulation, and V&V methods. Second, each additional index to the APE also requires weights in the augmented plant which rapidly increases the complexity of the robust control design process, which was already substantial for the project. It was decided that the indices should only be part of the project as a future goal, once all higher priority tasks were finished. Third, currently there is no way to include these parameters in the design of linear and predictive controllers, making the concurrent design process biased towards robust solutions, which would be undesirable.

4.2. Outer loop control performance

The simulation parameters were all specified in previous sections of the thesis, but just to summarize: all simulations are carried out using an ode3 (Bogacki-Shampine) solver at a fixed step time of 1 ms, with a simulation length of 600 s. For testing the control loop, the satellite was initialized with two-line orbital elements measured during the morning pass and then pointed at an off-track city nearby (fig. 4.2). In this case, the target is the peak of the TU Delft Library, at latitude 52.00265 deg and longitude 4.3756 deg. This ensured a realistic test scenario with a large roll angle.

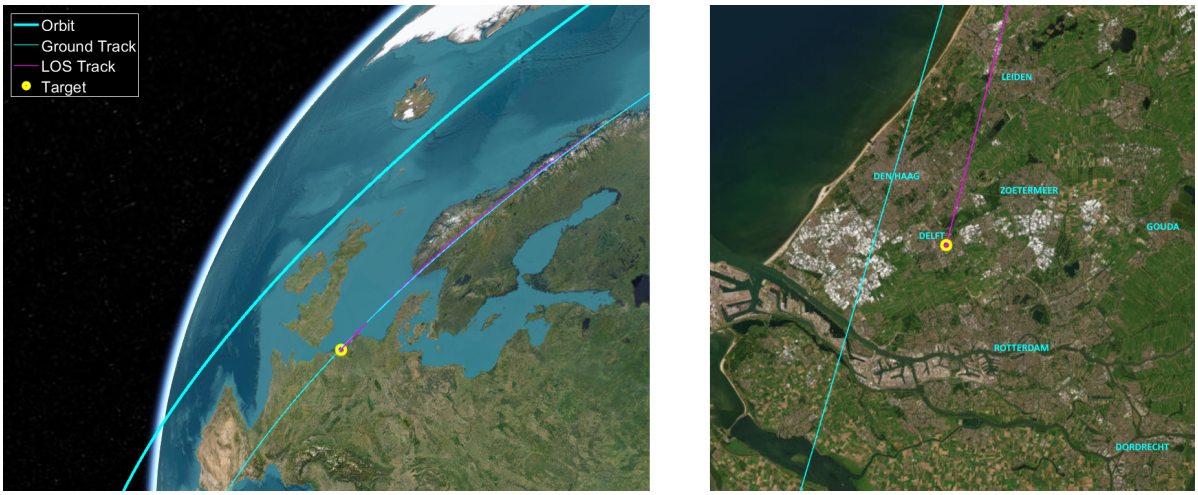

Figure 4.2: Smooth single-point tracking scenario layout.

After closely following the smooth portion of the trajectory and staying within the maximum torque range of the reaction wheels, the designed controllers enter the precise pointing stage where the APE requirements are below 2 arcsecs (fig. 4.3). In case of wheel failure, the smooth portion of the trajectory can be extended to further reduce the required torque. After intersecting the line-of-sight vector with the WGS84 reference ellipsoid, the ground errors are calculated and evaluated for all three control loops. The total length of the ground pass is 450 s which can be further extended by relaxing the performance requirements or optimizing the timing of the overpass.

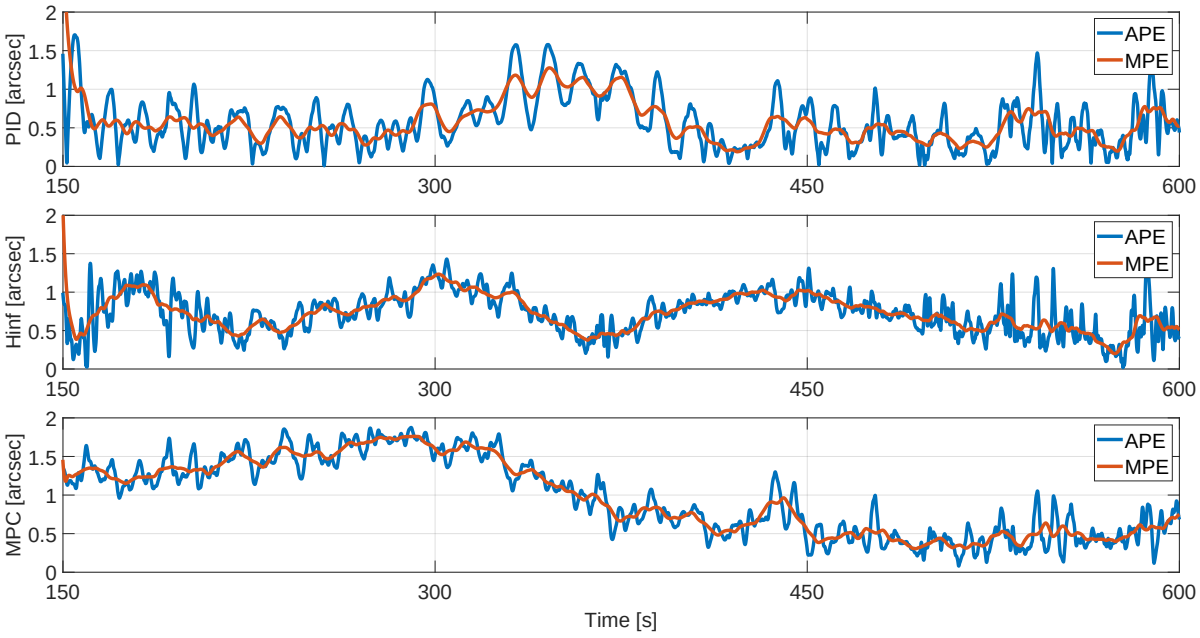

Figure 4.3: Pointing performance comparison for smooth single-point tracking.

For evaluating pointing performance, an RMS-like (root-mean-square) error signal is defined based on the norm of the roll and pitch errors, since yaw is relatively unimportant in precise pointing.

$$e(t) = \sqrt{\phi_e^2(t) + \theta_e^2(t)} \tag{4.2}$$

The first thing that's immediately visible is a slight delay in the MPE (fig. 4.4) of the linear and robust controllers that predictive control does not suffer from due to the length of the prediction horizon. The PID is performing very well in the first and last segments and only drops below 1 arcsec precision around the time of the ground target pass due to the large relative velocity between the target and the spacecraft. The H_∞-based robust controller is oscillating between 0.5 and 1 arcsec MPE, frequently meeting the accuracy of the PID. MPC starts off worst, barely meeting the main requirement (<2 arcsecs) at first, but then meets the other two controllers halfway through the observation time. This is likely caused by an error in the internal state estimator of the MPC, as mentioned in the design process.

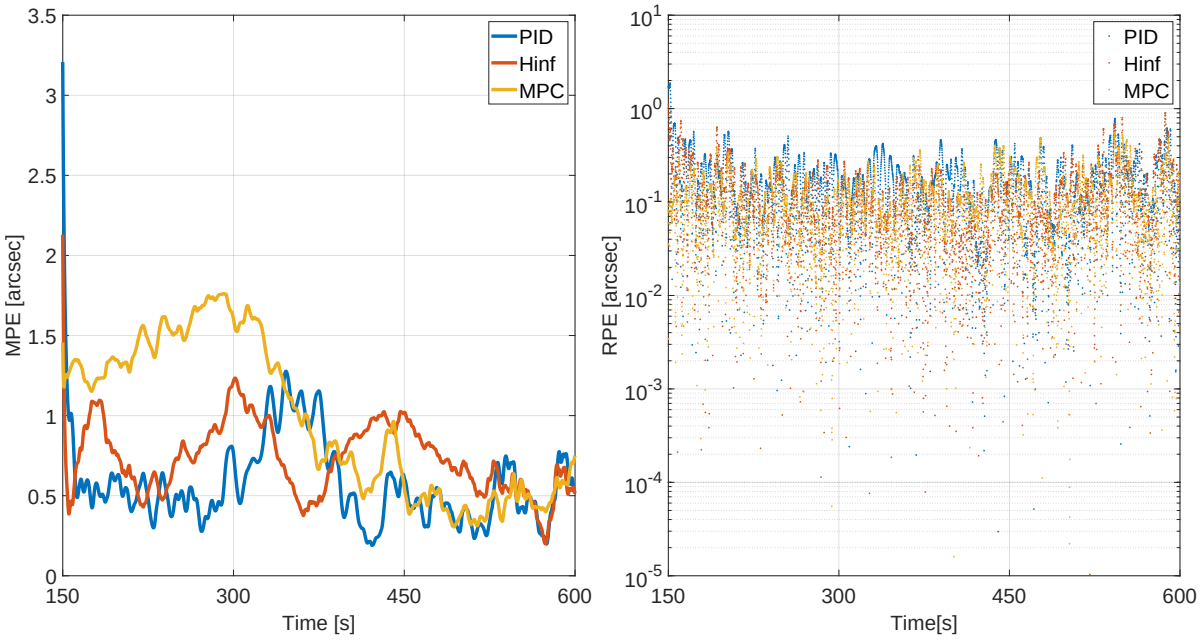

Figure 4.4: Pointing stability comparison for smooth single-point tracking.

To assess pointing stability, the RPE index is used (fig. 4.4). It shows generally worse performance for the PID and similar performances for the H_∞-based controller and the MPC, validating the main drawback of simple, linear controllers. While robust and predictive controllers have full access to the state-space model of the system, the PID has barely half the states of the other two. As a result, pointing stability degrades. Overall, the three controllers perform on similar orders of magnitude, which was the

Controller	APE_{avg} [arcsec]	MPE_{min} [arcsec]	MPE_{max} [arcsec]	RPE_{avg} [arcsec]
PID	0.5611	0.1900	1.2766	0.1598
H_∞	0.7355	0.2005	1.2340	0.1134
MPC	0.9659	0.3049	1.7611	0.1086

Table 4.2: Pointing performance error indices for smooth single-point tracking in terms of RMS error.

main goal of this first, simple experiment. The average APE is almost twice as much for the MPC than the PID, with H_∞ being in the middle. The minimum MPE is roughly similar for all three, with the peak values being smallest for H_∞ and largest for the MPC (by far). The average RPE is 50% higher for the PID than for the other two controllers, indicating the decreased pointing stability.

Ground track error

Pointing error is not everything in the performance of the optical system. While it is a strong indicator of imaging performance, the ground error of the LOS vector is the only measurement to give a precise result for a simulated camera image. Due to the geometry of the Earth and the spacecraft's orbit, not all segments of the precise pointing phase are made equal. Pointing errors and instabilities are greatly amplified while the spacecraft is far away from the target, and only reflect true performance in a roughly one-minute-long window of time while the spacecraft is passing over the ground target.
To calculate the exact location of the camera image centroid, three elements are needed: the line-of-sight vector of the spacecraft in terms of ECEF coordinates, the current ECEF position of the spacecraft, and a geometric representation of the planet. In this case, the WGS84 reference ellipsoid is used with a constant equatorial radius (semi-major axis) of 6378.137 km and a polar radius (semi-minor axis) of 6356.752. This gives an inverse flattening value of 298.257223563.
Then, to simplify the coordinate geometry problem significantly, these elements are made dimensionless with the WGS84 radii of the planet. First, the line-of-sight vector:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} R_\oplus & 0 & 0 \\ 0 & R_\oplus & 0 \\ 0 & 0 & R_{pol} \end{bmatrix}^{-1} \mathbf{R}_{FB} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.3}$$

Then, the ECEF position vector:

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} R_\oplus & 0 & 0 \\ 0 & R_\oplus & 0 \\ 0 & 0 & R_{pol} \end{bmatrix}^{-1} \mathbf{r}_{BF}^F \tag{4.4}$$

Finally, the reference ellipsoid is reduced to a perfect sphere in the dimensionless system:

$$x^2 + y^2 + z^2 = 1 \tag{4.5}$$

Now, consider a straight line that is defined by the LOS vector and ECEF position vector, representing all potential visible points from the camera:

$$LOS(k) = \mathbf{r}_{BF}^F + k \cdot \mathbf{R}_{FB} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + k \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{4.6}$$

This expression can be substituted into the equation of the perfect sphere to find the closest intersection.

$$(x_0 + kx_1)^2 + (y_0 + ky_1)^2 + (z_0 + kz_1)^2 = 1 \tag{4.7}$$

This can be reformulated as a quadratic formula for k:

$$\underbrace{\left(x_1^2 + y_1^2 + z_1^2\right)}_{a} k^2 + 2\underbrace{\left(x_0x_1 + y_0y_1 + z_0z_1\right)}_{b} k + \underbrace{\left(x_0^2 + y_0^2 + z_0^2\right)}_{c} = 0 \tag{4.8}$$

And can be solved with an adjusted form of the quadratic formula due to the 2b portion in the middle. Note that only a single root needs to be calculated, as the smaller one is guaranteed to be the closest intersection of the planet due to how the vectors have been set up. The exact ECEF coordinates of the target are then determined with the SI position and LOS vectors:

$$k_1 = \frac{-2b - \sqrt{4b^2 - 4ac}}{2a} = -\frac{b + \sqrt{b^2 - ac}}{a} \qquad b^2 - ac \geq 0 \tag{4.9}$$

$$\mathbf{r}_{GF}^F = \mathbf{r}_{BF}^F + k_1 \cdot \mathbf{R}_{FB} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.10}$$

Finally, the ground pointing error is quantified with the norm of the ground error:

$$\mathbf{r}_e = ||\mathbf{r}_{GF}^F - \mathbf{r}_{TF}^F|| \tag{4.11}$$

As expected, the period of the closest approach will yield the smallest ground errors for all controllers (fig. 4.5). This is roughly on the scale of 1-5 meters, about the size of a car. The amplification of pointing instability can also be observed, especially as the spacecraft moves away from the target. For another visualization, images were simulated at 1 Hz and plotted on top of the satellite map of Delft. The centroids of images taken with the PID loop have an ellipse-shaped spread, while the H_∞-based controller causes a barbell-shaped deformation, and MPC has a clear bias towards one side of the target (fig. 4.6). Interestingly enough, this transforms the spread into an L-shape.

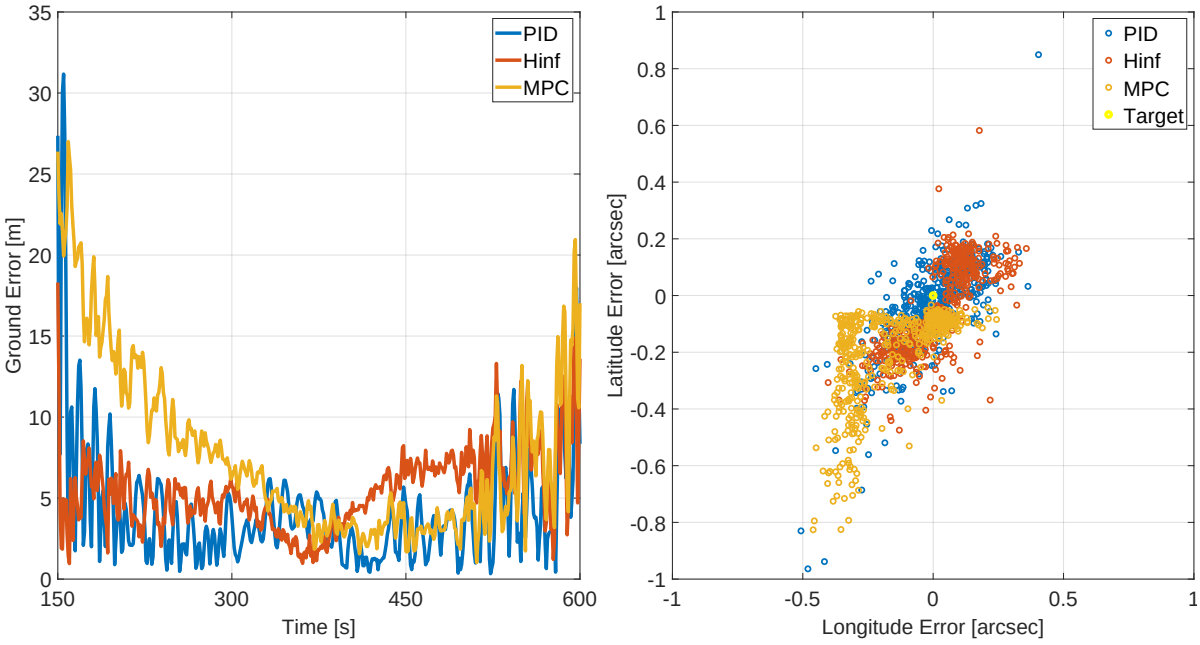

Figure 4.5: Ground error characteristics for smooth single-point tracking.

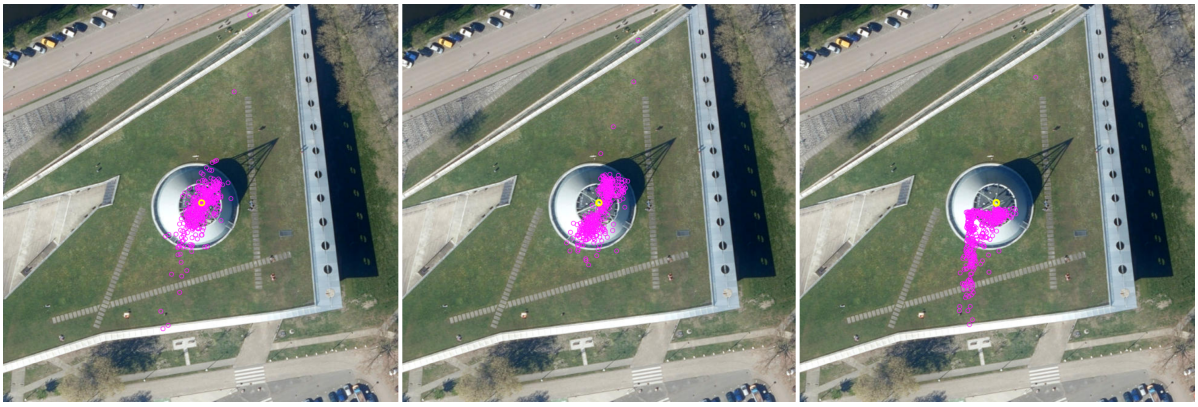

Figure 4.6: Camera images at 1 Hz for smooth single-point tracking, target in yellow. Left to right: PID, H_∞, MPC.

4.3. The impact of trajectory design

For the next experiment, the controllers will receive a much larger demand from the reference angle generator. Another target is added before the library, as the spacecraft will first point towards the bell tower of Nieuwe Kerk on the main square of Delft (lat 52.0123596 deg, lon 4.3608797 deg). Another change is that the simulation is started from steady-state, a full 3 minutes later than the previous experiment. Between the two targets, a 15-second long smoothing takes place to ensure that the controllers do not have to deal with a step-like change in the trajectory which would disturb the structural dynamics and push the actuators to torque saturation.

In this dual-point tracking scenario, some interesting changes can be observed. First, the pointing performance of the PID drops drastically once the change is made (fig. 4.7). This is caused by the much slower dynamics of the PID loop compared to the two other, more advanced controllers. There is also a residual oscillation in the PID response which is only attenuated to acceptable levels in about a minute. In contrast, the H_∞ and MPC loops show no immediately noticeable difference between the tracking of the two targets (fig. 4.8). While the PID takes about 30 seconds from the end of the smoothing period to reach the desired 2 arcsec threshold of the system requirements, both the H_∞ and MPC reach this in a few seconds. During the transient, the PID reaches a peak of 12 arcsecs of error, compared to the MPC's 5 arcsecs and the H_∞-based controller's 3 arcsecs. Pointing stability is also greatly reduced for the PID at 1-5 arcsecs of divergence from the windowed mean. As in the previous scenario, here the robust and predictive controllers also show similar RPE at 0.01-0.1 arcsec.

