Design of Attitude Controller for Alticube+

Design of LQR Controller for Alticube+ and Evaluation of the Pointing Performance Based on Reaction Wheel Jitter and Flexible Structure Interactions

J. Y. Shioda

Delft University of Technology

TU Delft

Design of Attitude Controller for Alticube+

Design of LQR Controller for Alticube+ and Evaluation of the Pointing Performance Based on Reaction Wheel Jitter and Flexible Structure Interactions

by

J. Y. Shioda

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Monday January 26, 2025 at 15:00.

Student number:	5050936	
Project duration:	July 10, 2025 – January 9, 2025	
Thesis committee:	Dr. J. Guo,	TU Delft, Responsible supervisor
	Dr. ir. E. Van Kampen,	TU Delft, Supervisor
	A. Cervone,	TU Delft, Chair
	Dr. ir. J. Bouwmeester,	TU Delft, Examiner

TU Delft

Abstract

Interest in oceanic climate change and to better understand the oceanic dynamics, a key interest in oceanic topography pushes for cheaper and smaller EO satellites which can achieve similar resolution and measurement accuracy. Alticube+ lies at the forefront of a new frontier, aggregated system of CubeSats connected by booms to fulfil the scientific objective of providing accurate ocean height measurements on par with large monolithic satellites. This configuration introduces strong coupling between attitude dynamics, flexible structural modes, and reaction wheel jitter, posing a challenge for attitude controller design.

This research aims to design a centralised LQR attitude controller that enables effective utilisation of reactions wheel to satisfy scientific motivated pointing control and knowledge requirements. The second objective is to research how reaction wheel jitter interacts with the large aggregated structure to degrade the pointing control, affecting the measurement accuracy. To achieve this, a comprehensive simulation framework was developed in MATLAB/Simulink, capable of modelling both rigid-body and flexible spacecraft dynamics within a unified environment. A centralised Linear Quadratic Regulator (LQR) combined with a Kalman filter was designed to address multi-axis attitude regulation and pointing knowledge requirements under realistic actuator and sensor constraints.

The flexible spacecraft model was formulated using Kane' s equations and a lumped-parameter representation of the dominant structural modes. Reaction wheel jitter was modelled via static imbalance effects, enabling realistic excitation of flexible modes. The simulation framework was verified through analytical comparisons for the rigid-body case and validated for the flexible model by comparison with a finite element model, including modal frequency alignment and structural response under controlled torque excitation.

Simulation results demonstrate that the centralised LQR controller achieves stable attitude convergence within the allocated $450\,\mathrm{s}$ operation window for the majority of initial conditions. Monte Carlo analysis shows that approximately 80% of the simulated cases satisfy the absolute pointing error requirement of $0.2\,\mathrm{deg}$, with performance primarily limited by reaction wheel saturation. Flexible dynamics introduce oscillations, particularly in the pitch axis, where reaction wheel jitter and inertia uncertainty result in pointing errors up to $0.3\,\mathrm{deg}$ under worst-case conditions. Despite this, internal antenna misalignment and pointing knowledge errors remain within mission requirements for nominal operating conditions.

The results indicate that centralised LQR-based control remains a viable and effective solution for flexible assembled CubeSat systems such as Alticube+, provided that actuator saturation, structural flexibility, and uncertainty effects are explicitly accounted for during design. This work provides insight into the achievable pointing performance envelope of Alticube+'s architecture.

Contents

List of Figures

List of Tables

Nomenclature

If a nomenclature is required, a simple template can be found below for convenience. Feel free to use, adapt or completely remove.

Abbreviations

Abbreviation	Definition
ADCS	Attitude Determination and Control System
AEKF	Additive Extended Kalman Filter
AKE	Absolute Knowledge Error
AOCS	Attitude and Orbital Control System
APE	Absolute Pointing Error
AT	Along-Track
CKF	Cuberate Kalman Filter
COTS	Components of the Shelf
CPU	Computer Processing Unit
CT	Cross-Track
DCM	Directional Cosine Matrix
DOF	Degrees of Freedom
ECI	Earth-Centred Inertial
ECSS	European Cooperation of Space Standardization
EKF	Extended Kalman Filter
EO	Earth Observations
ESA	European Space Agency
FEM	Finite Element Analysis
HJB	Hamilton-Jacobi-Bellman
KaRIn	Ka-band Radar Interferometry
KDE	Knowledge Drift Error
KF	Kalman Filter
LEO	Low Earth Orbit
LEOP	Launch and Early Orbit Phase
LQR	Linear Quadratic Regulator
LTI	Linear Time-Invariant
MEKF	Multiplicative Extended Kalman Filter
MEMS	Micro-Electro-Mechanical System
MKE	Mean Knowledge Error
MPC	Model Predictive Control
MPE	Mean Performance Error
NASA	National Aeronautics and Space Administration
ND	Nadir
NOAA	National Oceanic and Atmospheric Administration
OFRF	Orbital-Fixed Reference Frame
OPRF	Orbital Path Reference Frame
PID	Proportional Integral Derivative
PDE	Partial Drift Error
RAAN	Right Ascension of the Ascending Node
RCS	Reaction Control System
RL	Reinforcement learning
RMS	Root Mean Squared

Abbreviation	Definition
RW	Reaction Wheel
SAC	Simple Apaptive Control
SAR	Synthetic Aperture Radar
SC	Spacecraft
SD	Standard Deviation
SISO	Single-Input Single-Output
SSO	Sun Synchronous Orbit
SWOT	Surface Water and ocean Topography
UKF	Unscented Kalman Filter

Symbols

The Symbols nomenclature can be divided into three categories. Scalars, vectors and Matrices. Scalars can be represented by either capital or lower case letters or symbols (e.g. α or A). Vectors are represented as lower case symbols with a vector notation on top (e.g. $\bar{\alpha}$ or $\hat{a}$). Matrices are always capital letters or symbols and are in bold (e.g. $\boldsymbol{A}$ or $\boldsymbol{\Gamma}$).

Symbol	Definition	Unit
A	Area	[m^2]
B	Baseline length	[m]
B_0	Reference baseline length	[m]
c	Speed of light	[$\frac{m}{s}$]
C	Cross-track length	[km]
C_D	Drag coefficient	[-]
f	Frequency	[Hz]
F_s	Static imbalance force	[N]
h	Ocean surface height	[km]
H	Orbit height	[km]
$i_\odot$	Earth inclination angle	[deg]
i	Orbit inclination angle	[deg]
J	Cost function value	[-]
L	Angular momentum	[Nms]
n	Angular rate of orbit	[$\frac{rad}{s}$]
Q_q	Attitude LQR cost weights	[-]
Q_ω	Angular rate LQR cost weights	[-]
r	Distance between points	[km]
$R_\oplus$	Radius of Earth	[km]
S	Solar constant	[$\frac{W}{m^2}$]
t	Time	(s)
T_d	Dynamic imbalance torque	[Nm]
u_d	Dynamic imbalance constant	[$kg \cdot m^2$]
u_s	Static imbalance constant	[$kg \cdot m$]
Δt	Time interval	[s]
ϕ	Phase angle	[rad]
λ	Wave length	[m]
$\mu_\oplus$	Gravitational constant of Earth	[$\frac{km^3}{s^2}$]
ρ	Density	[$\frac{km}{m^3}$]
σ	Standard deviation	[-]
θ	Rotation angle	[rad]
θ_0	Reference roll angle	
$\theta_\odot$	Sun phase angle	[rad]

Symbol	Definition	Unit
θ_p	Orbit Path angle	[rad]
ω	Angular rate	$[\frac{rad}{s}]$
$\omega_{\odot}$		
ω_{RW}	Reaction wheel angular rate	$[\frac{rad}{s}]$
$\hat{a}$	OPRF unit vector	[-]
$\hat{b}$	Body reference frame vector	[-]
$\hat{e}_{\odot}$	Sun unit vector	[-]
$\hat{I}$	Inertial unit vector	[-]
$\bar{B}$	Magnetic field vector	[T]
$\bar{f}$	Force	[N]
$\bar{L}$	Reaction wheel Angular momentum state vector	$[Nms]$
$\bar{m}$	Residual dipole moment vector	$[Am^2]$
$\bar{q}$	Quaternion vector	
$\bar{r}$	Distance vector between two points	[m]
$\bar{x}$	State space vector	
$\bar{z}$	Measurement vector	
$\bar{\tau}$	Torque	[Nm]
$\bar{\theta}$	Euler angle rotation vector	[rad]
$\bar{\omega}$	Angular rate vector	$[\frac{rad}{s}]$
$\boldsymbol{A}$	State system matrix	
$\boldsymbol{A}_{RW}$	Reaction wheel Torque transformation matrix	
$\boldsymbol{B}$	State input matrix	
$\boldsymbol{C}$	Directional Cosine Matrix	[-]
$\boldsymbol{D}$	Throughput matrix	
$\boldsymbol{H}$	Observation matrix	
$\boldsymbol{I}$	Identity matrix	
$\boldsymbol{J}$	Mass moment of inertia matrix	$[kg \cdot m^2]$
$\boldsymbol{Q}$	LQR state cost matrix	
$\boldsymbol{R}$	LQR actuator cost matrix	
$\boldsymbol{S}$	LQR optimal cost function matrix	
$\boldsymbol{T}$	Transformation Matrix	
$\boldsymbol{\Lambda}_q$	Quaternion skew matrix	
$\boldsymbol{\Lambda}_{\omega}$	Quaternion augmented angular rate skew matrix	$[\frac{rad}{s}]$
$\boldsymbol{\Omega}$	Angular rate skew matrix	$[\frac{rad}{s}]$

1

Introduction

Interest in oceanic climate change and to better understand the oceanic dynamics, a key interest in oceanic topography pushes for cheaper and smaller EO satellites which can achieve similar resolution and measurement accuracy. Alticube+ lies at the forefront of a new frontier, aggregated system of CubeSats connected by booms to fulfil the scientific objective of providing accurate ocean height measurements on par with large monolithic satellites. Alticube+ uses Ku-band SAR interferometry to measure the ocean height. Although this aperture is an effective payload for this purpose, elevated sensitivity to roll pointing errors stresses the importance understanding both pointing control and knowledge capabilities. Pointing control error sources stem from the controller design and Alticube+'s reaction wheel jitter, possibly interacting with the structure.

This research aims to design an attitude controller that utilises reactions wheel to satisfy scientific motivated pointing control and knowledge requirements. The second objective is to research how reaction wheel jitter interacts with the large aggregated structure to degrade the pointing control, affecting the measurement accuracy. This research will achieve the objective by first expanding the research objective through literature informed supporting research questions. Then this research provides a modelling framework implementing Kane's equations to validate the controller design and analyse the reaction wheel jitter interaction. Bode analysis on the LQR cost matrix tuning will be used to identify a feasible range of tuning parameters that provides a stable response of the LQR controller. Then controller capabilities is further researched through variations in initial conditions and assumed inertia matrix. Lastly, the interaction between the reaction wheel jitter and the structure is then analysed using the simulation data obtained.

This MSc thesis is structured as follows. The second chapter reviews the relevant literature and expands on the research objective. Additional supporting research questions break down the global objective into a tractable research. The third chapter explores the ADCS requirements that can be derived from the scientific objectives. The fourth chapter provides a mathematical modelling framework, divided into four section: A orbital model framework, rigid body model, flexible model through the implementation of Kane's equations and a reaction wheel model framework that includes jitter generation. The fifth chapter designs and tunes the LQR controller and provides a mathematical implementation of the Kalman filter for attitude estimation. The sixth chapter presents the simulation setup, verification and validation, and simulation results. Chapter 7 concludes the research and chapter 8 provides recommendations for future research.

2

Literature review

2.1. Scientific Motivation

For decades, satellite altimetry missions have advanced measurements of the ocean topography. These missions have mapped variations in the ocean's surface, capturing the effects of gravity, ocean currents, and heat distribution. The data generated has been crucial for advancing scientific research and supporting practical applications such as weather forecasting, storm surge prediction, fisheries management, maritime navigation, and disaster risk assessment [1]. Radar interferometry technology can be used to obtain sub-kilometre resolution given a sufficiently long baseline that accommodates at least two antennas [2]. The Surface Water and Ocean Topography (SWOT) satellite launched in 2022 is one of the great examples. This satellite houses the KaRIn (Ka-band Radar Interferometry) using a $10\,m$ baseline and represents the state-of-the-art in radar altimetry providing sub-kilometre resolution [3]. Monolithic spacecraft utilizing KaRIn like SWOT does come at a cost as these missions are expensive (north to a billion dollars) and take huge effort (15 years) to design and built.

With current efforts heading towards the miniaturization of spacecraft, CubeSat missions have gained significant traction in the space industry. The development for the miniaturization of spacecraft is not aimed to replace monolithic spacecraft as a whole, but provide a standardized and flexible low-cost alternative for a subset of missions. The first CubeSats built were mainly educational and technology demonstration missions, but since 2017 they are used for Earth Observation (EO) science and include more flexible platforms, including constellations [4]. The Raincube satellite was one of the critical missions that contributed to these efforts. The success of the RainCube satellite opened up the possibility for a CubeSat swarm to serve as a low-cost alternative for EO science [5]. However, the realization of such standardized swarming solutions is not without challenges. from a study done, [6] concluded that "a distributed interferometric swarm for cross-track measurement would require extremely high accurate time/attitude synchronizations and centimetre level formation maintenance, which is not easier than building a large monolithic satellite". This issue could be circumvented if the CubeSats are physically connected to each other such that formation flying is not needed at all during science.

Alticube+ is a ESA funded mission led by TU Delft that wants to take advantage of this to build a capable CubeSat platform performing in-orbit assembly to perform EO science for Ocean and inland water applications [7]. This mission will perform high resolution EO hydrology sciences while keeping lower costs than monolithic spacecraft missions for EO sciences. The key technology used is the in-orbit assembly to increase the baseline for the Ku-band radar payloads to obtain higher resolution than ever possible for a single CubeSat. Alticube+ will consist of five 16U CubeSats connected with a boom as seen in Figure 2.1.

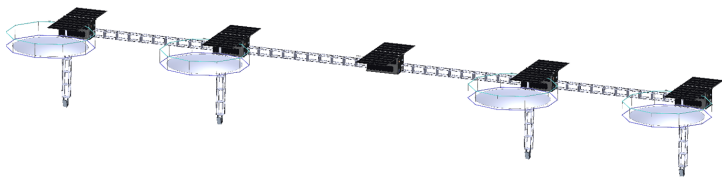


Figure 2.1: Representative model of the fully assembled Alticube+ spacecraft [7]

At LEOP the five individual CubeSats are inserted into a $500\,km$ SSO orbit. After deployment, the CubeSats will perform a rendezvous and docking sequence. Once the spacecraft is fully assembled and correctly oriented, scientific observations can begin. The total length of the assembled spacecraft will be 880 cm, with each CubeSat platform measuring 40 cm and each boom in between extending 180 cm. Figure 2.2 gives a visual representation on how the Alticube+ will measure high resolution data. Science operations will be conducted by either Sat #1 and Sat #4 or Sat #2 and Sat #5, both observing the same target. By positioning two antennas to measure two swath bands, the system effectively simulates a single large dish with a diameter equal to the spacing between the antennas. The spacing between these two nodes is equal to $660\,cm$. This configuration enables a swath width of 18 km, a resolution of 220 x 313 m, and a statistical height error of 1.3 cm, meeting the mission's stringent scientific objectives. An analysis study performed by [7] showed that the roll stability (rotation around the flight-direction axis) plays a critical roll in the height measurement error.

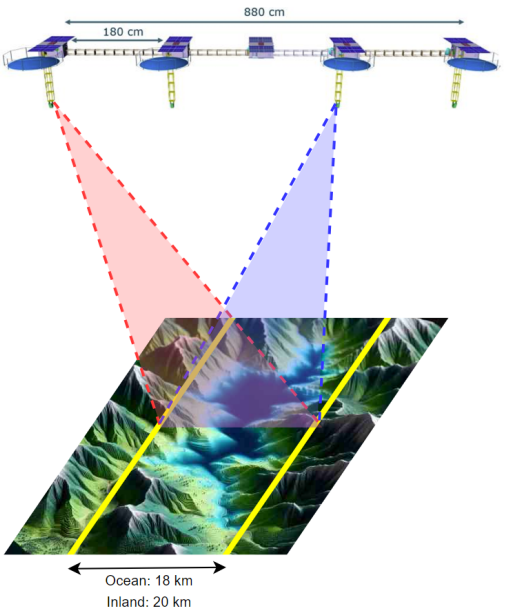


Figure 2.2: Alticube+ measuring inland water surface height using two antennas (single swath band represented)

2.2. Ku-band Radar Interferometry

Radar is a widely used instrument since early 1900. Its main applications lies in detecting ships and aircraft, laying the groundwork of today's instruments. The radar technology relied on the principles of reflected radio waves. To measure the position of an object (e.g. aircraft) using radar, a pulse of radio waves (ranging from $1\,mm - 1\,m$) is transmitted from the radar system. Radio waves that hit the object will reflect these waves back towards the direction of the radar system. The receiver then detects the reflected radio waves, computing the distance of that object depending how much time the wave took between transmission and detection. As this principle is simple, it is an effective method utilized by spacecraft for altimetry measurement. To effectively obtain high vertical precision measurements, this instruments needs to be nadir-pointing and have its transmitted beam tightly focussed. As a result, the swath width of this instrument is limited to the area direct under the satellite. Therefore temporal resolution (revisit times) is favoured over spatial coverage [8]. To measure the topography of the surface using a larger swath width, a side-looking radar is needed. Synthetic Aperture Radar (SAR) is an instrument that is able to measure the topography from roughly 20 degrees to 60 degrees away from the nadir-point as seen in Figure 2.3.

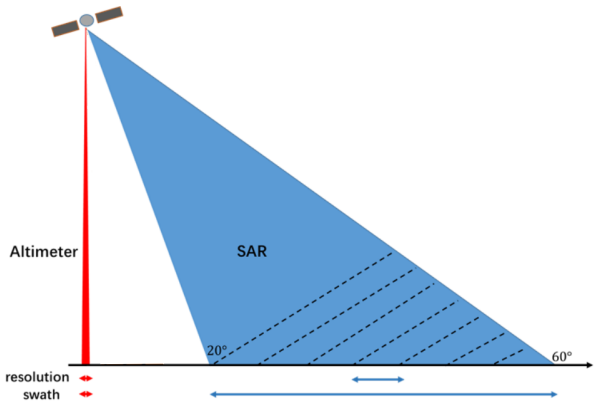


Figure 2.3: swath and angle of incidence difference between an altimeter and a SAR [9]

SAR can use multiple frequency bands ranging from $300\,kHz - 47\,GHz$ to observe different aspects. The lower frequencies are used for biomass and agriculture while higher frequencies are used for oceanic and glaciology applications. The C-band, X-band and Ku-band, band classifications for radar frequencies are commonly used for oceanic and inland water height measurements. The Ku-band, which will be used by Alticube+, has a frequency range of $12 - 18\,GHz$ (wavelength of $1.7 - 2.5\,cm$).
Now the measurement principles of SAR for oceanic applications will be explored. First some assumptions are made. First assumption is that the SAR will only measure 1 point. While normally the instrument is able to measure between a range of angles, only one of these points is looked into for simplicity. The second assumption is about the atmospheric effects, such as scattering and refraction of the radar signal. For simplicity this effect is neglected.
Imagine a satellite that has a baseline length B measuring a point with height h from the reference as seen in Figure 2.4. The two antenna's placed at each end of the baseline, need to be differentiated into the master antenna and slave antenna. In Figure 2.4 the master antenna is on the right side and is defined by r_1, which signifies the absolute distance between the master antenna and measuring point. r_2 is the absolute distance between the slave antenna and measuring point. To calculate the height, the following equation is used.

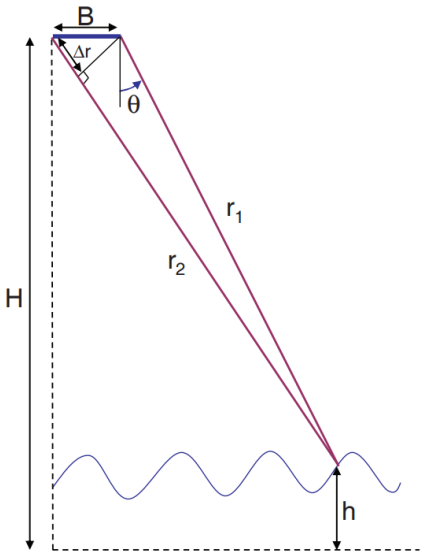


Figure 2.4: Alticube+

$$h = H - rcos(\theta) \tag{2.1}$$

Where H is spacecraft altitude and r the distance from the midpoint of the baseline to the measurement point. θ can be precisely estimated using the phase difference Φ of the signals between the master and slave antenna [10], which is described using the following two equations.

$$\Delta r = r_1 - r_2 = Bsin(\theta) \quad ; \quad \Delta r = \frac{\lambda}{2\pi}\Phi \tag{2.2}$$

Here λ is the wave length of the radar signal. Φ as mentioned before, is the unwrapped phase difference. This is different from the measured phase difference Φ'. Auxiliary data is needed to unwrap the phase difference by determining the n in $\Phi = 2\pi n + \Phi'$ [9]. It is apparent from this measurement principle that the accurate knowledge of θ is important to the accuracy of the height. accurateness of angle θ could be influenced by the attitude instability of the spacecraft. A small change in theta would mean a change in the calculated height. The change in height $(\delta h = h_{measured} - h_0)$ can be derived from equation 2.1. Expanding the $\cos(\theta)$ to $\cos(\theta_0) - \sin(\theta_0)\delta\theta$ using the Taylor expansion and substituting this into equation 2.1 will give equation 2.3.

$$\delta h \approx H\sin(\theta_0)\delta\theta \longleftrightarrow \delta h \approx C\delta\theta \tag{2.3}$$

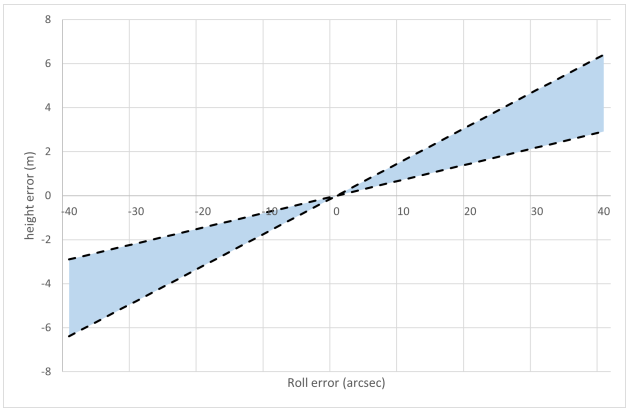


Figure 2.5: height error observed from the change in θ_0 due to roll

θ_0 represents the true angle of the measuring point and where $\delta\theta$ is defined as the roll error. In this equation, it is assumed that $H \approx r$ as the difference is negligible. The range of θ_0 can be derived from the missions specifics around the measurement objective. Alticube+ will use a swath width of $18\,km$ over oceans where the closest measurement point is $15\,km$ away from nadir. Based on the orbit height of $500\,km$ the measurement range lies within $1.7 \leq \theta_0 \leq 3.8$ degrees. Figure 2.5 shows he range of height errors as a function of $\delta\theta$. A range of $\pm 40\,arcsec$ is taken as roll error. The calculated height error for this range has a maximum of $\pm 6.4\,m$.

Another error lies within the length of the boom. A small change in this boom length would also impact the calculated height. Deriving this relation starts with equation 2.2 and let Δr be changed due to both an error in θ and B. Using small angle approximation, it is known that these changes are proportional to each other ($\delta\theta \propto \delta B$), generating the same $\delta\Delta r$. Thus the following equation holds for small angles

$$\sin(\theta_0)\delta B = B_0\delta\theta \tag{2.4}$$

substituting equation 2.4 into 2.3 will give equation 2.5

$$\delta h = \frac{H\sin^2(\theta_0)}{B_0}\delta B \longleftrightarrow \delta h = \frac{C^2}{B_0 H}\delta B \tag{2.5}$$

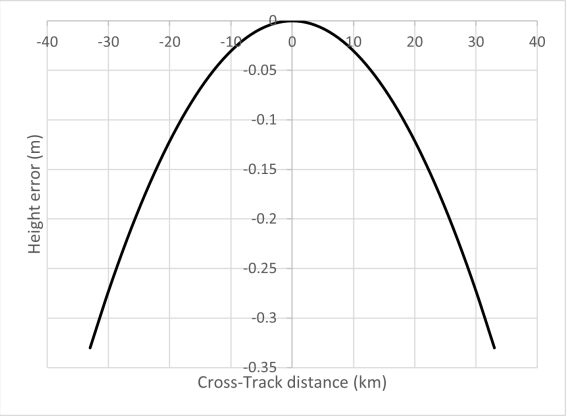


Figure 2.6: height error observed over the swath width with a $1\,mm$ boom length error

Figure 2.6 plots this height error due to a $1\,mm$ boom length error. The swath width range of the Alticube+'s SAR is taken. As seen from equation 2.5 and Figure 2.6, the height error has a quadratic relationship with the cross-track distance from nadir. Compared to the roll error, the height error is much smaller for a boom length error of $1\,mm$.

Impact of micro-vibrations on height error

In addition to the errors discussed, semi-rigid spacecraft vibrate which can cause errors as well. [7] covers a micro-vibrational analysis study on Alticube+ which will be briefly covered in this section. The two elastic modes that are relevant to this height measurement error analysis can be described in Figure 2.7. The left figure describes a symmetric bending of the spacecraft, which will shorten the effective baseline. The right figure shows an antisymmetric bending of the spacecraft which results in a roll error in addition to the shortening of the baseline. The vibration analysis of [7] shows that Alticube+'s aggregated system has a maximum deformation of $13.7\,arcsec$ and $7.0\,arcsec$ for the symmetric and antisymmetric bending respectively. To calculate how much the baseline has shortened, a conservative approach is taken. The new baseline is calculated using equation 2.6.

$$\delta B = B_0\left(1-\cos\left(\frac{\delta\alpha}{2}\right)\right) \tag{2.6}$$

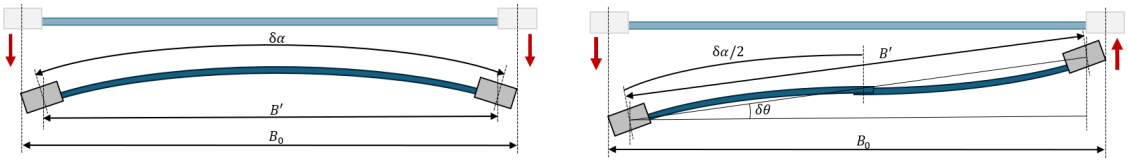
For the asymmetric case, equation 2.6 is substituted into equation 2.4, resulting is the equation below.

$$\delta\theta = \delta\alpha + \frac{C}{H}\left(1-\cos\left(\frac{\delta\alpha}{2}\right)\right) \tag{2.7}$$

Now as the vibration are small angles, the small angle approximation is used (i.e. $1-\cos(\delta\alpha/2)\approx 0$), the height error can be calculated using equations 2.3 and 2.5 - 2.7.

$$\delta h_{sym} \approx 0; \quad \delta h_{asym} \approx \frac{C\delta\alpha}{2} \tag{2.8}$$

As derived from equation 2.8, vibration cause negligible reduction of the effective boom length. The antisymmetric case, which causes a virtual roll of the boom, can create a maximum height error of $1.1\,m$ in addition to the baseline roll error.



(a) Symmetric bending modus

(b) Asymmetric bending modus

Figure 2.7: Alticube+ boom vibration modus

Inteferometry Data Calibration

As noted in the previous sections, roll, boom length, and vibration introduce significant errors to the height measurements. It is important to understand the scientific requirements on the height error. Alticube+'s criteria on the scientific measurement accuracy focusses on the relative height error rather than the absolute error. This means that the error between adjacent cells in the SAR instrument is closely investigated. Following from the scientific requirements, the total instrument SD height error of $2\,cm$ needs to be satisfied [7]. The instrument is composed of two error terms, a statistical height error and roll error of the baseline ($\sigma^2_{instrument} = \sigma^2_{statistical} + \sigma^2_{roll}$). Knowing the instrument error and statistical height error (refer to section 2.1), the height error due to roll is calculated: $\sigma_{roll} = \sqrt{2^2 - 1.3^2} = 1.52\,cm$. Now comparing this to the height error analysis in section 2.2, it can be inferred that a SD roll error of $0.095\,arcsec$ needs to be achieved. This roll error requirement is infeasible as even NASA's SWOT satellite is capable of reducing the roll error to $10\,arcsec$ using the ADCS system [11]. In order to reduce height errors, calibration techniques can be employed, increasing the height accuracy and therefore relaxing the roll error requirements. This section will explore the auto-calibration technique and lay out the implications this calibration technique has on the roll error.