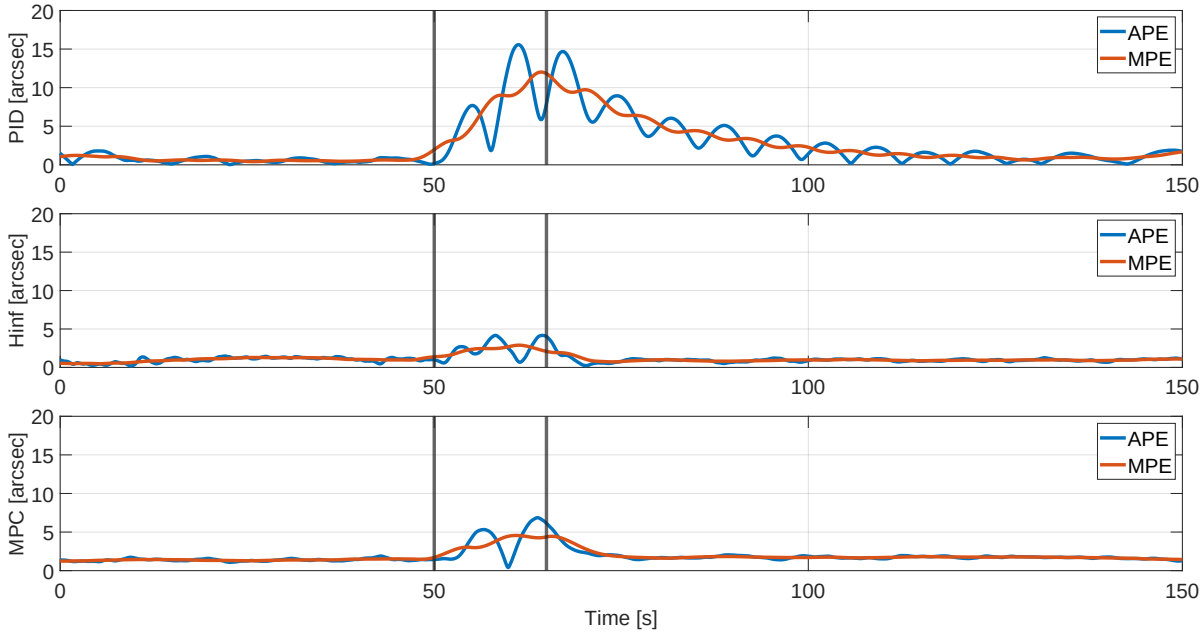

Figure 4.7: Pointing performance comparison for dual-point tracking.

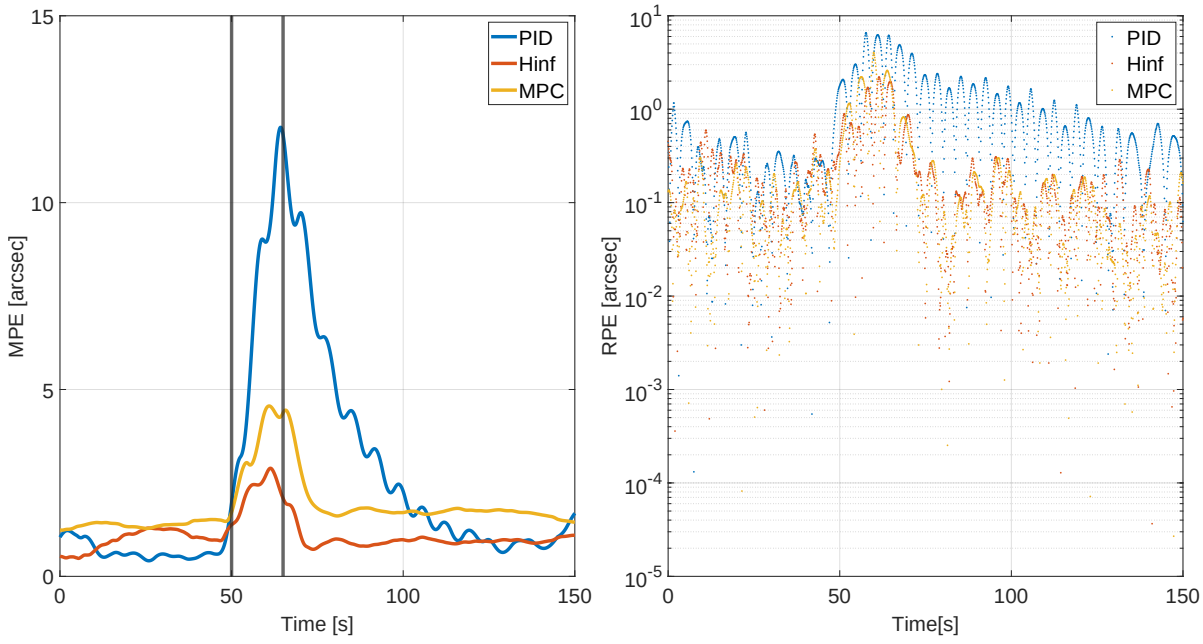

Figure 4.8: Pointing stability comparison for dual-point tracking.

Because of its residual error, the average APE for MPC is still worse than the PID loop, however H_∞ vastly outperforms both (table 4.3). Robust control is also a winner in the MPE bounds, with the PID explicitly failing the 2 arcsec requirement, and MPC's best MPE barely meeting H_∞'s worst.

Controller	APE_{avg} [arcsec]	MPE_{min} [arcsec]	MPE_{max} [arcsec]	RPE_{avg} [arcsec]
PID	1.5038	0.4167	6.4843	0.5524
H_∞	0.9609	0.4811	1.3788	0.1220
MPC	1.5771	1.2365	1.8301	0.0903

Table 4.3: Pointing performance error indices for dual-point tracking.

The ground errors also reflect these differences with all loops producing worse results due to the more aggressive reference signal. Both advanced controllers produce relatively stable tracking at 5-10 m ground error which is largely due to a lopsided bias. PID's spread keeps the same ellipse shape but at a loss of large oscillations that the controller has trouble dampening. As a result, even without bias, PID is vastly outperformed by both advanced controllers in a more demanding control scenario.

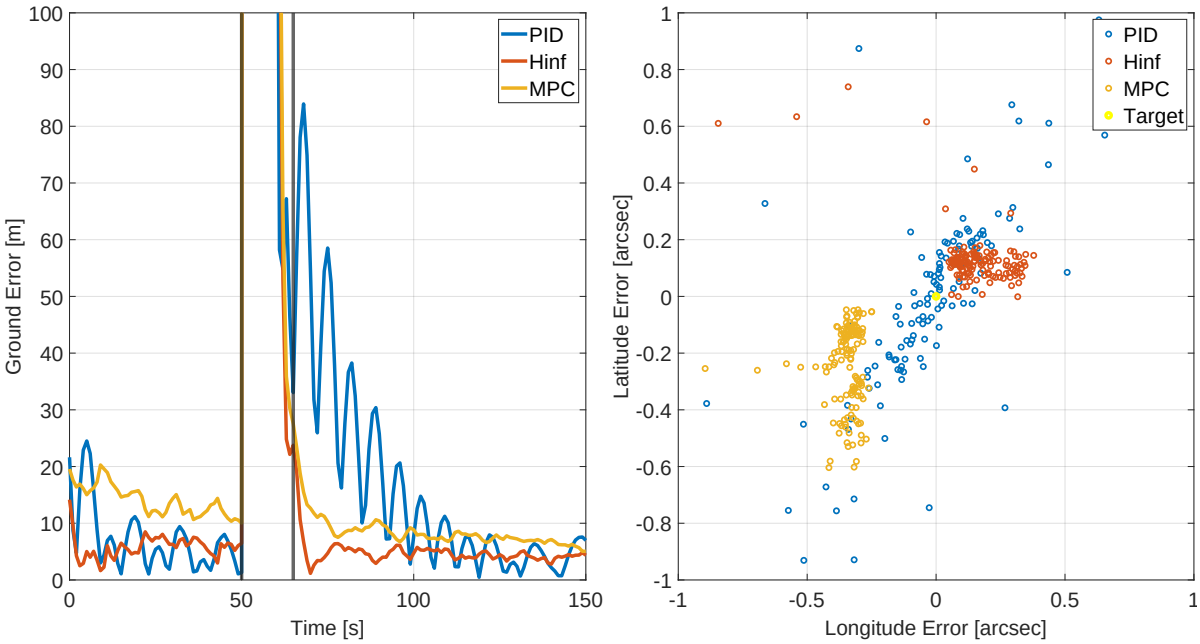

Figure 4.9: Overlaid ground error characteristics for dual-point tracking. Both targets in yellow.

Figure 4.10: Camera images at 1 Hz for dual-point tracking. Left to right: PID, H_∞, MPC.

4.4. Active-passive vibration isolation

Based on the outer loop tuning campaign, it can be determined that in a single-loop setup, the maximum frequency of decent reference tracking is roughly 5 rad/s, much lower than the estimated vibrational modes of structural dynamics or the microvibration environment of the reaction wheel assembly. This does not come as a surprise, considering the findings of the literature study.
In their 2014 overview of the field, Liu et al have identified several candidates for space applications. High-frequency (above 200 rad/s) microvibrations are usually handled via passive means, while low-frequency vibrations can be tackled with active control. Since a flexible spacecraft suffers from disturbing effects in the entire frequency range, sensitive payloads must be protected using both active and passive means. Some promising isolation techniques involve electro-rheological fluids, smart materials, or nonlinear dampers, however these methods are yet to go through the rigorous testing and verification process of space systems. Fortunately, there is a candidate with a proven track record and reliable pointing performance: isolation platforms with hybrid struts [14].

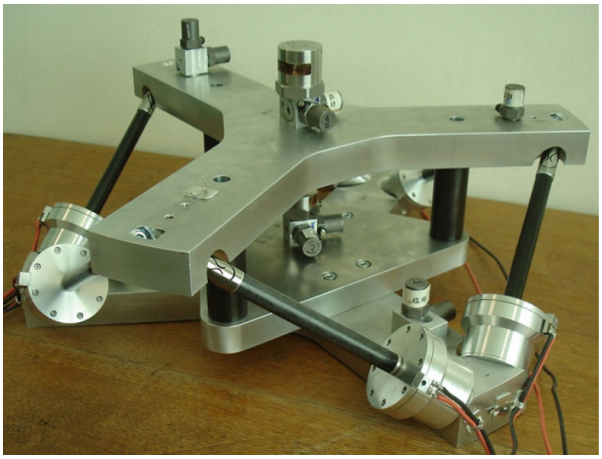

Figure 4.11: Active-passive isolation device based on a cubic Stewart-Gough platform design. [5]

A Stewart-Gough platform (fig. 4.11) is a parallel manipulator device with linear actuators and universal joints that allow for the 6-DOF control of the payload plate. Their use in space applications is supported by a large research interest due to their inherent capability to not only provide articulation between different subsystems but also isolate microvibrations. This has the added benefit of merging two functionalities in the same mechanical system, decreasing both the complexity and the mass of the vehicle. The platform should be supported by at least 6 struts (a hexapod platform), but for the purpose of redundancy, there are also designs for 8 struts (octopod) or more. The configuration of the struts also influences the degree of cross-coupling in the device, which should be taken into consideration during the design process. In the case of extremely sensitive payloads, multiple isolation layers can also be considered [16][47]. As part of this thesis, the design of a single hexapod Stewart platform is investigated in a cascade precise pointing control system, placed between the spacecraft bus and the imaging subsystem (fig. 4.12).

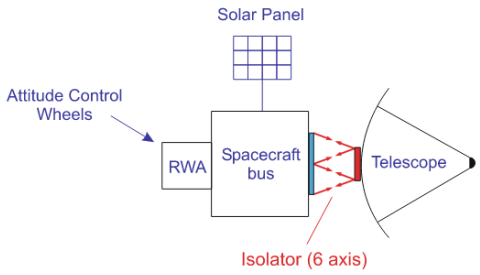

Figure 4.12: Spacecraft architecture with isolator between the bus and the telescope on a flexible spacecraft. [5]

Mathematical modelling of Stewart-Gough platforms

The mathematical model of the isolation platform is largely inspired by the work of Hanieh and Preumont. Their work also involved the construction of a hardware demonstrator, whose frequency-domain characteristics should be matched as closely as possible. The goal is to build an idealized model of the hexapod device with struts capable of active-passive isolation. First off, the hexapod should follow a cubical structure (fig. 4.13) with all struts being orthogonal to each other in the neutral position. This reduces the effects of cross-coupling between them to the smallest degree possible [5][48].

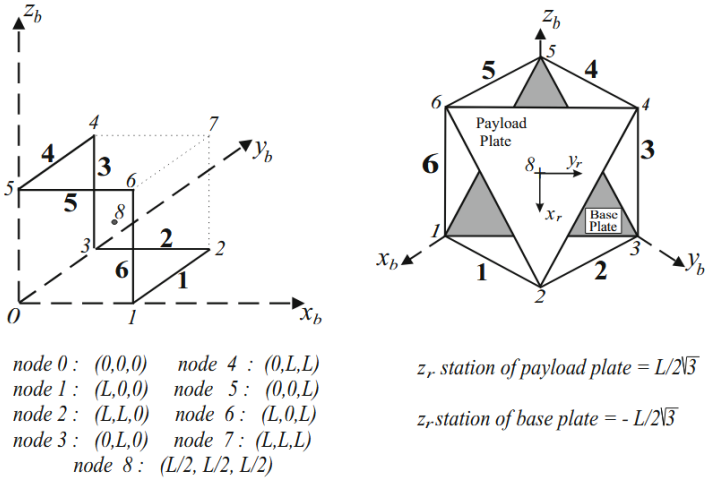

Figure 4.13: Geometry of cubic hexapod isolator. Bold numbers indicate actuated struts. [5]

Second, the length of a strut is fixed at 10 cm, and the mass of the payload platform at 1 kg. Struts are considered massless ideal spring-dampers to simplify the dynamical model with a stiffness coefficient of 1 N/m and a damping coefficient of 0.5 Ns/m. These values (summarized in table 4.4) are supported by the transfer functions identified in Hanieh's dissertation [5][48].

Symbol	Name	Value	Unit
ℓ_0	Neutral strut length	0.1	m
m_I	Mass of payload platform	1	kg
k_I	Strut stiffness coefficient	1	N/m
c_I	Strut damping coefficient	0.5	Ns/m

Table 4.4: Parameters of designed Stewart-Gough platform.

Based on this geometry, the initial states of the model can be constructed. Consider that $\mathbf{x}_S$ and $\mathbf{x}_P$ are state vectors containing the position and rotation quaternion of the support plate (centered on the S frame) attached to the body of the spacecraft, as well as the payload plate (centered on the P frame) containing the imaging device. The derivatives of the state vectors are the velocities and angular velocities of the frames, with all of these physical quantities expressed in the body frame (B) of the spacecraft.

$$\mathbf{x}_S = \Big[\underbrace{0 \quad 0 \quad 1.175}_{\mathbf{r}^B_{SB}} \Big| \underbrace{1 \quad 0 \quad 0 \quad 0}_{\mathbf{q}_{SB}} \Big]^T \qquad \dot{\mathbf{x}}_B = \Big[\underbrace{0 \quad 0 \quad 0}_{\mathbf{v}^B_{SB}} \Big| \underbrace{0 \quad 0 \quad 0}_{\boldsymbol{\omega}_{SB}} \Big]^T \tag{4.12}$$

$$\mathbf{x}_P = \Big[\underbrace{0 \quad 0 \quad 1.233}_{\mathbf{r}^B_{PB}} \Big| \underbrace{1 \quad 0 \quad 0 \quad 0}_{\mathbf{q}_{PB}} \Big]^T \qquad \dot{\mathbf{x}}_B = \Big[\underbrace{0 \quad 0 \quad 0}_{\mathbf{v}^B_{PB}} \Big| \underbrace{0 \quad 0 \quad 0}_{\boldsymbol{\omega}_{PB}} \Big]^T \tag{4.13}$$

From the rotation quaternion, rotation matrices can also be defined that will be useful in deriving the Jacobian matrix of the mechanical system:

$$\mathbf{q}_{PB}, \mathbf{q}_{SB} \longrightarrow \mathbf{R}_{PB}, \mathbf{R}_{SB} \qquad \mathbf{R}_{SP} = \mathbf{R}_{SB}\mathbf{R}^T_{PB} \tag{4.14}$$

Finally, the positions of each strut connection node are defined in the support and the payload frames, based on the corners of an equilateral triangle. The i-th column of these matrices corresponds to the start and end nodes of the i-th strut.

$$\mathbf{r}_{Si}^{S} = \frac{\ell_0}{\sqrt{6}} \begin{bmatrix} 2 & 2 & -1 & -1 & -1 & -1 \\ 0 & 0 & \sqrt{3} & \sqrt{3} & -\sqrt{3} & -\sqrt{3} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.15}$$

$$\mathbf{r}_{Pi}^{P} = \frac{\ell_0}{\sqrt{6}} \begin{bmatrix} 1 & 1 & 1 & -2 & -2 & 1 \\ -\sqrt{3} & \sqrt{3} & \sqrt{3} & 0 & 0 & -\sqrt{3} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.16}$$

Deriving the force Jacobian

Now, let us consider the suspension dynamics of the Stewart platform. There, the system can be thought of as an LTI MIMO 6-DOF mass-spring-damper system with six identical actuators:

$$\left(\mathbf{M}_I s^2 + \mathbf{C}_I s + \mathbf{K}_I\right) \mathbf{x}_P = \mathbf{J}_P \mathbf{u} \tag{4.17}$$

Here, $\mathbf{J}_P$ is the force Jacobian matrix, which is the transpose of the Jacobian connecting the state vector $\mathbf{x}_P$ to the spring extensions $\Delta\boldsymbol{\ell}_i$:

$$\Delta\boldsymbol{\ell}_i = \mathbf{J}_P^T \mathbf{x}_P \tag{4.18}$$

It can be calculated from the geometry of the isolator system. First, let us determine the positions of each strut's end node in the support frame:

$$\mathbf{r}_{Pi}^{S} = \mathbf{R}_{SP} \mathbf{r}_{Pi}^{P} + \mathbf{R}_{SB} \left(\mathbf{r}_{PB}^{B} - \mathbf{r}_{SB}^{B}\right) \tag{4.19}$$

From here, the length of individual struts can also be defined:

$$\boldsymbol{\ell}_i = \mathbf{r}_{Pi}^{S} - \mathbf{r}_{Si}^{S} \qquad \hat{\boldsymbol{\ell}}_i = \frac{\boldsymbol{\ell}_i}{||\boldsymbol{\ell}_i||} \tag{4.20}$$

And finally, the force Jacobian matrix in the neutral state is constructed as the well-known inverse kinematics solution for hexapods [5][16][48]. Note that $\mathbf{J}_P$ is rebuilt on every simulation time step:

$$\mathbf{J}_P = \begin{bmatrix} \hat{\boldsymbol{\ell}}_i \\ \mathbf{r}_{Pi}^{S} \times \hat{\boldsymbol{\ell}}_i \end{bmatrix} = \begin{bmatrix} 1 & 1 & -2 & 1 & 1 & -2 \\ \sqrt{3} & -\sqrt{3} & 0 & \sqrt{3} & -\sqrt{3} & 0 \\ \sqrt{2} & \sqrt{2} & \sqrt{2} & \sqrt{2} & \sqrt{2} & \sqrt{2} \\ 0 & 0 & \ell_0 & \ell_0 & -\ell_0 & -\ell_0 \\ -\frac{2\ell_0}{\sqrt{3}} & -\frac{2\ell_0}{\sqrt{3}} & \frac{\ell_0}{\sqrt{3}} & \frac{\ell_0}{\sqrt{3}} & \frac{\ell_0}{\sqrt{3}} & \frac{\ell_0}{\sqrt{3}} \\ \sqrt{2}\ell_0 & -\sqrt{2}\ell_0 & \sqrt{2}\ell_0 & -\sqrt{2}\ell_0 & \sqrt{2}\ell_0 & -\sqrt{2}\ell_0 \end{bmatrix} \tag{4.21}$$