The purpose of auto-calibration is to refine the measurements by reducing residual roll error after an initial calibration has been used. This first calibration layer uses roll knowledge to eliminate large height errors as a first step whereas the auto-calibration method uses interferometric data to eliminate the residual roll error. Imagine a ocean wave that has been measured. Let the true height of the ocean surface that will be measured be expressed by the blue line in Figure 2.8. The observed height is then a superposition of the true height and the height error due to residual roll (black line in Figure 2.8). The height error has a linear relationship with roll error (See Equation 2.3 and red line in Figure 2.8). The difficulty in unravelling the true height is that both slopes are not known over the cross-track.

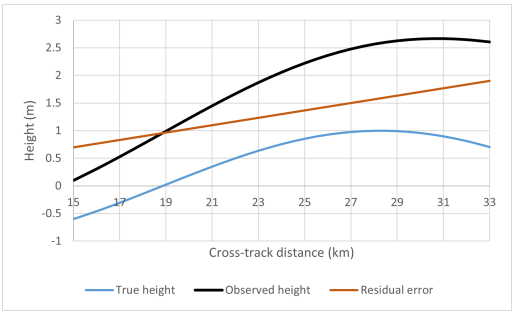


Figure 2.8: Observed height by SAR (black line) as a superposition of the true height (blue line) and height error due to residual roll (red line)

The principle of the auto-calibration method is to use along-track data to observe the rate of change in slope in the frequency domain and distinguish between the ocean slope and residual roll. Figure 2.9 visualizes the change in ocean height for the same cross-track position over time. This means that as waves change in the along-track direction, the ocean slope along this cross-track will change over time. It should be noted that the along-track evolution of the ocean height that is visualized in Figure 2.9 is not a realistic representation of the ocean wave dynamics and is purely for visual purposes. An ocean wave spectrum study shows that waves have a period range between $0 - 0.2\,Hz$, meaning that for every five seconds or more a wave will pass through [12]. Now distinguishing the height error slope due to the residual roll error from the ocean height slope is only possible if the frequency at which the residual error moves between $\pm\delta\theta_{residual}$ is higher than $0.3\,Hz$.

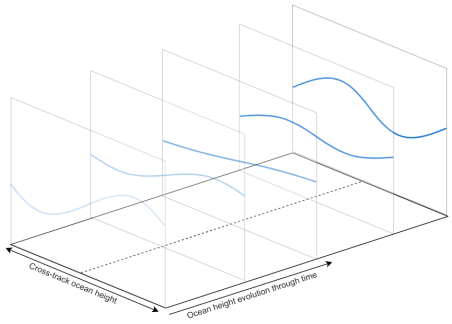


Figure 2.9: Cross-track ocean wave height change during along-track measurements

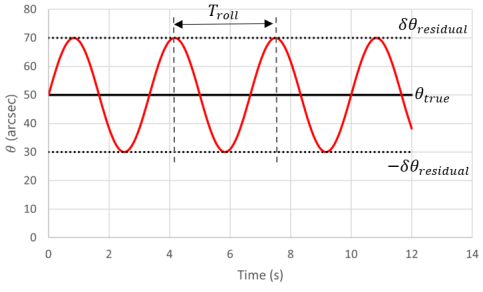


Figure 2.10: Another figure

The auto-calibration method does need to have a residual error less than $20\,arcsec$ RMS to success-

fully distinguish between the ocean slope and residual roll slope. If this requirement cannot be met, an unfiltered roll calibration can be used to reduce the height errors. This method will flatten the entire slope across the measurement track to zero, also filtering some of the finer ocean structures. Although less desirable, this filter technique does fall within the height error requirements. The roll knowledge requirement for this technique is $100\,arcsec$, less stringent than the auto-calibration method.

Roll Requirements

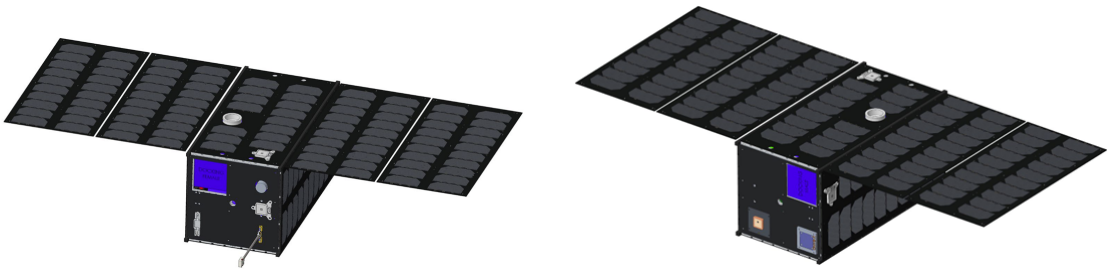
Using the information provided in this section the roll requirements based on scientific mission objectives can be generated. One last requirements is still left to be defined. To ensure that all antenna's have good overlap for both swath sites, the spacecraft should orient with an error not greater than $0.1 - 0.2\,deg$ [7]. Now all roll requirements have been generated and are summarised in Table 2.1 below.

Table 2.1: Roll requirements for Alticube+ derived from payload height error requirements

Requirement ID	Requirement description	Notes
Req-Roll-1	The aggregated system must have an APE of the roll of 0.1-0.2 deg.	Pointing error requirements
Req-Roll-2	The aggregated system must have a MKE of the roll angle of 20 arcsec RMS over the whole measurement track (up to 20 minutes), or max 100 arcsec if req 3 cannot be achieved.	Pointing knowledge requirements
Req-Roll-3	The aggregated system should keep a MKE on the roll rate larger than 0.3 Hz.	

2.3. Alticube+ Platform

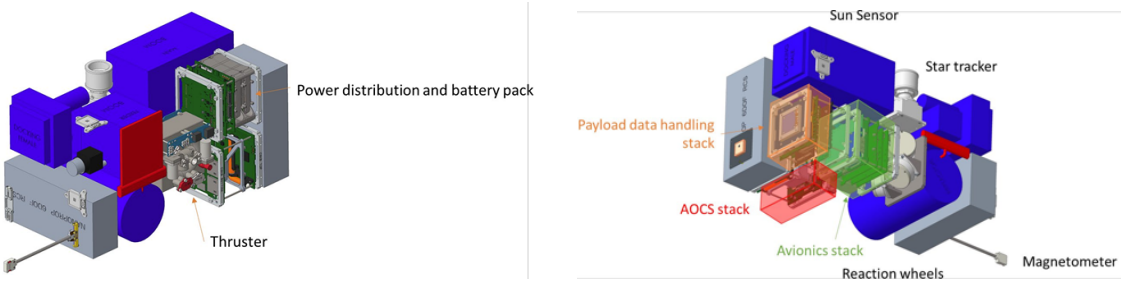
In the previous two sections, Alticube+'s aggregated system was outlined briefly. In this section a more in-depth overview of the CubeSat platforms (nodes) which Alticube+ consists of is presented. Figure 2.11 and Figure 2.12 shows the external and internal view of the Alticube+ CubeSat platform respectively. Here, the +X side points parallel to the along-track vector, the +Z points to nadir and +Y complements the other two vectors. The platform is a 16U CubeSat in a $2 \times 2 \times 4$ configuration, having a weight totalling $18\,kg$.



(a) View of -X/-Y/-Z side [7]

(b) View of +X/+Y/-Z side

Figure 2.11: External view of Alticube+ platform [7]



(a) View of -X/-Y/-Z side [7]

(b) View of +X/+Y/+Z side

Figure 2.12: Internal view of Alticube+ platform [7]

ADCS Hardware

Each CubeSat platform is equipped with an ADCS subsystem. A star tracker, sun sensor, gyroscope and magnetometer boom encompasses the sensors. In Figure 2.12b, the location of the various sensors can be seen. Four reaction wheels and a magnetotorquer give the CubeSat the ability to control the attitude in all three axis. Two sets of RCS thrusters provides a 6-DOF capabilities. This section covers each of the attitude sensors and control actuators.

Star tracker

The star tracker used in Alticube+'s CubeSat platform is an Auriga from Sodern (see Figure 2.13). This sensor is placed towards the -Z axis seen in Figure 2.12. It can provide quick attitude acquisition with arcsec level accuracy. This sensor can measure and output data at a rate of $10\,Hz$ or $5\,Hz$ depending on the CPU load. A star tracker is generally a heavy sensor and Auriga weighs $770\,g$. The data obtained contains some errors inherent to the instrument. A worst case sensor bias of $0.017\,deg$ and a worst case thermo-elastic error of $< 1.5\,arcsec/C^{\circ}$ can create a worst case periodic bias of $106\,arcsec$. next to the biases, there are uncertainties due to spatial and temporal noise which can create an additional error of $94.6\,arcsec\,@\,3\sigma$. The star tracker is only capable of determining one axis of reference for the spacecraft. To complete the three axis of reference, another sensor type or more star trackers on other axis are required.

Figure 2.13: Sodern Auriga star tracker [13]

Figure 2.14: Lens R&D MAUS sun sensor [14]

Sun sensor
Three MAUS sun sensors from Lens R&D are used that provide a three axis reference frame for the CubeSat platform (see Figure 2.14). One sensor is placed on each of the +X, -Y and -Z side of the CubeSat, which can be seen in both Figure 2.11 and Figure 2.12. Weighing at just 15 grams and an output rate of $2\,Hz$, the sun sensors can give the full attitude of the spacecraft, unlike the single star tracker. A disadvantage is that the attitude error observed is larger than that of a star tracker, giving attitude error of $3.5\,deg\,@\,3\sigma$ without calibration tables up to $0.5\,deg\,@\,3\sigma$ when calibrated. The data of these sensors could be fused with the more accurate star tracker data to give better estimations on the attitude of Alticube+'s CubeSat platform.

Gyroscope
The gyroscope selected for Alticube+ is the STIM277H MEMS gyroscope from Safran (see Figure 2.15). Placed in the AOCS stack, this sensor has the capability to directly measure the rotational velocities of Alticube+'s CubeSat platform. The sensor weighs $52\,g$ and can output at a rate of $1000\,Hz$. Two types of errors are present in the instrument depending on the type of data that is being output from the sensor. For angular measurements, the initial bias drift is $0.3\,°/h$. For angular velocity measurements, the root Allan variance is $0.15\,°/\sqrt{h}$.

Figure 2.15: Safran STIM277H MEMS gyroscope [15]

Magnetometer and magneto torquer
The magnetometer and magneto torquer used are from ISISpace [16].
The sensor is placed on a boom on the -Y size of the spacecraft as seen in Figure 2.11 and 2.12. The primary use for the magnetometer is to establish a reference frame relative to Earth's magnetic field and information source for the magneto torquer to correctly apply torque for the desired actuation. The magneto torquer weighs $196\,g$ and able to provide $0.2\,Am^2$. A feature of this magneto torquer is that it has three axis actuation with an accuracy of $< 3\,\mu T$. An additional feature is the actuator's capability to de-saturate reaction wheels. The magnitude of angular momentum per orbit depends on Earth's magnetic field strength. [17] shows that at $500\,km$ altitude the magnetic field strength varies from $20 - 50\,\mu T$. The angular momentum generated by the magneto torquer for each orbit then lies between $0.034 - 0.046\,Nms$.

Figure 2.16: Astrofein RW25 reaction wheels [18]

Reaction wheels
Astrofein's RW25 is the chosen reaction wheel for Alticube+ (see Figure 2.16). Four reaction wheels are used in a tetrahedral configuration which can be located behind the avionics stack seen in Figure 2.12b. The tetrahedral setup is capable of providing three axis attitude control using any three of the four reaction wheels. This means that this configurations already has redundancy in its setup, not needing any other redundant reaction wheels. A drawback is the necessitation of vector transformations through mathematical models to effectively utilise the reaction wheels. Each of the reaction wheels can provide $2\,mNm$. The maximum momentum storage each wheel can store is $0.03\,Nms\,@\,5000\,rpm$.

2.4. State-of-the-Art Control Methods

Alticube+'s distributed control is relatively novel with respect to the history of control theory and spacecraft. Alticube+ is a spacecraft that has multiple sets of actuators along the aggregated system, which necessitates addressing this control method. Attitude control for spacecraft started around 1960. Missions used Passive techniques to maintain spacecraft orientation. Spin stabilization involved rotating the spacecraft around its axis, utilizing gyroscopic principles to maintain a consistent orientation. This method was straightforward and reliable, making it suitable for early missions. Alternatively, gravity gradient stabilization exploited Earth's gravitational variations to align the spacecraft along a preferred orientation.

As mission objectives grew more complex, the limitations of passive systems became evident, unable to provide the precise control required for tasks such as Earth observation and astronomical studies. This necessitated the development of active attitude control systems capable of three-axis stabilization. These systems used actuators like reaction wheels, control moment gyroscopes, and thrusters to adjust the spacecraft's orientation dynamically. Three-axis stabilization allowed for precise pointing in any direction, significantly enhancing the versatility and capability of spacecraft. From 2010 onwards, micro-electromechanical systems (MEMS) started to become commercially available as Component of the Shelf (COTS) capable of performing attitude tasks previously reserved for large monolithic spacecraft.

The optimization of attitude stability is critical for Earth observation missions. The increasingly complex and high resolution criteria stretches the boundary for attitude stability optimization. Large monolithic spacecraft that employ rigid-body attitude control laws generally do not perform within the requirements any more as vibrations from the large solar panels hamper the performance. Attitude control for monolithic spacecraft is now generally simplified to rigid body with flexible appendages and has been extensively researched [19, 20, 21]. To ensure attitude stability, several control paradigms are based on adaptive control [22], H∞ control [23] and sliding mode control [24]. Although effective, it utilises a centralized attitude control system requiring continues information on the attitude trajectory.

In the recent advancements of spacecraft miniaturization, in particular CubeSats purposed for assembling into larger space structures, the option for a distributed attitude control system is researched. The distributed control comes from the notion that these spacecraft are deployed individually into space and each requires an attitude and orbit control system to rendezvous and dock with each other, assembling into a single spacecraft. The attitude stability criteria is then formed into a optimization of the control allocation problem and control algorithm problem. The control allocation problem is well defined and researched for spacecraft, and several algorithms have been proposed [25]. The aforementioned algorithms proposed by [25] uses a centralized node for computations of the control allocation. [26] notes that such a centralized method is challenging for a cellularized spacecraft. The centralized method requires the central node have enough data interfaces with all other nodes for adequate data transfer from distributed sensors to the centralized computer and back to the distributed actuators. This was supported with the research from [27] which gave a solution to this challenge using a distribution control allocation algorithm. Concluding, the allocation strategy used is dependent on the data interface availability and speed of inter-node communication.

With limited data interface structures, decentralised control takes the advantage over centralised control. Decentralised control has been extensively researched for satellite formations. [28] and [29] used Linear-Quadratic Gaussian control for formation flying successfully in numerical simulations. [30] researched decentralised finite-time attitude synchronisation using a modified leader/follower formation approach while [31] used local relative measurements only. The main strength in applying decentralised control techniques for formation flying is that each spacecraft is structurally decoupled and can follow a pre-determined path (or imaginary hub). Assembled spacecraft like Alticube+ will have structurally coupled control actuators, requiring more robust techniques accounting for the structural modes. [32] successfully implemented decentralised control for vibration suppression of solar panels, although the author did mention that control synchronization is needed for effective suppression of the structure.

While decentralised control architectures offer clear advantages for formation flying spacecraft, their applicability becomes more limited once individual spacecraft are mechanically assembled into a single structure. After assembly, systems such as Alticube+ no longer behave as independent agents

but instead form a structurally coupled spacecraft with shared inertia properties and flexible modes. In such cases, the spacecraft dynamics are governed by a single global state, making local control laws insufficient to fully capture the coupling between attitude motion and structural deformation.
The presence of structural flexibility in Alticube+'s aggregated system does not inherently invalidate the use of centralised control. Instead, it motivates the inclusion of dominant flexible modes within the system model. For many flexible spacecraft, only a limited number of low-frequency modes significantly influence attitude dynamics, while higher-order modes remain sufficiently decoupled or naturally damped [33, 34]. By incorporating these dominant modes into a centralised control framework, effective attitude regulation and vibration mitigation can be achieved without resorting to fully distributed control architectures [35, 36]. This approach has been widely adopted in the control of large space structures, flexible satellites, and spacecraft with deployable appendages.
In the specific case of Alticube+, the scientific operation phase assumes that the spacecraft has already been assembled and operates as a single, physically connected system. During this phase, the primary objective is to achieve high pointing accuracy and stability rather than autonomy of individual nodes. A centralised control architecture enables direct utilisation of global attitude measurements, coordinated reaction wheel actuation, and explicit handling of coupled flexible dynamics, all of which are critical to meeting the stringent pointing requirements imposed by the scientific mission.
Furthermore, practical considerations support the selection of centralised control for Alticube+. The availability of attitude sensors on the central node, combined with limited inter-node communication bandwidth, makes real-time distributed state estimation challenging. A centralised controller avoids the need for continuous high-rate data exchange between nodes and reduces implementation complexity while maintaining high control performance. These considerations collectively motivate the adoption of a centralised attitude control strategy for Alticube+ in this research.

Control Algorithm Selection and Motivation

In the context of a semi-rigid microsatellite such as Alticube+, which requires high pointing accuracy ($\leq 0.1^\circ$), sub-arcsecond pointing knowledge, and operates under strict onboard computational and sensing constraints, the selection of an appropriate attitude control algorithm needs to be decided on to fit within the scope of the research. Rather than being driven purely by controller performance, this selection must balance control authority, robustness, computational feasibility, and implementation maturity.

Based on the mission characteristics and system architecture of Alticube+, the control algorithm is required to satisfy the following criteria:

1. Low onboard computational load, compatible with microsatellite-class flight computers [37, 38].
2. Capability to handle multivariable, coupled attitude dynamics, including reaction wheel cross-coupling [39].
3. Compatibility with noisy, low-rate attitude sensors such as gyroscopes, sun sensors, and a single star tracker [40].
4. Ability to incorporate dominant flexible modes to enable modal damping and mitigate structural vibrations [41, 37].
5. Strong analytical foundation and flight heritage to reduce implementation and verification risk [38].

A wide range of attitude control strategies have been proposed in the literature, ranging from classical linear controllers such as Proportional-Integral-Derivative (PID) control to advanced optimal and nonlinear approaches including Model Predictive Control (MPC), $\mathcal{H}_\infty$ control, adaptive control, and reinforcement learning. In practice, PID controllers remain widely used for spacecraft attitude control due to their simplicity and intuitive tuning [42]. However, PID control lacks explicit model-based multivariable feedback and performs poorly when applied to strongly coupled systems or spacecraft with flexible appendages, where structural modes interact with rigid-body dynamics [37]. Furthermore, PID tuning becomes increasingly sensitive in the presence of parameter uncertainty and external disturbances, limiting achievable pointing precision.
Advanced control strategies such as MPC offer the theoretical advantage of explicit constraint handling and multi-objective optimization [43]. However, MPC generally requires the solution of constrained op-

timization problems online, resulting in a computational burden that exceeds the capabilities of typical microsatellite avionics unless significant model simplifications are applied [44, 45]. Similarly, $\mathcal{H}_\infty$ and other robust control techniques provide formal robustness guarantees but often lead to high-order controllers whose performance is sensitive to uncertainty modelling and whose tuning complexity increases rapidly when flexible modes are included [46, 37]. Adaptive and learning-based controllers, including reinforcement learning, have demonstrated promising results in simulation environments; however, their limited flight heritage, need for persistent excitation, and challenges in formal verification currently limit their applicability to risk-averse space missions [47, 48].
Within this context, Linear Quadratic Regulation (LQR) emerges as a particularly well-suited control strategy for Alticube+. LQR provides optimal full-state feedback for linearized multivariable systems, offering an explicit trade-off between control performance and actuator effort through the selection of weighting matrices [39]. Once the gain matrix is computed offline, the online implementation reduces to a simple matrix multiplication, resulting in an extremely low computational burden compatible with real-time operation on microsatellite hardware [37]. Furthermore, LQR naturally accommodates augmented state vectors, allowing dominant flexible modes to be included in the control design for active modal damping, an approach that has been widely adopted in the control of flexible spacecraft and large space structures [41, 49].
When combined with a Kalman filter, LQR forms control framework that enables optimal control performance under stochastic disturbances and noisy sensor measurements [40]. This is particularly relevant for Alticube+, where sensor quality and update rates are limited. Although LQR assumes linear dynamics about an equilibrium point and lacks intrinsic constraint handling, these limitations are acceptable within the scope of the scientific operation phase, which involves relatively small attitude deviations and well-characterized actuator limits.

A qualitative comparison of candidate control algorithms against the selection criteria is summarized in Table 2.2.

Table 2.2: Qualitative comparison of candidate attitude control algorithms for Alticube+.

Controller	Low Comp. Load	MIMO Handling	Flexible Modes	Constraint Handling	Flight Heritage
PID	✓✓	✗	✗	✗	✓✓
MPC	✗	✓	✓	✓	✗
$\mathcal{H}_\infty$	✗	✓	✓	✗	✓
Adaptive / RL	✗	✓	✓	✓	✗
LQR	✓✓	✓	✓	✗	✓✓

Based on this criteria-driven evaluation, LQR represents the most balanced and technically justified control strategy for Alticube+. It uniquely satisfies the combined requirements of low computational complexity, multivariable coupled dynamics, flexible mode damping capability, and strong analytical foundation with extensive flight heritage. For these reasons, a centralized LQR controller, augmented with Kalman filtering for state estimation, is selected as the baseline attitude control architecture for this research.
Despite its advantages, the application of Linear Quadratic Regulation (LQR) to Alticube+ also entails several inherent limitations. First, LQR is fundamentally a model-based control strategy that relies on an accurate linearised state-space representation of the spacecraft dynamics. For flexible spacecraft, uncertainties in mass properties, inertia tensors, and flexible mode characteristics can degrade control performance, particularly when these parameters deviate from their nominal values [37, 39]. Second, LQR is designed around a local linearisation of the dynamics about an equilibrium point, making its performance optimal only for small attitude errors and angular rates. While this assumption is valid during fine-pointing in the science operation phase, larger initial attitude offsets or transient disturbances may result in reduced performance or prolonged convergence, unless gain scheduling or additional control modes are introduced [37]. Third, standard LQR formulations do not explicitly account for actuator constraints such as reaction wheel torque and angular momentum saturation. In practice, this limitation can lead to periods of reduced or lost control authority when actuators saturate, as the controller continues to demand infeasible control inputs [39, 42].

2.5. Research Objective and Thesis Outline

In the previous sections the relevant literature on the scientific payload, calibration techniques, Alticube+'s CubeSat platform ADCS hardware, its capabilities and state of the art for control algorithms has been reviewed. After understanding the core aspects on Alticube+'s attitude control problem, a research framework can be formed. To completely frame the research, a scope of research needs to be outlined.

Firstly, the scientific requirements which are emphasised in the attitude control problem are only applicable during the science operation phase (refer to Section 2.2). Alticube+ has a dedicated mode called "Sci. Ops" in which the spacecraft will operate in during this phase. This mode will then have a certain set of ADCS requirements to satisfy, which is different with respect to other modes. Therefore, the scope of research will be the attitude control problem within the scientific operation phase. The ADCS requirements that need to be satisfied in this research is then limited to the scientific operation mode only. It means that during the phase that Alticube+ should satisfy the phase specific pointing control and pointing knowledge requirements, assuming that the phase is executed without interruption (e.g. changing to safe mode or nominal mode).

Secondly, the scope on the control method for Alticube+ needs to be worked out. Section 2.4 reviewed the state of the art on control allocation problem. Alticube+ being an aggregated system of assembled CubeSats gives it a distinct feature of having a distributed control system. It does not imply that distributed control is a necessity for Alticube+ during the scientific operation phase. Literature showed that distributed attitude control was beneficial using a centralised attitude controller, given ample data interfaces and data transfer were present. Knowing Alticube+ has limited data interfaces with other nodes, it cannot be guaranteed that there is enough data transfer between nodes for sensors and actuators to work together in a decentralised or centralised configuration with distributed control. For this reason, it is important to understand if Alticube+ can be controlled adequately with just the central node's ADCS system available such that all pointing control and knowledge requirements are met. In addition to the control problem, the flexibility of Alticube+'s large baseline structure forms another key interest area. From section 2.2, it is evident that the flexibility influences pointing direction of the baseline and brings additional errors to the ocean height measurements. Alticube+ uses reaction wheel for the control actuation. The jitter produced from these reaction wheels can have interactions with the flexible structure of Alticube+, impacting the pointing control and height measurements. The effect of this interaction on the pointing stability is unknown for a long baseline multi-linked CubeSat configuration such as Alticube+, creating a need to analyse this. To address this key aspect of the control problem, this research will investigate the interaction between reaction wheel jitter and Alticube+'s flexible structure, by analysing the impact this interaction has on pointing stability.

With the research scope and framework outlined, the following research objective can be formed.

Research Objective
To Design a centralised attitude controller for Alticube+'s 2024 concept proposal during its science operation phase and analyse the impact of reaction wheel jitter on the pointing stability for a flexible structure

To achieve this research objective, the research is divided into four supporting research questions. The aim of the supporting research questions is to break down the research objective into key aspects. This creates a tractable research where all research questions can be answered using a clear methodology and a set of deliverables. The first research question stems from the literature, covered by section 2.1 - 2.3. From these sections, it is clear how the scientific requirements translates into the roll requirements. But in order to address the research objective, criteria on pointing stability requirements need to be specified. These criteria originate from the Attitude Determination and Control System (ADCS) requirements, which are derived from the scientific mission objectives. With this, the first research question can be defined.

The second research question focusses on the design of the centralised controller for Alticube+. To

Research Question 1

RQ-1: What are the attitude determination and control system requirements that meet Alticube+'s 2024 concept proposal scientific mission objectives during the scientific operation phase?

first establish a baseline, it is necessary to design and evaluate a centralised attitude controller for Alticube+ in the rigid-body case, prior to extending the analysis to the flexible structure. Based on the literature presented in 2.4, a LQR controller is chosen for the centralised attitude controller. LQR provides a systematic centralised control law for multi-axis spacecraft attitude dynamics, and balances pointing performance against actuator effort. Moreover, LQR can be applied to a rigid spacecraft model and possibly be extended to include flexible mode dynamics if needed. This research will limit its scope to a general LQR controller without the inclusion of flexible modes. LQR is a full-state feedback controller and including the flexible modes would mean that all flexible modes can be observed by the spacecraft sensors. As the scope of the research objective is limited to a centralised attitude controller with centralised sensor capabilities, a general LQR is used for this research. Next to the controller, a Kalman Filter (KF) is adopted to address the pointing knowledge requirements. This KF will be able to use the sensors available to estimate Alticube+'s attitude.
In addition to establishing a baseline performance, different tuning parameters of the LQR cost function be investigated. This allows the assessment of control and estimation performance with respect to the ADCS requirements and provides insight into the capabilities of the centralised controller design for Alticube+. Now the second research question can be presented.

Research Question 2

RQ-2: How can a centralised LQR controller and Kalman filter be designed and tuned for the rigid-body model of Alticube+, and how does this impact their performance to satisfy the ADCS pointing control and knowledge requirements?

- RQ-2a (Controller design): How can an LQR controller be formulated for Alticube+ to achieve centralised attitude regulation of all three axes in the rigid-body case?
- RQ-2b (Controller tuning): How do different weighting matrices in the LQR cost function influence the trade-off between pointing performance and actuator effort?

The third research question will address the interaction between reaction wheel jitter and the flexibility of the spacecraft. The aim of this research question is extend the rigid body model to a flexible model and analyse how reaction wheel jitter impacts pointing control and knowledge compared to the rigid spacecraft model. In practice, only a few modes significantly affect spacecraft attitude response, as high order modes are too fast and filtered out. next to this, including all modes will significantly increase model complexity and is increases computations exponentially. For this reason, this research is restricted to the most dominant flexible modes of Alticube+. Accordingly, research question 3 can be formulated.

Research Question 3

RQ-3: How do the dominant flexible modes of Alticube+ interact with reaction wheel jitter to affect pointing stability?

Having established the ADCS requirements in RQ-1, designed and tuned a centralised LQR controller with Kalman filter for the rigid-body case in RQ-2, and analysed the interaction of dominant flexible modes with reaction wheel jitter in RQ-3, it is important to assess how the the centralised LQR controller performs under uncertainties. In practice, actuator performance, sensor characteristics, and structural properties are never perfectly known and may deviate from the nominal values assumed in the design phase. These uncertainties can shift flexible mode frequencies, introduce bias or noise in attitude measurements, or alter the effective torque output of reaction wheels, all of which can degrade pointing performance. Reflecting on the research objective and the supporting research question RQ-1 to RQ-3, this research will limit the scope to uncertainties in the CubeSat platform mass and inertia matrix. Therefore, a final research question is formulated.

Research Question 4
RQ-4: What is the achievable envelope of pointing performance for Alticube+ under inertia uncertainties?

2.5.1. Thesis Outline

This section will briefly cover the thesis outline. Figure 2.17 shows a visual guide how the thesis is structured. This chapter covered the literature review and expanded on the research objective and its supporting research questions. The remaining chapters of this thesis are dedicated to achieve the research objective and answer the research questions. Chapter 3 will answer the first research question in finding the ADCS requirements. Chapter 4 covers the modelling framework and can be divided into four parts. First a general orbital model is created which will serve as a model framework bases for the satellite model(s). Then a rigid body spacecraft model is derived and expressed through two methods, a non-linear model and a linearized state-space model. The third part will extend to the flexible spacecraft model through the implementation of Kane's equations. Lastly, a reaction wheel model is created which can be integrated into the complete model framework. Chapter 5 creates a controller design using LQR and a Kalman filter. The state-space model from chapter 4 is then used to derive the LQR gain feedback and used in the Kalman filter for its internal prediction. Through the design and tuning of the LQR controller, **RQ-2** can be answered. The results from the simulations are presented in chapter 6 using the flexible model and controller designed in chapter 3 and 4. **RQ-3** and **RQ-4** can then be answered by comparing the performance against the requirements derived in chapter 2. Then having answered all research questions, conclusion on the research objective can be made in chapter 7. Lastly, chapter 8 reflects on the research and proposes recommendations for future research.

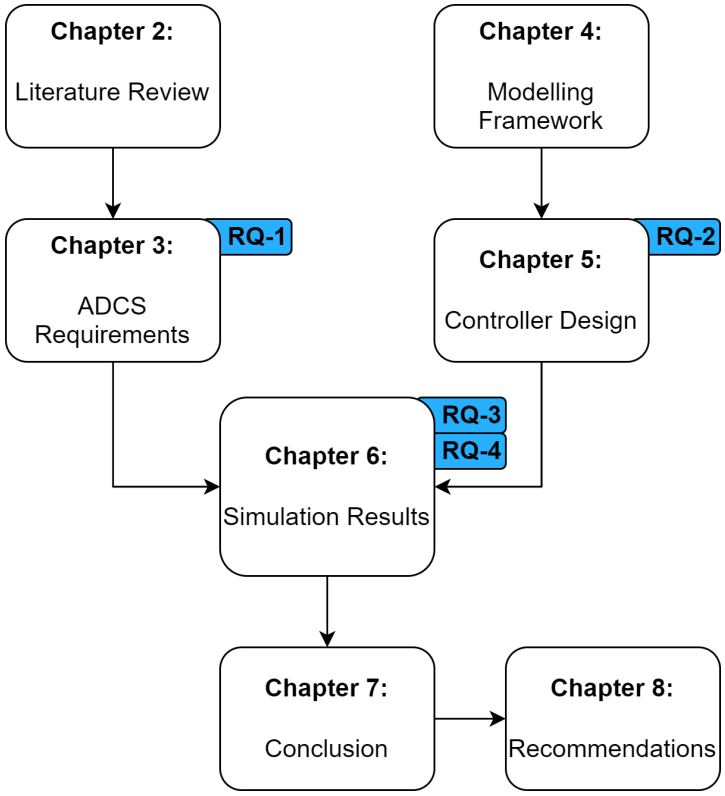


Figure 2.17: Thesis content overview to answer the research questions

3

ADCS Requirements

This chapter will cover the first research question RQ-1: What attitude determination and control system requirements are derived from Alticube+'s scientific mission objectives? To address this research question, the ADCS requirements will be derived from scientific requirements, payload requirements and ADCS hardware utility. As pointed out in section 2.5, the scope of the research is limited to the scientific operations phase. This will then limit the scope for the ADCS requirements derivation to just the scientific operations phase.
Presenting the layout of this chapter, this chapter first covers a brief definition of the pointing errors. Then the ADCS requirements are identified and derived for hardware usage, pointing control and pointing knowledge. Lastly, a summary of all the ADCS requirements is presented.

3.1. Pointing Error Definition

For a satellite in orbit during its lifetime or a certain observation period, different types of pointing errors can be highlighted and defined through the ECSS-E-ST-60-10C standardisation of control performance [50]. The characterisation of time-dependent pointing errors is illustrated in Figure 3.1.

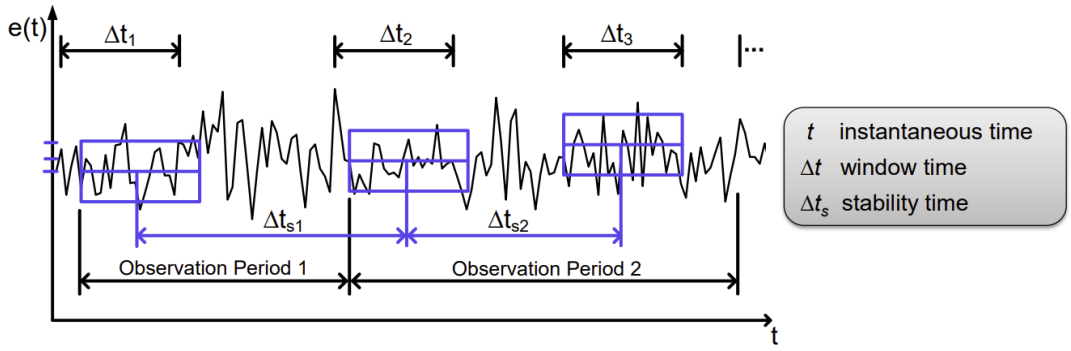


Figure 3.1: Illustration of time dependent pointing errors [51]

Here the **instantaneous time** t defines a pointing error at any point in time during a defined observation period. The **window time** Δt is the pointing error during a time window, which can occur at any moment in time within the defined observation period. The **stability time** Δt_s is the pointing error describing stability (relative error) between pointing errors in two successive time-windows of length Δt. From this framework, using the ECSS terms standard, the definitions of the relevant pointing errors are presented in Table 3.1.

Table 3.1: Pointing error index definition

Index	Name	Definition	Formulation
AKE	Absolute Knowledge Error	Difference between actual attitude and known (measured or estimated) attitude	$AKE(t) = e_K(t)$
APE	Absolute Performance Error	Difference between target attitude and actual attitude	$AKE(t) = e_K(t)$
MKE	Mean Knowledge Error	Mean value of AKE over time interval Δt	$MKE(\Delta t) = \bar{e}_K(\Delta t) = \frac{1}{\Delta t}\int_{\Delta t} e_K(t)dt$
MPE	Mean Performance Error	Mean value of APE over time interval Δt	$MPE(\Delta t) = \bar{e}_P(\Delta t) = \frac{1}{\Delta t}\int_{\Delta t} e_P(t)dt$
PDE	Performance Drift Error	Difference between two MPEs taken over two time intervals separated by a specified time Δt_s, within a single observation period	$PDE(\Delta t_1, \Delta t_2) = \bar{e}_P(\Delta t_2) - \bar{e}_P(\Delta t_1)$
KDE	Knowledge Drift Error	Difference between two MKEs taken over two time intervals separated by a specified time Δt_s, within a single observation period	$PDE(\Delta t_1, \Delta t_2) = \bar{e}_K(\Delta t_2) - \bar{e}_K(\Delta t_1)$

3.2. Requirement Identification and Derivation

Having defined the pointing errors, identification and derivation of the requirements from literature, payload requirements and ADCS hardware utility can be correctly indexed. The complete set of ADCS requirements can be divided up into three categories.

- ADCS hardware usage requirements
- Pointing control (performance) requirements
- Pointing knowledge requirements

Pointing Control

The pointing control requirements for Alticube+ are derived from a combination of mission-level scientific requirements reported in the Alticube+ 2024 concept study [7], hardware limitations of the attitude control subsystem, and additional engineering analyses performed in this thesis. For clarity, the requirements presented in this section are classified as either (i) directly adopted from literature, or (ii) analytically derived within this research to ensure consistency with the mission objectives and system constraints. The primary pointing control requirement is the absolute pointing error (APE), which defines the maximum allowable deviation between the spacecraft attitude and the desired line-of-sight. According to the scientific pointing requirements defined in the Alticube+ concept study [7], the spacecraft shall maintain an APE of no more than $0.2\,\mathrm{deg}$ during science operations. This requirement is adopted directly from literature and is formalised as: **ADCS-PC-REQ-1**: The ADCS system shall have an APE $\leq 0.2\,deg$.

In addition to absolute pointing accuracy, relative pointing errors between antennas forming the $6.6\,\mathrm{m}$ baseline must be constrained to ensure sufficient overlap of the illuminated areas. Guo et al. [7] report that a minimum overlap of 90% requires the combined static and dynamic relative error in torsion and bending to be less than $150\,\mathrm{arcsec}$ in the AT-ND plane and CT-ND plane. This requirement is directly adopted from literature. Bending in the AT-CT plane introduces significantly smaller height errors and is therefore neglected under the assumption of comparable deformation magnitudes. This then generated the following requirements. **ADCS-PC-REQ-2**: The ADCS system shall have a maximum relative error of $150\,arcsec$ between two antennas in the AT-ND plane. **ADCS-PC-REQ-3**: The ADCS system shall have a maximum relative error of $150\,arcsec$ between two antennas in the CT-ND plane.

The pitch stability requirement is derived analytically in this thesis based on the sampling resolution of the scientific payload. To avoid aliasing of the measured ground track, the pitch angle variation during a single measurement period shall not exceed 10% of the spatial sample resolution [7]. For a sample resolution of $2\,\mathrm{km}$ and a measurement period of $0.25\,\mathrm{s}$, this corresponds to a maximum allowable ground displacement of $200\,\mathrm{m}$. Using the equation ($\delta_{ground} = H \tan\theta_{pitch}$) that translates this to a maximum pitch rotation of $82.5\,arcsec$ per $0.25\,s$ measurement period. Now the requirement on pitch stability can be formulated: **ADCS-PC-REQ-4**: The ADCS system shall have a maximum pitch PDE of $330\,\frac{arcsec}{s}$.

Pointing Knowledge

The second category of attitude determination and control system (ADCS) requirements concerns pointing knowledge, which defines the accuracy with which the spacecraft attitude must be estimated. Pointing knowledge requirements are derived from a combination of mission-level scientific observability constraints reported in the Alticube+ concept study [7] and established ADCS design practices. Where requirements are not explicitly specified in the literature, conservative engineering assumptions are adopted and clearly identified.

The most stringent pointing knowledge requirement applies to the roll axis, which directly influences the synthetic aperture radar (SAR) observation geometry and the along-track phase coherence of the measurement. As discussed in Section 2.2, accurate roll attitude knowledge is essential to ensure proper geolocation and scientific data quality. Based on the SAR performance analysis presented in [7], the aggregated spacecraft system shall achieve a roll mean knowledge error (MKE) of $20\,\mathrm{arcsec}$ RMS over the full measurement track, with an allowable maximum MKE of $100\,\mathrm{arcsec}$ RMS. This leads to the following requirement: **ADCS-PK-REQ-1**: The aggregated system shall have a roll angle MKE of $20\,\mathrm{arcsec}$ RMS over the measurement track, or a maximum MKE of $100\,\mathrm{arcsec}$ RMS.

For the pitch and yaw axes, the literature does not provide explicit mission-level pointing knowledge requirements. However, accurate knowledge in these axes remains necessary to support stable closed-loop attitude control and post-processing of scientific data. In spacecraft ADCS design, it is common practice to require attitude knowledge accuracy to be significantly tighter than pointing control accuracy, typically by a factor of two to ten, in order to maintain sufficient observability margin and estimator robustness.In this research, a conservative and symmetric approach is adopted by setting the pitch and yaw knowledge requirements equal to the maximum allowable roll knowledge error defined in ADCS-PK-REQ-1. This assumption ensures that attitude estimation errors do not become a limiting factor for control performance or scientific data quality. Accordingly, the following requirements are defined: **ADCS-PK-REQ-2**: The aggregated system shall have a maximum pitch angle MKE of $100\,\mathrm{arcsec}$ RMS. **ADCS-PK-REQ-3**: The aggregated system shall have a maximum yaw angle MKE of $100\,\mathrm{arcsec}$ RMS.

In addition to absolute attitude knowledge, the relative pointing knowledge between two antennas forming a single baseline must be sufficiently accurate to enable correct reconstruction of the interferometric geometry. According to the Alticube+ concept study [7], the relative pointing knowledge error between antennas shall not exceed $100\,\mathrm{arcsec}$ RMS. This results in the final pointing knowledge requirement: **ADCS-PK-REQ-4**: The aggregated system shall have a maximum relative pointing knowledge error of $100\,\mathrm{arcsec}$ RMS between two antennas forming a baseline.

Hardware Usage

This subsection defines the hardware usage requirements adopted in this research. Unlike the pointing control and pointing knowledge requirements, which are primarily derived from scientific mission objectives and literature, the hardware usage requirements are defined explicitly within the scope of this master's thesis. Their purpose is to constrain the modelling and control design to a realistic and tractable configuration that is consistent with the Alticube+ concept study [7], while avoiding unnecessary complexity beyond the research objectives. As presented in Section 2.3, the ADCS architecture of Alticube+ consists of reaction wheels and magnetorquers as actuators, and a combination of star trackers, gyroscopes, and sun sensors for attitude determination. During the scientific operation phase, precise and continuous attitude control is required, while orbital control and momentum dumping activities are not expected to be active. The main purpose of the reaction wheels is to point Alticube+ in the correct direction. The other actuator, magneto torquer, is used for de-saturation of the reaction wheels. This leads to the following hardware usage requirement: **ADCS-HW-REQ-1**: The ADCS system shall utilize reaction wheels as the sole actuators for attitude control during the scientific operation phase.

Another important consideration is the available power budget. According to the Alticube+ concept

study [7], the average allocated power for the Attitude and Orbit Control System (AOCS) over an orbital period is $5.3\,\mathrm{W}$. During the scientific operation phase considered in this research, it is assumed that orbit control functions are inactive, allowing the full allocated AOCS power budget to be used by the ADCS. Furthermore, the central node of the aggregated Alticube+ system does not host scientific payloads, leaving an additional estimated $22\,\mathrm{W}$ of available power margin. Given that this research focuses on a centralised ADCS architecture rather than detailed power management, the available power margin is considered sufficient and non-limiting. As a result, no explicit power usage requirement is imposed within the scope of this thesis. For attitude determination, each CubeSat platform is equipped with a star tracker, gyroscope, and three sun sensors. However, due to limited inter-satellite data interfaces and the adoption of a centralised control architecture, it is not feasible to fuse sensor data from all nodes in real time during scientific operations. To maintain a realistic and implementable control framework, attitude estimation is therefore restricted to the sensors located on the central node. This results in the final hardware usage requirement: **ADCS-HW-REQ-2**: Attitude estimation of Alticube+ shall be limited to the attitude sensors located on the central node during the scientific operation phase.

3.3. ADCS Requirements Table

With all relevant requirements identified, a table summarising the ADCS requirements can be seen in 3.2.

Table 3.2: List of ADCS requirements applicable to the scientific operation phase

Requirement ID	Description	Notes
ADCS-PC-REQ-1	The ADCS system shall have an APE $\leq 0.2\,deg$.	Pointing control
ADCS-PC-REQ-2	The ADCS system shall have a maximum relative error of $150\,arcsec$ between two antennas in the ATND plane.	Pointing control
ADCS-PC-REQ-3	The ADCS system shall have a maximum relative error of $150\,arcsec$ between two antennas in the CTND plane.	Pointing control
ADCS-PC-REQ-4	The ADCS system shall have a maximum pitch PDE of $330\,\frac{arcsec}{s}$.	Pointing control
ADCS-PK-REQ-1	The aggregated system shall have a roll angle MKE of $20\,arcsec$ RMS over the whole measurement track, or a maximum MKE of $100\,arcsec$ RMS.	pointing knowledge
ADCS-PK-REQ-2	The aggregated system shall have a maximum pitch angle MKE of $100\,arcsec$ RMS.	pointing knowledge
ADCS-PK-REQ-3	The aggregated system shall have a maximum yaw angle MKE of $100\,arcsec$ RMS.	pointing knowledge
ADCS-PK-REQ-4	The aggregated system shall have a maximum relative error between two antenna's that make up one baseline of $100\,arcsec$ RMS.	pointing knowledge
ADCS-HW-REQ-1	The ADCS system shall only utilize reaction wheels during the scientific operation phase.	ADCS hardware
ADCS-HW-REQ-2	Attitude estimation of Alticube+ shall be limited to the attitude sensors from the central node.	ADCS hardware

4

Modelling Framework

Answering the remaining research questions will require a modelling framework. This framework consists of three model frameworks: orbit model, spacecraft model and reaction wheel model. The orbital model defines the orbital reference frame during the scientific operation phase and provides a baseline for the spacecraft's payload pointing. The spacecraft model defines the dynamics and kinematics of Alticube+ for both rigid space model and flexible model. Lastly, the reaction wheel model ensures realistic behaviour of the actuator, including hardware limitations and jitter.

4.1. Orbital Model

The orbit of Alticube+ has already been defined in section 2.1. To fully define the orbital model framework, several relevant reference frames need to be defined first. The first relevant reference frame is the Earth centred inertial reference frame (ECI), which forms the basis of all reference frames that are used in the complete modelling framework and defined as $\mathcal{F}_I \in \mathbb{R}^3$. Here, $\hat{I}_x$ and $\hat{I}_z$ are defined as unit vectors pointing towards the vernal equinox and celestial pole from the mean J2000 respectively, completed by $\hat{I}_y = \hat{I}_z \times \hat{I}_x$. Within this inertial reference frame, the sun vector $\hat{e}_\odot$ moves around the ecliptic plane as seen in Figure 4.1. The ecliptic plane is defined by the normal vector $\hat{e}_{\omega_\odot}$ that is rotated around the the $\hat{I}_x$ axis by $i_\odot = 23.5$ degrees from the $\hat{I}_z$ axis. Then the sun vector rotates around this plane. The sun vector $\hat{e}_\odot$ can be mathematically defined by equation 4.1.

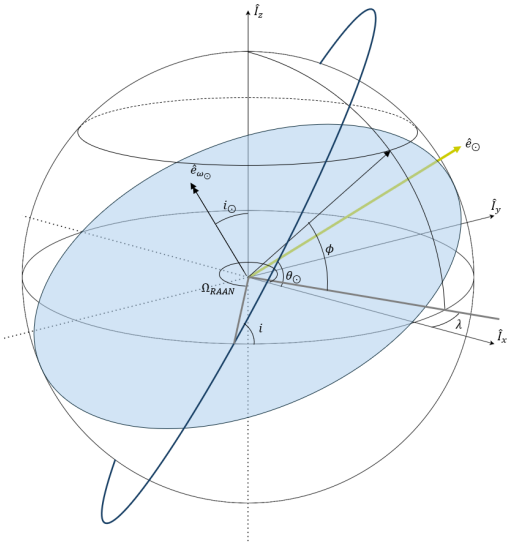


Figure 4.1: Reference Frame convention for an Earth centred inertial reference frame

$$\hat{e}_\odot = \boldsymbol{T}_{\mathcal{I}}^{\odot} \hat{I}_x \quad ; \quad \boldsymbol{T}_{\mathcal{I}}^{\odot} = \begin{bmatrix} \cos\theta_\odot & -\sin\theta_\odot & 0 \\ \cos i_\odot \sin\theta_\odot & \cos i_\odot \cos\theta_\odot & -\sin i_\odot \\ \sin i_\odot \sin\theta_\odot & \sin i_\odot \cos\theta_\odot & \cos i_\odot \end{bmatrix} \tag{4.1}$$

Here, the rotating motion of the sun vector can be expressed by equation 4.2.

$$\theta_\odot(t) = \omega_\odot(t - t_0) + \theta_\odot(t_0) \quad ; \quad \omega_\odot \approx 7.292 \cdot 10^{-5} \, \frac{rad}{s} \tag{4.2}$$

The second relevant reference frame is the Orbit-Fixed Reference Frame (OFRF). This reference frame is defined by several properties as presented in Figure 4.1. Obtaining the rotation matrix requires the

transformation matrix seen in Equation 4.3 where two consecutive rotations are performed, first around the $\hat{I}_x$ axis and then the $\hat{I}_z$ axis.

$$\boldsymbol{T}_{\mathcal{I}}^{OF} = \begin{bmatrix} \cos\Omega_{RAAN} & -\sin\Omega_{RAAN}\cos i & \sin\Omega_{RAAN}\sin i \\ \sin\Omega_{RAAN} & \cos\Omega_{RAAN}\cos i & -\cos\Omega_{RAAN}\sin i \\ 0 & \sin i & \cos i \end{bmatrix} \tag{4.3}$$

The inclination defined for Alticube is $i = 78\,deg$ and is a Sun Synchronous orbit (SSO) with the Right Ascension of the Ascending Node (RAAN) at 10 AM. This means that Ω_{RAAN} moves along the equator following the sun vector with the same speed, meaning that equation 4.2 can be used, including an offset that keeps the 10 AM RAAN $\Omega_{RAAN}(t) = \theta_{\odot}(t) - \frac{\pi}{6}$.

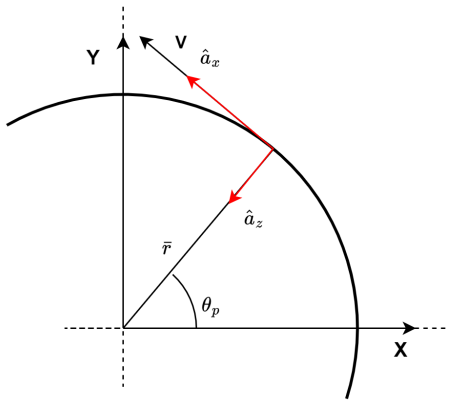


Figure 4.2: Example of nadir orbital path reference frame in orbit

The last reference frame within the orbital model is the Orbital Path Reference Frame (OPRF) defined as $\mathcal{F}_a \in \mathbb{R}^3$. This reference frame defines the reference frame of orbital path at the position of the satellite relative to the OFRF. The three OPRF axis are defined as $\hat{a}_x$, $\hat{a}_y$ and $\hat{a}_z$, where each are pointing along track, cross track and nadir respectively (See Figure 4.2 for reference). Achieving this reference frame requires two transformations. First an eigenaxis rotation around the normal vector of the orbital plane is performed. The rotation matrix is defined as the directional cosine matrix based on the eigenaxis rotation as described in equation 4.4.

$$\boldsymbol{T}_{OF}^{OP} = \begin{bmatrix} \cos\theta_p & -\sin\theta_p & 0 \\ \sin\theta_p & \cos\theta_p & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \tag{4.4}$$

In equation 4.4, the first reference rotation puts the $\hat{a}_x$, $\hat{a}_y$ and $\hat{a}_z$ axis towards the correct orientation for an orbit in the $\hat{I}_x - \hat{I}_y$ plane. Then the second matrix rotates the reference frame according with an angle θ_p according to the current orbit path and is dependent on the orbit properties. For a circular orbit, this equation would be:

$$\theta_p(t) = n(t - t_0) + \theta_p(t_0) \quad ; \quad n = \sqrt{\frac{\mu_{\oplus}}{(R_{\oplus} + H)^3}} \tag{4.5}$$

Where $\mu_{\oplus}$ is the gravitational constant of Earth, $R_{\oplus}$ is the Earth's radius and H is the orbital height. After this transformation, the last matrix from equation 4.3 is applied to reform is into the correct orbit using the following product convention: $\boldsymbol{T}_{\mathcal{I}}^{OP} = \boldsymbol{T}_{\mathcal{I}}^{OF}\boldsymbol{T}_{\mathcal{OF}}^{OP}$

External Disturbance Torques

Disturbance torques are unavoidable external driven forces that perturb a spacecraft from its desired attitude and must be counteracted by the attitude control system to maintain stable pointing. These torques arise from a variety of sources. Effects such as gravity-gradient, aerodynamic drag, solar radiation pressure, and Earth' s magnetic field. Although often small in magnitude, disturbance torques can accumulate over time, leading to attitude deviations if not properly mitigated. Therefore, the inclusion of disturbance torques is necessary to determine the attitude controller's capability in satisfying the ADCS requirements while dealing with quasi- real life disturbance torque scenario's (i.e. simplified model of disturbance torques) . The total disturbance torque is the sum of contributing disturbance torques as seen in equation 4.6.

$$\bar{\tau}_d = \bar{\tau}_{gg} + \bar{\tau}_{drag} + \bar{\tau}_{mag} + \bar{\tau}_{solar} \tag{4.6}$$

Gravity Gradient Torque

Earth's gravity is radially a non-homogeneous field, which can induce torques for spacecraft with non-symmetrical structure. Spacecraft structures which lie below the centre of mass will experience a

greater force of attraction to Earth than structures that lie farther away than the centre of mass. This very small differential in gravity introduces a rotational force, called gravity gradient torque. The derivation of this torque has been mathematically done before and is the resulting equation is given by [52].

$$\bar{\tau}_{gg} = 3n^2 \hat{a}_z \times \boldsymbol{J} \hat{a}_z \tag{4.7}$$

Here, $\boldsymbol{J}$ is the inertial matrix of the spacecraft in the inertial reference frame $\mathcal{F}_{\mathcal{I}}$.

Aerodynamic Torque

Although very small, the atmosphere is present in Low Earth Orbit (LEO) and impacts the spacecraft in the form of atmospheric drag. The drag formula can be written in the standard form as seen in equation 4.8.

$$\bar{f}_D = -\frac{1}{2}\rho \frac{\mu}{R_\oplus + H} C_D A_{eff}\, \hat{a}_x \tag{4.8}$$

Here, ρ is the atmospheric density. For an orbit at $500\,km$, the exosphere has a mean density is roughly $\rho \approx 4.76 \cdot 10^{-13}\ \frac{kg}{m^3}$ and a maximum density of $\rho \approx 2.82 \cdot 10^{-12}\ \frac{kg}{m^3}$ [53]. C_D is the drag coefficient, which is assumed to be 2.2 for general cubesats in LEO. Lastly, A_{eff} is the effective area of the spacecraft in the along track ($\hat{a}_x$) direction. Then the torque can be obtain by applying the cross product between the drag force and lever distance from the centre of mass to the centre of pressure.

$$\bar{\tau}_{drag} = \bar{r}_{cp} \times \bar{f}_D \tag{4.9}$$

Solar Pressure

The solar pressure that is experienced by the spacecraft during orbit generates a small force. This force can be expressed by equation 4.10.

$$\bar{f}_{solar} = -\frac{S A_S}{c} \hat{e}_\odot \tag{4.10}$$

Here, S is the solar constant in $\frac{W}{m^2}$, c is the speed of light and A_S is the area exposed to the solar pressure. From equation 4.10, it can be observed that the vector is driven by the sun vector, pointing in the opposite direction relative to the inertial reference frame $\mathcal{F}_{\mathcal{I}}$. For this research, the solar constant is assumed to be $1400\ \frac{W}{m^2}$, over estimating the solar pressure slightly. To obtain the resulting solar pressure torque, just like in equation 4.9, the cross product is taken with the centre of pressure arm.

$$\bar{\tau}_{solar} = \bar{r}_{cp} \times \bar{f}_{solar} \tag{4.11}$$

Magnetic Torque

For LEO satellites, Earth's magnetic field influences spacecraft through a residual dipole that is present for all satellites containing electrical wires or solar panels. The torque generated by the magnetic field and residual pole is described by equation 4.12.

$$\bar{\tau}_{mag} = \bar{m} \times \bar{B}_\oplus \tag{4.12}$$

Here, $\bar{m}$ is the residual dipole moment and $\bar{B}_\oplus$ is the magnetic field of Earth. Modelling Earth's magnetic field will be simplified with respect to the integral version of the geomagnetic model from the National Oceanic and Atmospheric Administration (NOAA). It will be assumed that the magnetic field vector will be in line with true north and orthogonal with respect to nadir. The magnetic field strength between latitudes $75^\circ\, S - 75^\circ\, N$ assumed to have a mean magnitude of $35\,\mu T$. The residual dipole moment of Alticube is based on [54] which states that the 3U Dart CubeSat had a measured residual dipole moment of $9.0 \cdot 10^{-3} Am$. Taking in to account that Alticube+ has five Cubesats, the residual dipole moment is assumed to be $0.05\,Am$.

4.2. Rigid Spacecraft Model

Section 4.1 presented the orbital model framework, where the various reference frames from the inertial reference frame to the orbital path reference frame was defined. This section will cover the framework of the spacecraft model, specifically the rigid spacecraft model. First, the definition on Alticube+'s spacecraft reference frame is defined. Then the mathematical convention that couples the orbital path reference frame to the spacecraft reference frame is derived for both Euler angels and quaternions. Then the relevant rigid body dynamics and kinematics are derived which will provide the mathematical basis of the spacecraft model framework. Lastly, the non-linear dynamics are linearised into a linear state-space model.