Modelling system dynamics

In order to expand the previously derived linear dynamics to a nonlinear simulation model that can be included in the high-fidelity `Simulink` environment, some adjustments are needed to the mass-spring-damper system. First off, let us define the rate of change in the length of a strut based on a body-fixed support plate and a free payload plate:

$$\delta\boldsymbol{\ell}_i = \hat{\boldsymbol{\ell}}_i \cdot \mathbf{R}_{SB} \left(\mathbf{v}_{PB}^{B} + \boldsymbol{\omega}_{PB}^{B} \times \mathbf{R}_{SP} \mathbf{r}_{Pi}^{P}\right) \tag{4.22}$$

Then, consider the force transformation matrix from the Body frame to the Payload frame that accounts for the shifted center of the payload frame as well as its orientation compared to the body frame:

$$\mathbf{J}_{BP} = \begin{bmatrix} \mathbf{R}_{SB}^{T} & \mathbf{0}_{3x3} \\ \mathbf{R}_{PB} \mathbf{R}_{SB}^{T} \left[\mathbf{r}_{SB}^{B} - \mathbf{r}_{PB}^{B}\right]_{\times} & \mathbf{R}_{PB} \mathbf{R}_{SB}^{T} \end{bmatrix} \tag{4.23}$$

Then, the forces acting on the payload plate can be expressed in the body frame with the use of the force transformation and force Jacobian matrices:

$$\begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{bmatrix}_{P}^{B} = \mathbf{J}_{BP} \mathbf{J}_P \left[\mathbf{f}_a - c_I \delta\boldsymbol{\ell}_i - k_I \left(\boldsymbol{\ell}_i - \ell_0\right)\right] \tag{4.24}$$

Finally, the equations of motion for the payload plate are expressed in terms of relative dynamics, considering the rotating frame of the spacecraft Body and the arising Coriolis, Euler, and centrifugal forces. Rotational dynamics are also expressed in the body frame, $\mathbf{J}_I$ is the mass moment of inertia matrix of the payload plate, approximated as a flat equilateral triangle:

$$\ddot{\mathbf{x}}_P = \begin{bmatrix} \dot{\mathbf{v}}^B_{PB} \\ \dot{\boldsymbol{\omega}}^B_{PB} \end{bmatrix} = \begin{bmatrix} \frac{1}{m_I}\mathbf{f}^B_P + \mathbf{g}^B - \dot{\mathbf{v}}^B_{BF} - 2\boldsymbol{\omega}^B_{BI} \times \mathbf{v}^B_{PB} - \dot{\boldsymbol{\omega}}^B_{BI} \times \mathbf{r}^B_{PB} - \boldsymbol{\omega}^B_{BI} \times \left(\boldsymbol{\omega}^B_{BI} \times \mathbf{r}^B_{PB}\right) \\ \mathbf{J}_I^{-1}\left(\boldsymbol{\tau}^B_P - \boldsymbol{\omega}^B_{PB} \times \mathbf{J}_I \boldsymbol{\omega}^B_{PB}\right) \end{bmatrix} \tag{4.25}$$

Despite the complexity of the dynamical equations, the kinematics are relatively simple with similar formulas used in the EOM of the spacecraft body as well:

$$\dot{\mathbf{x}}_P = \begin{bmatrix} \dot{\mathbf{r}}^B_{PB} \\ \dot{\mathbf{q}}_{PB} \end{bmatrix} = \begin{bmatrix} \mathbf{v}^B_{PB} \\ \frac{1}{2}\mathbf{q}_{PB} \circ \mathring{\boldsymbol{\omega}}^B_{PB} \end{bmatrix} \tag{4.26}$$

There are two possible linearizations to construct from this dynamical model, because of the way the inputs and outputs are set up: for a single output channel (state vector of P), there are two input channels. The first input channel corresponds to the active forces of the struts and is represented by the transfer function $\mathbf{G}^I_F(s)$, to be used in control design. The second input channel covers the disturbances from the Body frame and is represented by the transfer function $\mathbf{G}^I_X(s)$, to be used in the evaluation of cascade performance and to match the passive isolation characteristics of the isolator model to the design of Hanieh. First, an auxiliary transfer function is defined to make the derivation clearer:

$$\mathbf{G}^I_A(s) = \mathbf{M}_I^{-1}\mathbf{J}_{BP}\mathbf{J}_P\left(c_I s + k_I\right)\mathbf{J}_P^T\mathbf{R}_{SB} \qquad \mathbf{M}_I = \begin{bmatrix} m_I\mathbf{I}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{J}_I \end{bmatrix} \tag{4.27}$$

Then, the two transfer functions characterizing the Stewart platform are:

$$\mathbf{G}^I_F(s) = \left(s^2\mathbf{I}_{6x6} + \mathbf{G}^I_A(s)\right)^{-1}\mathbf{M}_I^{-1} \qquad \mathbf{G}^I_X(s) = \left(s^2\mathbf{I}_{6x6} + \mathbf{G}^I_A(s)\right)^{-1}\mathbf{G}^I_A(s) \tag{4.28}$$

The Bode plot of the isolator for disturbance rejection from the Body frame shows similar frequency-domain characteristics to the measurements of Hanieh et al (fig. 4.14). In all directions, they are low-pass filters with a cut-off frequency of roughly 1 rad/s and -20 dB/decade of attenuation.

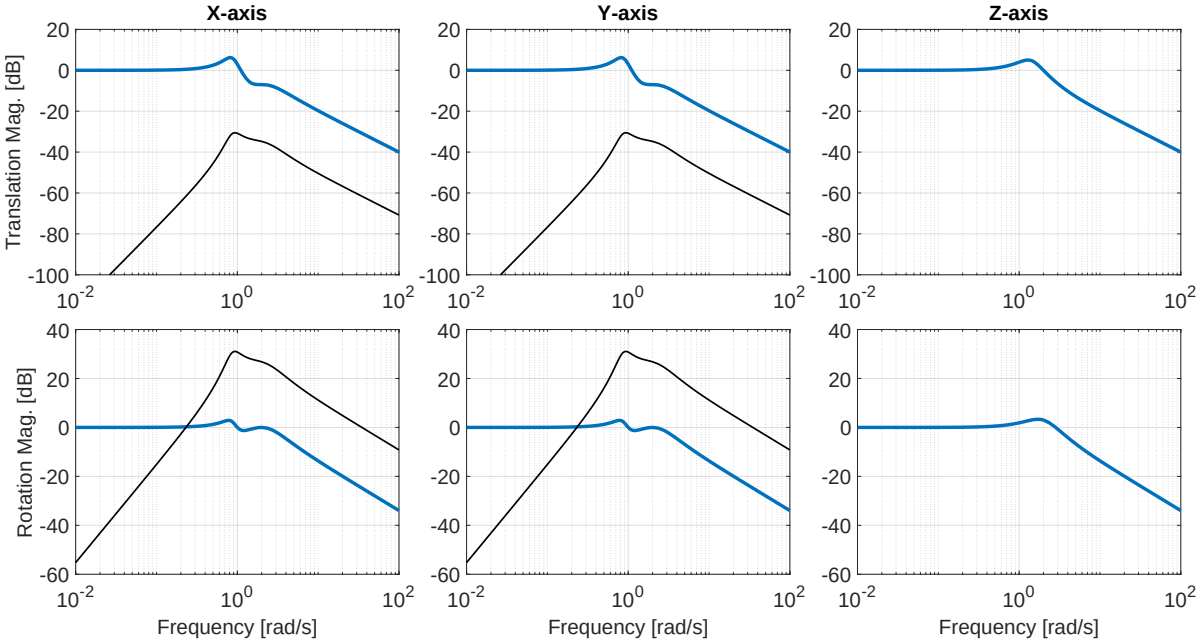

Figure 4.14: Bode response of isolator plant model for Body-frame disturbances. Cross-coupling effect between opposite-axis forces and torques in black.

Linearization for control design

The linear actuators of the struts are saturated at ±0.05 N and have a 0.01-sec delay which is reduced in the control-oriented linearized model with a 5-th order Padé approximant:

$$e^{-0.01s} \approx G^I_{Pade}(s) = -\frac{s-729.3}{s+729.3}\frac{s^2-1341s+57090}{s^2+1341s+57090}\frac{s^2-929.9s+72630}{s^2+929.9s+72630} \tag{4.29}$$

The actuators themselves are driven by electric motors with a cutoff frequency of 100 rad/s that form a 6x6 diagonal matrix to represent all struts:

$$\mathbf{G}^{6x6,diag}_{Strut}(s) = \frac{100}{s+100} \tag{4.30}$$

And body-frame forces and torques are converted to strut forces with the use of a decoupling matrix based on the force Jacobian and the force transform matrices:

$$\mathbf{D}_I = (\mathbf{J}_{BP}\mathbf{J}_P)^{-1} \tag{4.31}$$

The final model is an expansion of $\mathbf{G}^I_F(s)$ with actuator dynamics and time delay characteristics, as presented in fig. 4.15. It allows the design of a collocated controller by defining the outputs as the position and orientation of the payload plate while system inputs are the overall forces and torques acting on the payload plate instead of individual strut forces.

$$\mathbf{G}^I_{Act}(s) = \mathbf{G}_{Strut}(s)\mathbf{D}(s)\mathbf{G}^I_{Pade}(s) \tag{4.32}$$

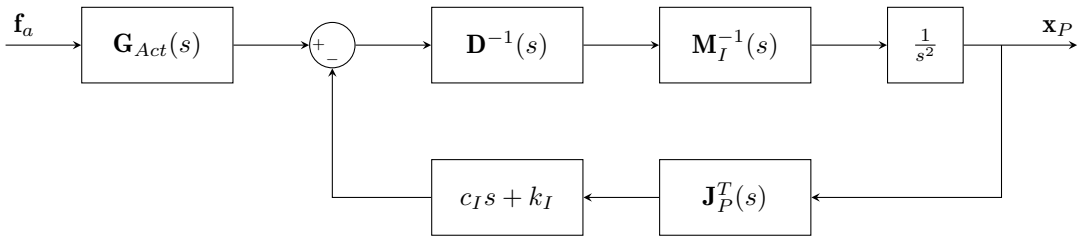

Figure 4.15: Linearized isolator plant model for control design.

4.5. Control design of pointing platform

The inner loop is chosen to be driven by an H_∞-based robust controller. A PID was tested but its performance was found insufficient for the complexity of the isolator system, and a predictive controller would not have allowed for the precise definition of frequency ranges to be tackled by the controller. However, there is an important change from the robust control design presented in the previous chapter. For the isolator loop, the controller is tuned for mimicking a reference system in its closed-loop behaviour. This reference system is also an integral part of the augmented plant (fig. 4.16) with its own filters.

$$\mathbf{G}^I_{ref,diag}(s) = \frac{100}{s^2+20s+100} \tag{4.33}$$

This reference system is the exact same for the entire 6x6 diagonal. However, to somewhat speed up the pitch dynamics, the damping coefficient of the transfer function is adjusted for output 5:

$$\mathbf{G}^I_{ref,5}(s) = \frac{100}{s^2+4s+100} \tag{4.34}$$

The weights of the reference system are a constant gain of 1 as any adjustment to this value was found to produce undesirable reference tracking characteristics:

$$\mathbf{W}^I_r(s) = \mathbf{I}_{6x6} \tag{4.35}$$

The rest of the filters are various low-pass and high-pass filters, as well as zero gains for off-diagonal elements (fig. 4.17). Similarly to the previous design, the filter parameters were tuned using a combination of intuitive initial values selected based on desired system performance and iterative tuning based on the achieved disturbance rejection and reference tracking characteristics.
There were two main goals in the tuning process. First, to diminish the cross-coupling effects from X-displacement to Pitch and Y-displacement to Roll. Second, to bring the cutoff frequency of the pointing loop beyond 17 rad/s in order to actively control at least the first flexible mode of the system and achieve at least -40 dB/decade of attenuation for higher frequencies. While some of the poles ended up too fast, they can be slowed down in future design iterations.

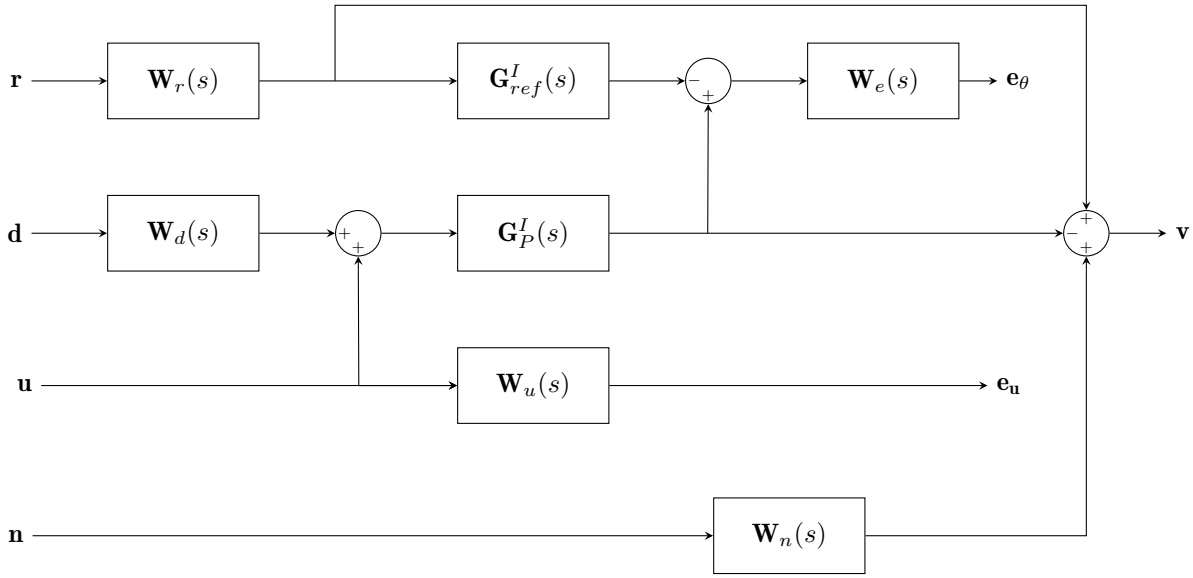

Figure 4.16: Augmented plant for isolator control loop design.

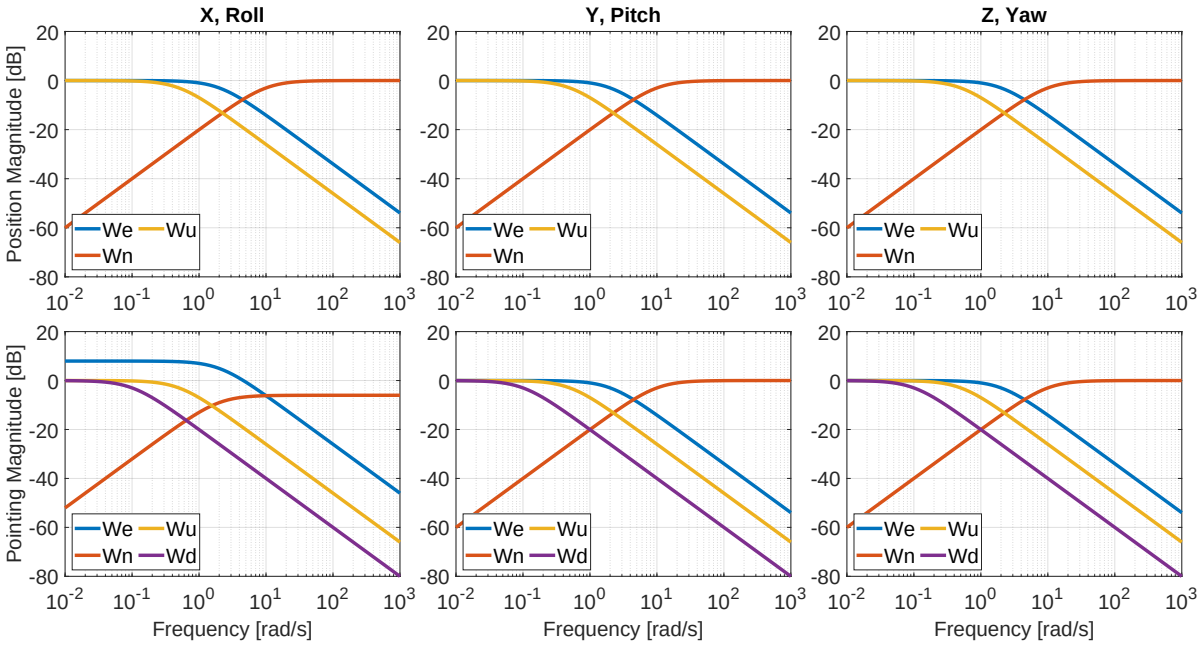

Figure 4.17: Tuning weights of isolator control loop.

The control synthesis algorithms concluded with a final γ of 1.0058. The final controller had 81 states, which was reduced to the 18 largest ones in terms of Hankel singular value, with the methods described in the last chapter. The pole-zero map of the closed-loop is stable, with the highest frequency points corresponding to the Padé-approximant (fig. 4.19). Most system modes can be found below 100 Hz, with some poles placed near the imaginary axis, but still on the stable side.
The closed-loop shows low-pass reference tracking characteristics with a cut-off slightly above 20 rad/s for Roll and Pitch, and 30 for Yaw (fig. 4.18). The cross-coupling between position and pointing dynamics has been significantly reduced in magnitude to sit below the tracking modes, and disturbance rejection for all other transfers is sufficiently low at -100 dB peaks. The position loop is slightly lower, which is acceptable as the only requirement there is to fit within the operational bounds of the isolator.

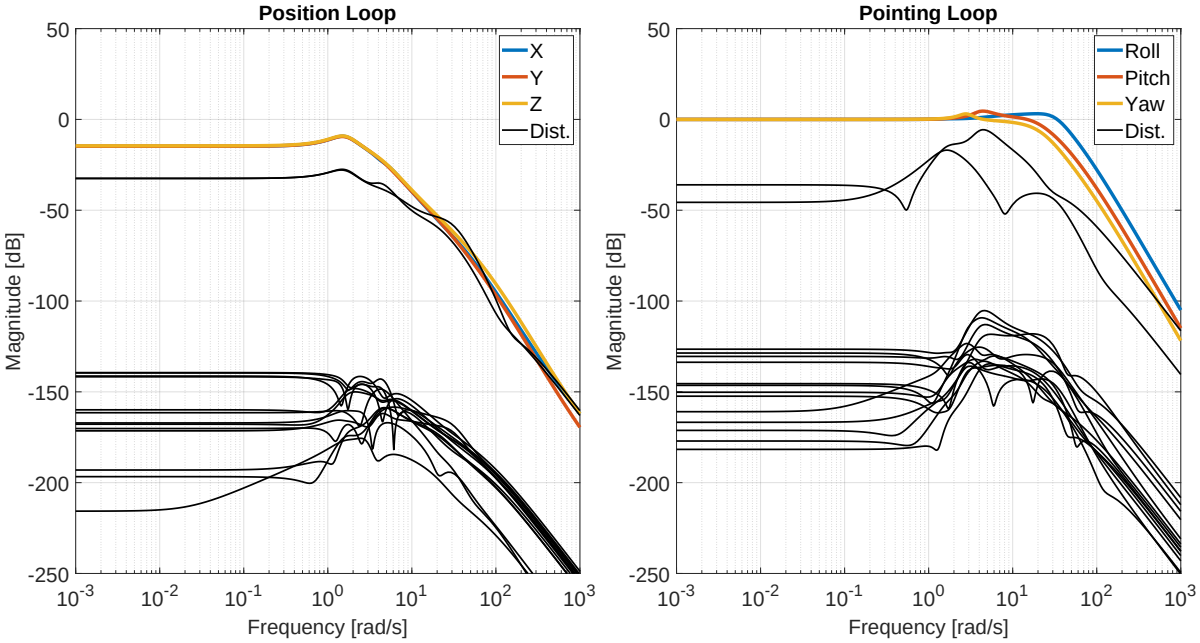

Figure 4.18: Reference tracking and disturbance rejection closed-loop Bode plots on the position and pointing outputs.