4.2.1. Spacecraft Reference Frame

Relating the Spacecraft body reference frame to the orbital reference frame is essential for spacecraft kinematics. This can be done in two ways, using Euler angles and quaternions. Euler angles are used to validate the performance of the spacecraft related to the requirements discussed in chapter 3. Quaternions are utilised for the spacecraft model to describe the dynamics and kinematics. But first, the basis of the reference frame needs to be defined. The body reference frame using the vector basis $\{\hat{b}_x, \hat{b}_y, \hat{b}_z\} \in \mathbb{R}^3$ is related relative to the OPRF with the vector basis $\{\hat{a}_x, \hat{a}_y, \hat{a}_z\} \in \mathbb{R}^3$. Transformation between these reference frames can be done using a Directional Cosine Matrix (DCM) as seen in equation 4.13.

$$\begin{bmatrix} \hat{b}_x \\ \hat{b}_y \\ \hat{b}_z \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{bmatrix} \tag{4.13}$$

The DCM in terms of the Euler angle rotations $\bar{\theta} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T$ can be expressed by three individual rotation matrix around each individual OPRF axis.

$$\begin{bmatrix} \hat{b}_x \\ \hat{b}_y \\ \hat{b}_z \end{bmatrix} = \boldsymbol{C}_z(\theta_3)\boldsymbol{C}_x(\theta_1)\boldsymbol{C}_y(\theta_2) \begin{bmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{bmatrix} \tag{4.14}$$

Where,

$$\boldsymbol{C}_x(\theta_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 \\ 0 & \sin\theta_1 & \cos\theta_1 \end{bmatrix} \quad \boldsymbol{C}_y(\theta_2) = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \quad \boldsymbol{C}_z(\theta_3) = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The DCM can also be used to compute the Euler angles if both orbital path and body reference frame vector basis are known: $\bar{\theta} = \begin{bmatrix} \arcsin(C_{32}) & \arctan\left(-\frac{C_{31}}{C_{33}}\right) & \arctan\left(-\frac{C_{12}}{C_{22}}\right) \end{bmatrix}^T$ With the Euler angle method defined, the quaternion method will be defined next. First a definition of the quaterion vector and its constraint is given as follows.

$$\bar{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T \quad ; \quad \bar{q}^T\bar{q} = 1 \tag{4.15}$$

For the quaternions, a DCM using the quaternions is given by equation 4.16 [55].

$$\begin{bmatrix} \hat{b}_x \\ \hat{b}_y \\ \hat{b}_z \end{bmatrix} = \boldsymbol{C}_a^b \begin{bmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{bmatrix} \quad ; \quad \boldsymbol{C}_a^b = \left(q_4^2 - \bar{q}^T\bar{q}\right)\boldsymbol{I} + 2\bar{q}\bar{q}^T - 2q_4\bar{q}^\times \tag{4.16}$$

Expanding both sides to its matrix form will give the following equation

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2+q_3q_4) & 2(q_1q_3-q_2q_4) \\ 2(q_1q_2-q_3q_4) & 1-2(q_1^2+q_3^2) & 2(q_2q_3+q_1q_4) \\ 2(q_1q_3+q_2q_4) & 2(q_2q_3-q_1q_4) & 1-2(q_1^2+q_2^2) \end{bmatrix} \tag{4.17}$$

4.2.2. Rigid Body Dynamics and Kinematics

In this section, the rigid body dynamics for spacecraft is discussed along with the associated kinematics. As presented in section 4.2.1, the kinematics between two reference frames can be expressed by Euler angles or quaternions through the DCM. For the remainder of this section, quaternions will be used to express the spacecraft kinematics of this spacecraft. Starting with the kinematics, a relationship between the the rotation rate for the body reference frame $\mathcal{F}_b$ relative to the OPRF $\mathcal{F}_a$ is derived to fully describe the kinematics for a rigid spacecraft. Let the angular rate of the body reference frame $\mathcal{F}_b$ relative to OPRF $\mathcal{F}_a$ be described as $\bar{\omega}^{\mathcal{F}_b/\mathcal{F}_a}$. For equation 4.18 until equation 4.20, the notation for the angular rate will be simplified to: $\bar{\omega} = \bar{\omega}^{\mathcal{F}_b/\mathcal{F}_a}$. The first order derivative of the body reference frame vectors $\begin{bmatrix}\hat{b}_x & \hat{b}_y & \hat{b}_z\end{bmatrix}^T$ can be expressed by equation 4.18.

$$\begin{bmatrix}\dot{\hat{b}}_x \\ \dot{\hat{b}}_y \\ \dot{\hat{b}}_z\end{bmatrix} = \begin{bmatrix}\bar{\omega} \times \hat{b}_x \\ \bar{\omega} \times \hat{b}_y \\ \bar{\omega} \times \hat{b}_z\end{bmatrix} \quad ; \quad \bar{\omega} = \omega_1\hat{b}_x + \omega_2\hat{b}_y + \omega_3\hat{b}_z \tag{4.18}$$

Remembering equation 4.13, inverting the DCM to the other side of the equation and taking the derivative will give equation 4.19.

$$\begin{bmatrix}\hat{a}_x \\ \hat{a}_y \\ \hat{a}_z\end{bmatrix} = \boldsymbol{C}^{-1}\begin{bmatrix}\hat{b}_x \\ \hat{b}_y \\ \hat{b}_z\end{bmatrix} = \boldsymbol{C}^T\begin{bmatrix}\hat{b}_x \\ \hat{b}_y \\ \hat{b}_z\end{bmatrix} \Longrightarrow \begin{bmatrix}\bar{0} \\ \bar{0} \\ \bar{0}\end{bmatrix} = \dot{\boldsymbol{C}}^T\begin{bmatrix}\hat{b}_x \\ \hat{b}_y \\ \hat{b}_z\end{bmatrix} + \boldsymbol{C}^T\begin{bmatrix}\bar{\omega} \times \hat{b}_x \\ \bar{\omega} \times \hat{b}_y \\ \bar{\omega} \times \hat{b}_z\end{bmatrix} = \dot{\boldsymbol{C}}^T\begin{bmatrix}\hat{b}_x \\ \hat{b}_y \\ \hat{b}_z\end{bmatrix} - \boldsymbol{C}^T\begin{bmatrix}0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & \omega_1 \\ -\omega_2 & \omega_1 & 0\end{bmatrix}\begin{bmatrix}\hat{b}_x \\ \hat{b}_y \\ \hat{b}_z\end{bmatrix}$$

By reducing this to the matrix form, collecting terms and transposing the equation simplifies the equation.

$$\boldsymbol{0}_{3\times3} = (\dot{\boldsymbol{C}}^T - \boldsymbol{C}^T\boldsymbol{\Omega})\begin{bmatrix}\hat{b}_x \\ \hat{b}_y \\ \hat{b}_z\end{bmatrix} \Longrightarrow \dot{\boldsymbol{C}}^T = \boldsymbol{C}^T\boldsymbol{\Omega} \Longrightarrow \dot{\boldsymbol{C}} = \boldsymbol{\Omega}^T\boldsymbol{C} \Longrightarrow \dot{\boldsymbol{C}} - \boldsymbol{\Omega}^T\boldsymbol{C} = 0 \quad ; \quad \dot{\boldsymbol{C}} = \begin{bmatrix}\dot{C}_{11} & \dot{C}_{12} & \dot{C}_{13} \\ \dot{C}_{21} & \dot{C}_{22} & \dot{C}_{23} \\ \dot{C}_{31} & \dot{C}_{32} & \dot{C}_{33}\end{bmatrix}$$

$$\dot{\boldsymbol{C}} + \boldsymbol{\Omega}\boldsymbol{C} = 0 \quad ; \quad \boldsymbol{\Omega}^T = -\boldsymbol{\Omega} \quad ; \quad \boldsymbol{\Omega} = \begin{bmatrix}0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & \omega_1 \\ -\omega_2 & \omega_1 & 0\end{bmatrix} \tag{4.19}$$

Computing the derivative of equation 4.17 and substituting equation 4.17 and its derivative into equation 4.19, $\dot{\bar{q}}$ can be computed.

$$\dot{\bar{q}} = \boldsymbol{\Lambda}_q\begin{bmatrix}\omega_1 \\ \omega_2 \\ \omega_3 \\ 0\end{bmatrix} = \boldsymbol{\Lambda}_\omega\bar{q} \quad \Longleftrightarrow \quad \begin{bmatrix}\omega_1 \\ \omega_2 \\ \omega_3 \\ 0\end{bmatrix} = \boldsymbol{\Lambda}_q^{-1}\dot{\bar{q}} \tag{4.20}$$

Where

$$\boldsymbol{\Lambda}_q = \begin{bmatrix}q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4\end{bmatrix} \quad ; \quad \boldsymbol{\Lambda}_\omega = \begin{bmatrix}0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0\end{bmatrix}$$

Rotating OPRF Kinematics

The kinematics described above relate the angular rates of the body with respect to $\mathcal{F}_a$. To describe the dynamics in the Inertial frame $\mathcal{F}_\mathcal{I}$, the total angular rate of the spacecraft relative to the inertial reference frame $\bar{\omega}^{\mathcal{F}_b/\mathcal{F}_\mathcal{I}}$ needs to be explored. For Alticube+, as described in section 4.1, the spacecraft will enter a circular orbit, where the OPRF ($\mathcal{F}_a$) points along-track, cross-track and nadir (See Figure 4.2). Knowing this information, the angular rate of the body relative to the inertial reference frame can be described by.

$$\bar{\omega}^{\mathcal{F}_b/\mathcal{F}_\mathcal{I}} = \bar{\omega}^{\mathcal{F}_b/\mathcal{F}_a} + \bar{\omega}^{\mathcal{F}_a/\mathcal{F}_\mathcal{I}} \tag{4.21}$$

The angular rate of the OPFR with respect to $\mathcal{F}_\mathcal{I}$ is a clockwise rotation around the $\hat{a}_y$ with a magnitude of n. Equation 4.22 then describes the full angular rate of the body.

$$\bar{\omega}^{\mathcal{F}_b/\mathcal{F}_\mathcal{I}} = \mathbf{\Lambda}_q^{-1}\dot{\bar{q}} - n\begin{bmatrix}\hat{a}_y \\ 0\end{bmatrix} \tag{4.22}$$

Here, $\hat{a}_y = C_{12}\hat{b}_x + C_{22}\hat{b}_y + C_{32}\hat{b}_z$Now isolating $\dot{\bar{q}}$ to the left side and substituting the DCM elements from equation 4.17, equation 4.23 can be described.

$$\dot{\bar{q}} = \frac{1}{2}\mathbf{\Lambda}_q\bar{\omega} + \frac{n}{2}\mathbf{\Lambda}_q\begin{bmatrix}2(q_1q_2 + q_3q_4) \\ 1 - 2(q_1^2 + q_3^2 \\ 2(q_2q_3 - q_1q_4) \\ 0\end{bmatrix} \tag{4.23}$$

Reducing the second term of the right hand-side of the equation will give the following. Derivation of equation 4.24 can be found in Appendix A.

$$\dot{\bar{q}} = \frac{1}{2}\mathbf{\Lambda}_q\bar{\omega} + \boldsymbol{N}_q\bar{q} \quad ; \quad \boldsymbol{N}_q = \frac{n}{2}\begin{bmatrix}\mathbf{0}_{2\times2} & \boldsymbol{I}_{2\times2} \\ -\boldsymbol{I}_{2\times2} & \mathbf{0}_{2\times2}\end{bmatrix} \tag{4.24}$$

Dynamics

Now that the full kinematics for Alticube+ in a circular orbit is described, rigid body dynamics is looked at. The equation of motion of a satellite that rotates in space is presented in equation 4.25 [55].

$$\boldsymbol{J}\dot{\bar{\omega}} + \mathbf{\Omega}\boldsymbol{J}\bar{\omega} = \bar{\tau} \tag{4.25}$$

Where $\bar{\omega} = \bar{\omega}^{\mathcal{F}_b/\mathcal{F}_\mathcal{I}}$, $\boldsymbol{J}$ is the inertia matrix of Alticube+ and $\bar{\tau}$ is the torque. Isolating $\dot{\bar{\omega}}$ to the left side, equation 4.26 is obtained.

$$\dot{\bar{\omega}} = \boldsymbol{J}^{-1}\bar{\tau} - \boldsymbol{J}^{-1}\mathbf{\Omega}\boldsymbol{J}\bar{\omega} \tag{4.26}$$

4.2.3. Non-linear Plant

With both kinematics and spacecraft dynamics described, they can be merged into a non-linear differential equation as described in equation 4.27.

$$\dot{\bar{x}} = f(\bar{x}) + g(\bar{u}) \tag{4.27}$$

Here, $f(\bar{x})$ represents the dynamics function with $\bar{x} = \begin{bmatrix}\bar{q} & \bar{\omega}\end{bmatrix}^T$ being the state vector of the spacecraft. $g(\bar{u})$ represents the input function for the torque input $\bar{u} = \begin{bmatrix}\bar{\tau}_c & \bar{\tau}_d\end{bmatrix}^T$ which consists of both control and disturbance torques. The functions $f(\bar{x})$ and $g(\bar{u})$ are defined by using the derived equations 4.24 and 4.26.

$$f(\bar{x}) = \begin{bmatrix}\frac{1}{2}\mathbf{\Lambda}_q\bar{\omega} + \boldsymbol{N}_q\bar{q} \\ -\boldsymbol{J}^{-1}\mathbf{\Omega}\boldsymbol{J}\bar{\omega}\end{bmatrix} \quad ; \quad g(\bar{u}) = \begin{bmatrix}0 & 0 \\ \boldsymbol{J}^{-1} & \boldsymbol{J}^{-1}\end{bmatrix}\bar{u}$$

The function of the non-linear plant is to model the spacecraft dynamics and kinematics accurately. Figure 4.3 is a schematic of the block digram for the non-linear model in Simulink.

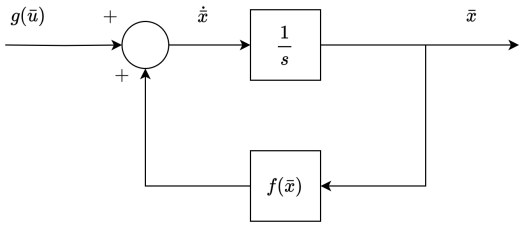


Figure 4.3: Simulink block diagram of non-linear plant model.

4.2.4. Linearised State-Space Model

Although the non-linear model accurately represents the dynamics and kinematics of Alticube, a linearised model simplifies the spacecraft dynamics and allows for LQR control. The other function of a linearised state-space model is that the Kalman filter can use this to perform predictions of the true attitude modelled by the non-linear dynamics. Before linearising the state space, a couple of changes and assumptions should be clarified. Firstly, a change in the input function from equation 4.27 is made. The linearised model cannot directly model disturbance torques as LQR and the Klaman filter cannot directly influence and predict disturbance torques. This means that the input function from 4.27 is changed to $g(\bar{u}) = \boldsymbol{J}^{-1}\bar{u}$ where $\bar{u} = \bar{\tau}_c$. Secondly, an assumption is made on the linearisation. The state-space will balance around the $\mathcal{F}_a$ reference frame. It then follows from equation 4.14 that the state-space model needs to be linearised around $\bar{\theta}_{\ell,0} = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$. The angular rate then needs to be linearised around $\bar{\omega}_{\ell,0} = \begin{bmatrix}0 & -n & 0\end{bmatrix}^T$ to account for the orbital rotation around the $\hat{a}_y$ axis. Although the linearisation point is obtained in terms of Euler angles, it is required to be transformed in terms of quaternions in order to apply this to the non-linear plant. Starting from the transformation between linearised Euler angles and quaternions.

$$\delta\bar{q} = \frac{\delta\bar{\theta}}{2} \tag{4.28}$$

To linearise the non-linear plant, a Taylor expansion is taken around the linearisation point, which is $\bar{x}_{\ell,0} = \begin{bmatrix}\bar{\theta}_{\ell,0}^T & \bar{\omega}_{\ell,0}^T\end{bmatrix}^T$. For a more neat derivation, the non-linear plant, from equation 4.27 can be separated again, and a Taylor expansion can be done for equation 4.24 and equation 4.26 and will be represented in the following manner.

$$\dot{\bar{q}} = f(\bar{q},\bar{\omega}) \quad ; \quad \dot{\bar{\omega}} = g(\bar{\omega},\bar{u})$$

Starting with the linearisation of $\dot{\bar{q}}$, equation 4.29 shows the Taylor expansion of this function.

$$\dot{\bar{q}}_\ell = f(\bar{q}_{\ell,0},\bar{\omega}_{\ell,0}) + \frac{\partial f}{\partial\bar{q}}(\bar{q}_{\ell,0},\bar{\omega}_{\ell,0})\delta\bar{q} + \frac{\partial f}{\partial\bar{\omega}}(\bar{q}_{\ell,0},\bar{\omega}_{\ell,0})\delta\bar{\omega} + \mathcal{O}(\delta\bar{q}^2) + \mathcal{O}(\delta\bar{\omega}^2) \tag{4.29}$$

Here, the second order terms will be neglected. Now first the derivation of the first order derivative with respect to $\bar{q}$ is done.

$$\frac{\partial f}{\partial\bar{q}} = \frac{1}{2}\frac{\partial(\boldsymbol{\Lambda}_q\bar{\omega})}{\partial\bar{q}} + \frac{\partial(\boldsymbol{N}_q\bar{q})}{\partial\bar{q}} \tag{4.30}$$

Remembering equation 4.20, the first term of equation 4.30 can be changed: $\boldsymbol{\Lambda}_q\bar{\omega} \leftrightarrow \boldsymbol{\Lambda}_\omega\bar{q}$. Applying this to the first term will give:

$$\frac{\partial f}{\partial\bar{q}} = \frac{1}{2}\boldsymbol{\Lambda}_\omega + N_q \tag{4.31}$$

Now filling in the values of the linearisation point $\bar{x}_{\ell,0}$ into equation 4.31 gives equation 4.32.

$$\frac{\partial f}{\partial\bar{q}}(\bar{q}_{\ell,0},\bar{\omega}_0) = \frac{1}{2}\begin{bmatrix}0 & 0 & n & 0\\ 0 & 0 & 0 & -n\\ -n & 0 & 0 & 0\\ 0 & n & 0 & 0\end{bmatrix} + \frac{n}{2}\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -1 & 0 & 0 & 0\\ 0 & -1 & 0 & 0\end{bmatrix} = \begin{bmatrix}0 & 0 & n & 0\\ 0 & 0 & 0 & 0\\ -n & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix} \tag{4.32}$$

It is interesting to note that the fourth row and colum are zero valued. This is a logical consequence from the linearisation of the quaternions where q_4 is a constant, as seen from equation 4.15. Equation 4.32 can ben by simplified to a 3×3 matrix.

$$\frac{\partial f}{\partial\bar{q}}(\bar{q}_{\ell,0},\bar{\omega}_{\ell,0}) = \begin{bmatrix}0 & 0 & n\\ 0 & 0 & 0\\ -n & 0 & 0\end{bmatrix} \tag{4.33}$$

Looking at the linearisation of $f(\bar{q}, \bar{\omega})$ with respect to $\bar{\omega}$ gives the following.

$$\frac{\partial f}{\partial \bar{\omega}} = \frac{1}{2}\mathbf{\Lambda}_q \Rightarrow \frac{\partial f}{\partial \bar{\omega}}(\bar{q}_{\ell,0}, \bar{\omega}_{\ell,0}) = \frac{1}{2}\boldsymbol{I}_{3\times 3} \tag{4.34}$$

Again, because q_4 is constant, this matrix can be reduced to a 3×3 matrix. The last part is to fill in the first term of equation 4.29. For this, as q_4 is constant, the resulting vector of $f(\bar{q}_{\ell,0}, \bar{\omega}_{\ell,0})$ should give a 3×3 matrix. For this reason, the $\mathbf{\Lambda}_q$ will be reduced to a 3×3 matrix by removing the fourth row and column as done in equation 4.33. In addition to this, the fourth row of $\boldsymbol{N}_q$ is also removed as q_4 is already known, which makes the fourth row redundant. Now filling in $f(\bar{q}_{\ell,0}, \bar{\omega}_{\ell,0})$ gives.

$$f(\bar{q}_{\ell,0}, \bar{\omega}_{\ell,0}) = \frac{1}{2}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}0\\ -n\\ 0\end{bmatrix} + \frac{n}{2}\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}0\\ 0\\ 0\\ 1\end{bmatrix} = \begin{bmatrix}0\\ -\frac{n}{2}\\ 0\end{bmatrix} + \begin{bmatrix}0\\ \frac{n}{2}\\ 0\end{bmatrix} = \bar{0} \tag{4.35}$$

With both first order derivatives known, equation 4.29 can be finalised by filling in the values for all terms.

$$\dot{\bar{q}}_\ell = \begin{bmatrix}0 & 0 & n\\ 0 & 0 & 0\\ -n & 0 & 0\end{bmatrix}\delta\bar{q} + \frac{1}{2}\boldsymbol{I}_{3\times 3}\delta\bar{\omega} \tag{4.36}$$

Now with the first equation linearised, $\dot{\bar{\omega}}$ can also be linearised as is expanded using the Taylor expansion as seen here below.

$$\dot{\bar{\omega}}_\ell = g(\bar{\omega}_{\ell,0}, \bar{u}) + \frac{\partial g}{\partial \bar{\omega}}(\bar{\omega}_{\ell,0}, \bar{u})\delta\bar{\omega} + \mathcal{O}(\delta\bar{\omega}^2) \tag{4.37}$$

Here, the second order term is neglected and $\bar{u} = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$. The first term from equation 4.37 is then computed.

$$g(\bar{\omega}_{\ell,0}, \bar{u}) = \boldsymbol{J}^{-1}\bar{u} - \boldsymbol{J}^{-1}\begin{bmatrix}0 & 0 & -n\\ 0 & 0 & 0\\ n & 0 & 0\end{bmatrix}\boldsymbol{J}\begin{bmatrix}0\\ -n\\ 0\end{bmatrix} = \boldsymbol{J}^{-1}\bar{u} \tag{4.38}$$

Now deriving the first order derivative term of equation 4.37.

$$\frac{\partial g}{\partial \bar{\omega}} = \frac{\partial \boldsymbol{J}^{-1}\bar{u}}{\partial \bar{\omega}} - \frac{\partial(\boldsymbol{J}^{-1}\mathbf{\Omega}(\bar{\omega})\boldsymbol{J}\bar{\omega})}{\partial \bar{\omega}} = -\boldsymbol{J}^{-1}\frac{\partial(\mathbf{\Omega}(\bar{\omega})\boldsymbol{J}\bar{\omega})}{\partial \bar{\omega}} \tag{4.39}$$

The term $-\mathbf{\Omega}(\bar{\omega})\boldsymbol{J}\bar{\omega}$ needs to be simplified into one vector in order to compute the derivative.

$$-\mathbf{\Omega}(\bar{\omega})\boldsymbol{J}\bar{\omega} = -\begin{bmatrix}0 & \omega_3 & -\omega_2\\ -\omega_3 & 0 & \omega_1\\ \omega_2 & -\omega_1 & 0\end{bmatrix}\begin{bmatrix}J_{11} & 0 & 0\\ 0 & J_{22} & 0\\ 0 & 0 & J_{33}\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix} = \begin{bmatrix}(J_{22}-J_{33})\omega_2\omega_3\\ (J_{33}-J_{11})\omega_1\omega_3\\ (J_{11}-J_{22})\omega_1\omega_2\end{bmatrix}$$

Now computing the derivative of this vector will give the following matrix.

$$\frac{\partial(\mathbf{\Omega}(\bar{\omega})\boldsymbol{J}\bar{\omega})}{\partial \bar{\omega}} = \begin{bmatrix}0 & (J_{22}-J_{33})\omega_3 & (J_{22}-J_{33})\omega_2\\ (J_{33}-J_{11})\omega_3 & 0 & (J_{33}-J_{11})\omega_1\\ (J_{11}-J_{22})\omega_2 & (J_{11}-J_{22})\omega_1 & 0\end{bmatrix}$$

Putting this result back into equation 4.39 will give the following equation.

$$\frac{\partial g}{\partial \bar{\omega}} = \begin{bmatrix} 0 & \frac{J_{22}-J_{33}}{J_{11}}\omega_3 & \frac{J_{22}-J_{33}}{J_{11}}\omega_2 \\ \frac{J_{33}-J_{11}}{J_{22}}\omega_3 & 0 & \frac{J_{33}-J_{11}}{J_{22}}\omega_1 \\ \frac{J_{11}-J_{22}}{J_{33}}\omega_2 & \frac{J_{11}-J_{22}}{J_{33}}\omega_1 & 0 \end{bmatrix} \tag{4.40}$$

With all terms defined, equation 4.37 can be completed and simplified, which results in equation 4.41.

$$\dot{\bar{\omega}}_\ell = \begin{bmatrix} 0 & 0 & \frac{J_{33}-J_{22}}{J_{11}}n \\ 0 & 0 & 0 \\ \frac{J_{22}-J_{11}}{J_{33}}n & & \end{bmatrix} \delta\bar{\omega} + \boldsymbol{J}^{-1}\bar{u} \tag{4.41}$$

Equation 4.36 and 4.41 described the spacecraft dynamics for a linearised state. Now these two equation can be transformed into a state-space model described by equation 4.42.

$$\dot{\bar{x}}_\ell = \boldsymbol{A}\bar{x}_\ell + \boldsymbol{B}\bar{u} \quad ; \quad \bar{z} = \boldsymbol{H}\bar{x}_\ell + \boldsymbol{D}\bar{u} \tag{4.42}$$

Where $\boldsymbol{A}$ is the state-matrix, $\boldsymbol{B}$ is the input matrix, $\boldsymbol{H}$ is the observation matrix and $\boldsymbol{D}$ is the throughput matrix. For this state-space model, the state vector here can be described as $\bar{x}_\ell = \begin{bmatrix}\delta\bar{q}^T & \delta\bar{\omega}^T\end{bmatrix}^T$ and input vector as $\bar{u} = \bar{\tau}_c$. The respective matrices are constructed based on equation 4.36 and 4.41.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & n & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ -n & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & n\frac{J_{33}-J_{22}}{J_{11}} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & n\frac{J_{22}-J_{11}}{J_{33}} & 0 & 0 \end{bmatrix} \quad ; \quad \begin{aligned} \boldsymbol{B} &= \begin{bmatrix} \boldsymbol{I}_{3\times3} \\ \boldsymbol{J}^{-1} \end{bmatrix} \\ \boldsymbol{H} &= \boldsymbol{I}_{6\times6} \\ \boldsymbol{D} &= \boldsymbol{0}_{6\times3} \end{aligned}$$

The state space model can then be transformed into an open-loop block diagram as seen in Figure 4.4. This model then has the torque $\bar{\tau}$ as input and ten propagates the state to the next step, which is then outputted.

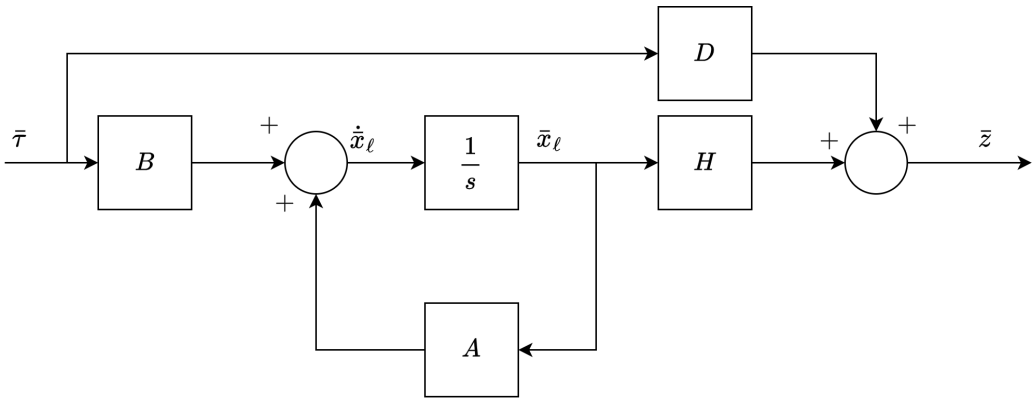


Figure 4.4: Open-loop state space block diagram

4.3. Flexible Model

The rigid body dynamics used in section 4.2 were sufficient for the Rigid body case of Alticube+. It is not sufficient to describe the flexible dynamics of Alticube+ due to the nature of the problem. Alticube+ can be identified as multiple rigid bodies (the five CubeSats) which are constraint by the booms linking the five CubeSats. Kane's equations is a great and simple method to implement, completely defining a framework for the flexible dynamics problem. [56] brings an effective implementation of Kane's equations for a set of rigid body satellites which closely relates to configuration of Alticube+. This section will adapt this implementation used by [56] to suit the structural setup of Alticube+. The derivation of the flexible dynamics can be divided into three parts. The first part will define the dynamics between two rigid bodies connected by a torsional joint that represents the boom. The second part constructs the full dynamics of all five cubesats through matrix representation relative to the central CubeSat's motion. Lastly, the full dynamics of the entire spacecraft, including the central cubesat is outlined through Kane's equations.

4.3.1. Two Rigid Body Case

In order to create the complete framework of flexible dynamics for Alticube, the fundamental basis of the implementation of Kane's equation is presented. This dynamcis entails two rigid body spacecraft which are connected by a boom. This boom can be represented by a torional spring that is able to rotate around all three axis (roll, yaw and pitch). Figure 4.5 gives a graphical representation of the two linked spacecraft.