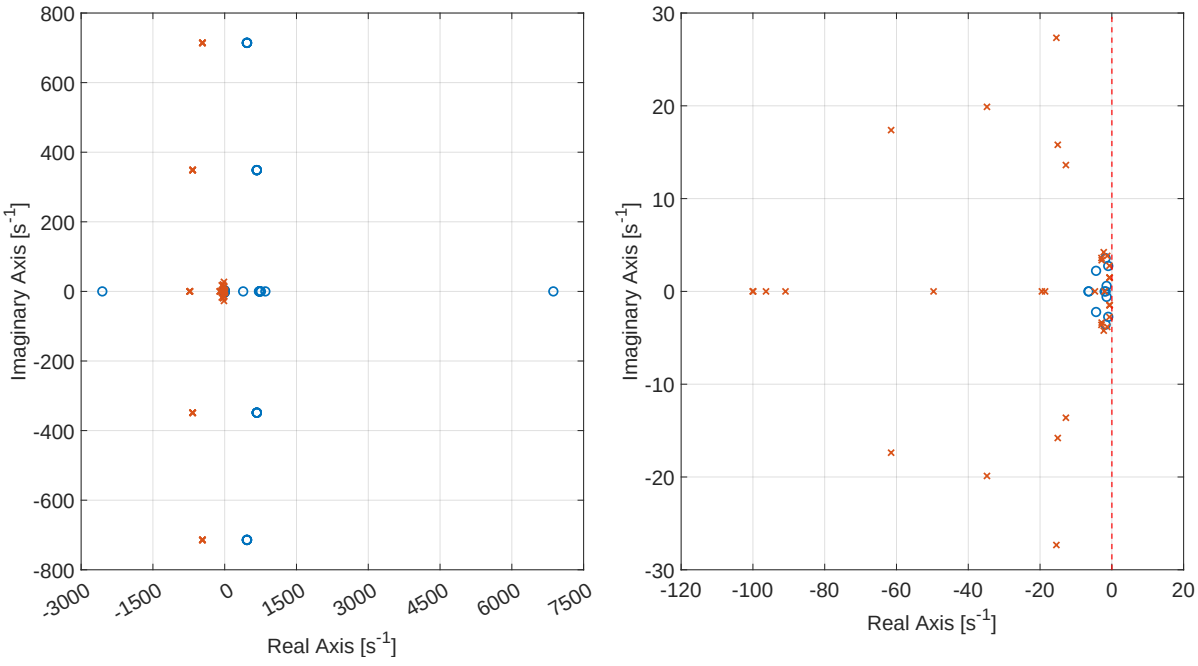

Figure 4.19: Pole-zero map of closed isolator control loop. Right side: Zoom on origin.

The step response of the position loop shows a greatly reduced capacity to match a reference signal of 1. As explained for the Bode diagram, this is acceptable behaviour. The position loops all reach a steady state within 1-5 seconds with no residual error, and disturbances from other reference signals have a noticeably diminished magnitude compared to the main tracking signal.

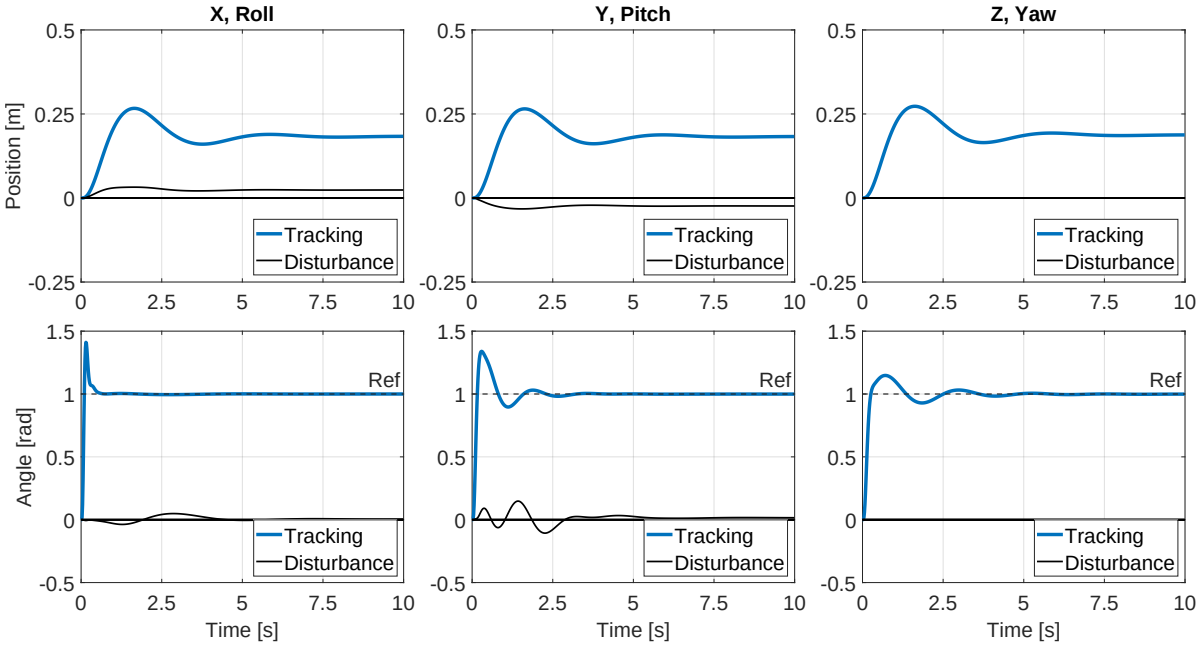

Figure 4.20: Step response and disturbance rejection for designed H_∞-based isolator control loop.

Virtual sensor in the isolator loop

Due to the highly idealized model of the isolator loop, a virtual sensor is used that could be based on AI-driven image-based solutions or a laser-based method analogous to communication terminals. It measures the difference between the real angle of the target and the current orientation of the spacecraft. In order to limit the potential performance of the control loop, the virtual sensor has a 5-degree wide range in roll, pitch, and yaw, and its output is digitized in a signed 16-bit integer. In this process, a zero-mean Gaussian noise is added to all channels with 10^{-13} variance, which puts the virtual sensor higher than an IMU but lower than a gyro in terms of performance. Overall, this results in roughly 0.15 arcsec of noise at 100 Hz.

To improve system dynamics around the saturation of the sensor, an arctangent-based smoothing algorithm is applied between ± 2 and ± 2.5 degrees instead of a harsh cutoff, which would cause a slight overshooting of the sensor bounds with a fast controller. The equation is a simple linear transformation of the function, visualized in figure 4.21.

$$\phi_{sat} = \pm 2 + \frac{1}{\pi} \arctan\left(\pi \left(\phi \mp 2\right)\right) \qquad \phi \notin [-2, 2] \tag{4.36}$$

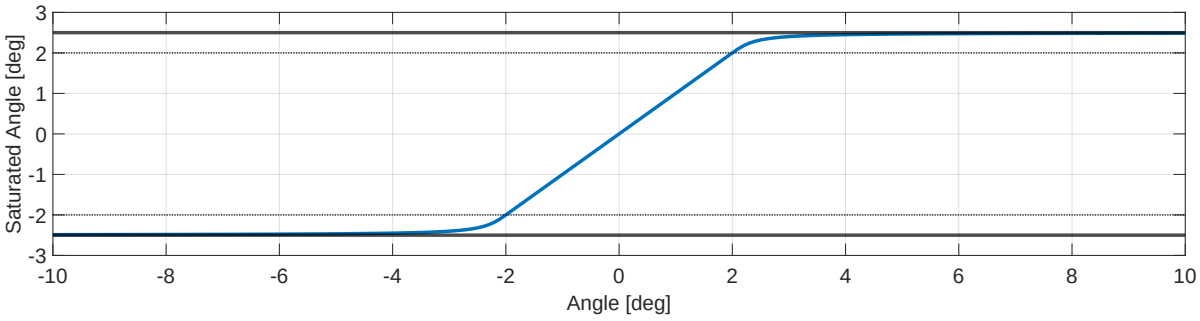

Figure 4.21: Arctangent-based saturation function for the isolator loop.

4.6. Cascade loop control performance

In the cascade loop, the isolator's robust controller (as the inner loop) is combined with one of the three designed controllers for the spacecraft ADCS (as the outer loop). This section compares the time-domain performances of various combinations and assesses the improvements in pointing precision from the inclusion of the payload isolator.

Dual-point tracking

In the purely outer loop case, the H_∞-based controller had the best APE with PID and MPC tied slightly behind it, while MPC and the H_∞-based controller produced the best RPE with PID far behind them. The addition of the isolator changes these standings quite significantly (table 4.5, fig. 4.22).
In all indicators, the PID and MPC-based cascade loops performed best, at roughly similar average APE (0.17-0.19) and RPE (0.06-0.08), as well as a similar MPE ranges (0.12-0.6 for PID, 0.14-0.55 for MPC). Surprisingly, the H_∞-based controller benefited much less from the expansion of the control loop. While its APE was brought down to 0.27 arcsec, and the MPE range was lowered to similar levels as the other two controllers (0.18-0.69), its average RPE barely changed: dropping from 0.122 to 0.115. This could be caused by the H_∞ tuning algorithm producing a more aggressive controller to achieve better performance indicators in the outer loop. As a result, the general noisiness and speed of this controller cause disturbances on the payload platform that are harder to attenuate for the isolator loop. This is an important conclusion that will be elaborated on later in this section: with the addition of the isolator, there is no need to provide precise control in the outer loop at all costs, and there should be more emphasis on robust and loose control that makes the inner loop's operation easier.

Controller	APE_{avg} [arcsec]	MPE_{min} [arcsec]	MPE_{max} [arcsec]	RPE_{avg} [arcsec]
PID	1.5031	0.4174	6.4772	0.5520
PID+Isol	0.1701	0.1272	0.6008	0.0654
H_∞	0.9612	0.4832	1.3772	0.1220
H_∞+Isol	0.2659	0.1844	0.6861	0.1154
MPC	1.5930	1.2444	1.8495	0.0902
MPC+Isol	0.1890	0.1372	0.5510	0.0867

Table 4.5: Pointing performance error indices for dual-point tracking with cascade control.

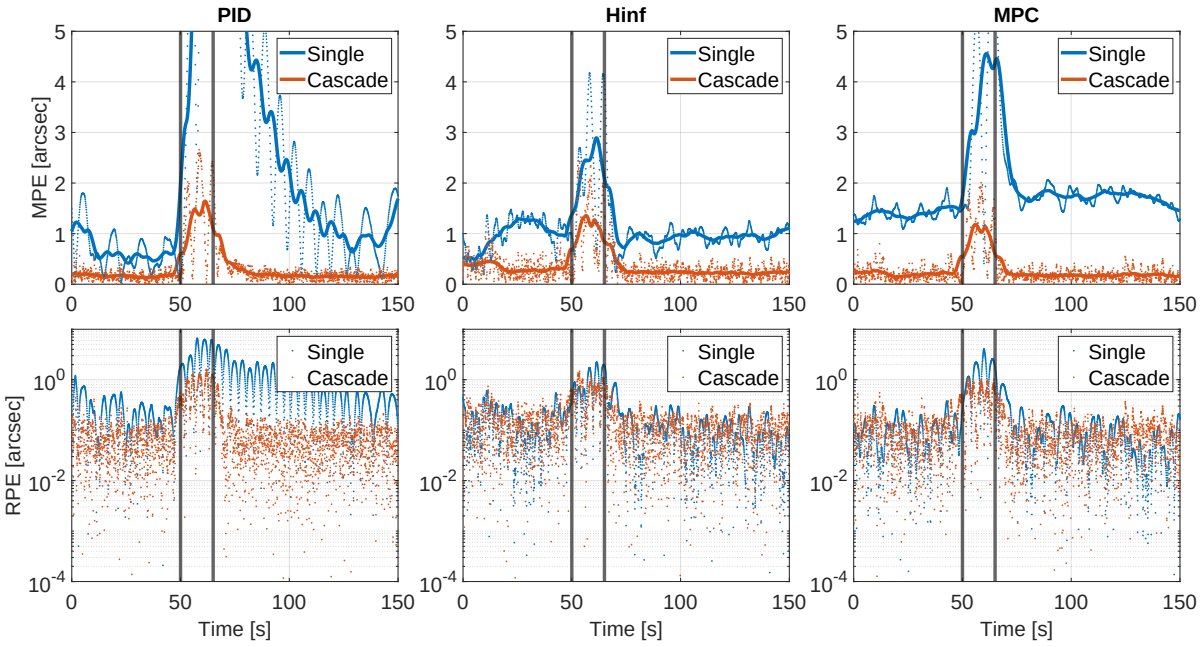

Figure 4.22: Performance indicators of cascade control loop for dual-point tracking. Solid MPE, dashed APE.

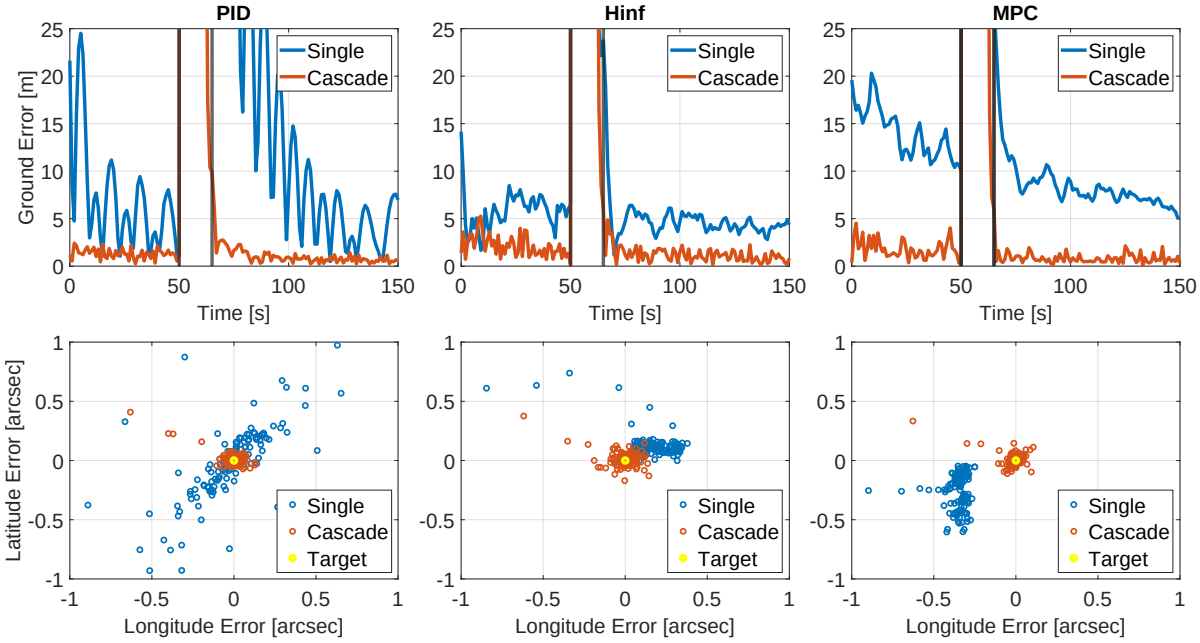

Figure 4.23: Ground error characteristics of cascade control loop for dual-point tracking.

However, there is a limit to relaxing control requirements in the outer loop. The slow settling time of the PID causes some extra strain in the isolator loop, which adds some extra seconds in eliminating residual errors. This is especially visible on the ground error chart (fig. 4.23), where the two advanced controllers have little issue reaching the target in time while the PID struggles.

Multi-point tracking experiment

To expand on the conclusion of the cascade loop dual-pointing scenario, new targets are defined. In a multi-tracking experiment, the satellite would be observing the town centers of Terschelling (in the West Frisian Islands), Den Helder, Amsterdam, and Delft (fig. 4.24). The spacecraft takes about 40 seconds to pass through the entire country of the Netherlands, with its ground track progressing roughly 400 km in a minute. At this speed, it would be impossible for the outer loop to observe four targets with the given actuator limits, and even two of them posed a challenge to the PID. However, if the outer loop follows an approximate center line through the country, the inner loop can switch rapidly from one target to another, with the only limitation being that they can be no further than 2.5 degrees off track. The simulation is initialized from steady-state, 2 minutes after the dual-tracking mode's start time, right above the maritime boundary of the country. Switches between the targets are made at 18, 30, and 43 seconds after the start time. The roll angle is kept at a constant 1 degree, while the pitch angle decreases linearly from 4 to -1 degrees.

Figure 4.24: Multi-point tracking trajectory. Ground track in cyan, outer loop in red, inner loop in blue.

Each switch takes about 3 seconds to complete, this length was found to result in the best tracking performance. The inner pitch loop is starting to saturate its sensors at some points in the scenario, but this is expected and did not seem to cause any issues (fig. 4.25). The overall pointing performance is worse than in the dual-tracking scenario, with the MPE settling around 10-15 arcsec error after 3-5 seconds (fig. 4.26). Surprisingly, performance is almost the exact same for all three outer loops, indicating that the reference generator does not pose any large demands for the controllers. This translates to a ground error of 50 meters, about the size of a large building. Note that the settling time delay corresponds to some short-term oscillations in the ground error which last half the imaging time. This worsened performance is expected, considering the much harsher demands from the inner loop compared to the dual-tracking case. In conclusion, the best candidate for the cascade control would be some combination of an approximative, but not linear outer loop where the operational range of the isolator is reduced to a few hundred arc seconds.

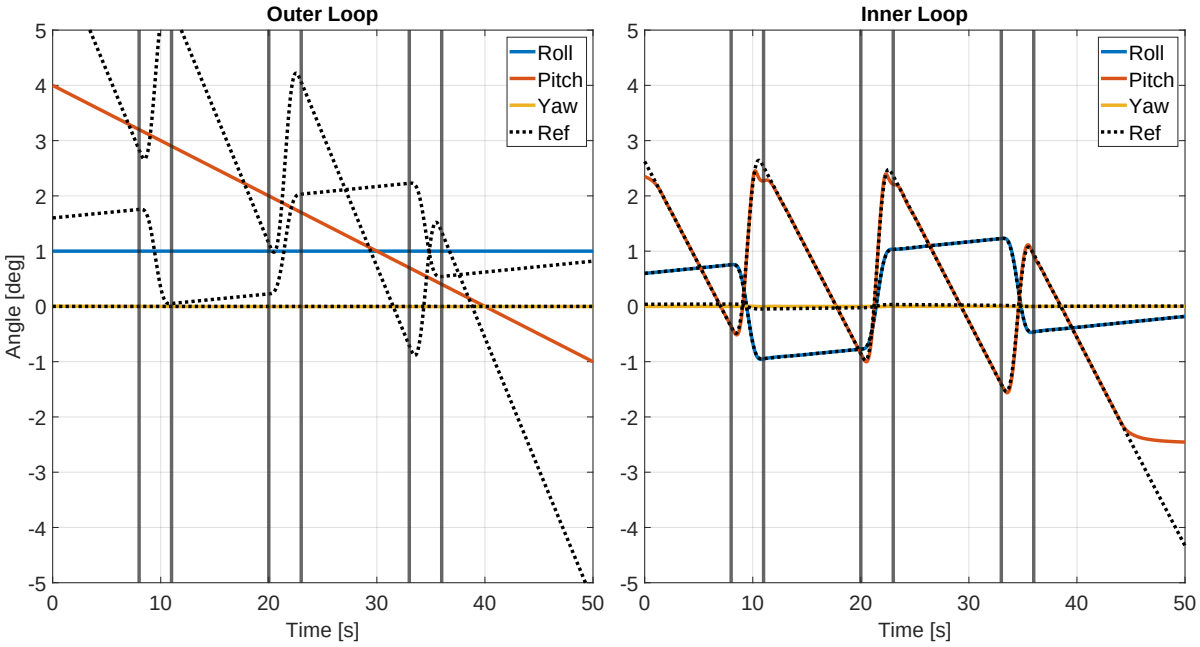

Figure 4.25: Simulation results in multi-point tracking scenario. Target switches in grey.

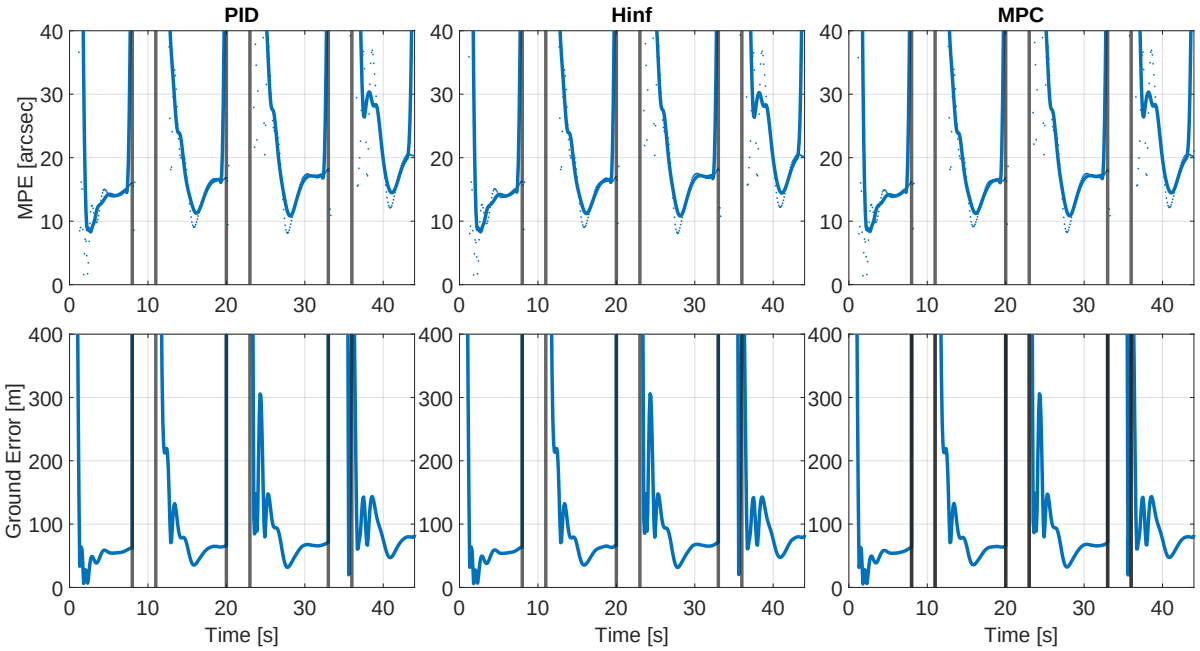

Figure 4.26: Pointing performance of cascade control loop for multi-point tracking. Target switches in grey.

5

Verification and Validation

The main obstacle in the implementation of a precise pointing control system for flexible spacecraft is the difficulty of verification and validation (V&V). This usually covers the estimation of uncertain parameters and dynamics of the system and guaranteeing correct functionality under different scenarios. There is an intensive ongoing research effort to solve the problems arising from this uncertainty, which tends to be larger in space-bound vehicles due to the differences in the gravity environment compared to a ground-based test chamber. This chapter presents two approaches to validate the three designed control loops based on the research done during the literature study, which answered the following questions:

- How to evaluate and validate the model, taking uncertainty into account?
  - What are the most modern approaches to nonlinear Monte Carlo simulations of flexible spacecraft?
  - How is control performance evaluated in probabilistic analysis?

The first one is a linear analysis to identify worst-case combinations of uncertain parameters using analytical tools, then evaluating the results in the nonlinear simulation environment. The second one also involves model uncertainty in a multi-frequency worst-case analysis. This method combines time-domain and frequency-domain performance indicators in order to validate the system faster than conventional Monte Carlo methods with higher confidence in the final results.

5.1. Parameter uncertainty analysis

One way of verification and validation is to use μ-analysis techniques, which rely on advanced approaches to LFT modelling (linear fractional transformation) and a significant effort in model reduction. However, even with reduction, these methods generate high-order models, which severely complicate the process of verification. As a result, it is standard practice in the aerospace industry (in Europe, as of 2010) to use Monte Carlo simulations to test the performance of a robust controller. This method essentially evaluates a large number of simulations with random dynamics, in order to provide statistical guarantees for the worst-case behaviour of the closed-loop system. However, while large iterations (approx. 1000) would be required for a hard proof, typical simulation numbers are much lower (around 300) due to computational costs, which can render the results of the analysis unreliable [19][20].
To give a ballpark estimate, the current runtime of a single dual-pointing simulation is roughly 100 seconds on a high-end processor manufactured in 2020. To reach any significant statistical result from a purely random sampling of uncertain parameters, several days' worth of constant simulations would be required. High-fidelity models in the industry suffer from similar constraints. As a result, Wang et al developed a hybrid optimization algorithm in the early 2010s to analyze robustness in flexible satellites. Besides the large computation effort, there were other serious shortcomings identified in Monte Carlo-based methods, largely involving uncertain high-frequency flexible modes, which further motivated their commitment to exploring optimization methods. In 2012, the V&V methods developed by Gasbarri et al also came to the same conclusions [2].

Modelling for uncertainty

In all of these verification campaigns, four parameters were considered to be uncertain. The primary ones are the frequency and damping of the flexible modes of the spacecraft structure, and the secondary ones are the inertia and overall mass of the vehicle. To limit the complexity of the analysis, some assumptions are made. First, the uncertainty of frequency and damping ratio is controlled by two multiplicators with 5% and 15% range respectively, which were defined according to the aerospace verification practices of SZTAKI (the host institute for this thesis). Second, the contribution of uncertainty in the inertia and mass of the system is assumed to be largely caused by the rotation angle of the solar panel and the filling ratio of the propellant tank. For the tank, the range is set up to cover any possible options between empty and full. For the panel, the rotation is constrained between 0 and 90 degrees, which sufficiently covers all options due to the symmetry of the flexible model. These values are summarized in table 5.1, with their nominal values used during the setup of the simulator [22].

Parameter	Nominal value	Range of uncertainty
Tank mass multiplier	0.5	0-1
Panel rotation angle	0	0 - 90 deg
Panel damping ratio multiplier	1	0.85-1.15
Panel frequency multiplier	1	0.95-1.05

Table 5.1: Uncertain parameters in the linear analysis.

For each simulation run, the relevant parameters and models of the spacecraft model are re-initialized with the controllers left the same. The calculations of the start sequence are decoupled in a way that this can be done in a fast and efficient manner, with no unnecessary steps taken (i.e. redefinition of initial conditions in orbital dynamics or rebuilding trajectories).

Frequency-domain analysis

To verify the designed controllers, they need to be robust concerning parameter uncertainty. The targets of this analysis are the four variables defined in the previous section. First, their effect is individually assessed on system stability using Bode diagrams and pole-zero plots, where the most important performance indicators are the frequency-domain stability margins and the real distance of the poles from the half-plane of instability. It was found that the panel rotation angle is markedly different from the other three parameters since it provides a way for flexible behaviour to be transferred from purely yaw dynamics to purely roll dynamics (and anything in between). This shows up as large dips on the Bode diagrams of both rotation axes at the vibration frequencies.
This pointed out two optimization strategies in finding the worst-case parameter set: one for purely yaw disturbances (no rotation, "Param 1"), and another for purely roll disturbances (fully rotated, "Param 2"). The remaining three parameter ranges are then sliced up with 10% intervals to form two cubes of potential parameter sets, one for each rotation angle. Then, all of the cases are analyzed in an exhaustive search to identify the case with the worst characteristics in phase margin, gain margin, and pole locations. These were found to be at the edges, as presented in table 5.2.

Parameter	Param 1	Param 2
Tank mass multiplier	0	0
Panel rotation angle (deg)	0	90
Panel damping ratio multiplier	0.85	0.85
Panel frequency multiplier	1.05	1.05

Table 5.2: Worst-case parameter sets for parameter uncertainty analysis.

Two figures showcase the effect of uncertainty on the PID loop's performance. Figure 5.1 presents Bode diagrams and pole-zero maps for both the yaw and the roll responses. Several systems are shown on the graphs at the same time, transforming the nominal case parameter set (blue) into the worst-case parameter set (red) in a linear manner. The main feature is the gradual loss of phase margin at the vibration frequencies as system dynamics move away from the nominal case, visible as a deepening dip on the phase diagrams. There is also a noticeable migration of the closer to the imaginary axis.

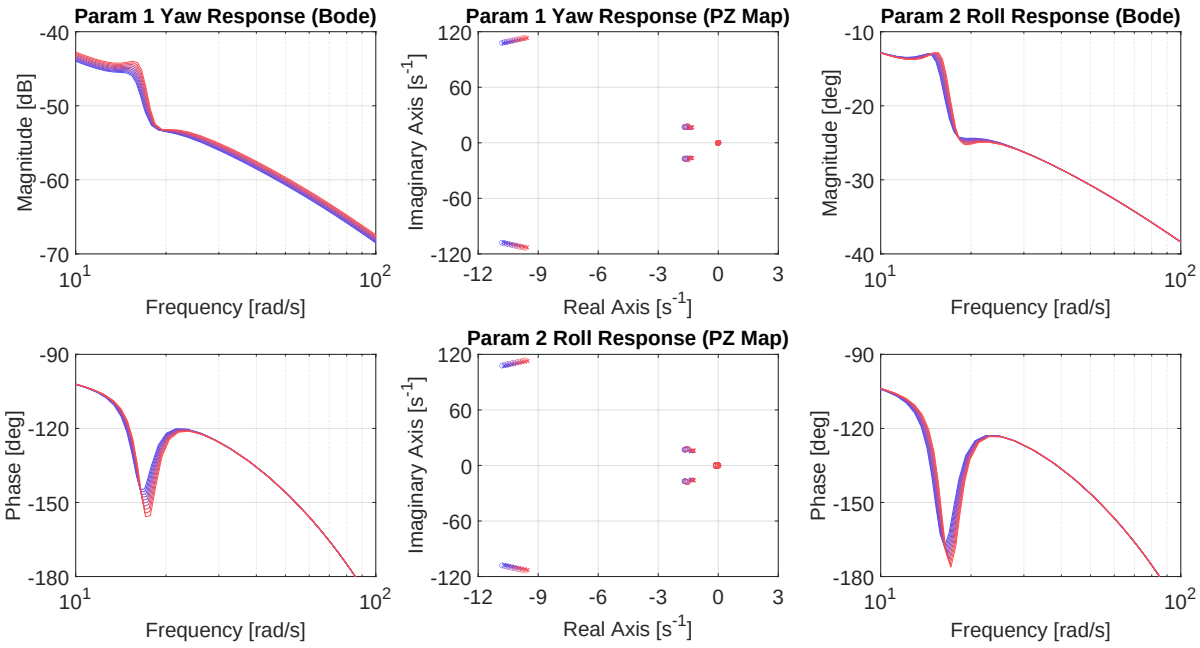

Figure 5.1: The effect of uncertainty on linearized yaw and roll dynamics for parameter sets between nominal (blue) and worst-case (red)

Time-domain analysis

Regarding performance in the nonlinear simulation, little difference was found between the three cases, despite the controller retaining its tuning. Table 5.3 summarizes the results in terms of pointing performance indicators, which would prove that all control loops are sufficiently resilient to parameter uncertainty.

Unfortunately, there are also some major issues in the nonlinear simulation results (figs. 5.2, 5.3). It is hard to miss that the worst-case systems actually perform better with the nominal controllers than the nominal systems themselves. While only finding the edges of a parameter set in a semi-random analysis is expected, the edge cases performing better than the nominal system is unusual. This is likely caused by two factors. First, the controller tuning was largely done manually, which means that the best possible performance with a given controller might not actually be achieved with the system it was designed for. Second, there is an insufficient degree of uncertainty in the spacecraft model with the current parameter set, and other sources of uncertainty should be investigated to validate the designed controllers.

Controller	APE_{avg} [arcsec]	MPE_{min} [arcsec]	MPE_{max} [arcsec]	RPE_{avg} [arcsec]
Nominal PID	1.5038	0.4167	6.4843	0.5524
PID - WC1	1.2341	0.3571	5.7280	0.4088
PID - WC2	1.2248	0.3578	5.6204	0.4145
Nominal H_∞	0.9609	0.4811	1.3788	0.1220
H_∞ - WC1	0.9267	0.5029	1.3130	0.1200
H_∞ - WC2	0.9642	0.5100	1.3689	0.1207
Nominal MPC	1.5771	1.2365	1.8301	0.0903
MPC - WC1	1.5568	1.1437	1.8310	0.0895
MPC - WC2	1.5689	1.1504	1.8623	0.0896

Table 5.3: Pointing performance error indices for parameter uncertainty analysis.

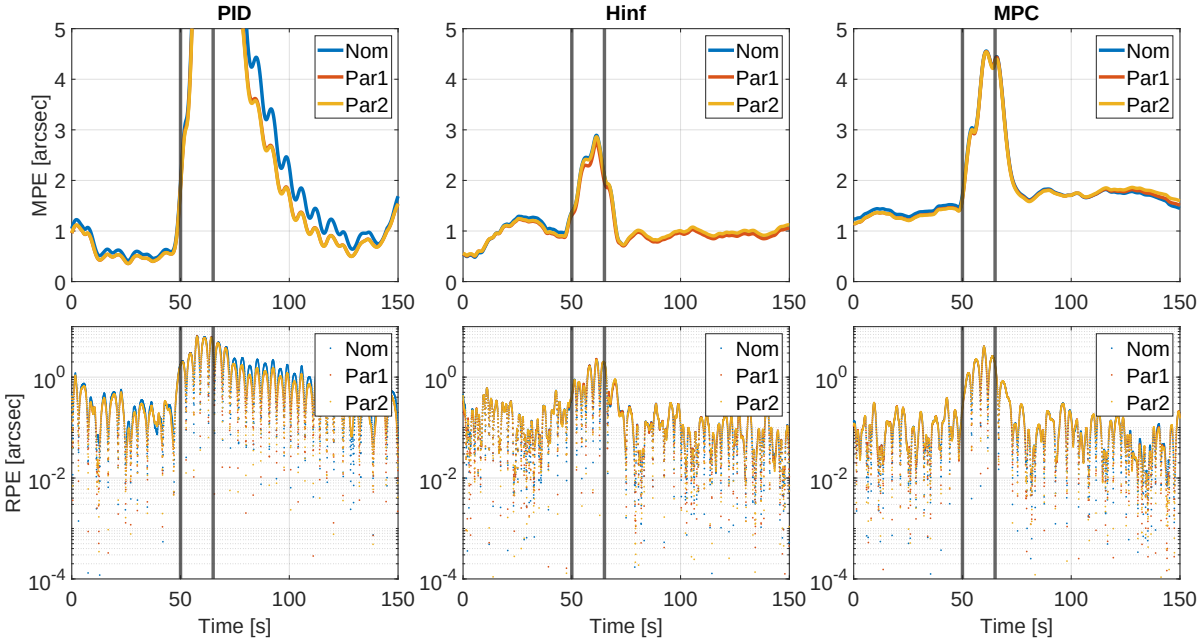

Figure 5.2: Pointing performance indicators of the three control loops in the nominal case and the two worst cases.

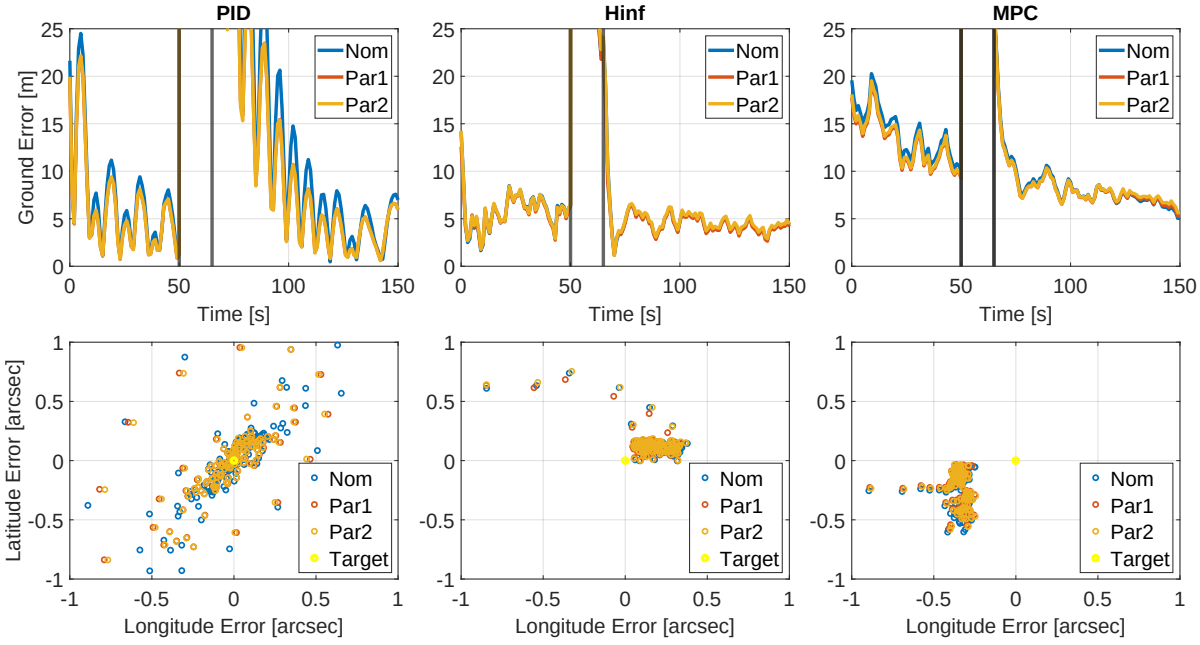

Figure 5.3: Ground error of the three control loops in the nominal case and the two worst cases.

5.2. Validation using worst-case analysis

To expand the capabilities of the V&V process without resorting to a Monte Carlo campaign whose drawbacks have been detailed in the previous sections, a worst-case uncertainty construction method is used that was developed by Patartics et al. It maximizes the gain of a system at multiple user-defined frequencies and is currently applied in the analysis of flutter suppression in flexible aircraft [23][49]. Consider the block diagram in figure 5.4, with the stable MIMO LTI system $\mathbf{M}(s)$ and its corresponding MIMO uncertainty block $\mathbf{\Delta}(s)$. There, $\mathbf{M}(s)$ is an augmented version of the nominal system model with widened output and input channels to tie the effect of uncertainty into the model. The exact description of how this uncertainty is acting in the system dynamics is described by $\mathbf{\Delta}(s)$, which has two components.

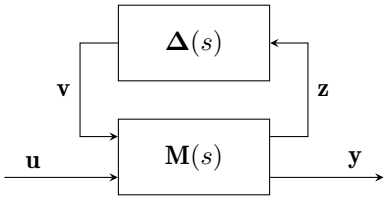

Figure 5.4: Interconnection of an uncertain system for upper LFT.

The first one is parametric uncertainty, which is the main type of uncertainty introduced to the system so far. It describes deviations in the value of certain physical parameters. The second one is dynamic uncertainty, which represents unmodelled behaviour in the system. Examples include high-frequency vibrations or residual dynamics from model reduction. The most important criterion is that uncertain dynamics must be stable and unit norm-bounded, but they can be weighted to adjust their magnitude. Then, similarly to the lower LFT defined in the ADCS chapter, partition the system model so that the upper left block has compatible dimensions with the uncertainty block:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \tag{5.1}$$

The upper LFT of $\mathbf{M}$ and $\mathbf{\Delta}$ is then defined as:

$$\mathbf{F}_u\left(\mathbf{M}, \mathbf{\Delta}\right) = \mathbf{M}_{22} + \mathbf{M}_{21}\mathbf{\Delta}\left(\mathbf{I} - \mathbf{M}_{11}\mathbf{\Delta}\right)^{-1}\mathbf{M}_{12} \tag{5.2}$$

With the overall uncertain system being:

$$\mathbf{P}\left(\mathbf{\Delta}, s\right) = \mathbf{F}_u\left(\mathbf{M}(s), \mathbf{\Delta}(s)\right) \tag{5.3}$$

Now, assume the following: $\mathbf{\Delta}(s) \in \mathbb{\Delta}$, $\mathbf{P}\left(\mathbf{\Delta}, s\right)$ is robustly stable (i.e. stable for $\forall \mathbf{\Delta}(s) \in \mathbb{\Delta}$), given are a set frequencies $[\omega_k]_{k=1}^{N_\omega}$ and worst-case gain lower bounds $[L_k]_{k=1}^{N_\omega}$. Then, define the function $J : \mathbb{\Delta} \to \mathbb{R}$ that it is the sum of gains at the given frequencies:

$$J\left(\mathbf{\Delta}(s)\right) = \sum_{k=1}^{N_\omega} \bar{\sigma}\left(\mathbf{F}_u\left(\mathbf{M}(j\omega_k), \mathbf{\Delta}(j\omega_k)\right)\right) \tag{5.4}$$

The worst-case uncertainty sample construction problem is to find $\mathbf{\Delta}_m(s) \in \mathbb{\Delta}$ for which $J\left(\mathbf{\Delta}_m(s)\right)$ is maximized. The function J has a theoretical upper bound, which is the sum of the worst-case gain lower bounds at the given frequencies:

$$J_U = \sum_{k=1}^{N_\omega} L_k, \quad J(\mathbf{\Delta}(s)) \leq J_U, \quad \forall \mathbf{\Delta}(s) \in \mathbb{\Delta} \tag{5.5}$$

If the system is only subjected to dynamic uncertainty, then it is possible to find a $\mathbf{\Delta}_m(s)$ that $J(\mathbf{\Delta}_m(s)) = J_U$, based on the boundary Nevanlinna-Pick interpolation [50]. However, in the case of mixed uncertainty with both dynamic and parametric components, it is generally not possible to find such a $\mathbf{\Delta}_m(s)$, because parametric uncertainty couples the frequencies together and different values of the same parameter may be required to achieve the lower bound at each frequency.
The worst-case uncertainty construction method applied in this section addresses these issues by using a nonlinear optimization method that performs an efficient search of the full range of parametric uncertainty as well as a boundary Nevanlinna-Pick interpolation of dynamic uncertainty [23][49].