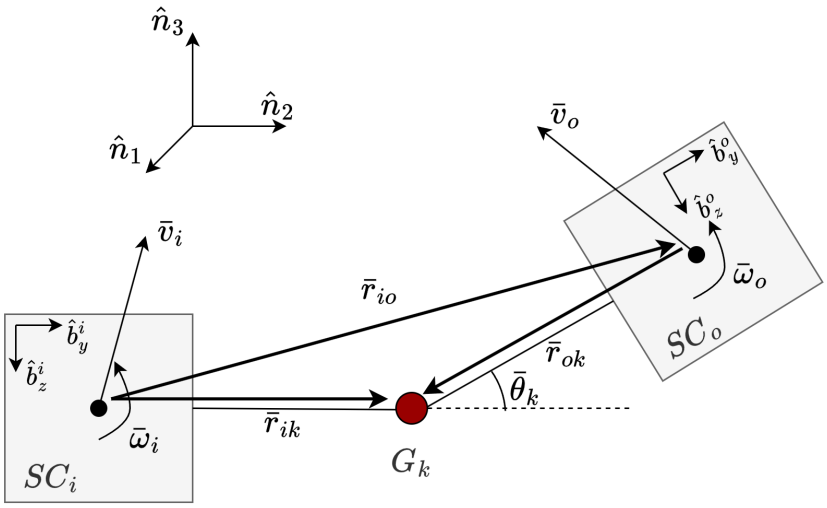


Figure 4.5: Two rigid body spacecraft dynamics linked by torsional spring

During the derivation of the dynamics, the two spacecraft can be separated through the notion of an inner and outer spacecraft. Here, the inner spacecraft (labelled SC_i) motion can be identified with the translational motion $\bar{v}_i$ and angular motion $\bar{\omega}_i$, while the outer spacecraft (Labelled SC_o) has the motion variables $\bar{v}_o$ and $\bar{\omega}_o$. The torsional joint linking the two spacecraft is labelled as G_k and has the angular displacement $\bar{\theta}_k$. The dynamics is derived in the inertial space, meaning that both translational and angular motion is represented in the inertial reference frame similar to the rigid body dynamics derived in section 4.2. But as the angular rates for a specific spacecraft are best described in the body reference frame of that spacecraft, the motion can also be described in the form of a partial velocities as seen in equation 4.43.

$$\bar{\omega}^{\mathcal{F}_\mathrm{I}} = u_1\hat{b}_x + u_2\hat{b}_y + u_3\hat{b}_z \tag{4.43}$$

Here, u_1, u_2 and u_3 represent the angular rates with respect to the spacecraft reference frame with as basis the spacecraft body reference frame $\mathcal{F}_b$ as the basis vectors. Similarly, the velocity of a spacecraft can be expressed using equation 4.44.

$$\bar{v}^{\mathcal{F}_\mathbf{I}} = u_4\hat{n}_x + u_5\hat{n}_y + u_6\hat{n}_z \tag{4.44}$$

Where u_4, u_5 and u_6 represent the translational velocities with respect to the inertial reference frame $\mathcal{F}_\mathbf{I}$. This notation can be reduced to a more convenient form with the use of dyads.

$$\bar{\omega} = \underline{B}\begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T \quad ; \quad \underline{B} = \begin{bmatrix} \hat{b}_x & \hat{b}_y & \hat{b}_z \end{bmatrix}$$
$$\bar{v} = \underline{N}\begin{bmatrix} u_4 & u_5 & u_6 \end{bmatrix}^T \quad ; \quad \underline{N} = \begin{bmatrix} \hat{n}_x & \hat{n}_y & \hat{n}_z \end{bmatrix}$$

The dyads are essentially a matrix of the basis vectors which construct a particular reference frame. For example, the angular velocity dyad $\underline{B}$ is created from the body reference frame basis vectors. Letting $\bar{u} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{bmatrix}^T$ be the vector that represents all partial velocities, then the translational and angular velocities can be expressed by equation 4.45 and 4.46.

$$\bar{\omega} = \underline{\Omega}\bar{u} \quad ; \quad \underline{\Omega} = \begin{bmatrix} \underline{B} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{4.45}$$

$$\bar{v} = \underline{V}\bar{u} \quad ; \quad \underline{V} = \begin{bmatrix} \mathbf{0}_{3\times3} & \underline{N} \end{bmatrix} \tag{4.46}$$

Single Link Formulation

Now that all mathematical notations have been expressed, derivation of the two rigid body dynamics can be done. Starting with the inner spacecraft, where the translational and angular velocity can be expressed similar to equation 4.43 and 4.44.

$$\bar{\omega}_i^{\mathcal{F}_\mathcal{I}} = \underline{\underline{B_i}}\bar{\omega}_i \quad ; \quad \bar{v}_i^{\mathcal{F}_\mathcal{I}} = \underline{N}\bar{v}_i \tag{4.47}$$

Where, $\underline{\underline{B_i}}$ is the dyad of the inner body reference frame. $\bar{\omega}_i$ is the angular velocities with respect to the inner body reference frame. The angular velocity of the outer spacecraft can be described as the sum of angular velocities as seen in equation 4.48

$$\bar{\omega}_o^{\mathcal{F}_\mathcal{I}} = \bar{\omega}_i^{\mathcal{F}_\mathcal{I}} + \bar{\omega}^{\mathcal{F}_o/\mathcal{F}_i} \tag{4.48}$$

This equation is similar to equation 4.21, where the angular velocity of the joint can be related to the kinematics using the Euler angle approach. Then the relative angular velocity between the outer and inner spacecraft can be described by equation 4.49

$$\bar{\omega}^{\mathcal{F}_o/\mathcal{F}_i} = \underline{\underline{B_o}}\mathbf{\Lambda}_k\dot{\bar{\theta}}_k \quad ; \quad \mathbf{\Lambda}_k = \begin{bmatrix} 1 & -\sin\theta_3^k & 0 \\ 0 & \cos\theta_1^k\cos\theta_3^k & -\sin\theta_1^k \\ 0 & \sin\theta_1^k\cos\theta_3^k & \cos\theta_3^k \end{bmatrix} \tag{4.49}$$

Having derived all components needed to described the angular velocity of the outer spacecraft, equation 4.50 can be generated.

$$\bar{\omega}_o^{\mathcal{F}_\mathcal{I}} = \underline{\underline{B_i}}\bar{\omega}_i + \underline{\underline{B_o}}\mathbf{\Lambda}_k\dot{\bar{\theta}}_k \tag{4.50}$$

The translational velocity of the outer spacecraft can be defined by adding the angular velocity contribution of the inner spacecraft and the joint together with the translational velocity of the inner spacecraft. By careful examination of the arm directions in Figure 4.5, equation 4.51 is obtained.

$$\bar{v}_o^{\mathcal{F}_\mathcal{I}} = \bar{v}_i^{\mathcal{F}_\mathcal{I}} + \bar{\omega}_i^{\mathcal{F}_\mathcal{I}} \times \bar{r}_{ik} - \bar{\omega}_o^{\mathcal{F}_\mathcal{I}} \times \bar{r}_{ok} \tag{4.51}$$

substituting equation 4.48 into 4.51 and rearranging terms by the velocity components and using the dyads for each of the velocity components will give the following.

$$\bar{v}^{\mathcal{F}_i/\mathcal{F}_\mathcal{I}} = \underline{N}\bar{v}_i + (\underline{\underline{B_i}}\bar{\omega}_i) \times (\underline{N}\bar{r}_{ik} - \underline{N}\bar{r}_{ok}) - (\underline{\underline{B_o}}\mathbf{\Lambda}_k\dot{\bar{\theta}}_k) \times \underline{N}\bar{r}_{ok} \tag{4.52}$$

Now that all velocity components are defined for the two spacecraft, the system of equations can be transformed into a matrix form, which will then depend on the partial velocity vector $\bar{u} = \begin{bmatrix}\bar{\omega}_i^T & \dot{\bar{\theta}}_k^T & \bar{v}_i^T\end{bmatrix}^T$.

$$\begin{bmatrix}\bar{\omega}_i^{\mathcal{F}_\mathcal{I}} \\ \bar{\omega}_o^{\mathcal{F}_\mathcal{I}}\end{bmatrix} = \begin{bmatrix}\underline{\underline{B_i}} & 0 & 0 \\ \underline{\underline{B_i}} & \underline{\underline{B_o}}\Lambda_k & 0\end{bmatrix} \begin{bmatrix}\bar{\omega}_i \\ \dot{\bar{\theta}}_k \\ \bar{v}_i\end{bmatrix} \quad ; \quad \{\bar{\omega}^{\mathcal{F}_\mathcal{I}}\} = \underline{\Omega}\bar{u} \tag{4.53}$$

$$\begin{bmatrix}\bar{v}_i^{\mathcal{F}_\mathcal{I}} \\ \bar{v}_o^{\mathcal{F}_\mathcal{I}}\end{bmatrix} = \begin{bmatrix}0 & 0 & \underline{N} \\ \underline{N}\bar{\beta}_{oi}^{\times}\underline{\underline{B_i}} & \underline{N}\bar{r}_{ok}^{\times}\underline{\underline{B_o}}\Lambda_1 & \underline{N}\end{bmatrix} \begin{bmatrix}\bar{\omega}_i \\ \dot{\bar{\theta}}_k \\ \bar{v}_i\end{bmatrix} \quad ; \quad \{\bar{v}^{\mathcal{F}_\mathcal{I}}\} = \underline{V}\bar{u} \tag{4.54}$$

Where $\bar{\beta}_{oi} = \bar{r}_{ok} - \bar{r}_{ik}$. The last step is to eliminate the dyad notation. This can be achieved through using the following basis vector transformation matrix notation seen in equation 4.55.

$$\underline{\underline{B_o}}^T\underline{\underline{B_i}} = T_i^o \quad ; \quad \underline{\underline{B_i}}^T\underline{\underline{B_i}} = \underline{\underline{B_o}}^T\underline{\underline{B_o}} = I \quad ; \quad \underline{N}^T\underline{\underline{B_i}} = T_i^N \quad ; \quad \underline{N}^T\underline{\underline{B_o}} = T_o^N \tag{4.55}$$

Multiplying the $\bar{\omega}_i$ row with $\underline{\underline{B_i}}^T$, $\bar{\omega}_o$ row with $\underline{\underline{B_o}}^T$ and $\bar{v}$ rows with $\underline{N}^T$ will give equation 4.56 and 4.57 which can be described in pure matrix notation.

$$\begin{bmatrix}\bar{\omega}_i \\ \bar{\omega}_o\end{bmatrix} = \begin{bmatrix}I & 0 & 0 \\ I & T_i^o\Lambda_k & 0\end{bmatrix} \begin{bmatrix}\bar{\omega}_i \\ \dot{\bar{\theta}}_k \\ \bar{v}_i\end{bmatrix} \quad ; \quad \{\bar{\omega}\} = \Omega\bar{u} \tag{4.56}$$

$$\begin{bmatrix}\bar{v}_i \\ \bar{v}_o\end{bmatrix} = \begin{bmatrix}0 & 0 & I \\ \bar{\beta}_{oi}^{\times}T_i^N & \bar{r}_{ok}^{\times}T_o^N\Lambda_k & I\end{bmatrix} \begin{bmatrix}\bar{\omega}_i \\ \dot{\bar{\theta}}_k \\ \bar{v}_i\end{bmatrix} \quad ; \quad \{\bar{v}\} = V\bar{u} \tag{4.57}$$

4.3.2. Assembly of Constrained Multi-rigid Spacecraft Dynamics

Section 4.3.1 has described the dynamics of a single link with respect to the centre spacecraft's motion ($\bar{\omega}$ and $\bar{v}$) and joint motion $\dot{\bar{\theta}}$. This serves as a core component to assemble the full state motion for all five CubeSats based on the centre spacecraft's motion and joint motion. First an additional derivation is made that connects three spacecraft together to outline a pattern. Then the full matrix in the form of equation 4.56 and 4.57 can be derived for both translational and angular velocity.

Let the notation SC_i, SC_m, SC_o be an identification for the inner, middle and outer spacecraft. The middle spacecraft is linked to the inner spacecraft by joint G_k. Then the outer spacecraft is linked to the middle spacecraft by joint G_{k+1}. The goal is to describe the motion of all spacecraft based on the partial velocity vector $u = \begin{bmatrix}\bar{\omega}_i^T & \dot{\bar{\theta}}_k^T & \dot{\bar{\theta}}_{k+1}^T & \bar{v}_i^T\end{bmatrix}^T$. The motion of the outer spacecraft can be described by the motion of the middle spacecraft through equations 4.47 - 4.52. Starting with the angular velocity:

$$\bar{\omega}_o^{\mathcal{F}_\mathcal{I}} = \bar{\omega}_m^{\mathcal{F}_\mathcal{I}} + \underline{\underline{B_o}}\mathbf{\Lambda}_{k+1}\dot{\bar{\theta}}_{k+1} \tag{4.58}$$

Similarly for the middle spacecraft, equation 4.48 and 4.49 can be used to describe the angular velocity with respect to the inner spacecraft.

$$\bar{\omega}_m^{\mathcal{F}_\mathcal{I}} = \bar{\omega}_i^{\mathcal{F}_\mathcal{I}} + \underline{\underline{B_m}}\mathbf{\Lambda}_k\dot{\bar{\theta}}_k \tag{4.59}$$

Now substituting equation 4.59 into equation 4.58 gives equation 4.60.

$$\bar{\omega}_o^{\mathcal{F}_\mathcal{I}} = \bar{\omega}_i^{\mathcal{F}_\mathcal{I}} + \underline{\underline{B_m}}\mathbf{\Lambda}_k\dot{\bar{\theta}}_k + \underline{\underline{B_o}}\mathbf{\Lambda}_{k+1}\dot{\bar{\theta}}_{k+1} \tag{4.60}$$

The pattern here is that the angular velocity is the summation of contributing joint motions plus the inner spacecraft's angular velocity. The same method is used for the translational velocities, starting with the translational velocity of the outer spacecraft with respect to the middle spacecraft.

$$\bar{v}_o^{\mathcal{F}_\mathcal{I}} = \bar{v}_m^{\mathcal{F}_\mathcal{I}} + \bar{\omega}_m^{\mathcal{F}_\mathcal{I}} \times \bar{r}_{m,k+1} - \bar{\omega}_o^{\mathcal{F}_\mathcal{I}} \times \bar{r}_{o,k+1} \tag{4.61}$$

$$\bar{v}_m^{\mathcal{F}_\mathcal{I}} = \bar{v}_i^{\mathcal{F}_\mathcal{I}} + \bar{\omega}_i^{\mathcal{F}_\mathcal{I}} \times \bar{r}_{ik} - \bar{\omega}_m^{\mathcal{F}_\mathcal{I}} \times \bar{r}_{mk} \tag{4.62}$$

Substituting equations 4.62, 4.59 and 4.60 into equation 4.61 gives equation 4.63.

$$\bar{v}_o^{\mathcal{F}_\mathcal{I}} = \bar{v}_i^{\mathcal{F}_\mathcal{I}} + \bar{\omega}_i^{\mathcal{F}_\mathcal{I}} \times \bar{r}_{ik} - \bar{\omega}_m^{\mathcal{F}_\mathcal{I}} \times \bar{r}_{mk} + \left(\bar{\omega}_i^{\mathcal{F}_\mathcal{I}} + \underline{\underline{B_m}}\mathbf{\Lambda}_k\dot{\bar{\theta}}_k\right) \times \bar{r}_{m,k+1} - \left(\bar{\omega}_i^{\mathcal{F}_\mathcal{I}} + \underline{\underline{B_m}}\mathbf{\Lambda}_k\dot{\bar{\theta}}_k + \underline{\underline{B_o}}\mathbf{\Lambda}_{k+1}\dot{\bar{\theta}}_{k+1}\right) \times \bar{r}_{o,k+1} \tag{4.63}$$

Regrouping terms under the partial velocities and substituting equation 4.59 gives:

$$\bar{v}_o^{\mathcal{F}_\mathcal{I}} = \bar{v}_i^{\mathcal{F}_\mathcal{I}} + (\bar{r}_{o,k+1} - \bar{r}_{m,k+1} + \bar{r}_{mk} - \bar{r}_{ik}) \times \bar{\omega}_i^{\mathcal{F}_\mathcal{I}} + (\bar{r}_{o,k+1} - \bar{r}_{m,k+1} + \bar{r}_{mk}) \times \underline{\underline{B_m}}\mathbf{\Lambda}_k\dot{\bar{\theta}}_k + \bar{r}_{o,k+1} \times \underline{\underline{B_o}}\mathbf{\Lambda}_{k+1}\dot{\bar{\theta}}_{k+1} \tag{4.64}$$

Similar as before, this can be put into the matrix form and a patterns can be observed.

$$\begin{bmatrix} \bar{\omega}_i \\ \bar{\omega}_m \\ \bar{\omega}_o \end{bmatrix} = \begin{bmatrix} \underline{\underline{B_i}} & 0 & 0 & 0 & 0 & 0 \\ \underline{\underline{B_i}} & \underline{\underline{B_m}}\Lambda_k & 0 & 0 & 0 & 0 \\ \underline{\underline{B_i}} & \underline{\underline{B_m}}\Lambda_k & \underline{\underline{B_o}}\Lambda_{k+1} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{\omega}_i \\ \dot{\bar{\theta}}_k \\ \dot{\bar{\theta}}_{k+1} \\ \bar{v}_i \end{bmatrix} \tag{4.65}$$

$$\begin{bmatrix} \bar{v}_i \\ \bar{v}_m \\ \bar{v}_k \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \underline{\underline{N}} \\ \underline{\underline{N}}\bar{\beta}_{mi}^{\times}\underline{\underline{B_i}} & \underline{\underline{N}}\bar{r}_{mi}^{\times}\underline{\underline{B_m}}\Lambda_k & 0 & 0 & 0 & \underline{\underline{N}} \\ \underline{\underline{N}}\bar{\beta}_{oi}^{\times}\underline{\underline{B_i}} & \underline{\underline{N}}\bar{\gamma}_{ok}^{\times}\underline{\underline{B_m}}\Lambda_1 & \underline{\underline{N}}\bar{r}_{om}^{\times}\underline{\underline{B_o}}\Lambda_{k+1} & 0 & 0 & \underline{\underline{N}} \end{bmatrix} \begin{bmatrix} \bar{\omega}_i \\ \dot{\bar{\theta}}_k \\ \dot{\bar{\theta}}_{k+1} \\ \bar{v}_i \end{bmatrix} \tag{4.66}$$

In order to fully describe the dynamics of all five CubeSats, a new notation is introduced, supported by Figure 4.6. Starting with the spacecraft identification, let the central spacecraft be described as SC_1. Then following from the body frame reference $\mathcal{F}_b$ of SC_1, the spacecraft in the positive $\hat{b}_y^1$ direction are noted as SC_2 and SC_3 ordered from closest to farthest. The spacecraft in the negative $\hat{b}_y^1$ direction are identified as SC_4 and SC_5 ordered from closest to farthest from SC_1. Next, the joints that represent the booms are numbered from one to four, where G_1 is the joint connecting spacecraft SC_1 and SC_2. G_2 connects SC_2 with SC_3, G_3 connects SC_1 with SC_4 and G_4 connects SC_4 with SC_5. Lastly, like in Figure 4.5, the vector going from the spacecraft's centre of mass to the joint location G_k is defined as $\bar{r}_{ik}$, where i is the spacecraft number and k is the joint number. This means that the vector pointing to joint one from spacecraft one is defined as $\bar{r}_{11}$, while the vector going from spacecraft four to joint three is described as $\bar{r}_{43}$.

With all variables being defined, equations 4.47 to 4.66 can be used to construct the full matrix in the form of 4.53 and 4.54.

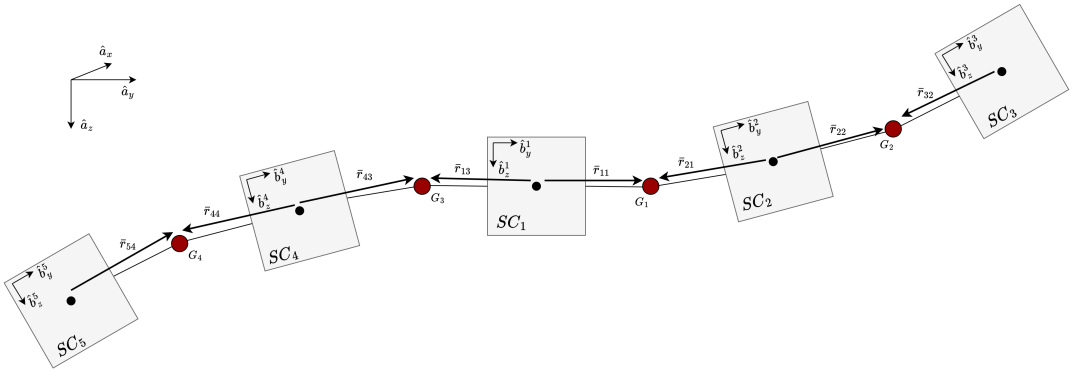


Figure 4.6: Flexible model representation of Alticube+

$$\begin{bmatrix}\bar{\omega}_1\\ \bar{\omega}_2\\ \bar{\omega}_3\\ \bar{\omega}_4\\ \bar{\omega}_5\end{bmatrix} = \begin{bmatrix}\underline{\underline{B_1}} & 0 & 0 & 0 & 0 & 0\\ \underline{\underline{B_1}} & \underline{\underline{B_2}}\Lambda_1 & 0 & 0 & 0 & 0\\ \underline{\underline{B_1}} & \underline{\underline{B_2}}\Lambda_1 & \underline{\underline{B_3}}\Lambda_2 & 0 & 0 & 0\\ \underline{\underline{B_1}} & 0 & 0 & \underline{\underline{B_4}}\Lambda_3 & 0 & 0\\ \underline{\underline{B_1}} & 0 & 0 & \underline{\underline{B_4}}\Lambda_3 & \underline{\underline{B_5}}\Lambda_4 & 0\end{bmatrix}\begin{bmatrix}\bar{\omega}_1\\ \dot{\bar{\theta}}_1\\ \dot{\bar{\theta}}_2\\ \dot{\bar{\theta}}_3\\ \dot{\bar{\theta}}_4\\ \dot{\bar{\theta}}_5\\ \bar{v}_1\end{bmatrix} \tag{4.67}$$

$$\begin{bmatrix}\bar{v}_1\\ \bar{v}_2\\ \bar{v}_3\\ \bar{v}_4\\ \bar{v}_5\end{bmatrix} = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & \underline{\underline{N}}\\ \underline{\underline{N}}\bar{\beta}_{21}^{\times}\underline{\underline{B_1}} & \underline{\underline{N}}\bar{r}_{21}^{\times}\underline{\underline{B_2}}\Lambda_1 & 0 & 0 & 0 & \underline{\underline{N}}\\ \underline{\underline{N}}\bar{\beta}_{31}^{\times}\underline{\underline{B_1}} & \underline{\underline{N}}\bar{\gamma}_{31}^{\times}\underline{\underline{B_2}}\Lambda_1 & \underline{\underline{N}}\bar{r}_{32}^{\times}\underline{\underline{B_3}}\Lambda_2 & 0 & 0 & \underline{\underline{N}}\\ \underline{\underline{N}}\bar{\beta}_{41}^{\times}\underline{\underline{B_1}} & 0 & 0 & \underline{\underline{N}}\bar{r}_{41}^{\times}\underline{\underline{B_4}}\Lambda_3 & 0 & \underline{\underline{N}}\\ \underline{\underline{N}}\bar{\beta}_{51}^{\times}\underline{\underline{B_1}} & 0 & 0 & \underline{\underline{N}}\bar{\gamma}_{51}^{\times}\underline{\underline{B_4}}\Lambda_1 & \underline{\underline{N}}\bar{r}_{54}^{\times}\underline{\underline{B_5}}\Lambda_4 & \underline{\underline{N}}\end{bmatrix}\begin{bmatrix}\bar{\omega}_1\\ \dot{\bar{\theta}}_1\\ \dot{\bar{\theta}}_2\\ \dot{\bar{\theta}}_3\\ \dot{\bar{\theta}}_4\\ \dot{\bar{\theta}}_5\\ \bar{v}_1\end{bmatrix} \tag{4.68}$$

Converting this to the matrix form using the same method as with the derivation of the single link, the following matrix equations can be constructed for both angular and translational velocities.

$$\begin{bmatrix}\bar{\omega}_1\\ \bar{\omega}_2\\ \bar{\omega}_3\\ \bar{\omega}_4\\ \bar{\omega}_5\end{bmatrix} = \begin{bmatrix}\boldsymbol{I}_{3\times3} & 0 & 0 & 0 & 0 & 0\\ \boldsymbol{T}_1^2 & \Lambda_1 & 0 & 0 & 0 & 0\\ \boldsymbol{T}_1^3 & \boldsymbol{T}_2^3\Lambda_1 & \Lambda_2 & 0 & 0 & 0\\ \boldsymbol{T}_1^4 & 0 & 0 & \Lambda_3 & 0 & 0\\ \boldsymbol{T}_1^5 & 0 & 0 & \boldsymbol{T}_4^5\Lambda_3 & \Lambda_4 & 0\end{bmatrix}\begin{bmatrix}\bar{\omega}_1\\ \dot{\bar{\theta}}_1\\ \dot{\bar{\theta}}_2\\ \dot{\bar{\theta}}_3\\ \dot{\bar{\theta}}_4\\ \dot{\bar{\theta}}_5\\ \bar{v}_1\end{bmatrix} \tag{4.69}$$

$$\begin{bmatrix}\bar{v}_1\\ \bar{v}_2\\ \bar{v}_3\\ \bar{v}_4\\ \bar{v}_5\end{bmatrix} = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & \boldsymbol{I}_{3\times3}\\ \bar{\beta}_{21}^{\times}\boldsymbol{T}_1^N & \bar{r}_{21}^{\times}\boldsymbol{T}_2^N\Lambda_1 & 0 & 0 & 0 & \boldsymbol{I}_{3\times3}\\ \bar{\beta}_{31}^{\times}\boldsymbol{T}_1^N & \bar{\gamma}_{31}^{\times}\boldsymbol{T}_2^N\Lambda_1 & \bar{r}_{32}^{\times}\boldsymbol{T}_3^N\Lambda_2 & 0 & 0 & \boldsymbol{I}_{3\times3}\\ \bar{\beta}_{41}^{\times}\boldsymbol{T}_1^N & 0 & 0 & \bar{r}_{41}^{\times}\boldsymbol{T}_4^N\Lambda_3 & 0 & \boldsymbol{I}_{3\times3}\\ \bar{\beta}_{51}^{\times}\boldsymbol{T}_1^N & 0 & 0 & \bar{\gamma}_{51}^{\times}\boldsymbol{T}_4^N\Lambda_1 & \bar{r}_{54}^{\times}\boldsymbol{T}_5^N\Lambda_4 & \boldsymbol{I}_{3\times3}\end{bmatrix}\begin{bmatrix}\bar{\omega}_1\\ \dot{\bar{\theta}}_1\\ \dot{\bar{\theta}}_2\\ \dot{\bar{\theta}}_3\\ \dot{\bar{\theta}}_4\\ \dot{\bar{\theta}}_5\\ \bar{v}_1\end{bmatrix} \tag{4.70}$$

This can then be reduced to the simplified form.

$$\{\bar{\omega}\} = \boldsymbol{\Omega}\bar{u} \tag{4.71}$$

$$\{\bar{v}\} = \boldsymbol{V}\bar{u} \tag{4.72}$$

Reference Frame Rotation Matrix convention

Multiple reference frame rotation matrices can be observed in equation 4.69 and 4.70 which takes the form of either T_i^N or T_i^j where i and j refer to the spacecraft reference. For T_i^N, $i = 1, 2, ..., 5$. For T_i^j, $i = 1, 2, ..., 4$ and $j = 2, 3, ..., 5$ constrained by $j > i$.

$$\boldsymbol{T}_i^j = \prod_{k=1}^{j-i} \boldsymbol{T}_{B_{j-k}}^{B_{j-k+1}} \tag{4.73}$$

$$\boldsymbol{T}_j^{\mathcal{F}_\mathcal{I}} = \left(\boldsymbol{T}_{\mathcal{F}_\mathcal{I}}^j\right)^T = \left(\boldsymbol{T}_1^j \boldsymbol{T}_{\mathcal{F}_\mathcal{I}}^1\right)^T = \left(\boldsymbol{T}_{\mathcal{F}_\mathcal{I}}^{B_1}\right)^T \left(\prod_{i=1}^{j} \boldsymbol{T}_{B_i}^{B_{i+1}}\right)^T \tag{4.74}$$

4.3.3. Implementation of Kane's Equations

Kane's equations provide an efficient framework for deriving equations of motion for complex, multi-body spacecraft systems. Starting from Newton-Euler equations, Kane's method projects the translational and rotational dynamics into the subspace spanned by the generalized speeds using partial velocities [56]. In matrix form, the governing equation can be expressed as

$$\boldsymbol{V}^T\{\bar{F}\} + \boldsymbol{\Omega}^T\{\bar{\tau}\} = \boldsymbol{V}^T\boldsymbol{M}\{\bar{a}\} + \boldsymbol{\Omega}^T\left(\boldsymbol{J}\{\bar{\alpha}\} + \{\bar{\omega}\}^\times \boldsymbol{J}\{\bar{\omega}\}\right) \tag{4.75}$$

where $\boldsymbol{V}$ and $\boldsymbol{\Omega}$ are the linear and angular partial velocity matrices, $\boldsymbol{M}$ and $\boldsymbol{J}$ represent the mass and inertia matrices, $\{\bar{F}\}$ and $\{\bar{\tau}\}$ are external forces and torques, and $\{\bar{a}\}$ and $\{\bar{\alpha}\}$ are the linear and angular accelerations of each body. This formulation naturally incorporates both translational and rotational dynamics while avoiding the explicit calculation of constraint forces and Lagrange multipliers. For flexible spacecraft, the method is particularly advantageous because it allows the coupling of rigid-body hub dynamics with flexible appendages modelled as interconnected rigid links with torsional springs and dampers. The partial velocity formulation enables efficient construction of minimum-order dynamic equations suitable for onboard computation and control design, facilitating adaptive estimation of unknown flexible parameters during operation. For the Alticube+ system, which consists of five identical CubeSats, the inertia matrix $\boldsymbol{J}$ in Kane's equations naturally takes a block diagonal form, since each spacecraft's rotational dynamics can be described independently in its body frame. The structure of the total inertia matrix is

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_1 & 0 & 0 & 0 & 0 \\ 0 & \boldsymbol{J}_2 & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{J}_3 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{J}_4 & 0 \\ 0 & 0 & 0 & 0 & \boldsymbol{J}_5 \end{bmatrix} \tag{4.76}$$

where each ($\boldsymbol{J}_i \in \mathbb{R}^{3\times3}$) represents the inertia matrix of the (i)-th CubeSat about its body-fixed principal axes.

Since all five CubeSats are identical, we have

$$\boldsymbol{J}_1 = \boldsymbol{J}_2 = \boldsymbol{J}_3 = \boldsymbol{J}_4 = \boldsymbol{J}_5 = \boldsymbol{J}_* \tag{4.77}$$

which simplifies computation and system modelling, and also ensures symmetry in the dynamic equations.

For a single CubeSat modelled as a uniform solid cuboid with mass (m), length (L), width (W), and height (H), the mass moment of inertia about the principal axes through the centre of mass is given by the standard rigid body formulation:

$$J_{xx} = \frac{1}{12} m \left(W^2 + H^2\right), \qquad J_{yy} = \frac{1}{12} m \left(L^2 + H^2\right), \qquad J_{zz} = \frac{1}{12} m \left(L^2 + W^2\right). \tag{4.78}$$

Thus, the inertia matrix of a single CubeSat is

$$\boldsymbol{J}_* = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \tag{4.79}$$

Similarly, the mass matrix ($\boldsymbol{M}$) in Kane' s formulation, which accounts for the translational dynamics of each body, is also block diagonal:

$$\boldsymbol{M} = \begin{bmatrix} m_1 \boldsymbol{I}_{3\times3} & 0 & 0 & 0 & 0 \\ 0 & m_2 \boldsymbol{I}_{3\times3} & 0 & 0 & 0 \\ 0 & 0 & m_3 \boldsymbol{I}_{3\times3} & 0 & 0 \\ 0 & 0 & 0 & m_4 \boldsymbol{I}_{3\times3} & 0 \\ 0 & 0 & 0 & 0 & m_5 \boldsymbol{I}_{3\times3} \end{bmatrix} \tag{4.80}$$

with each ($m_i = m_*$).

Another notation that needs to be described is the cross-product term in equation 4.75. To create the matrix form of this cross product, a block diagram is formed of five individual skew matrices for each of the angular velocities as seen below.

$$\{\bar{\omega}\}^{\times} = \begin{bmatrix} \bar{\omega}_1^{\times} & 0 & 0 & 0 & 0 \\ 0 & \bar{\omega}_2^{\times} & 0 & 0 & 0 \\ 0 & 0 & \bar{\omega}_3^{\times} & 0 & 0 \\ 0 & 0 & 0 & \bar{\omega}_4^{\times} & 0 \\ 0 & 0 & 0 & 0 & \bar{\omega}_5^{\times} \end{bmatrix} \tag{4.81}$$

Derivations of Acceleration Terms in Kane's Equations

Starting from the kinematic relations that define the system' s angular and linear velocities in terms of the partial velocity matrices and generalized speeds, we have

$$\{\bar{\omega}\} = \boldsymbol{\Omega} \bar{u} \tag{4.82}$$

$$\{\bar{v}\} = \boldsymbol{V} \bar{u} \tag{4.83}$$

where $\boldsymbol{\Omega}$ and $\boldsymbol{V}$ are the angular and linear partial velocity matrices, respectively, and $\bar{u}$ is the vector of generalized speeds. It is important to note that both $\boldsymbol{\Omega}$ and $\boldsymbol{V}$ are functions of the generalized coordinates and therefore are time-varying. To obtain the angular and linear accelerations, the above relations are differentiated with respect to time. Applying the product rule to the angular velocity expression yields

$$\frac{d}{dt}\{\bar{\omega}\} = \frac{d}{dt}\left(\boldsymbol{\Omega}\bar{u}\right) = \dot{\boldsymbol{\Omega}}\bar{u} + \boldsymbol{\Omega}\dot{\bar{u}} \tag{4.84}$$

Rearranging terms gives the expression for the angular acceleration,

$$\{\bar{\alpha}\} = \boldsymbol{\Omega}\dot{\bar{u}} + \dot{\boldsymbol{\Omega}}\bar{u} \tag{4.85}$$

For convenience, the second term can be expressed using a compact notation as $(\{\bar{f}(\dot{\boldsymbol{\Omega}}\bar{u})\} = \dot{\boldsymbol{\Omega}}\bar{u})$ leading to the final form

$$\boxed{\bar{\alpha} = \boldsymbol{\Omega}\dot{\bar{u}} + \{\bar{f}(\dot{\boldsymbol{\Omega}}\bar{u})\}} \tag{4.86}$$

A similar procedure is followed for the linear velocity. Differentiating with respect to time gives

$$\frac{d}{dt}\{\bar{v}\} = \frac{d}{dt}\big(\boldsymbol{V}\bar{u}\big) = \dot{\boldsymbol{V}}\bar{u} + \boldsymbol{V}\dot{\bar{u}} \tag{4.87}$$

Rewriting, the linear acceleration is expressed as

$$\{\bar{a}\} = \boldsymbol{V}\dot{\bar{u}} + \dot{\boldsymbol{V}}\bar{u} \tag{4.88}$$

Introducing the shorthand $(\{\bar{g}\dot{\boldsymbol{V}}\bar{u})\} = \dot{\boldsymbol{V}}\bar{u})$, we obtain

$$\boxed{\{\bar{a}\} = \boldsymbol{V}\dot{\bar{u}} + \{\bar{g}(\dot{\boldsymbol{V}}\bar{u})\}} \tag{4.89}$$

These two equations represent the decomposition of the angular and linear accelerations into terms dependent on the generalized accelerations ($\dot{\bar{u}}$) and configuration-dependent terms $\dot{\boldsymbol{\Omega}}\bar{u}$ and $\dot{\boldsymbol{V}}\bar{u}$. The first terms, $\boldsymbol{\Omega}\dot{\bar{u}}$ and $\boldsymbol{V}\dot{\bar{u}}$, contribute directly to the inertia-related components of the equations of motion, while the latter terms, $\dot{\boldsymbol{\Omega}}\bar{u}$ and $\dot{\boldsymbol{V}}\bar{u}$, account for the configuration and velocity-dependent effects such as gyroscopic and Coriolis terms. These expressions form the kinematic foundation used to construct the final dynamic equations in Kane' s formulation.

The algebraic derivation of the acceleration terms starts from the kinematic relations defining the angular and translational velocities of the inner and outer spacecraft with respect to the inertial frame $\mathcal{F}_{\mathcal{I}}$. The goal is to obtain the corresponding acceleration expressions and to separate the results into the generalized acceleration dependent part, $\boldsymbol{\Omega}\dot{\bar{u}}$, and the remainder terms $\bar{f}$ and $\bar{g}$.

Inner Angular Acceleration

The inner angular velocity in the inertial frame is expressed as

$$\bar{\omega}_i^{\mathcal{F}_{\mathcal{I}}} = \underline{\underline{B_i}}\,\bar{\omega}_i \tag{4.90}$$

Taking the time derivative gives

$$\dot{\bar{\omega}}_i^{\mathcal{F}_{\mathcal{I}}} = \dot{\underline{\underline{B_i}}}\,\bar{\omega}_i + \underline{\underline{B_i}}\,\dot{\bar{\omega}}_i \tag{4.91}$$

The time derivative of a direction cosine dyadic $\underline{\underline{B_i}}$ acting on a vector can be written as a cross product with the angular velocity, such that $\dot{\underline{\underline{B_i}}}\,\bar{x} = \bar{\omega}_i^{\mathcal{F}_{\mathcal{I}}} \times \underline{\underline{B_i}}\,\bar{x}$. Applying this to $\bar{x} = \bar{\omega}_i$ yields

$$\dot{\underline{\underline{B_i}}}\,\bar{\omega}_i = \bar{\omega}_i^{\mathcal{F}_{\mathcal{I}}} \times \underline{\underline{B_i}}\,\bar{\omega}_i \tag{4.92}$$

Since any vector crossed with itself is zero, the term $\underline{\underline{B_i}}\,\bar{\omega}_i \times \underline{\underline{B_i}}\,\bar{\omega}_i = 0$. Thus, the inner angular acceleration in the inertial frame becomes

$$\bar{\alpha}_i^{\mathcal{F}_{\mathcal{I}}} = \underline{\underline{B_i}}\,\dot{\bar{\omega}}_i \tag{4.93}$$

This shows that the inner body has no remainder term and its angular acceleration depends solely on $\dot{\bar{\omega}}_i$.

Outer Angular Acceleration

The angular velocity of the outer spacecraft is given by

$$\bar{\omega}_o^{\mathcal{F}_{\mathcal{I}}} = \bar{\omega}_i^{\mathcal{F}_{\mathcal{I}}} + \underline{\underline{B_o}}\,\boldsymbol{\Lambda}_k\,\dot{\bar{\theta}}_k \tag{4.94}$$

Taking the time derivative and applying the product rule results in

$$\dot{\bar{\omega}}_o^{\mathcal{F}_{\mathcal{I}}} = \dot{\bar{\omega}}_i^{\mathcal{F}_{\mathcal{I}}} + \dot{\underline{\underline{B_o}}}\,\boldsymbol{\Lambda}_k\,\dot{\bar{\theta}}_k + \underline{\underline{B_o}}\,\dot{\boldsymbol{\Lambda}}_k\,\dot{\bar{\theta}}_k + \underline{\underline{B_o}}\,\boldsymbol{\Lambda}_k\,\ddot{\bar{\theta}}_k$$

Each $\dot{\underline{B}}$ term can be expressed as a cross product with the corresponding angular velocity, for example $\dot{\underline{\underline{B_o}}}\,\bar{x} = \bar{\omega}_o^{\mathcal{F}_\mathcal{I}} \times \underline{\underline{B_o}}\,\bar{x}$. Substituting and rearranging terms gives

$$\bar{\alpha}_o^{\mathcal{F}_\mathcal{I}} = \underline{\underline{B_i}}\,\dot{\bar{\omega}}_i + \underline{\underline{B_o}}\,\mathbf{\Lambda}_k\,\ddot{\bar{\theta}}_k + \bar{\alpha}_r \tag{4.95}$$

where the remainder term $\bar{\alpha}_r$ collects all velocity-dependent components,

$$\bar{\alpha}_r = \underline{\underline{B_o}}\,\bar{\omega}_o \times \underline{\underline{B_o}}\,\mathbf{\Lambda}_k\,\dot{\bar{\theta}}_k + \underline{\underline{B_o}}\,\dot{\mathbf{\Lambda}}_k\,\dot{\bar{\theta}}_k \tag{4.96}$$

Using the compact notation T_i^o for $\underline{\underline{B_i}}$ expressed in the outer frame, the above can be rewritten as

$$\bar{\alpha}_o = T_i^o\,\dot{\bar{\omega}}_i + \mathbf{\Lambda}_k\,\ddot{\bar{\theta}}_k + \bar{\alpha}_r \tag{4.97}$$

$$\bar{\alpha}_r = \bar{\omega}_o \times \mathbf{\Lambda}_k\,\dot{\bar{\theta}}_k + \dot{\mathbf{\Lambda}}_k\,\dot{\bar{\theta}}_k \tag{4.98}$$

Stacking the remainder angular accelerations of all spacecraft gives

$$\{\bar{f}(\dot{\mathbf{\Omega}}\,\bar{u})\} = \begin{bmatrix} \bar{0} \\ \bar{\omega}_2^{\times}\,\Lambda_1\,\dot{\bar{\theta}}_1 + \dot{\Lambda}_1\,\dot{\bar{\theta}}_1 \\ \bar{\omega}_2^{\times}\,\Lambda_1\,\dot{\bar{\theta}}_1 + \dot{\Lambda}_1\,\dot{\bar{\theta}}_1 + \bar{\omega}_3^{\times}\,\Lambda_2\,\dot{\bar{\theta}}_2 + \dot{\Lambda}_2\,\dot{\bar{\theta}}_2 \\ \bar{\omega}_4^{\times}\,\Lambda_3\,\dot{\bar{\theta}}_3 + \dot{\Lambda}_3\,\dot{\bar{\theta}}_3 \\ \bar{\omega}_4^{\times}\,\Lambda_3\,\dot{\bar{\theta}}_3 + \dot{\Lambda}_3\,\dot{\bar{\theta}}_3 + \bar{\omega}_5^{\times}\,\Lambda_4\,\dot{\bar{\theta}}_4 + \dot{\Lambda}_4\,\dot{\bar{\theta}}_4 \end{bmatrix} \tag{4.99}$$

Inner Translational Acceleration

The translational velocity of the inner spacecraft is

$$\bar{v}_i^{\mathcal{F}_\mathcal{I}} = \underline{\underline{N}}\,\bar{v}_i \tag{4.100}$$

Taking the time derivative yields

$$\dot{\bar{v}}_i^{\mathcal{F}_\mathcal{I}} = \dot{\underline{\underline{N}}}\,\bar{v}_i + \underline{\underline{N}}\,\dot{\bar{v}}_i \tag{4.101}$$

Since $\underline{\underline{N}}$ represents the fixed inertial frame, its derivative is zero, $\dot{\underline{\underline{N}}} = 0$. The inner translational acceleration then becomes

$$\bar{a}_i^{\mathcal{F}_\mathcal{I}} = \underline{\underline{N}}\,\dot{\bar{v}}_i \tag{4.102}$$

Outer Translational Acceleration

The outer translational velocity is defined by

$$\bar{v}_o^{\mathcal{F}_\mathcal{I}} = \bar{v}_i^{\mathcal{F}_\mathcal{I}} + \bar{\omega}_i^{\mathcal{F}_\mathcal{I}} \times \bar{r}_{ik} - \bar{\omega}_o^{\mathcal{F}_\mathcal{I}} \times \bar{r}_{ok} \tag{4.103}$$

Differentiating and applying the product rule gives

$$\dot{\bar{v}}_o^{\mathcal{F}_\mathcal{I}} = \dot{\bar{v}}_i^{\mathcal{F}_\mathcal{I}} + \dot{\bar{\omega}}_i^{\mathcal{F}_\mathcal{I}} \times \bar{r}_{ik} + \bar{\omega}_i^{\mathcal{F}_\mathcal{I}} \times \dot{\bar{r}}_{ik} - \dot{\bar{\omega}}_o^{\mathcal{F}_\mathcal{I}} \times \bar{r}_{ok} - \bar{\omega}_o^{\mathcal{F}_\mathcal{I}} \times \dot{\bar{r}}_{ok}$$

For a rigid connection, $\dot{\bar{r}} = \bar{\omega} \times \bar{r}$, leading to double cross-product terms. Substituting the angular acceleration expressions gives

$$\bar{a}_o^{\mathcal{F}_\mathcal{I}} = \underline{\underline{N}}\,\dot{\bar{v}}_i + \underline{\underline{N}}\,\bar{\beta}_{oi} \times \underline{\underline{B_i}}\,\dot{\bar{\omega}}_i + \underline{\underline{N}}\,\bar{r}_{ok} \times \underline{\underline{B_o}}\,\mathbf{\Lambda}_k\,\ddot{\bar{\theta}}_k + \bar{a}_r \tag{4.104}$$

where the remainder term is

$$\bar{a}_r = \underline{\underline{B_i}}\,\bar{\omega}_i \times \left(\underline{\underline{B_i}}\,\bar{\omega}_i \times \underline{\underline{N}}\,\bar{r}_{ik}\right) + \underline{\underline{B_o}}\,\bar{\omega}_o \times \left(\underline{\underline{B_o}}\,\bar{\omega}_o \times \underline{\underline{N}}\,\bar{r}_{ok}\right) + \bar{\alpha}_r \times \underline{\underline{N}}\,\bar{r}_{ok} \tag{4.105}$$

Stacking all remainder translational terms leads to

$$\{\bar{g}(\dot{\boldsymbol{V}}\,\bar{u})\} = \begin{bmatrix} \bar{0} \\ T_1^N\,\bar{\omega}_1^\times\,T_1^N\,\bar{\omega}_1^\times\,\bar{r}_{11} + T_2^N\,\bar{\omega}_2^\times\,T_2^N\,\bar{\omega}_2^\times\,\bar{r}_{21} - \bar{r}_{21}^\times\,\bar{\alpha}_r^2 \\ T_1^N\,\bar{\omega}_1^\times\,T_1^N\,\bar{\omega}_1^\times\,\bar{r}_{11} + T_2^N\,\bar{\omega}_2^\times\,T_2^N\,\bar{\omega}_2^\times\,\bar{r}_{21} - \bar{r}_{21}^\times\,\bar{\alpha}_r^2 + T_2^N\,\bar{\omega}_2^\times\,T_2^N\,\bar{\omega}_2^\times\,\bar{r}_{22} + T_3^N\,\bar{\omega}_3^\times\,T_3^N\,\bar{\omega}_3^\times\,\bar{r}_{32} - \bar{r}_{32}^\times\,\bar{\alpha}_r^3 \\ T_1^N\,\bar{\omega}_1^\times\,T_1^N\,\bar{\omega}_1^\times\,\bar{r}_{13} + T_4^N\,\bar{\omega}_4^\times\,T_4^N\,\bar{\omega}_4^\times\,\bar{r}_{43} - \bar{r}_{43}^\times\,\bar{\alpha}_r^4 \\ T_1^N\,\bar{\omega}_1^\times\,T_1^N\,\bar{\omega}_1^\times\,\bar{r}_{13} + T_4^N\,\bar{\omega}_4^\times\,T_4^N\,\bar{\omega}_4^\times\,\bar{r}_{43} - \bar{r}_{43}^\times\,\bar{\alpha}_r^4 + T_4^N\,\bar{\omega}_4^\times\,T_4^N\,\bar{\omega}_4^\times\,\bar{r}_{44} + T_5^N\,\bar{\omega}_5^\times\,T_5^N\,\bar{\omega}_5^\times\,\bar{r}_{54} - \bar{r}_{54}^\times\,\bar{\alpha}_r^5 \end{bmatrix} \tag{4.106}$$

Substitution into Kane's Equations

Finally, substituting the kinematic acceleration decompositions

$$\{\bar{\alpha}\} = \boldsymbol{\Omega}\,\dot{u} + \{\bar{f}\} \qquad \{\bar{a}\} = \boldsymbol{V}\,\dot{u} + \{\bar{g}\}$$

into Kane's matrix equation

$$\boldsymbol{V}^T\{\bar{F}\} + \boldsymbol{\Omega}^T\{\bar{\tau}\} = \boldsymbol{V}^T\boldsymbol{M}\{\bar{a}\} + \boldsymbol{\Omega}^T\left(\boldsymbol{J}\{\bar{\alpha}\} + \{\bar{\omega}\}^\times\boldsymbol{J}\{\bar{\omega}\}\right)$$

yields the expanded form

$$\boldsymbol{V}^T\{\bar{F}\} + \boldsymbol{\Omega}^T\{\bar{\tau}\} = (\boldsymbol{V}^T\boldsymbol{M}\boldsymbol{V} + \boldsymbol{\Omega}^T\boldsymbol{J}\boldsymbol{\Omega})\dot{u} + \boldsymbol{V}^T\boldsymbol{M}\{\bar{g}\} + \boldsymbol{\Omega}^T\boldsymbol{J}\{\bar{f}\} + \boldsymbol{\Omega}^T\{\bar{\omega}\}^\times\boldsymbol{J}\{\bar{\omega}\}$$

Defining

$$\mathbf{W} = \boldsymbol{V}^T\boldsymbol{M}\boldsymbol{V} + \boldsymbol{\Omega}^T\boldsymbol{J}\boldsymbol{\Omega} \quad \text{and} \quad \mathbf{RHS} = \boldsymbol{V}^T\boldsymbol{M}\{\bar{g}\} + \boldsymbol{\Omega}^T\boldsymbol{J}\{\bar{f}\} + \boldsymbol{\Omega}^T\{\bar{\omega}\}^\times\boldsymbol{J}\{\bar{\omega}\}$$

the final first-order system is obtained:

$$\mathbf{W}\,\dot{u} = \boldsymbol{V}^T\{\bar{F}\} + \boldsymbol{\Omega}^T\{\bar{\tau}\} - \mathbf{RHS} \tag{4.107}$$

This completes the derivation, clearly showing the separation between the $\boldsymbol{\Omega}\dot{u}$ and $\boldsymbol{V}\dot{u}$ components and the remainder terms $\{\bar{f}\}$ and $\{\bar{g}\}$ that capture gyroscopic, Coriolis, and configuration-dependent effects in Kane's equations.

4.3.4. Application of Kane's Equations with Internal Torques

In the present system, it is assumed that no direct external forces act on the spacecraft structure. Therefore, the generalized force vector corresponding to translational motion is zero,

$$\{\bar{F}\} = \bar{0} \tag{4.108}$$

Under this assumption, the motion of the multi-body system is governed solely by the control torques acting about the centralised node and the internal joint torques. Each joint torque is modelled using a linear torsional spring‑damper representation to capture the elastic and dissipative behaviour of the joint:

$$\bar{\tau}_{j,k} = -K_k\,\bar{\theta}_k - C_k\,\dot{\bar{\theta}}_k \tag{4.109}$$

Here, K_k is the torsional stiffness coefficient of the k-th joint, and C_k is the torsional damping coefficient. The vector $\bar{\theta}_k$ represents the relative angular displacement between the connected bodies, while $\dot{\bar{\theta}}_k$ is the relative angular velocity across the same joint. The negative sign indicates that the generated torque opposes the relative rotation, providing a restoring moment proportional to the angular displacement (spring term) and an energy-dissipating moment proportional to the relative angular velocity (damping term). This formulation enables realistic modelling of structural flexibility and joint dynamics within the multi-body system. The torques present can be described as a toque vector defined as

$$\bar{\tau}_{SC} = \begin{bmatrix}\bar{\tau}_c^T & \bar{\tau}_{j,1}^T & \bar{\tau}_{j,2}^T & \bar{\tau}_{j,3}^T & \bar{\tau}_{j,4}^T\end{bmatrix}^T \tag{4.110}$$

where $\bar{\tau}_c$ represents the control torque applied to the central (node) CubeSat, and where each term $\bar{\tau}_{j,k}$ corresponds to the torque acting about the k-th joint connecting two spacecraft bodies (or links).

While the control torque only affects the central node, the joint torques both inner and outer node. The outer node will receive a positive torque feedback from the joint torque. The inner node will receive a negative feedback torque fro the joint torque because of the free rotation for both inner and outer nodes. A matrix $\boldsymbol{\Xi}$ can be used to map the torques such that each nodes experiences the correct torque in the correct sign. This mapping matrix for Alticube+ can be designed as the following.

$$\boldsymbol{\Xi} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & -\boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} & -\boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} & -\boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} & -\boldsymbol{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \tag{4.111}$$

Using this mapping matrix, equation 4.112 then computes the torque vector that can be applied in Kane's equation.

$$\{\bar{\tau}\} = \boldsymbol{\Xi}\bar{\tau}_{SC} \tag{4.112}$$

Finally Substituting $\{\bar{F}\} = \bar{0}$ and the above torque vector into equation 4.107 gives the final dynamics equation.

$$\mathbf{W}\,\dot{\bar{u}} = \boldsymbol{\Omega}^T\boldsymbol{\Xi}\bar{\tau}_{SC} - \mathbf{RHS} \tag{4.113}$$

Equation 4.113 defines the system dynamics as driven purely by control torques and internal joint torques. The first term on the right-hand side represents the direct contributions from commanded actuator torques and internal joint torques, while the second term captures the remaining effects from the motion of each nodes.

4.4. Reaction Wheel Model

As mentioned in section 2.3, the reaction wheels from Astrofein are used in Alticube+. Four of these reaction wheels are configured in a tetrahedral formation (see Figure 4.7). A tetrahedral configuration ensures three-axis pointing control while providing redundancy. The orientation of the reaction wheel assembly is provided in Figure 4.7, where the normal vector of the baseplate is pointed towards the positive Y-axis of the spacecraft reference frame (see Figure 2.12 for reference).

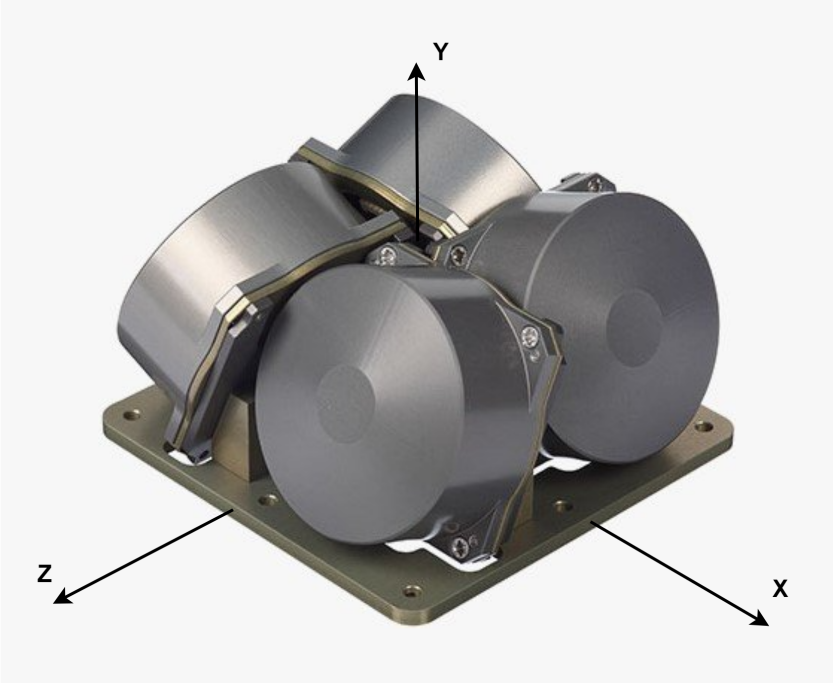


Figure 4.7: Tetrahedral configuration of reaction wheels

The assembly is located in the geometric centre along the Y-axis, but is shifted in both X and Z-axis. Knowing that the reaction wheel assembly is located in the centre of the +X and +Z cells, the shift is $100\,mm$ in both directions.

4.4.1. Reaction Wheel Control Logic

As the reaction wheels take an important role in controlling the spacecraft, a sub-model of the reaction wheel assembly is needed to ensure that certain hardware limitations are not exceeded. The first hardware limitation is derived from the maximum torque that can be delivered. Using the assumption that all reaction wheels are operable, the maximum torque for for each axis is described implicitly by first converting the requested torque to the right combination of reaction wheel activations. Then for each individual reaction wheel it is checked whether the maximum torque threshold is exceeded. The basic framework starts with equation 4.114.

$$\bar{\tau}_{cmd} = \boldsymbol{A}_{RW}\bar{\tau}_{RW} \tag{4.114}$$

Here, $\bar{\tau}_{cmd}$ is the vector of commanded torques, $\bar{\tau}_{RW}$ the vector of the four reaction wheel torques and $\boldsymbol{A_{RW}}$ a 3×4 matrix that translates the applied torque by the reaction wheels to the spacecraft body reference frame. The columns represent the unit torque vector for each of the reaction wheels. With this matrix having a Rank$(\boldsymbol{A_{RW}}) = 3$, inverting this matrix is done through the Moore-Penrose Pseudo-inverse. The allocated torques are then obtained by inverting equation 4.114. Equation 4.115 gives the allocated torques based on the requested torques.

$$\bar{\tau}_{RW} = \boldsymbol{A}_{RW}^{T}\left(\boldsymbol{A}_{RW}\boldsymbol{A}_{RW}^{T}\right)^{-1}\bar{\tau}_{cmd} \quad ; \quad \tau_{RW,i} \leq 2\,mNm \tag{4.115}$$

After allocation of the torques for each of the reaction wheels, a boundary condition on the maximum torque is placed as seen in equation 4.115. To define matrix $\boldsymbol{A}_{RW}$, each of the unit torque vectors needs to be identified. For the tetrahedral configuration seen in Figure 4.7, it can be seen that two of the four reaction wheels have their unit torque vector pointed in the X-Y plane. The remaining two reaction wheels have their unit torque vector pointed in the Z-Y plane. Another property of this configuration is that the angle from the Y-axis is equal for all reaction wheels. Figure 4.8 shows how the reaction wheel unit torque vector can be described by their angle from the y-axis. Assuming an angle $\alpha_{RW} = 15\,deg$ gives the following matrix.