System analysis

Analyzing the performance of a closed-loop control system with the worst-case uncertainty construction method is an iterative process which offers increased speed and reliability compared to random sampling-based Monte Carlo methods. First, an uncertain model is constructed (to be detailed in the next section) with both parameter and dynamic uncertainty. Then, the uncertain transfer function of the cascade loop from the 3 reference angles to the 6 pointing angles is converted to an LFT model that is passed

to the optimization script along with two parameter sets: the relevant frequencies and the input-output pair where the effect of uncertainty should be maximized. The selected channel can either correspond to reference tracking or disturbance rejection.
The result is an exact parametrization of the uncertain block, which is used to update the high-fidelity nonlinear simulation environment and evaluate pointing performance. To speed up the process, the most interesting frequencies are selected iteratively by first checking the step-tracking characteristics of the adjusted closed-loop and shifting the set of frequencies as needed. Usually, most pessimistic models are reached when the frequencies are placed in the peaks and inflection points of the Bode magnitude plot. A disadvantage of the method is that controllers without an LTI model (MPC) can not be evaluated.

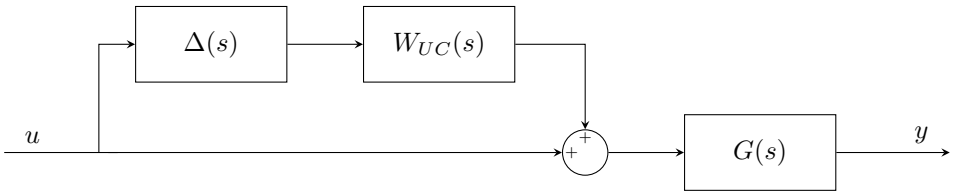

Figure 5.5: Block diagram of dynamic uncertainty applied to some plant G(s).

Modelling for uncertainty

Considering the varied dynamics of reaction wheels, it was determined that their model should be adjusted with uncertain dynamics (fig. 5.5). First, $\Delta(s)$ is created with Matlab's `ultidyn` function which generates a fourth-order LTI model with a gain bound of 1. Then, a filtering function $W_{UC}(s)$ is defined that produces a roughly 1 dB variance in the DC gain and a 5 dB/decade variance in the rolloff of the reaction wheel model (fig. 5.6):

$$W_{UC}(s) = \frac{1.89s + 100}{3.78s + 1000} \tag{5.6}$$

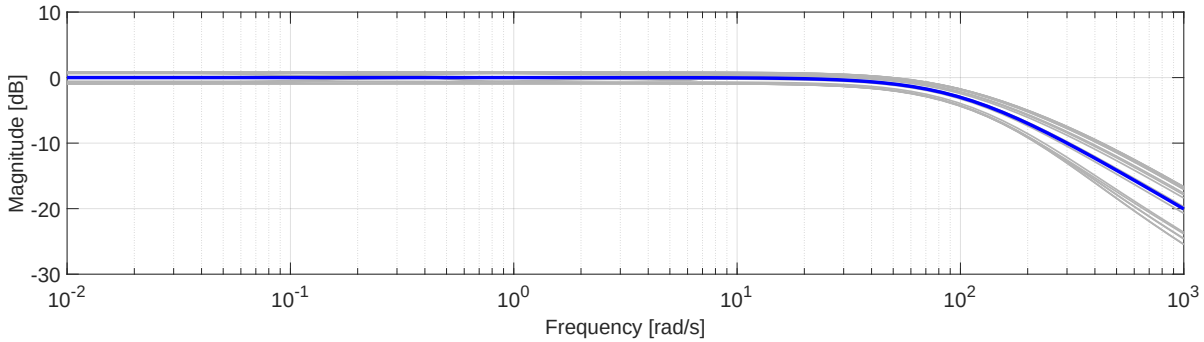

Figure 5.6: Bode magnitude plot of uncertain reaction wheel model. Nominal in blue, samples in grey.

Finally, the previous parameter sets are slightly shrunk to improve the granularity of the search, as it was found that the method tends to find solutions away from the edges (unlike the previous analysis). The ranges for the damping ratio and frequency are still in line with SZTAKI's usual assumptions, and the rotation angle was calculated based on the uncertainty in the exact rotation angle of the panel.

Parameter	Value	Range
Tank mass multiplier	0.5	0-1
Panel rotation angle	0	-0.2 - 0.2 rad
Panel damping ratio multiplier	1	0.90-1.10
Panel frequency multiplier	1	0.98-1.02

Table 5.4: Uncertain parameters for multifrequency worst-case uncertainty construction.

PID validation

The PID loop has proven itself to be robust enough against uncertainty in the multi-frequency analysis. Note that the Bode plot for Yaw dynamics only shows the isolator response, as the Yaw loop is too slow to show up on the chart. Using the method, the magnitude of disturbances on the roll and pitch outputs increased by 15 dB, which translates to a ten-fold time-domain rise in the oscillations caused by cross-coupling. Yaw dynamics were also perturbed to a significant degree, reaching up to 2.5 degrees of error in response to a roll or pitch reference. While step-tracking was not affected much for roll, the settling time and damping ratio for the pitch were significantly higher in the worst case. From all plots, it is clearly visible that the worst cases found by the method (in red) produce far worse performance than random samples (in grey), further motivating the use of the algorithm.

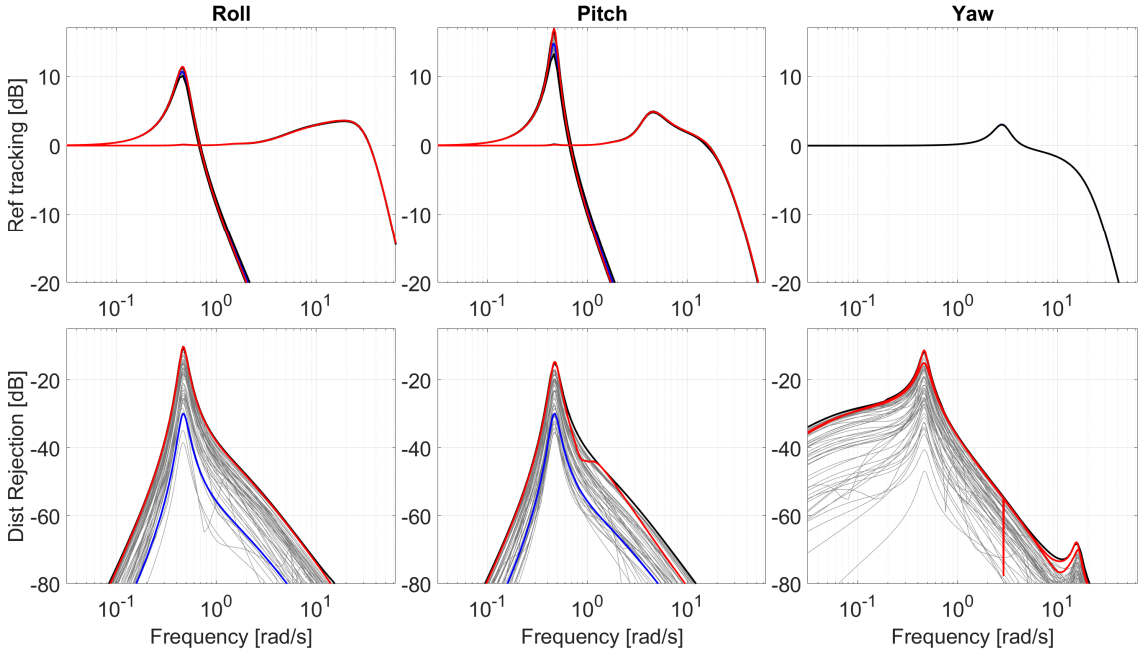

Figure 5.7: Bode magnitude plots of the cascade PID loop in the nominal (blue), worst (red), and random cases (grey).

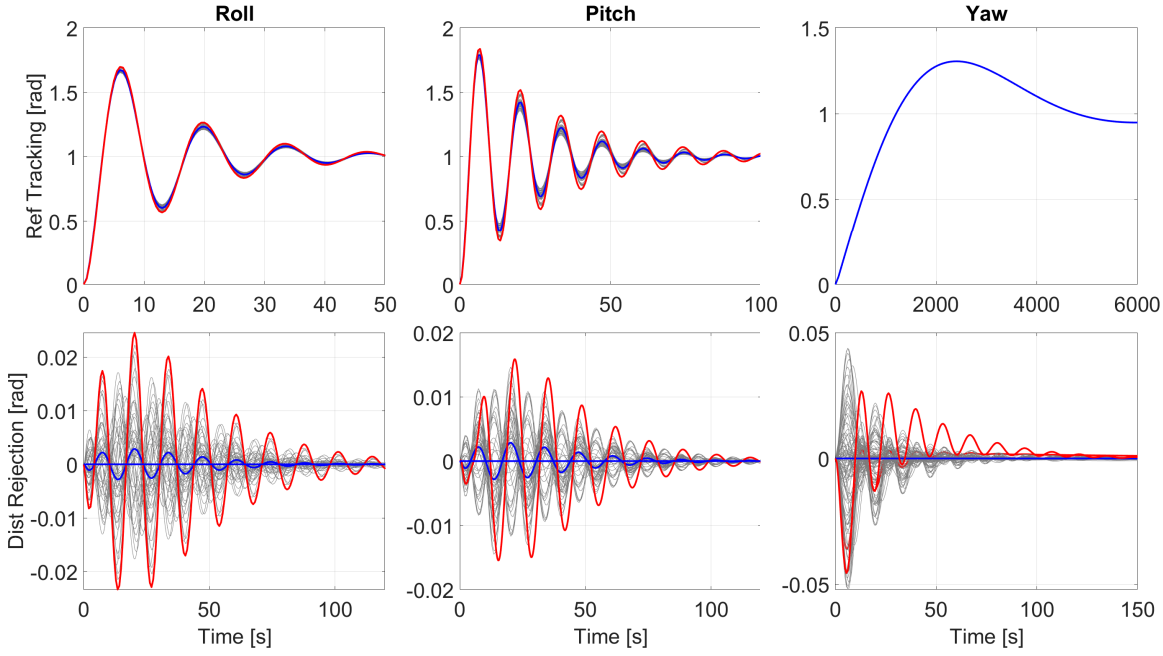

Figure 5.8: Step response of the PID outer loop in the nominal (blue), worst (red), and random cases (grey).

In the nonlinear environment, some of the cases produce similar, slightly improved results as seen in the parameter uncertainty analysis. While the isolator loop did not produce any bad responses, the outer pitch loop is very sensitive to worst cases optimized for any combinations except for $\theta_{ref} - \theta_{isol}$. There is an increase of about 50% in average APE, minimum MPE, and average RPE, even in reference tracking. This is the main contributor to the increase in outer loop errors seen in the plots below.

Worst cases	APE_{avg} [%]	MPE_{min} [%]	MPE_{max} [%]	RPE_{avg} [%]
ϕ	5.7692	36.4967	13.3539	-16.8891
ϕ_i	-7.3312	-5.1719	5.5011	-27.84
θ	15.4798	11.463	17.0646	6.8491
θ_i	9.5391	9.6623	10.5315	1.7747
ψ	-8.3419	-15.5535	4.3249	-26.8561
ϕ	3.4734	2.2609	-0.5763	2.3426
ϕ_i	3.0224	6.4758	0.4347	5.2049
θ	1.716	-6.4621	-1.3918	2.4789
θ_i	2.3448	-0.3614	-1.1395	0.7831
ψ	4.1801	-0.4102	0.355	3.5919

Table 5.5: Worst-case pointing indices of output channels for Roll tracking in the outer PID and inner H_∞ loops.

Worst cases	APE_{avg} [%]	MPE_{min} [%]	MPE_{max} [%]	RPE_{avg} [%]
ϕ	46.1561	44.6434	23.747	63.8876
ϕ_i	46.7421	40.3529	23.1732	66.7845
θ	48.313	52.6568	26.2055	58.2175
θ_i	-21.0268	4.9021	-20.0858	-28.4494
ψ	46.9345	43.742	23.2947	66.2167
ϕ	6.9861	1.677	-1.8676	6.184
ϕ_i	5.691	-4.4375	-3.2708	6.6777
θ	9.7447	6.4081	-1.7015	6.841
θ_i	2.1466	5.1227	-1.6103	8.981
ψ	5.6184	-4.1579	-2.3021	9.3367

Table 5.6: Worst-case pointing indices of output channels for Pitch tracking in the outer PID and inner H_∞ loops.

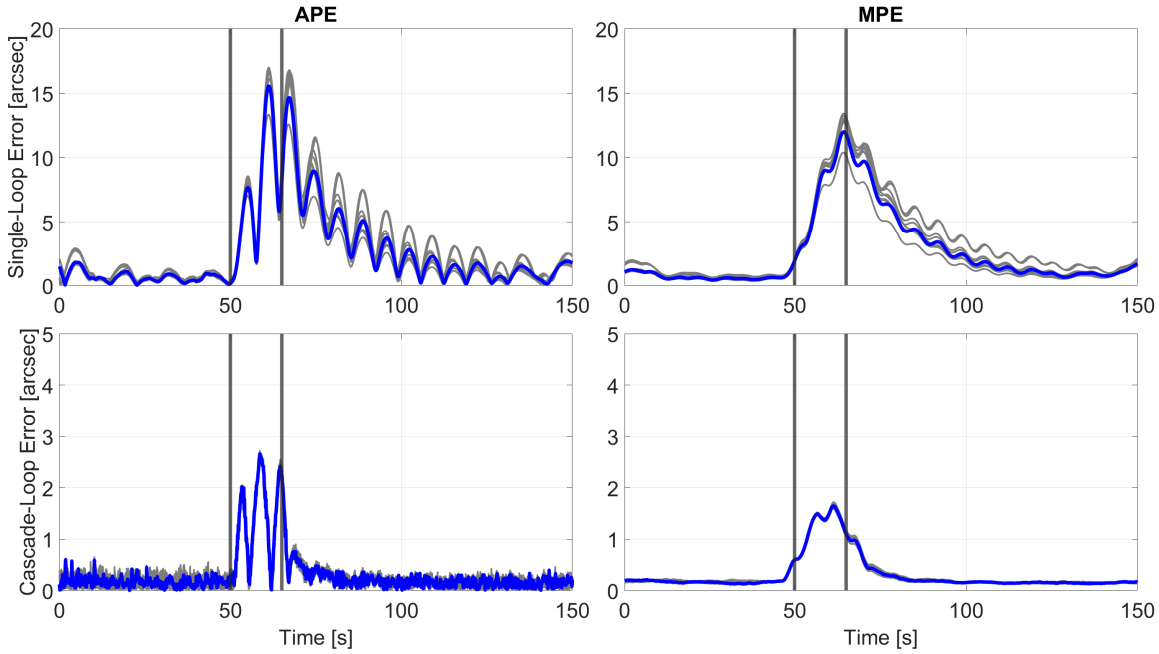

Figure 5.9: Simulation results for worst cases (grey) and nominal case (blue) with the cascade PID loop.

H_∞ validation

The H_∞ loop has also proven itself to be robust enough against uncertainty in the multi-frequency analysis. Using the method, the magnitude of disturbances on the roll and pitch outputs increased by 35 dB and 20 dB, respectively. This translates to a major time-domain rise in the oscillations in all axes, which still stay lower than the PID loop's. Step tracking is slightly affected in the form of some lightly dampened oscillations in the roll and pitch loops. As with the PID, it is clearly visible on the plots that the worst cases found by the method (in red) produce far worse performance than random samples (in grey), further motivating the use of the algorithm. One exception would be the disturbance rejection of the yaw loop, which was optimized for settling time instead of overshoot. This was a deliberate choice, as long-term errors have a larger impact on dynamics compared to transients.

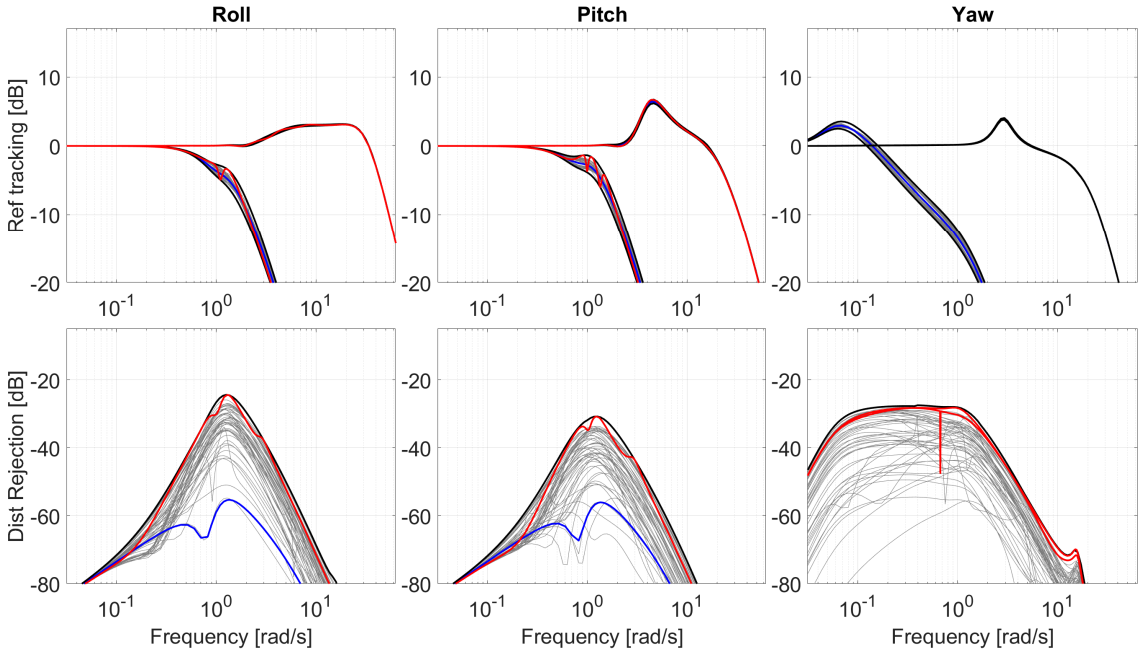

Figure 5.10: Bode magnitude plots of the cascade H_∞ loop in nominal (blue), worst (red), and random cases (grey).