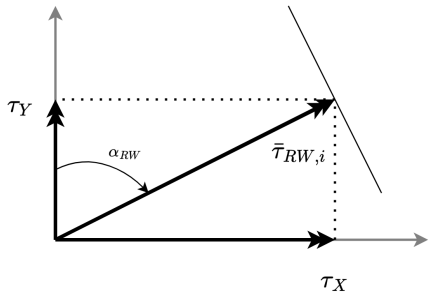


Figure 4.8: Torque unit vector decomposition of reaction wheel

$$\boldsymbol{A}_{RW} = \begin{bmatrix} \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} \end{bmatrix}$$

The reason for this specific angle is it closely matches the angle seen in Figure 4.7. Next to this, the torque outputted in the X or Z direction is around $97\,\%$ of the maximum torque available, which is needed to maximize the torque in these two directions.
The second hardware limitation is the maximum angular momentum storage for each of the reaction wheels. The increase in angular momentum of the reaction wheel can be calculated using equation 4.116.

$$L_{i,k+1} = L_{i,k} + \Delta L_i \tag{4.116}$$

$$\Delta L_i = \tau_{i,k} \Delta t \quad ; \quad |L_{i,k+1}| \leq 0.03\,Nms$$

Here, L_i is the angular momentum for one of four reaction wheels. So this calculation of the angular momentum storage is done for each reaction wheel separately. If the angular momentum storage of a specific reaction wheel is exceeded, then the reaction wheel will not be able to give the commanded torques. Only torques that decrease its angular momentum storage can be given by the reaction wheel. A flow chart for this reaction wheel control logic can be seen in Figure 4.9. After computing the resulting torque output for each of the reaction wheels, this is then transformed back into an control torque $\bar{\tau}_c$ using equation 4.114.

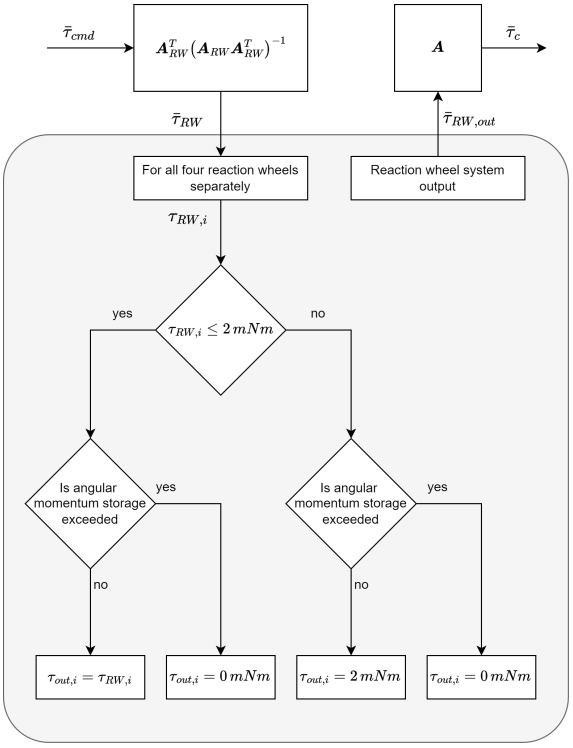


Figure 4.9: Flow chart for reaction wheel control logic

4.4.2. Reaction Wheel Jitter

A key aspect of answering **RQ-3** is the reaction wheel jitter. This section will develop a jitter model that describes the jitter behaviour of the reaction wheel system, starting with a definition of what reaction wheel jitter is. Jitter is a product from the mass imbalance of a reaction wheel during manufacturing. In theory, the mass of a reaction wheel is homogenous and will provide a pure torque. In practice, due to slight inconsistencies, the centre of mass is slightly off from the theoretical centre of mass. Figure 4.10 shows the two possible mass imbalances for a reaction wheel.

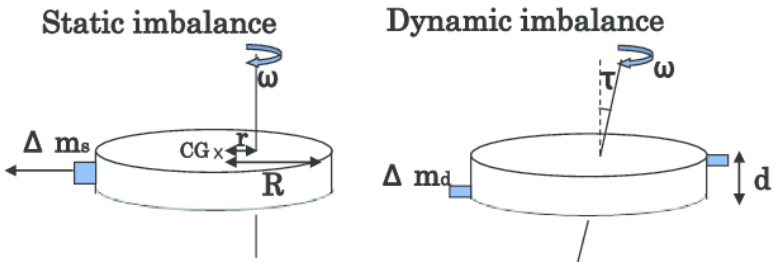


Figure 4.10: Reaction wheel Static imbalance (left) and Dynamic imbalance (right) [57]

The left figure shows the static imbalance for a reaction wheel, where the distribution of mass is not equal in all directions. This causes the centre of mass for the reaction wheel to be off-centre from the rotational axis, producing a jitter force. The magnitude of this force can be calculated using equation 4.117.

$$F_s = u_s \omega_{RW}^2 \tag{4.117}$$

Where u_s is the static imbalance constant for a reaction wheel. For a Cubesat which utilised reaction wheels, this value lies between $0.1 - 1.2\, g \cdot mm$ with an average of $0.25\, g \cdot mm$ [58]. The right figure shows a dynamic imbalance, where the distribution of mass is different along the rotation axis. When the reaction wheel rotates around the axis, the mass imbalance causes the reaction wheel to want to tilt, producing a jitter torque. This torque can be expressed by equation 4.118.

$$T_d = u_d \omega_{RW}^2 \tag{4.118}$$

Here, u_d is the dynamic imbalance constant for the reaction wheel. For a Cubesat this value lies between $1 - 10\, g \cdot mm^2$ with an average of $2.5\, g \cdot mm^2$ [58]. With these two equations, the magnitude of these forces and torques can be calculated for a range of reaction wheel speeds. Since the reaction wheel can wind up to a speed of $5000\, rpm$, figure 4.11 shows the produced jitter forces and torques up until this speed for the entire range of static and dynamic imbalance constants.

From the figure, it can be observed that the jitter force can reach up to $0.33\, N$ in magnitude. On the other hand, the jitter torque produced due to the dynamics imbalance is significantly lower and reaches a maximum of $2.7 \cdot 10^{-3}\, Nm$. For the remainder of this study, only the static imbalance is investigated as the dynamic imbalance of the reaction wheel is at least an order of magnitude lower.
To complete the reaction wheel model framework, the produced jitter by a single reaction wheel needs to be transformed from the local frame to the body reference frame $\mathcal{F}_b$. First, the force vector for the reaction wheel jitter can be expressed by equation 4.119.

$$\bar{F}_{flat}(t) = \begin{bmatrix} u_s \omega_{RW}^2 \cos(\omega_{RW} t + \phi) \\ 0 \\ u_s \omega_{RW}^2 \sin(\omega_{RW} t + \phi) \end{bmatrix} \tag{4.119}$$

The force vector here simulates a reaction wheel that lays flat on the X-Z plane, resulting in a force that rotates around the Y axis (See Figure 4.7 for reference on the axis direction) with the same rotational speed as the reaction wheel. Furthermore, in the sine and cosine is a phase ϕ which represents the

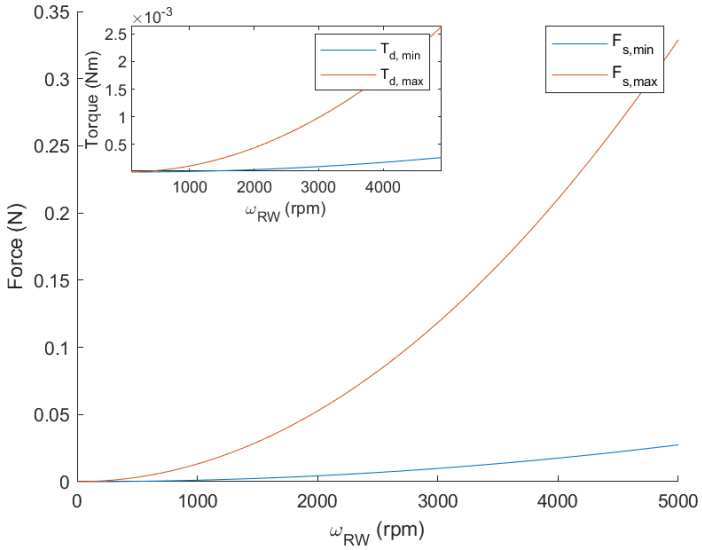


Figure 4.11: Produced jitter force from static imbalance and jitter torque from dynamic imbalance

offset of the mass distribution. From figure 4.7, each of the reaction wheels just has to be rotated with respect to either X or Z axis by $\pm\alpha_{RW}$. The transformation of the jitter force to the body reference frame can be done using either equation 4.120 or equation 4.121 depending on which reaction wheel.

$$\bar{F}_{jitter,z}(t) = \begin{bmatrix} \cos\pm\alpha_{RW} & -\sin\pm\alpha_{RW} & 0 \\ \sin\pm\alpha_{RW} & \cos\pm\alpha_{RW} & 0 \\ 0 & 0 & 1 \end{bmatrix} \bar{F}_{flat}(t) \tag{4.120}$$

$$\bar{F}_{jitter,x}(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\pm\alpha_{RW} & -\sin\pm\alpha_{RW} \\ 0 & \sin\pm\alpha_{RW} & \cos\pm\alpha_{RW} \end{bmatrix} \bar{F}_{flat}(t) \tag{4.121}$$

The framework of the reaction wheel jitter model can be put into a block diagram which shows how the jitter is calculated from the angular momentum of a individual reaction wheel (see Figure 4.12). Here, the angular momentum for the reaction wheels can be defined as $\bar{L}_{RW} = \begin{bmatrix} L_1 & L_2 & L_3 & L_4 \end{bmatrix}^T$. As seen from the arrow at the top, this block diagram goes through all reaction wheels individually. Then the resulting force can be computed by summing the produced jitters together into one vector.

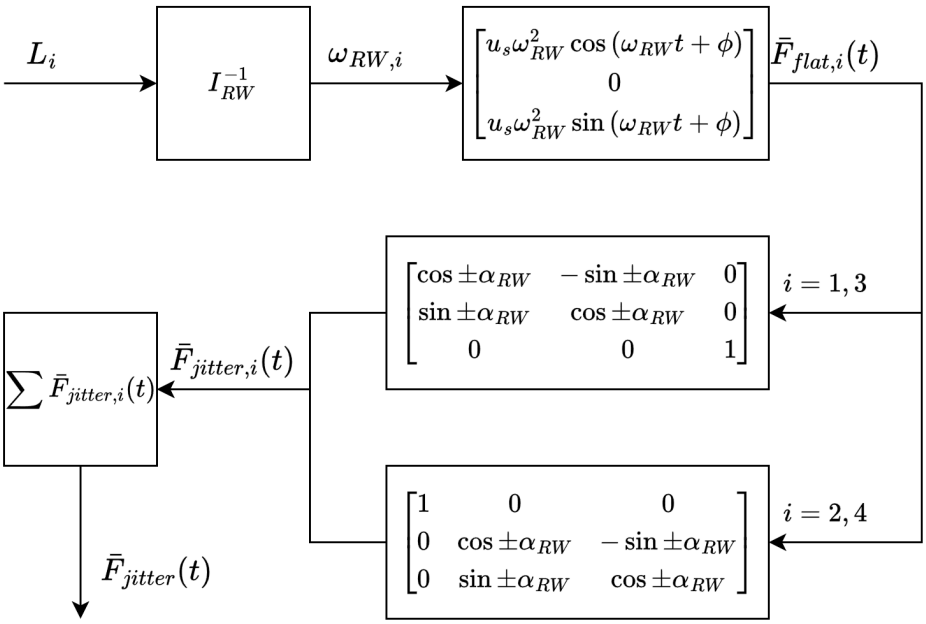


Figure 4.12: Block diagram for the reaction wheel jitter vector

Finally, the force needs to be translated into torques and can be written as.

$$\bar{\tau}_{jitter}(t) = \bar{r}_{RW} \times \bar{F}_{jitter}(t) \tag{4.122}$$

Where $\bar{r}_{RW} = \begin{bmatrix} 0.1 & 0 & 0.1 \end{bmatrix}^T$ is the arm between the centre of mass and the centre of the reaction wheel system.

5
Controller Design

Chapter 3 answered **RQ-1** and presented the ADCS requirements for Alticube+'s scientific operations phase. Chapter 4 defined a modelling framework and derived the spacecraft dynamics for the rigid body case. This chapter will aim to answer **RQ-2** by exploring the controller design using LQR and a Kalman filter. Answering **RQ-2** will be done by using two smaller sub-questions. First, a formulation for a centralised attitude controller is made that can control all three axis for the rigid body case. Then various weighting matrices for the cost function are explored to analyse the trade-off between pointing performance and actuator effort. Lastly, then a Kalman filter for attitude estimation is explored and the mathematical steps are presented.

5.1. Linear Quadratic Regulator

The linear Quadratic Regulator (LQR) is a widely used optimal full-state feedback controller for spacecraft attitude control. LQR balances regulation performance against actuation effort in a tunable quadratic-cost sense. For small attitude errors around a nominal pointing, the non-linear rotational dynamics can be linearized and transformed into a standard linear time-invariant (LTI) state-space, after which the LQR computes a constant gain matrix $\boldsymbol{K}$ which will minimise the quadratic cost function seen in equation 5.1.

$$J = \int_0^\infty (\bar{x}^T \boldsymbol{Q} \bar{x} + \bar{u} \boldsymbol{R} \bar{u}) dt \quad ; \quad \boldsymbol{Q} = \boldsymbol{Q}^T \succeq \mathbf{0} \, , \, \boldsymbol{R} = \boldsymbol{R}^T \succ \mathbf{0} \tag{5.1}$$

Where $\bar{x}$ is the state vector of Alticube+ and $\bar{u}$ the input vector of the reaction wheel system. The matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are user-chosen positive semi-/definite weighting matrices. The goal for this problem is to find the optimal cost function $J^*(\bar{x})$ that satisfies the Hamilton-Jacobi-Bellman (HJB) equation.

$$\forall \bar{x}, \quad 0 = \min_{\bar{u}} \left[\bar{x}^T \boldsymbol{Q} \bar{x} + \bar{u}^T \boldsymbol{R} \bar{u} + \frac{\partial J^*}{\partial \bar{x}} (\boldsymbol{A} \bar{x} + \boldsymbol{B} \bar{u}) \right] \tag{5.2}$$

Now let the optimal cost function be in the form of

$$J^*(\bar{x}) = \bar{x}^T \boldsymbol{S} \bar{x} \quad ; \quad \boldsymbol{S} = \boldsymbol{S}^T \succeq 0 \tag{5.3}$$

Where $\boldsymbol{S}$ is a positive semi-definite matrix. The gradient of equation 5.2 is

$$\frac{\partial J^*}{\partial \bar{x}} = 2 \bar{x}^T \boldsymbol{S} \tag{5.4}$$

Substituting equation 5.4 into equation 5.2 and solving for $\bar{u}$ gives equation 5.5

$$\frac{\partial}{\partial \bar{u}} = 2 \bar{u}^T \boldsymbol{R} + 2 \bar{x}^T \boldsymbol{S} \boldsymbol{B} = 0$$

$$\bar{u}^* = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S}\bar{x} = -\boldsymbol{K}\bar{x} \tag{5.5}$$

Equation 5.5 shows how the the optimal input can be achieved, by using the feedback gain matrix $\boldsymbol{K}$. To find the solution for $\boldsymbol{K}$, equation 5.5 is substituted back into equation 5.2.

$$0 = \bar{x}^T \left[\boldsymbol{Q} - \boldsymbol{S}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{S}\right]\bar{x} \tag{5.6}$$

Because this equation needs to hold for all $\bar{x}$, the part inside the brackets just needs to be solved for $\boldsymbol{S}$. After solving $\boldsymbol{S}$, the feedback gain matrix $\boldsymbol{K}$ can then be computed using equation 5.5 in a MATLAB solver.

Transfer Function

With the method to compute the LQR gain feedback matrix, the cost functions $\boldsymbol{Q}$ and $\boldsymbol{R}$ can shape the response of the LQR controller for Alticube+. As stated in section 2.5, the tuning of the LQR cost functions is done using the rigid spacecraft model. Figure 5.1 shows the block diagram of the closed-loop LQR controller for the state space model.

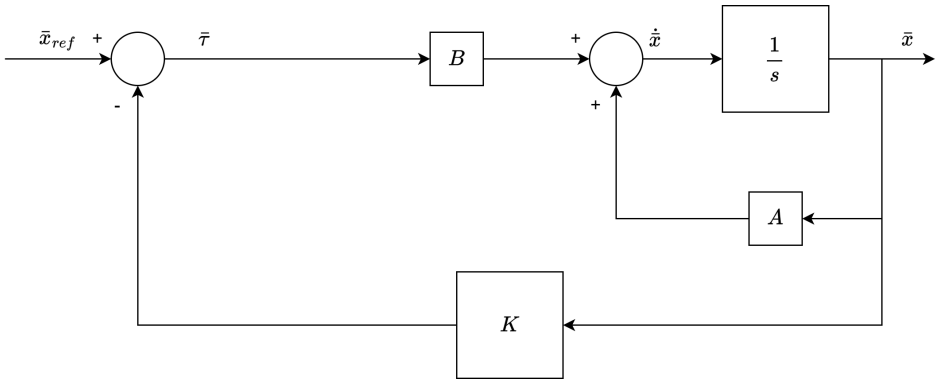


Figure 5.1: Closed-loop state space control block diagram with LQR feedback gain

To perform proper analysis of the cost tuning process, this closed-loop block diagram is transformed into a transfer function. The first step is to use the Laplace transform on equation 4.42, resulting in equation 5.7

$$sX(s) = \boldsymbol{A}X(s) + \boldsymbol{B}U(s) \tag{5.7}$$

Where $X(s)$ is the Laplace transform of $\bar{x}$ and $U(s)$ the Laplace transform of $\bar{\tau}$. substituting $U(s)$ with $X_{ref}(s) - \boldsymbol{K}X(s)$, the transfer function can be derived.

$$sX(s) = \boldsymbol{A}X(s) + \boldsymbol{B}(X_{ref}(s) - \boldsymbol{K}X(s)) \rightarrow X(s) = (s\boldsymbol{I} - (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}))^{-1}\boldsymbol{B}X_{ref}(s)$$

$$\frac{X(s)}{X_{ref}(s)} = (s\boldsymbol{I} - (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}))^{-1}\boldsymbol{B} \tag{5.8}$$

5.2. Q Matrix Tuning

Having formulated a centralised LQR controller for Alticube+ that can perform attitude regulation for all three axis, it is an important aspect to research how the Rigid-body dynamics system reacts to different cost functions. The first objective is to find a set of cost functions that produce a stable system for Alticube+. The second objective is to assess the pointing control performance against the actuator effort. Several methods can be utilised to obtain a good tuned feedback controller. One of these methods which will be utilised in this tuning process is the Q matrix tuning process, where the actuator

cost matrix $\boldsymbol{R}$ is set to a constant value. While tuning the state cost matrix $\boldsymbol{Q}$, the weights in matrix $\boldsymbol{R}$ will be set to one.

$$\boldsymbol{R} = R_u \boldsymbol{I}_{3\times 3} \quad ; \quad R_u = 1 \tag{5.9}$$

As an initial tuning process, matrix $\boldsymbol{Q}$ is split up into the attitude weights and angular rate weights.

$$\boldsymbol{Q} = \begin{bmatrix} Q_q \boldsymbol{I}_{3\times 3} & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & Q_\omega \boldsymbol{I}_{3\times 3} \end{bmatrix} \tag{5.10}$$

Here, Q_q is the weight factor for the attitude part of the state and Q_ω is the weight factor for the angular rate part of the state. By changing these factors, a different feedback gain $\boldsymbol{K}$ is calculated which will influence the closed-loop behaviour of the spacecraft.

Q_q Tuning, $Q_\omega = 1$
To analyse the stability of the closed-loop LQR system, a wide range of tuning parameters is taken for Q_q. The range can be defined as follows.

$$Q_q = 10^{\beta_q} \quad ; \quad \beta_q = -8, -7, ..., 0, ...7, 8 \tag{5.11}$$

Taking a large range of values for Q_q will show how the system reacts to for extremely small costs to extremely large costs. Plotting the response in a bode plot gives great insight into the close-loop stability and gain/phase margins. Figure 5.2 shows the bode diagram of θ_1 (roll) response. The remaining bode plots for the other states can be seen in Appendix B.

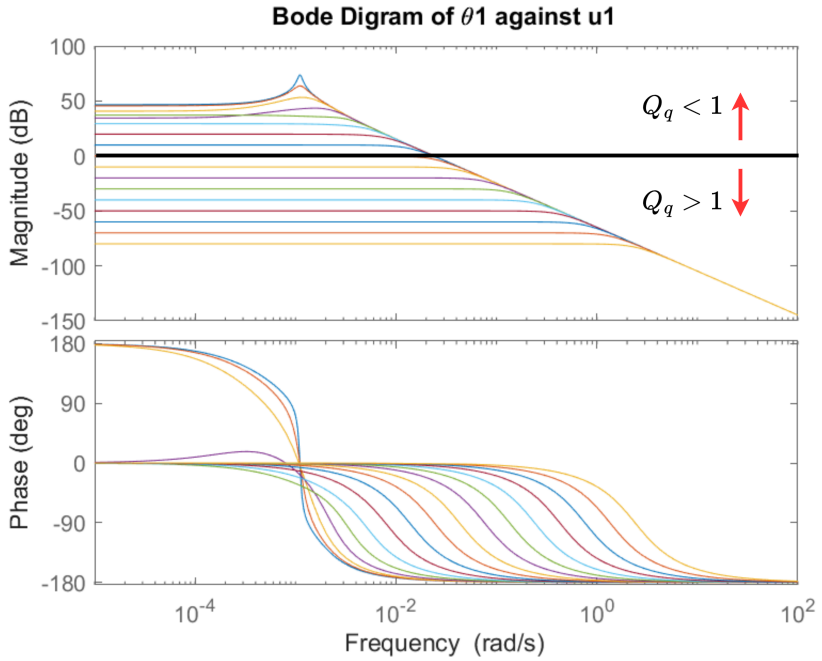


Figure 5.2: Bode diagram of θ_1 (roll) for various tuning values of Q_q

From this figure, it can be observed that low values of Q_q increases the magnitude and high values give very small magnitudes which are below the zero dB line. To keep a stable system for a particular axis, the magnitude needs to have a cross-over, requiring a positive magnitude for the lower frequencies. From this bode plot, the weighting values $Q_q < 1$ ensure a positive magnitude in the lower frequencies. The same goes for the bode plots that look at θ_2 (pitch) and θ_3 (yaw) which also require the Q_q to be smaller than one (See Figures in Appendix B.1).

Another property of the bode plots is that the gain and phase margin can be computed, which will give an idea on the robustness to uncertainty.

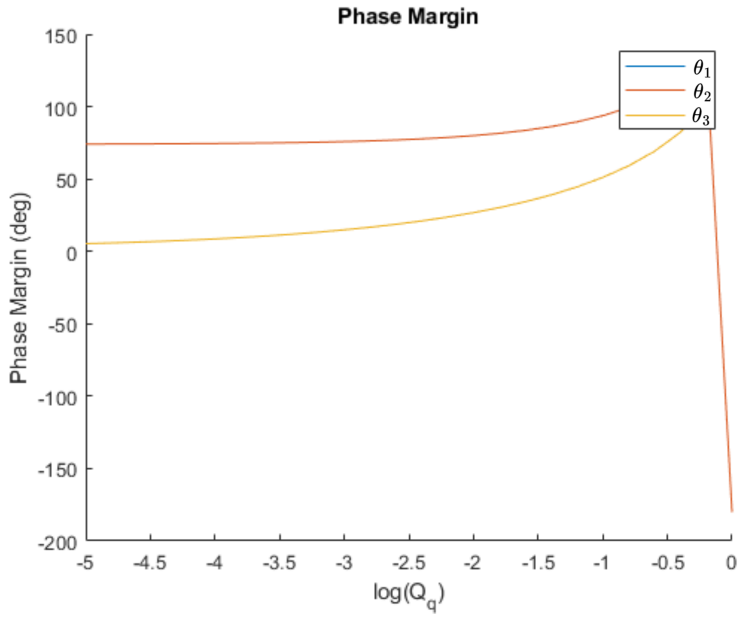


Figure 5.3: Phase margin of all axis for various values of Q_q

Figure 5.3 plots the Phase margin of all three attitudes. From this graph, it can be seen that the phase margin increases as the weighting factor increases to $Q_q = 1$. In order to have a stable system with sufficient margin for uncertainties, a phase margin greater than $30\,deg$ would be required. The weighting factor range that can satisfy the phase margin together with the gain requirement from above lies between $10^{-1.8} \leq Q_q < 1$. Specifically, having a weighting factor of $Q_q = 10^{-0.8} \approx 0.15$ would give a smooth damped response for both roll and yaw. Figure 5.3 shows that the phase margin for pitch is greater with respect to roll and yaw for all weighting factors. Using the same weighting factor for pitch will result in a phase margin of $> 80\,deg$. Having such larger phase margin for this axis creates a slow and unresponsive system and limits performance of the pointing control in this direction. To limit the phase margin above $60\,deg$ as much as possible, a lower weighting factor than $Q_q = 0.15$ would benefit the controllability of the pitch axis. This weighting factor would then be towards the $10^{-3} \leq Q_{q_2} \leq 10^{-2}$ which will give a phase margin of $75\,deg$.

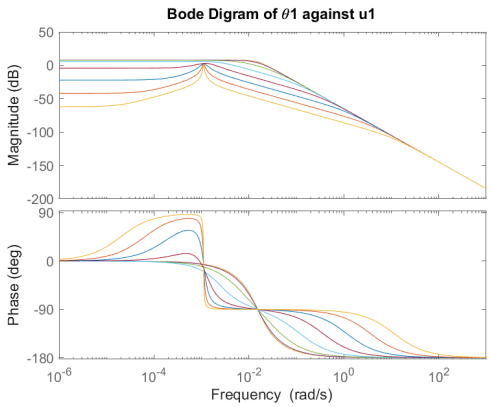
Q_ω Tuning, $Q_q = 0.15$

In the previous section, the tuning of Q_q was performed, and the resulting response from the system was explored. This section looks into how tuning the Q_ω weighting factor influences the response of the system. The range of weighting factors taken for Q_q can be defined as.

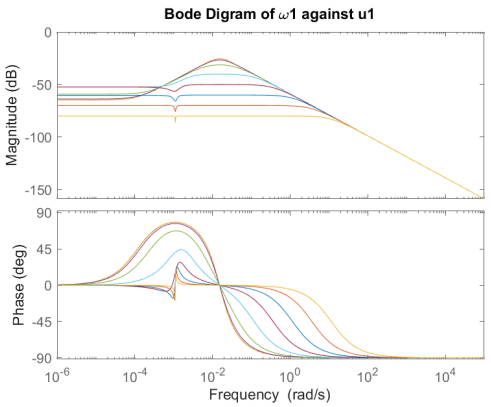
$$Q_\omega = 10^{\beta_\omega} \quad ; \quad \beta_\omega = -8, -7, ..., 0, ...7, 8 \tag{5.12}$$

Taking a large range of weighting factors will highlight the effect of the relative cost between the angular rates and attitudes. The response is plotted in figure 5.4a for θ_1 and figure 5.4b plots the response for ω_1.

From the response of θ_1, it can be observed that higher order values for Q_ω negatively impact the magnitude. However, it can also be observed that lower order values for Q_ω have negligible impact on the entire response for roll. The same can also be observed for pitch and yaw (See Figures in B.2). Similar effects can be observed in figure 5.4b, where higher order weighting factors lower the magnitude response, while low order weighting factors present little impact. Another observation is that the magnitude in figure 5.4b are below the $0\,dB$ line. This means that the angular rates are harder



(a) Bode diagram of θ_1 (roll) for various tuning values of Q_ω



(b) Bode diagram of ω_1 (roll rate) for various tuning values of Q_ω

Figure 5.4: Spacecraft state of Alticube+'s central node

to control. it also means that the weights in the feedback gain matrix $\boldsymbol{K}$ do not change much for the angular rates when tuning the cost function weights Q_ω.