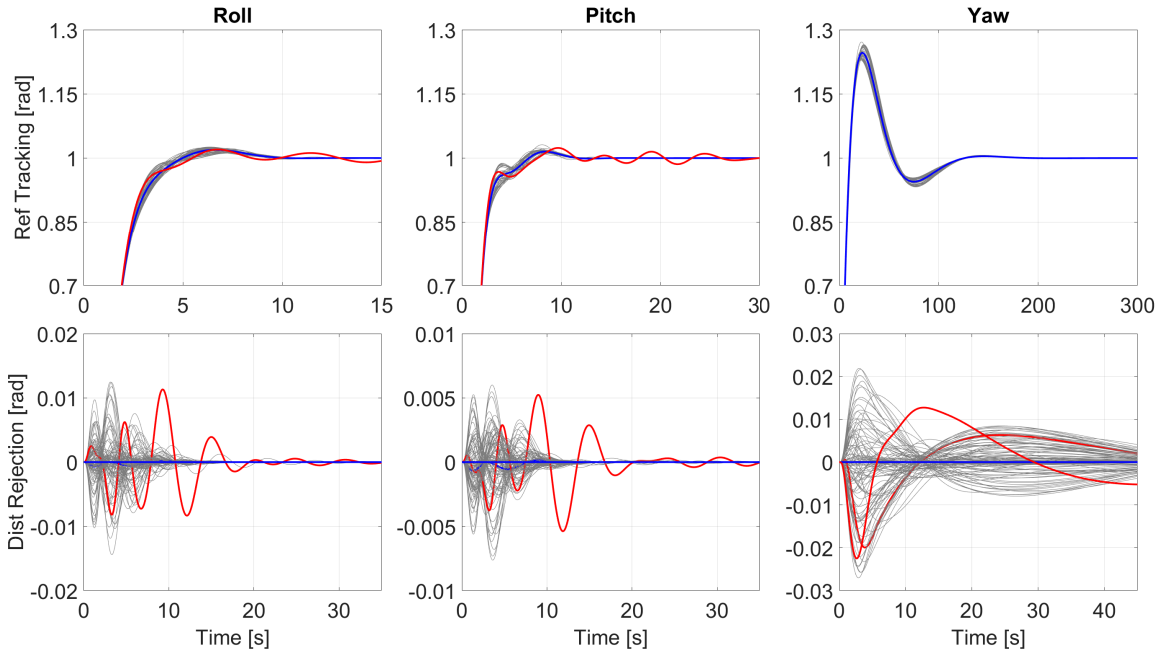

Figure 5.11: Step response of the H_∞ outer loop in the nominal (blue), worst (red), and random cases (grey).

In the nonlinear environment, most cases produce similar, slightly improved results as seen in the parameter uncertainty analysis. Unlike the PID, both loops have very low susceptibility to uncertainty, highlighting the importance of robust control. The most significant effect is perhaps a 29% increase in APE and 10% in RPE for roll reference tracking, and the effect of uncertainty seems to be diminishing with time as the controller brings system dynamics close to nominal for all cases, even with large initial errors.

Worst cases	APE_{avg} [%]	MPE_{min} [%]	MPE_{max} [%]	RPE_{avg} [%]
ϕ	29.1943	-6.749	77.511	10.5943
ϕ_i	-5.4575	-2.6953	-6.3912	-0.3264
θ	6.1577	-2.3725	9.8517	-0.5488
θ_i	8.315	-5.8852	15.7487	-0.7847
ψ	-17.2412	-30.6985	-5.6572	-10.1902
ϕ	-4.0103	-0.7906	2.9481	-5.4492
ϕ_i	4.5953	2.1361	0.1395	5.2307
θ	-2.8671	-4.93	-0.6716	-7.1753
θ_i	-1.1565	-7.8672	0.6591	-2.6894
ψ	-1.7688	-4.0452	2.3889	-2.4326

Table 5.7: Worst-case pointing indices of output channels for Roll tracking in the outer H_∞ and inner H_∞ loops.

Worst cases	APE_{avg} [%]	MPE_{min} [%]	MPE_{max} [%]	RPE_{avg} [%]
ϕ	7.3854	12.127	11.7233	0.6467
ϕ_i	6.3038	3.2714	10.4044	0.8901
θ	-1.2012	39.8143	-1.6015	3.2027
θ_i	-6.0763	-7.4852	-3.5371	-2.8062
ψ	17.7994	2.0899	40.7737	1.7107
ϕ	-2.978	-0.8368	-3.6655	-5.6737
ϕ_i	1.5117	1.1538	0.5031	1.9138
θ	-5.046	-0.111	-5.2367	-4.6331
θ_i	1.2504	1.2171	1.1474	2.5651
ψ	-2.792	-7.1414	0.8685	-4.3276

Table 5.8: Worst-case pointing indices of output channels for Pitch tracking in the outer H_∞ and inner H_∞ loops.

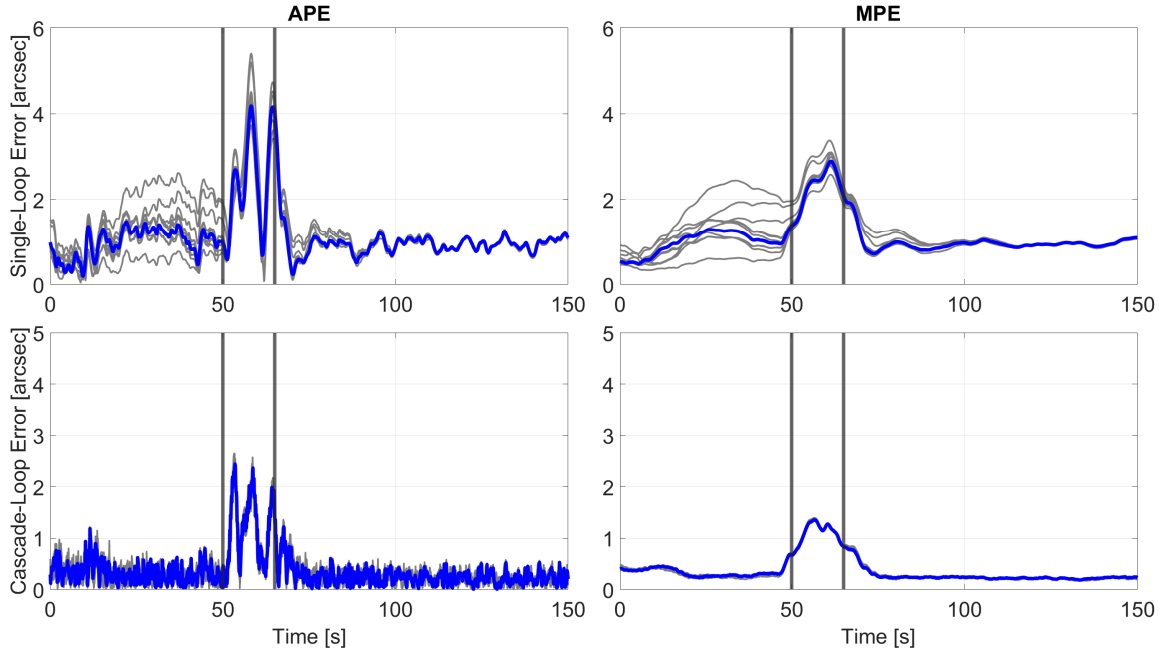

Figure 5.12: Simulation results for worst cases (grey) and nominal case (blue) with the cascade H_∞ loop.

6

Conclusion

The main research question of the thesis was: How can the pointing precision of flexible spacecraft be improved, compared to the current state of the art? Before it can be answered, it would be useful to go over the general conclusions of the project.

First off, it was determined that even in the case of weak structural dynamics, single-loop control design has to take the flexible modes into account. Despite not being able to actively dampen these disturbances due to the limitations of actuator suites, it is of utmost importance that the structural dynamics of the system are not excited by the ADCS loop. In this, all three tested controllers have proven themselves to be sufficiently robust with lower than 2 arcsecs of absolute pointing error. In both single and dual-tracking scenarios, the advanced controllers showed similar performance characteristics in terms of pointing indices and ground-tracking errors. While the MPC showed greater pointing stability, it was prone to residual errors in APE which led to an offset in the simulated camera images. The H_∞-based controller on the other hand showed consistently good APE and MPE results at the cost of slightly more noise being transmitted to the outputs. While the PID was able to keep up with both advanced controllers in the simple single-point tracking case, its weaknesses showed quickly in the more demanding dual-tracking scenario where it produced significantly worse results, especially in the 30 seconds following the transient phase. This can be attributed to the lower complexity of the PID, along with its slightly lower cutoff frequency in reference tracking.

However, all of these shortcomings were eliminated with the implementation of a cascade control loop. The H_∞-based controller running the active-passive vibration isolation loop of the hexapod pointing platform vastly expanded the reference tracking capabilities of the control loop. Both in terms of controllable frequencies and pointing indices, the cascade loops were superior to single-loop control in all tested scenarios. They rapidly diminished any residual error and noise in the outer control loops and even made it possible to speed up the trajectory generator and expand its operations to several tracked ground points. This was done by offloading control requirements from the outer loop to the isolation loop, leading to an overall more relaxed ADCS which was much less susceptible to inducing microvibrations, which would have led to a degradation of pointing characteristics.

This also highlights the importance of reference angle generation. The use of smoothing algorithms and quaternion operations massively improved the chances of a simple controller performing demanding maneuvers, but it quickly became evident that beyond 1-2 targets in a small country such as the Netherlands, a dedicated pointing system is required on any Earth-observing spacecraft. As a result, it is not sufficient to rely on simple step-tracking scenarios when evaluating control performance, as a more detailed look into the real-world use cases of pointing systems can eliminate many desirable candidates for control.

This is especially true for verification & validation efforts, which often rely heavily on frequency-domain analysis and only a random sampling of time-domain behaviour. With the addition of realistic reference tracking scenarios, high-fidelity nonlinear modelling, and standardized pointing indices, the time domain turned out to be a far more important indicator for the robustness of a control system to uncertainty. This is especially true for mixed uncertainty problems as a combination of parametric and dynamic uncertainty, where Monte Carlo consistently fails to find the worst-case models. This means that V&V processes must be augmented with a heavily optimized methodology that relies on the

evaluation of time-domain performance besides frequency-domain reference tracking and disturbance rejection characteristics. In this project, this was solved by the multi-frequency worst-case uncertainty construction method presented by Patartics et al, which resulted in a more thorough identification of the boundaries of pointing characteristics than what would have been possible with a significantly more time-intensive Monte Carlo campaign.
In the literature study that was carried out prior to this thesis, plenty of articles were identified that were dealing with complex problems in high-fidelity modelling, advanced control, or V&V techniques for flexible spacecraft, however most prominent publications only focus on one or maybe two of these fields. As a result, there can be significant blind spots in their conclusions due to the complexity of the precise pointing problem of these systems. The combination of all three fields yields a robust framework for the control engineering of flexible space structures, whose first iteration was developed in this thesis.
In conclusion, to improve the pointing performance of flexible spacecraft, there must be more synergy between modelling, validation, and control efforts. A cascade control system is an absolute necessity that offloads most of the control effort to an active-passive isolation loop, and various reference angle generation methods should be investigated and optimized in order to bring the most performance out of the designed control loops.

6.1. Critical review and recommendations

Of course, in case of a complex design problem, the work is never fully done. There are several points in the development process where work was cut short at a satisfactory level in order to bring the project to the finish line. This section will go over each of these points to provide an explanation for some of the decisions and explore ideas for future work.

Structural modelling

While the selected spacecraft model (Sentinel-2) was a good choice due to the vast availability of documentation on its systems and operations, its structural dynamics did not have the desired strength that would risk destabilising the system. There were some experiments done to adjust the parameters of the solar panel to emphasize its flexible characteristics, but they were still insufficient in causing critical disturbances. At this point, the outer loop controllers were already tuned, so it was decided that instead of a redo of the previous work packages, more effort should go into future-proofing the simulation environment. This led to a redesign of the structural dynamics implementation that allows for easy exchange of models in the future. There is an ongoing effort at SZTAKI to develop a vastly more flexible FEM model of an interplanetary spacecraft which will eventually be tested in the Simulink environment.
However, it should be pointed out that the lack of strong vibrations does not equal an easy control and validation problem. The flexible modes are still there, and they can still be excited to levels that are detrimental to pointing precision. In essence, the current thesis is more about building a robust, future-proof control design and verification environment than proposing specific controllers for a specific mission.

Control design

After the initial mathematical setup of the controllers, parameter tuning is a time-intensive process that follows the Pareto principle: 80% of the results stem from 20% of the effort. At the start of the thesis, it was decided to place an arbitrary limit as the pointing requirement for all controllers: 2 arcsec. The reasoning goes that the current star tracker model can only reliably measure angles with 1 arcsec accuracy, so a doubling of this error is an acceptable result for pointing systems. As a result, controllers are not as optimal as they can possibly be. Additionally, the cascade control loop would be better tuned in a single step, using robust control in a structured way with the process focusing on the sensitivity and complementary sensitivity matrices, as well as the pointing index weighing functions introduced by Ott and Pittelkau. Each new controller tuned in the future will follow these guidelines.

State estimation

A major shortcoming of the current closed loop is the lack of a reliable state estimator. Properly setting up a Kalman filter in the nonlinear environment would have gone too far beyond the scope of the thesis, and the ”simulator-based” estimator was implemented as a backup. There is no significant

loss in absolute pointing performance, but pointing stability could be vastly improved with Kalman filtering. Additionally, predictive control suffered from greatly increased pointing errors due to relying on its internal estimator algorithm. As a result, besides implementing a properly flexible FEM model in the structural dynamics, the next most important future task is the implementation of a Kalman filter.

Isolation platform

A future work package should deal with the high-fidelity modelling and proper design of the Stewart platform, complete with a sensor suite and accurate models for the noise processes. The current one is a good approximation of the real hardware and its parameters have a strong basis in empirical research, but it is not yet reliable enough to move any designed controllers into a hardware-in-the-loop environment.

6.2. Outlook

With the conclusion of this thesis, work does not stop on the project. The already heavily modular codebase is being further de-hardcoded in preparation for future internship topics and research activities. Some of the plans for industry applications involve carrying out the system identification and ADCS validation campaign of an interplanetary flexible spacecraft or developing AI-based AOCS algorithms using the developed framework.

References

[1] Valentin Preda. "Robust microvibration control and worst-case analysis for high pointing stability space missions". PhD thesis. Université de Bordeaux, 2017.

[2] Paolo Gasbarri et al. "Control-oriented modelization of a satellite with large flexible appendages and use of worst-case analysis to verify robustness to model uncertainties of attitude control". In: *Acta Astronautica* 81.1 (2012), pp. 214–226.

[3] Cornelius J Dennehy. "A survey of reaction wheel disturbance modeling approaches for spacecraft line-of-sight jitter performance analysis". In: *Proc. European Space Mechanisms and Tribology Symp. Munich, Germany*. 2019.

[4] F Landis Markley and John L Crassidis. *Fundamentals of spacecraft attitude determination and control*. Vol. 1286. Springer, 2014.

[5] Andre Preumont. *Vibration control of active structures*. Vol. 2. Springer, 1997.

[6] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: analysis and design*. john Wiley & sons, 2005.

[7] William Singhose. "Command shaping for flexible systems: A review of the first 50 years". In: *International journal of precision engineering and manufacturing* 10 (2009), pp. 153–168.

[8] Bing Xiao, Qinglei Hu, and Youmin Zhang. "Adaptive sliding mode fault tolerant attitude tracking control for flexible spacecraft under actuator saturation". In: *IEEE Transactions on Control Systems Technology* 20.6 (2011), pp. 1605–1612.

[9] T Loquen et al. "Attitude control of satellites with flexible appendages: a structured H_∞ control design". In: *AIAA Guidance, Navigation, and Control Conference*. 2012, p. 4845.

[10] Hyochoong Bang and Choong-Seok Oh. "Predictive control for the attitude maneuver of a flexible spacecraft". In: *Aerospace science and technology* 8.5 (2004), pp. 443–452.

[11] Utku Eren et al. "Model predictive control in aerospace systems: Current state and opportunities". In: *Journal of Guidance, Control, and Dynamics* 40.7 (2017), pp. 1541–1566.

[12] Takanori Iwata, Haruyuki Ishida, and Yuji Osawa. "Advanced Land Observing Satellite (ALOS): enabling technologies and platform performance". In: *Sensors, Systems, and Next-Generation Satellites XII*. Vol. 7106. SPIE. 2008, pp. 113–125.

[13] Marcio AA Fialho and Daniele Mortari. "Theoretical limits of star sensor accuracy". In: *Sensors* 19.24 (2019), p. 5355.

[14] Chunchuan Liu et al. "Recent advances in micro-vibration isolation". In: *Mechanical Systems and Signal Processing* 56 (2015), pp. 55–80.

[15] Valentin Preda et al. "Robust microvibration mitigation and pointing performance analysis for high stability spacecraft". In: *International Journal of Robust and Nonlinear Control* 28.18 (2018), pp. 5688–5716.

[16] John E McInroy, John F O'Brien, and Gregory W Neat. "Precise, fault-tolerant pointing using a Stewart platform". In: *IEEE/ASME transactions on mechatronics* 4.1 (1999), pp. 91–95.

[17] Baolin Wu, Danwei Wang, and Eng Kee Poh. "High precision satellite attitude tracking control via iterative learning control". In: *Journal of Guidance, Control, and Dynamics* 38.3 (2015), pp. 528–534.

[18] A El Hady M Elbeltagy et al. "Fixed ground-target tracking control of satellites using a nonlinear model predictive control". In: (2018).

[19] Wenfei Wang et al. "Verification and validation of attitude and orbit control systems for flexible satellites". In: *AIAA Guidance, Navigation, and Control Conference*. 2009, p. 5953.

[20] W Wang et al. “Robustness analysis of attitude and orbit control systems for flexible satellites”. In: *IET control theory & applications* 4.12 (2010), pp. 2958–2970.

[21] Daniel Alazard and Francesco Sanfedino. “Satellite dynamics toolbox for preliminary design phase”. In: *43rd Annual AAS Guidance and Control Conference*. Vol. 30. 2020, pp. 1461–1472.

[22] János Bezsilla et al. “Parameter Uncertainty Analysis in Precise Pointing Control of Flexible Spacecraft”. In: *IFAC-PapersOnLine* 55.20 (2022), pp. 241–246.

[23] Bálint Patartics et al. “Worst Case Uncertainty Construction via Multifrequency Gain Maximization With Application to Flutter Control”. In: *IEEE Transactions on Control Systems Technology* 31.1 (2022), pp. 155–165.

[24] MathWorks. *About Aerospace Coordinate Systems*. 2022. URL: `https://www.mathworks.com/help/aeroblks/about-aerospace-coordinate-systems.html`.

[25] Aryslan Malik, Troy Henderson, and Richard Prazenica. “Using Products of Exponentials to Define (Draw) Orbits and More”. In: *arXiv preprint arXiv:2203.01182* (2022).

[26] Jawhar Chebbi et al. “Linear dynamics of flexible multibody systems”. In: *Multibody System Dynamics* 41.1 (2017), pp. 75–100.

[27] Francesco Sanfedino. “Experimental validation of a high accuracy pointing system”. PhD thesis. Toulouse, ISAE, 2019.

[28] Francesco Sanfedino et al. “Finite element based N-Port model for preliminary design of multibody systems”. In: *Journal of Sound and Vibration* 415 (2018), pp. 128–146.

[29] Robert Lyle et al. “Spacecraft aerodynamic torques”. In: *NASA SP-8058* (1971).

[30] Arnaud Chulliat et al. “The US/UK world magnetic model for 2020-2025”. In: (2020).

[31] Wiley J Larson, James Richard Wertz, et al. *Space mission analysis and design*. Vol. 3. Springer, 1992.

[32] Rebecca A Masterson, David W Miller, and Robert L Grogan. “Development and validation of reaction wheel disturbance models: empirical model”. In: *Journal of sound and vibration* 249.3 (2002), pp. 575–598.

[33] Dae-Kwan Kim. “Micro-vibration model and parameter estimation method of a reaction wheel assembly”. In: *Journal of Sound and Vibration* 333.18 (2014), pp. 4214–4231.

[34] Georg Wiedermann et al. “The SENTINEL-2 satellite attitude control system–challanges and solutions”. In: *presented at 9th International ESA Conference on Guidance, Navigation & Control Systems*. 2014.

[35] David S Bayard. “High-Precision Three-Axis Pointing and Control”. In: *Encyclopedia of Aerospace Engineering* (2010).

[36] Oliver Montenbruck, Florian Kunzi, and André Hauschild. “Performance assessment of GNSS-based real-time navigation for the Sentinel-6 spacecraft”. In: *GPS Solutions* 26.1 (2022), p. 12.

[37] Oliver J Woodman. *An introduction to inertial navigation*. Tech. rep. University of Cambridge, Computer Laboratory, 2007.

[38] Airbus. *Astrix inertial measurement / IRU Series*. 2022. URL: `https://www.airbus.com/en/products-services/space/equipment/avionics/astrix-inertial-measurement-iru-series`.

[39] Honeywell. *Inertial Measurement Units*. 2022. URL: `https://aerospace.honeywell.com/us/en/products-and-services/product/hardware-and-systems/sensors/inertial-measurement-units`.

[40] Yijun Lian, Yudong Gao, and Guoqiang Zeng. “Staring imaging attitude control of small satellites”. In: *Journal of Guidance, Control, and Dynamics* 40.5 (2017), pp. 1278–1285.

[41] Przemyslaw Bogacki and Lawrence F Shampine. “A 3 (2) pair of Runge-Kutta formulas”. In: *Applied Mathematics Letters* 2.4 (1989), pp. 321–325.

[42] Xiaojiang Chen, Willem Steyn, and Yoshi Hashida. “Ground-target tracking control of earth-pointing satellites”. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2000, p. 4547.

[43] Ricardo Rodrigues et al. “Modeling, robust control synthesis and worst-case analysis for an on-orbit servicing mission with large flexible spacecraft”. In: *Aerospace Science and Technology* (2022), p. 107865.

[44] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer science & business media, 2013.

[45] Mark Pittelkau and William McKinley. “Pointing error metrics: displacement, smear, jitter, and smitter with application to image motion MTF”. In: *AIAA/AAS Astrodynamics Specialist Conference*. 2012, p. 4869.

[46] T Ott et al. “Precision pointing H∞ control design for absolute, window-, and stability-time errors”. In: *CEAS Space Journal* 4 (2013), pp. 13–30.

[47] LK Liu, GT Zheng, and WH Huang. “Octo-strut vibration isolation platform and its application to whole spacecraft vibration isolation”. In: *Journal of sound and vibration* 289.4-5 (2006), pp. 726–744.

[48] A Abu Hanieh. “Active isolation and damping of vibrations via Stewart platform”. PhD thesis. PhD Thesis, Université Libre de Bruxelles, Active Structures Laboratory, 2003.

[49] Bálint Patartics. “Uncertain systems: analysis and synthesis with application to flutter suppression control”. In: (2022).

[50] Joseph Ball, Israel Gohberg, et al. *Interpolation of rational matrix functions*. Vol. 45. Birkhäuser, 2013.

A

Simulink Model Blocks

The simulation environment forms a large feedback loop with three components. To obtain results, their outputs are exported into the Matlab workspace, where post-processing is done with Matlab scripts written individually for specific figures. During runtime, the outputs are observed with `Simulink Scope`s to take a quick, ballpark estimation of performance without needing to wait for the entire simulation length to pass.
Besides base `Simulink` blocks like `integrator`s, `state-space` blocks, `gain`s, `multiplexer`s, etc. all of the detailed systems here have been created by the graduating student under the supervision of the thesis advisors for the purpose of this project. The chapters containing the equations behind the blocks are referenced for each functionality.

A.1. Spacecraft dynamics

Description: The system dynamics block contains the nonlinear equations of motion for a point mass system (ch.2), as well as the calculation of the real pointing error (ch.3).
Inputs:

- $\mathbf{f}^B$: Body-frame forces acting on the COG
- $\boldsymbol{\tau}^B$: Body-frame torques acting on the COG
- $\mathbf{h}_{rw}$: Body-frame angular momentum vector of the RWA
- $\mathbf{x}_P$: State of the isolation platform

Outputs:

- $\dot{\mathbf{v}}^B_{BF}$: Body-frame accelerator vector of the COG (real and virtual component vectors)
- $\mathbf{v}^B_{BF}$: Body-frame velocity vector of the COG
- $\mathbf{r}^F_{BF}$: ECEF-frame position vector of the COG
- λ, ϕ, h: Latitude, Longitude, Altitude of the spacecraft
- $\dot{\boldsymbol{\omega}}^B_{BI}$: Body-frame angular acceleration vector of the COG
- $\boldsymbol{\omega}^B_{BI}$: Body-frame angular velocity vector of the COG
- $\mathbf{q}_{BL}$: Rotation quaternion between from the ECEF to Local frames (i.e. Euler angles ϕ, θ, ψ)
- $\mathbf{q}_{BF}$: Rotation quaternion from the ECEF to Body frames
- $\mathbf{q}_{PL}$: Rotation quaternion from the Local to Payload frames
- $\mathbf{q}_{TL}$: Rotation quaternion from the Local to Target frames
- $\mathbf{q}_{TB}$: Rotation quaternion from the Body to Target frames

Component: Translational Equations of Motion

Description: The translational EOM block contains the nonlinear equations for the orbit propagation of a point mass in Earth orbit (ch.2), the calculation of the base vectors for the local frame (ch.2), as well as the conversion of the spacecraft position vector to LLA coordinates (ch.2).
Inputs:

- $\mathbf{f}^B$: Body-frame forces acting on the COG
- $\mathbf{q}_{BF}$: Rotation quaternion from the ECEF to Body frames

Outputs:

- $\dot{\mathbf{v}}^B_{BF}$: Body-frame accelerator vector of the COG (real and virtual component vectors)
- $\mathbf{v}^B_{BF}$: Body-frame velocity vector of the COG
- $\mathbf{r}^F_{BF}$: ECEF-frame position vector of the COG
- λ, ϕ, h: Latitude, Longitude, Altitude of the spacecraft
- $\mathbf{q}_{LF}$: Rotation quaternion from the ECEF to Local frames

Component: Rotational Equations of Motion

Description: The rotational EOM block contains equations for the rotational kinematics and dynamics of a point mass system (ch.2), as well as the calculations of various rotation quaternions using the chain rule (ch.2).

Inputs:

- $\boldsymbol{\tau}^B$: Body-frame torques acting on the COG
- $\mathbf{h}_{rw}$: Body-frame angular momentum vector of the RWA
- $\mathbf{q}_{LF}$: Rotation quaternion from the ECEF to Local frames
- $\mathbf{x}_P$: State of the isolation platform

Outputs:

- $\dot{\boldsymbol{\omega}}^B_{BI}$: Body-frame angular acceleration vector of the COG
- $\boldsymbol{\omega}^B_{BI}$: Body-frame angular velocity vector of the COG
- $\mathbf{q}_{BL}$: Rotation quaternion between from the ECEF to Local frames (i.e. Euler angles ϕ, θ, ψ)
- $\mathbf{q}_{BF}$: Rotation quaternion from the ECEF to Body frames
- $\mathbf{q}_{PL}$: Rotation quaternion from the Local to Payload frames

Component: Real Pointing Angle

Description: The pointing angle block uses the equations for reference angle generation to define an idealized LOS vector (ch.3) for evaluating pointing precision.

Inputs:

- $\mathbf{r}^F_{BF}$: ECEF-frame position vector of the COG
- $\mathbf{q}_{BL}$: Rotation quaternion between from the ECEF to Local frames (i.e. Euler angles ϕ, θ, ψ)
- $\mathbf{q}_{LF}$: Rotation quaternion from the ECEF to Local frames

Outputs:

- $\mathbf{q}_{TL}$: Rotation quaternion from the Local to Target frames
- $\mathbf{q}_{TB}$: Rotation quaternion from the Body to Target frames

A.2. Forces and torques

Description: The force block contains the mathematical models for the effects of the environment (ch.2), the structural dynamics of the spacecraft (ch.2), as well as the actuators of the reaction wheel assembly (ch.3) and the isolator platform (ch.4).

Inputs:

- $\boldsymbol{\tau}_c$: Control demand of the outer loop in terms of reaction wheel torques.
- $\mathbf{f}_a$: Control demand of the inner loop in terms of actuator forces in the isolator.
- $\dot{\mathbf{v}}^B_{BF}$: Body-frame accelerator vector of the COG (real and virtual component vectors)
- $\mathbf{v}^B_{BF}$: Body-frame velocity vector of the COG
- $\mathbf{r}^F_{BF}$: ECEF-frame position vector of the COG
- λ, ϕ, h: Latitude, Longitude, Altitude of the spacecraft
- $\dot{\boldsymbol{\omega}}^B_{BI}$: Body-frame angular acceleration vector of the COG
- $\boldsymbol{\omega}^B_{BI}$: Body-frame angular velocity vector of the COG

- $\mathbf{q}_{BF}$: Rotation quaternion from the ECEF to Body frames

Outputs:

- $\mathbf{f}^B$: Body-frame forces acting on the COG, the sum of 3 components
- $\boldsymbol{\tau}^B$: Body-frame torques acting on the COG, the sum of 5 components
- $\mathbf{h}_{rw}$: Body-frame angular momentum vector of the RWA
- $\mathbf{x}_P$: State of the isolation platform

Component: Gravity Model

Description: Calculates the gravitational acceleration (and force) affecting the spacecraft at its current position along its orbit (ch.2).

Inputs:

- $\mathbf{r}^F_{BF}$: ECEF-frame position vector of the COG
- $\mathbf{q}_{BF}$: Rotation quaternion from the ECEF to Body frames

Outputs:

- $\mathbf{a}^B_{grav}$: Gravitational acceleration in the Body frame
- $\mathbf{f}^B_{grav}$: Gravitational force in the Body frame

Component: Structural Dynamics

Description: Contains the reduced state-space model that describes the flexible structural dynamics of the spacecraft (ch.2).

Inputs:

- $\dot{\mathbf{v}}^B_{BF}$: Body-frame accelerator vector of the COG (real and virtual component vectors)
- $\mathbf{v}^B_{BF}$: Body-frame velocity vector of the COG
- $\dot{\boldsymbol{\omega}}^B_{BI}$: Body-frame angular acceleration vector of the COG
- $\boldsymbol{\omega}^B_{BI}$: Body-frame angular velocity vector of the COG

Outputs:

- $\mathbf{f}^B_{flex}$: Body-frame structural forces acting on the COG
- $\boldsymbol{\tau}^B_{flex}$: Body-frame structural torques acting on the COG

Component: Environmental Torques

Description: This block calculates torque disturbances from the orbital environment of the spacecraft based on approximations of gravitational, magnetic, and atmospheric data (ch.2).

Inputs:

- $\mathbf{v}^B_{BF}$: Body-frame velocity vector of the COG
- $\mathbf{r}^F_{BF}$: ECEF-frame position vector of the COG
- λ, ϕ, h: Latitude, Longitude, Altitude of the spacecraft
- $\mathbf{q}_{BF}$: Rotation quaternion from the ECEF to Body frames

Outputs:

- $\boldsymbol{\tau}^B_{gg}$: Body-frame gravity gradient torque acting on the COG
- $\boldsymbol{\tau}^B_{mag}$: Body-frame magnetic torque acting on the COG
- $\boldsymbol{\tau}^B_{aero}$: Body-frame aerodynamic torque acting on the COG

Component: Reaction Wheel Assembly

Description: Contains the decoupling matrices, DC motor characteristics, and friction dynamics of the reaction wheel assembly, as well as an implementation of the microvibration environment generated by the component wheels (ch.3).

Inputs:

- $\boldsymbol{\tau}_c$: Control demand of the outer loop in terms of reaction wheel torques.

Outputs:

- $\mathbf{f}^B_{rw}$: Body-frame RWA forces acting on the COG
- $\boldsymbol{\tau}^B_{rw}$: Body-frame RWA torques acting on the COG
- $\mathbf{h}_{rw}$: Body-frame angular momentum vector of the RWA

Component: Hexapod Stewart-Gough Platform

Description: Contains the nonlinear equations of motion for the hybrid vibration isolation and pointing platform (ch.3).
Inputs:

- $\mathbf{f}_a$: Control demand of the inner loop in terms of actuator forces in the isolator.
- $\dot{\mathbf{v}}_{BF}^{B}$: Body-frame accelerator vector of the COG (real and virtual component vectors)
- $\dot{\boldsymbol{\omega}}_{BI}^{B}$: Body-frame angular acceleration vector of the COG
- $\boldsymbol{\omega}_{BI}^{B}$: Body-frame angular velocity vector of the COG
- $\mathbf{a}_{grav}^{B}$: Gravitational acceleration in the Body frame

Outputs:

- $\mathbf{x}_P$: State of the isolation platform

A.3. Onboard computer

Description: The onboard computer block is the heart of the control loop, where all the functionalities are separated in the simulation which would be carried out autonomously by the spacecraft electronics. This was done to prepare the environment for a future processor-in-the-loop testing campaign to assess computational speed. It contains the sensor models, state estimator, reference angle generator, and outer loop controllers (ch.2), as well as the isolator loop controller and virtual sensor (ch.3).
Inputs:

- $\mathbf{v}_{BF}^{B}$: Body-frame velocity vector of the COG
- $\mathbf{r}_{BF}^{F}$: ECEF-frame position vector of the COG
- $\boldsymbol{\omega}_{BI}^{B}$: Body-frame angular velocity vector of the COG
- $\mathbf{x}_P$: State of the isolation platform
- $\mathbf{q}_{BF}$: Rotation quaternion from the ECEF to Body frames
- $\mathbf{q}_{TB}$: Rotation quaternion from the Body to Target frames

Outputs:

- $\boldsymbol{\tau}_c$: Control demand of the outer loop in terms of reaction wheel torques.
- $\mathbf{f}_a$: Control demand of the inner loop in terms of actuator forces in the isolator.

Component: Sensor Models

Description: Contains the various sensor models (ch.3) which add noise to the precise, simulated states according to the characteristics of specific devices.
Inputs:

- $\mathbf{v}_{BF}^{B}$: Body-frame velocity vector of the COG
- $\mathbf{r}_{BF}^{F}$: ECEF-frame position vector of the COG
- $\boldsymbol{\omega}_{BI}^{B}$: Body-frame angular velocity vector of the COG
- $\mathbf{q}_{BF}$: Rotation quaternion from the ECEF to Body frames
- $\mathbf{x}_P$: State of the isolation platform

Outputs:

- $\mathbf{r}_{GPS}$: GPS measurement of spacecraft position
- $\mathbf{v}_{IMU}$: Integrated IMU measurement of spacecraft acceleration
- $\boldsymbol{\omega}_{IMU}$: IMU measurement of spacecraft angular velocity
- $\mathbf{q}_{SS}$: Star tracker measurement of spacecraft orientation
- $\mathbf{x}_{P,virt}$: Measured isolation platform states using virtual sensor

Component: State Estimation

Description: Contains the state estimator (ch.3) which calculates estimated states from noisy sensor measurements.
Inputs:

- $\mathbf{r}_{GPS}$: GPS measurement of spacecraft position
- $\mathbf{v}_{IMU}$: Integrated IMU measurement of spacecraft acceleration
- $\boldsymbol{\omega}_{IMU}$: IMU measurement of spacecraft angular velocity
- $\mathbf{q}_{SS}$: Star tracker measurement of spacecraft orientation
- $\mathbf{x}_{P,virt}$: Measured isolation platform states using virtual sensor

Outputs:

- $\mathbf{r}_{sen}$: Estimated spacecraft position
- $\mathbf{v}_{sen}$: Estimated spacecraft acceleration
- $\boldsymbol{\omega}_{sen}$: Estimated spacecraft angular velocity
- $\mathbf{q}_{sen}$: Estimated spacecraft orientation
- $\mathbf{x}_{P,sen}$: Estimated isolation platform states

Component: Controllers

Description: Contains all designed controllers (ch.3-4) in a single block, that can be individually turned on or off. The outer loop controllers all share a reference angle generator which calculates the desired orientation based on the estimated states (ch.3).
Inputs:

- $\mathbf{r}_{sen}$: Estimated spacecraft position
- $\mathbf{v}_{sen}$: Estimated spacecraft acceleration
- $\boldsymbol{\omega}_{sen}$: Estimated spacecraft angular velocity
- $\mathbf{q}_{sen}$: Estimated spacecraft orientation
- $\mathbf{x}_{P,sen}$: Estimated isolation platform states
- $\mathbf{q}_{TB}$: Rotation quaternion from the Body to Target frames

Outputs:

- $\boldsymbol{\tau}_c$: Control demand of the outer loop in terms of reaction wheel torques.
- $\mathbf{f}_a$: Control demand of the inner loop in terms of actuator forces in the isolator.

B

Code Snippets

Calculating LOS Vector Intersection with the WGS84 Ellipsoid

The function `LOSintersect` calculates the closest intersection of the WGS84 reference ellipsoid with the line-of-sight vector. It takes the LOS vector and the current position of the spacecraft in the ECEF frame as inputs, using the equatorial and polar radii of the planet as a parameter. Using dimensionless coordinate geometry, it returns the ECEF coordinates of the ground location at which the spacecraft is pointing.

```
function intersect = LOSintersect(Xecef,LOSecef,geo)
    x0 = Xecef(1)/geo.Req;
    y0 = Xecef(2)/geo.Req;
    z0 = Xecef(3)/geo.Rpol;
    x1 = LOSecef(1)/geo.Req;
    y1 = LOSecef(2)/geo.Req;
    z1 = LOSecef(3)/geo.Rpol;

    a = x1^2+y1^2+z1^2;
    b = x0*x1+y0*y1+z0*z1;
    c = x0^2+y0^2+z0^2-1;

    d = b^2-a*c;
    if (d < 0)
        interd = [NaN NaN NaN];
    else
        k = -(b+sqrt(d))/a;
        interd = Xecef+k*LOSecef;
    end
    intersect = [geo.Req geo.Req geo.Rpol].*interd
end
```

Magnetic Field Approximation

The function `geomag` calculates the magnetic field vector at the ECEF coordinates of the spacecraft, according to a lower-order approximation of the 2020 World Magnetic Model. The function uses the geomagnetic coefficients g and h as parameters in the calculation, which are tabulated in Chapter 2.

```
function B = geomag(Xecef, geo)
    r = sqrt(Xecef(1)^2+Xecef(2)^2+Xecef(3)^2);
    lam = atan2(Xecef(2),Xecef(1));
    phi = asin(Xecef(3)/r);

    Bx = 0; By = 0; Bz = 0;

    cp = cos(phi);
    sp = sin(phi);

    P = [cp,0,0;
        3*sp*cp,3*cp^2,0;
        6*cp-(15*cp^3)/2,-15*sp*(sp^2-1),15*cp^3];
    Pd = [-sp,0,0;
         3*(cp^2-sp^2),-6*sp*cp,0;
         -(3*sp*(15*sp^2-11))/2,45*cp^3-30*cp,45*sp*(sp^2-1)];

    for n = 1:3
        switch n
            case 1
                P0 = sp;
                Pd0 = cp;
            case 2
                P0 = (3/2)*sp^2-1/2;
                Pd0 = 3*sp*cp;
            case 3
                P0 = (5/2)*sp^3-(3/2)*sp;
                Pd0 = cp*(3/2)*(5*sp^2-1);
            otherwise
                P0 = 0;
                Pd0 = 0;
        end
        rr = (geo.Rm/r)^(n+2);
        Bx = Bx - rr * (geo.g0(n)) * Pd0;
        Bz = Bz - (n+1) * rr * (geo.g0(n)) * P0;
        for m = 1:n
            Schmidt = sqrt(2*factorial(n-m)/factorial(n+m));
            Bx = Bx - rr * (geo.g(n,m)*cos(m*lam)+geo.h(n,m)*sin(m*lam)) * Schmidt * Pd(n,m);
            By = By + (1/cp) * rr * m * (geo.g(n,m)*sin(m*lam)-geo.h(n,m)*cos(m*lam)) *
                Schmidt * P(n,m);
            Bz = Bz - (n+1) * rr * (geo.g(n,m)*cos(m*lam)+geo.h(n,m)*sin(m*lam)) * Schmidt *
                P(n,m);
        end
    end
    B = 1e-9*[Bx;By;Bz];
end
```