5.3. Kalman Filter

Remembering the dynamics equation from equation 4.42, the derivation of the Kalman procedure starts with an slightly altered version of this.

$$\dot{\bar{x}}(t) = \boldsymbol{A_\ell}\bar{x}(t) + \boldsymbol{B_\ell}\bar{u}(t) + \boldsymbol{G}\bar{w}(t) \tag{5.13}$$

In addition to equation 4.42, $\bar{w}(t)$ represents the system noise with $\boldsymbol{G}$ being the system noise input matrix. As the Kalman filter uses measurements for attitude estimation, the measurement vector equation can be written.

$$\bar{z}(t) = \boldsymbol{H}\bar{x}(t) + \boldsymbol{D}\bar{u}(t) + \bar{v}(t) \tag{5.14}$$

Here, $\bar{z}(t)$ is the measurement vector observed from the sensors. $\boldsymbol{H}$ is the observation matrix, representing which states can be observed. $\bar{v}(t)$ is the measurement noise. Equation 5.13 and 5.14 make up a continuous set of equations. As this Kalman filter is used for onboard computer estimation, the system of equations needs to be discretised. Equation 5.15 and 5.16 present the discretised versions.

$$\bar{x}_{k+1} = \boldsymbol{\Phi}_{k+1,k}\bar{x}_k + \boldsymbol{\Psi}_{k+1,k}\bar{\tau}_k + \boldsymbol{\Gamma}_{k+1,k}\bar{w}_k \quad ; \quad \bar{w}_k \sim \mathcal{N}(0, \boldsymbol{\Sigma}) \tag{5.15}$$

$$\bar{z}_{k+1} = \boldsymbol{H}_{k+1}\bar{x}_{k+1} + \boldsymbol{D}_{k+1}\bar{u}_{k+1} + \bar{v}_{k+1} \quad ; \quad \bar{v}_k \sim \mathcal{N}(0, \boldsymbol{\Pi}) \tag{5.16}$$

Where $\bar{x}_k$ is the current state at step k and $\bar{x}_{k+1}$ is the next state for the step $k+1$. $\boldsymbol{\Phi}_{k+1,k}$ is the system transition matrix, while $\boldsymbol{\Psi}_{k+1,k}$ is the input distribution matrix. Matrices $\boldsymbol{H}_{k+1}$ and $\boldsymbol{D}_{k+1}$ remain the same compared to 4.42. For equation 5.15 and 5.16, both system- and sensor noise can be represented by white noise. Then this means that the following assumptions hold true.

$$\mathbb{E}[\bar{w}_k] = 0 \quad ; \quad \mathbb{E}[\bar{w}_k\bar{w}_k^T] = \boldsymbol{\Sigma}_{ij}$$

$$\mathbb{E}[\bar{v}_{k+1}] = 0 \quad ; \quad \mathbb{E}[\bar{v}_{k+1}\bar{v}_{k+1}^T] = \boldsymbol{\Pi}_{ij} \quad ; \quad \mathbb{E}[\bar{w}_k\bar{v}_{k+1}^T] = 0$$

How defining the mean and covariance of the state vector $\bar{x}_{k+1}$ as

$$\mathbb{E}[\bar{x}_{k+1}] = \hat{x}_{k+1,k} \quad ; \quad \mathbb{E}[(\bar{x}_{k+1} - \hat{x}_{k+1,k})(\bar{x}_{k+1} - \hat{x}_{k+1,k})^T] = \boldsymbol{P}_{k+1,k} \tag{5.17}$$

By defining the error vector as the error between the estimated state and the true state for step $k+1$, the covariance definition can be simplified.

$$\hat{\epsilon}_{k+1,k} = \hat{x}_{k+1,k} - \bar{x}_{k+1} \quad ; \quad \mathbb{E}[\hat{\epsilon}_{k+1,k}\hat{\epsilon}_{k+1,k}^T] = \boldsymbol{P}_{k+1,k} \tag{5.18}$$

In order to obtain a good estimate of $\bar{x}$, a cost function J can be created based on the quadratic weighted least squares estimate.

$$J = \frac{1}{2}\hat{\epsilon}_{k+1,k}^T\boldsymbol{P}_{k+1,k}^{-1}\hat{\epsilon}_{k+1,k} + \frac{1}{2}(\bar{z}_{k+1} - \boldsymbol{H}_{k+1}\hat{x}_{k+1})^T\boldsymbol{\mathcal{R}}_{k+1}^{-1}(\bar{z}_{k+1} - \boldsymbol{H}_{k+1}\hat{x}_{k+1}) \tag{5.19}$$

Then by minimising the cost of this quadratic equation leads to the first order derivative of equation 5.19 which can be seen in equation 5.20.

$$\frac{\partial J}{\partial \bar{x}_{k+1}} = \hat{\epsilon}_{k+1,k}^T\boldsymbol{P}_{k+1,k}^{-1} - (\bar{z}_{k+1} - \boldsymbol{H}_{k+1}\hat{x}_{k+1})^T\boldsymbol{R}_{k+1}\boldsymbol{H}_{k+1} = 0 \tag{5.20}$$

then by rearranging the terms, an update rule for the next predicted state based on the information up until $k+1$ [59].

$$\hat{x}_{k+1,k+1} = \hat{x}_{k+1,k} + \boldsymbol{\mathcal{K}}_{k+1}(\bar{z}_{k+1} - \boldsymbol{H}_{k+1}\hat{x}_{k+1,k}) \tag{5.21}$$

Here, $\boldsymbol{\mathcal{K}}_{k+1}$ is the Kalman filter gain matrix which updates the the predicted state $\hat{x}_{k+1,k}$ to obtain the corrected state $\hat{x}_{k+1,k+1}$. [59] Showed how a set of five equations can be used to update the Kalman filter, which are outlined here below.

Step 1: From the previous state $\bar{x}_{k,k}$, predict the next step using the discretized state space from equation 5.15.

$$\hat{x}_{k+1,k} = \boldsymbol{\Phi}_{k+1,k}\hat{x}_{k,k} + \boldsymbol{\Psi}_{k+1,k}\bar{\tau}_k; \quad \bar{x}_{0,0} = \bar{x}_0$$

Step 2: Compute the covariance matrix $\boldsymbol{P}_{k+1,k}$

$$\boldsymbol{P}_{k+1,k} = \boldsymbol{\Phi}_{k+1,k}\boldsymbol{P}_{k,k}\boldsymbol{\Phi}_{k+1,k}^T + \boldsymbol{\Sigma}; \quad \boldsymbol{P}_{0,0} = \boldsymbol{P}_0$$

Step 3: Determine the Kalman Gain $\boldsymbol{\mathcal{K}}_{k+1}$ which will be used to correct the predicted state form step 1.

$$\boldsymbol{\mathcal{K}}_{k+1} = \boldsymbol{P}_{k+1,k}\boldsymbol{H}_{k+1}^T\left(\boldsymbol{H}_{k+1}\boldsymbol{P}_{k+1,k}\boldsymbol{H}_{k+1}^T + \boldsymbol{\Pi}_{k+1}\right)$$

Step 4: Update the predicted state

$$\hat{x}_{k+1,k+1} = \hat{x}_{k+1,k} + \boldsymbol{\mathcal{K}}_{k+1}\left(\bar{z}_{k+1} - \boldsymbol{H}_{k+1}\hat{x}_{k+1,k}\right)$$

Step 5: Update the covariance matrix based on the Kalman gain

$$\boldsymbol{P}_{k+1,k+1} = (\boldsymbol{I}_{6\times6} - \boldsymbol{\mathcal{K}}_{k+1}\boldsymbol{H}_{k+1})\boldsymbol{P}_{k+1,k}(\boldsymbol{I}_{6\times6} - \boldsymbol{\mathcal{K}}_{k+1}\boldsymbol{H}_{k+1})^T + \boldsymbol{\mathcal{K}}_{k+1}\boldsymbol{\Pi}_{k+1}\boldsymbol{\mathcal{K}}_{k+1}^T$$

6

Simulation Results

Chapter 6 represents the culmination of the previous chapters which will answer the research questions **RQ-3** and **RQ-4**. To answer the research questions, the flexible model framework of Alticube+ presented in section 4.3 is used to investigate the interaction between the reaction wheel jitter and the effect this has on the pointing performance, thereby answering **RQ-3**. To answer **RQ-4**, a quantitive analysis is performed under the effect of an uncertain inertia matrix. This Chapter can be divided in the following sections. First the simulation setup is outlined. The second part discusses the verification methods used for the modelling simulation setup and spacecraft model. Thereafter, the results from the simulation cases are discussed.

6.1. Simulation Setup

All simulations of the Alticube+ spacecraft, conducted to address the research questions of this study, are implemented within the Simulink/MATLAB environment. The developed simulation framework allows both the rigid and flexible spacecraft models to be executed under the same environment, with minor modifications applied to the latter to account for the flexible modes, as described later in this section.

The overall simulation framework is organized as a single integrated environment composed of several interconnected sub-models. The top-level simulation loop, shown in Figure 6.1, consists of five main components. The first component is the LQR feedback gain block, which provides the control law based on the cost matrices defined in Section 5.1. The second component is the reaction wheel model, detailed in Section 4.4, which incorporates the physical limitations of the reaction wheel assembly, such as the maximum torque and angular momentum capacity.

The generated control torques are then combined with the external disturbance torques to form the total torque acting on the spacecraft. This total torque is passed to the spacecraft dynamics model, the core of the simulation loop, which can represent either the rigid-body or flexible-body dynamics depending on the simulation case. Using the applied torque, the model computes and propagates the spacecraft state vector $\bar{x}$.

Subsequently, the sensor model simulates the measurement process of the on-board sensors defined in this framework, introducing realistic sensor noise (See section 2.3 for reference on the sensor noise magnitudes). The final component of the loop is the Kalman filter, which estimates the spacecraft state based on the noisy sensor measurements. Each loop simulates $0.1\,s$ in real time, where the model propagates the state by this time increment. Each time increment, the sensors, Kalman Filter and control feedback gain are computed.

For the rigid spacecraft model, the simulation loop can be left as is, but an additional change in the simulation loop is required to implement the flexible model. The first difference between the two models is the additional flexible modes present in the dynamics. A time increment of $0.1\,s$ is not sufficient to effectively model the flexible dynamics of the spacecraft. In addition to the flexible modes, reaction wheel jitter needs to be modelled up to a frequency of $10\,Hz$. Then to accurately simulate the sinusoidal wave, a smaller $\Delta t = 0.01$ is required. However, sensor and control frequency is still limited to $10\,Hz$, still needing the $0.1\,s$ time increment. To satisfy both time increment criteria, a time increment

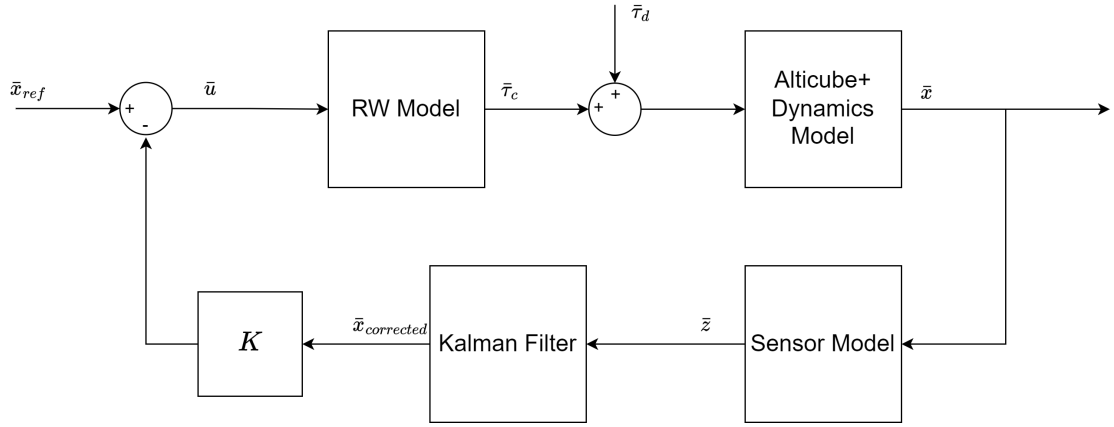


Figure 6.1: Block diagram of the rigid spacecraft simulation

conversion block is added in two places. The first block is added between the feedback gain block and the reaction wheel model, changing the time increment from $0.1\,s$ to $0.01\,s$. The second block is added after the spacecraft dynamics block, converting the simulation time back to $0.1\,s$ in preparation for the sensor model. Figure 6.2 gives a visual representation of this conversion in the form of a block diagram.

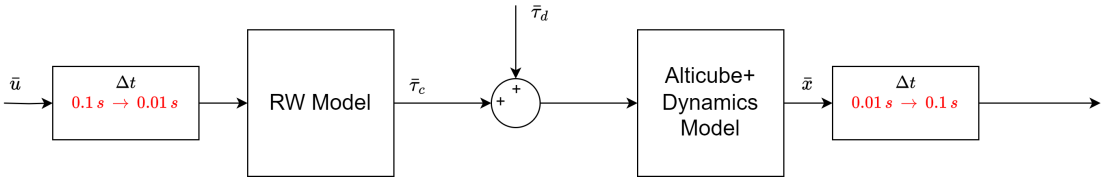


Figure 6.2: Modified part of the block diagram with time conversion blocks

Simulation Configuration

The simulation timeline is designed to emulate the operational sequence of the Alticube+ spacecraft under realistic mission conditions. The total simulation duration is set to 1200 s, corresponding to the maximum time window available for the scientific operation phase. Out of this period, 750 s are allocated for active payload operations. Consequently, the first 450 s of the simulation are reserved for the attitude control system to transition the spacecraft from its initial state to the desired operational attitude, in accordance with the performance requirements defined in Chapter 3.

To initiate each simulation, the spacecraft must be assigned a defined set of initial conditions. Throughout this study, multiple simulation cases are presented, each using distinct initial states to evaluate the performance of the attitude determination and control system (ADCS). Due to time constraints, it was not feasible to explore the complete range of possible initial conditions. Therefore, this work focuses on a limited but representative subset of parameters.

The primary configurable initial condition in this framework is the central node state vector, denoted as $\bar{x}_0$. By varying $\bar{x}_0$, the simulations assess how the ADCS responds to different initial states and investigate the range of initial conditions over which the system can achieve the required attitude accuracy and stability.

Other parameters in the flexible spacecraft model are treated as either constrained or quasi-constrained initial conditions. The constrained parameters—which remain fixed across all simulation cases—include the initial joint angles (θ_j), initial joint angular rates $(\dot{\theta}_j)$, and the initial partial velocity vector $\mathbf{u}$. Additionally, the initial state vector of the internal Kalman filter model is initialized as the central node state vector $\bar{x}_0$, but with an added estimation error sampled from a normal distribution $N(\bar{\mu}, \mathbf{\Sigma})$, representing realistic initialization uncertainty in the state estimation process. For the normal distribution,

the mean is equal to the initial state vector ($\bar{\mu} = \bar{x}_0$) and the uncertainty is $10\,\%$ of the initial state ($\Sigma = diag(0.1x_{0,1}, 0.1x_{0,2}, 0.1x_{0,3}, 0.1x_{0,4}, 0.1x_{0,5}, 0.1x_{0,6})$).These constraints ensure consistency in the simulation setup while maintaining a realistic representation of sensor and estimation uncertainties.

The quasi-constrained parameters are held constant unless explicitly stated otherwise. These include the inertia matrix $\mathbf{J}$, the flexible model stiffness coefficient k, and the stored angular momentum H. While these parameters influence the system dynamics, they are only adjusted in dedicated simulation cases aimed at sensitivity analysis or model validation.

Finally, the LQR cost matrices are fixed to the nominal values $\mathbf{Q}_i$ and $\mathbf{R}_i$ for all baseline simulations, unless specified otherwise in dedicated test cases. This ensures consistent control performance across simulations and isolates the effects of other varying parameters on system behaviour. The cost matrices used are expressed here below.

$$\boldsymbol{Q} = \begin{bmatrix} 0.15 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{-4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix} \quad ; \quad \boldsymbol{R} = \boldsymbol{I}_{3\times 3}$$

This configuration strategy ensures that the simulations remain focused on evaluating the ADCS performance under representative and physically consistent conditions, while maintaining sufficient flexibility for targeted investigations when required.

6.2. Verification

To ensure the accuracy and reliability of the developed simulation framework and the models implemented within it, a comprehensive verification process was carried out. The verification consisted of three main stages: sanity checks, unit tests, and model-level analysis. These steps were applied to both the rigid and flexible spacecraft models to confirm the correct implementation of their respective dynamics and interactions within the overall simulation environment.

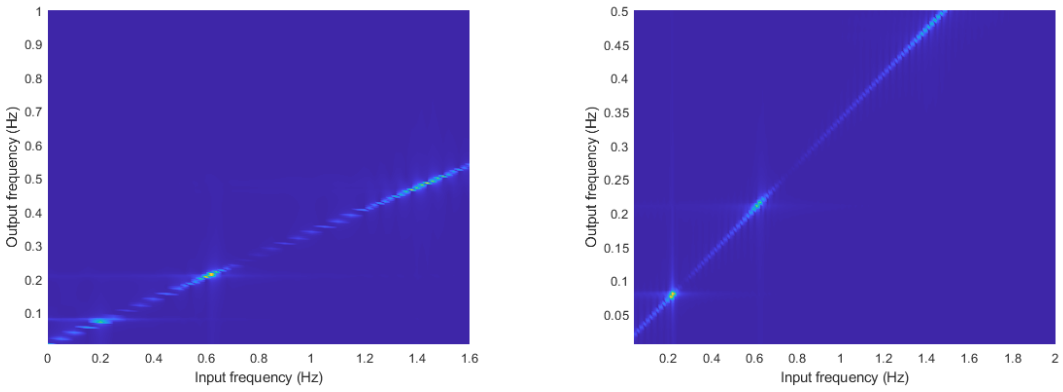
Initial sanity checks were performed to confirm the basic functionality and consistency of the simulation framework. These checks included verifying the correct flow of signals between sub-models, confirming physical consistency (e.g., conservation of angular momentum under torque-free conditions), and ensuring that the simulation responded logically to input perturbations.

Subsequently, unit tests were conducted on individual subsystems, such as the attitude control law, reaction wheel model, and sensor models. Each subsystem was tested independently to verify its expected behaviour under controlled conditions, allowing potential implementation errors to be identified and corrected before integration.

Verification of the Kalman filter focused on assessing its estimation performance under controlled conditions that reflect the expected operational behavior of the spacecraft. The filter was first tested using idealized sensor data without noise to confirm that it converged exactly to the true state, ensuring that the implementation of the prediction and update equations was correct. Subsequently, sensor noise was introduced to evaluate the filter' s convergence. The estimated states were compared against the true simulated states to verify that the estimation errors remained within the theoretical bounds predicted by the covariance matrices. These tests confirmed that the Kalman filter provides stable and accurate state estimates consistent with theoretical expectations across a range of operating conditions.

For the rigid spacecraft model, a detailed comparison was made between the simulation results and corresponding analytical solutions derived from rigid-body dynamics equations. This step ensured that the numerical integration and attitude kinematics were implemented correctly and that the model produced physically accurate responses under various torque inputs.

Validation of the flexible spacecraft model was carried out through a detailed comparison with an independent finite element model (FEM) developed by COMET. First, a modal frequency comparison was performed to ensure that the implemented lumped-parameter model reproduced the correct flexi-



(a) Roll/yaw vibration spectrum

(b) Pitch vibration spectrum

Figure 6.3: Vibration response spectrum from reaction wheel jitter applied on node #1

ble mode frequencies in roll, pitch, and yaw. From the FEM frequency analysis, the dominant flexible mode frequencies were $0.08\,Hz$, $0.41\,Hz$ and $0.47\,Hz$ for pitch, roll and yaw respectively [2]. In order to accurately compare the two models, a simulation environment was created in MATLAB. Alticube+'s flexible model was excited using sinusoidal torques swept across a frequency range of $0.01 - 10\,Hz$. The applied torque magnitude was set to $(0.9 \times 10^{-4} \cdot \sqrt{2}\,Nm)$, matching the excitation used in the COMET study to enable a direct comparison of frequency-domain characteristics. Figure 6.3 shows two vibration spectra for roll/yaw (left figure) and pitch (right figure). The reason behind combining the results for roll and yaw is due to the flexile model having a symmetry between those two planes, meaning that both roll and yaw will produce the same results giving the same inputs. The roll/yaw vibration spectrum shows two large peaks. The largest peak measured is at the output frequency of $0.48\,Hz$, comparable to the yaw resonance frequency of $0.47\,Hz$. However, this frequency is approximately $15\,\%$ higher compared to the roll frequency of $0.41\,Hz$ from the FEM model. Another smaller peak in the roll/yaw spectrum can be observed at $0.21\,Hz$. From analysis, this peak represents the rigid body resonance with the reaction wheel jitter as a change in torsional stiffness parameters did not shift the peak which was the case for the structural resonance peak at $0.48\,Hz$.
The pitch vibration spectrum shows that the largest peak is at $0.08\,Hz$, which aligns with COMET's model. A second validation step evaluated the maximum displacement of the flexible endpoints rela-

Table 6.1: Maximum absolute pointing errors [2]

Response point	Load point	COMET FEM (arcsec)	Lumped parameter (arcsec)	Error between models
CubeSat #1	CubeSat #1	7.02	8.07	$16.4\,\%$
CubeSat #5	CubeSat #1	7.00	7.12	$1.71\,\%$

tive to their nominal orientation under jitter excitation. The same torque magnitude and a jitter frequency of $3.75\,Hz$ was used, consistent with the conditions in COMET's FEM analysis. Table 6.1 presents the maximum pointing errors for COMET's FEM model and the lumped-parameter model. From the table, the lumped-parameter model shows larger maximum pointing errors compared to COMET's FEM model. This discrepancy is expected to some extent, as lumped-parameter models typically approximate distributed structural behaviour using fewer degrees of freedom and therefore tend to overestimate deflections when higher-order modes or local stiffness effects are not fully captured. Additionally, the simplifications in modelling the reaction wheel interface and the reduced geometric fidelity may contribute to higher predicted tip deflections. Another discrepancy is the inclusion of rigid body modes, together with the application of the torque during this simulation. Furthermore, the present model includes rigid-body modes in addition to flexible dynamics whereas COMET's FEM-based model only looks at the flexible deformation. The coupling between rigid-body motion and the flexible degrees of freedom can introduce small asymmetries in the predicted pointing errors, particularly when the excitation frequency lies near a combined rigid-flexible resonance. This coupling may therefore amplify

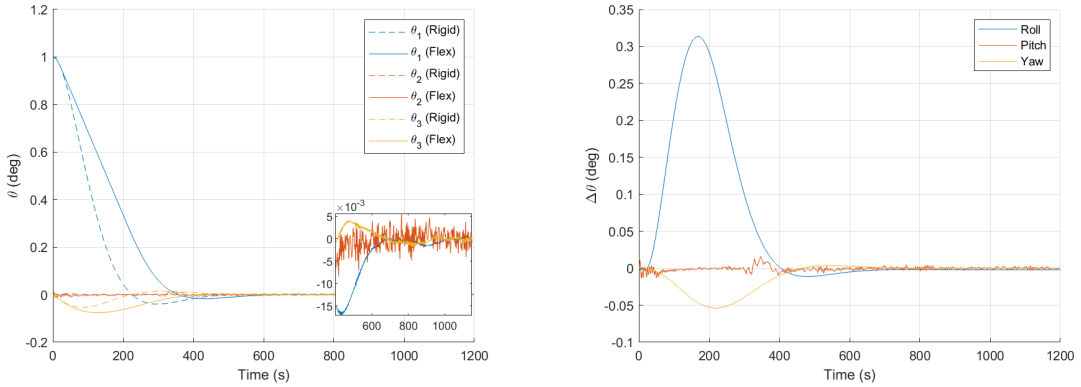
certain axis responses beyond what is observed in a flexible isolated deformation results by COMET. Despite these differences, the lumped model preserves the correct qualitative behaviour and the magnitude of the errors remains within the same order. This indicates that while the lumped-parameter approach is conservative in estimating maximum pointing errors, it remains sufficiently accurate for controller design, performance analysis, and jitter-propagation studies.

6.3. Alticube+ Manoeuvrer Simulation

Using the simulation environment described in section 6.1, research question **RQ-3** can addressed in this section. The aim is to explore the effects Alticube+'s reaction wheel system has on the flexible structure and its attitude stability. This research question is answered by comparing the flexible spacecraft model against the rigid spacecraft model, where the rigid model will be the baseline. Two simulations cases are compared. The first case is an angle manoeuvrer in the roll direction ($1\,deg$). The second case is a manoeuvrer in all directions. Lastly, initial condition parameters are varied to address the performance of the ADCS systems in performing manoeuvrers.

6.3.1. Case: Roll Manoeuvrer

The first case is a pure roll manoeuvrer. The spacecraft starts from the initial conditions $\bar{\theta}_0 = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^T$ and $\bar{\omega}_0 = \bar{0}$. This simulation will explore how the tuned LQR controller controls Alticube+ and how pitch and yaw are affected to this manoeuvrer for both rigid and flexible model.



(a) Roll manoeuvrer case for Rigid (dashed line) and flexible model (solid line)

(b) State difference of Alticube+'s central node between rigid and flexible model

Figure 6.4: Spacecraft attitude of the roll case for rigid and flexible model

Figure 6.4 presents the results from the roll manoeuvre simulation for both the rigid and flexible spacecraft models. As shown, the rigid-body model converges more quickly to the desired reference attitude compared to the flexible model. This difference arises from the fundamental characteristics of the two models: in the rigid case, the entire spacecraft responds instantaneously to the applied torque, whereas in the flexible model, the torque is applied at the central node and only directly affects the central node's rotation. The rotational motion must propagate through the flexible joints to affect other nodes. As a result, energy is first stored as elastic deformation within the boom joint before the motion is transmitted to the rest of the structure, introducing a clear dynamic lag in the overall response.

This lag is illustrated in the right-hand plot of Figure 6.4, which shows the difference in angular displacement between the flexible and rigid models, expressed as $\Delta\theta = \theta_{\text{Flex}} - \theta_{\text{Rigid}}$. The maximum lag in the roll direction reaches approximately 0.3° around 190 s into the simulation. A similar delay is observed in the yaw angle, where both models exhibit an increase in magnitude, but the flexible model again shows a phase lag relative to the rigid one.

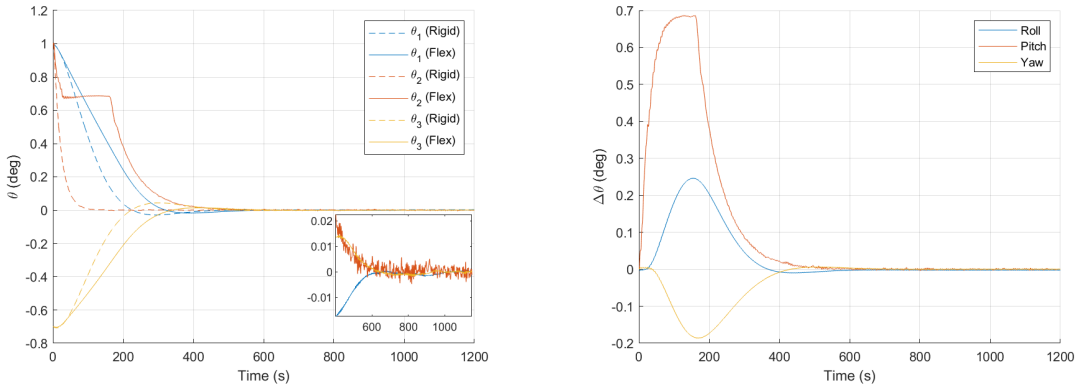
The deviation of the yaw response from the nominal zero offset can be attributed to the closed-loop system dynamics. Analysis of the system roots in the yaw direction reveals the presence of a right-half-plane zero in the Single-Input Single-Output (SISO) root locus between the roll torque input and the yaw angular rate output. Although the system remains stable, this non-minimum-phase zero introduces

an initial response opposite to the commanded direction, explaining the transient deviation observed in yaw.

Overall, the simulation clearly demonstrates the delay in system response introduced by the flexible dynamics. Nevertheless, both spacecraft models successfully achieve the attitude control requirements within the allocated 450 s manoeuvre period. A final noteworthy observation is that the flexible model exhibits a smaller overshoot in the roll response compared to the rigid model, indicating that the structural flexibility introduces a damping effect that slightly reduces the system's transient peak.

6.3.2. Case: Full Manoeuvrer

This second case follows from the first simulation case. The difference is that now different initial conditions are used for the manoeuvrer. The initial conditions use are $\bar{\theta}_0 = \begin{bmatrix} 1 & 1 & -0.7 \end{bmatrix}^T$ and $\bar{\omega}_0 = \bar{0}$. The aim here is to observe how the LQR controller performs when tasked with a multi-axis manoeuvrer.



(a) Full state manoeuvrer case for Rigid (dashed line) and flexible model (solid line)

(b) State difference of Alticube+'s central node between rigid and flexible model

Figure 6.5: Spacecraft Attitude of the full case for rigid and flexible model

Figure 6.5 presents the results from the full manoeuvre simulation, where the performance of the rigid and flexible spacecraft models is compared. The roll and yaw responses, shown in the left plot, exhibit similar qualitative behavior to that observed in the roll manoeuvre case (Figure 6.4). However, the pitch response in Figure 6.5 demonstrates a markedly different behavior from that seen previously.

After approximately 30 s, the pitch motion, initially offset by 1°, is partially corrected and stabilizes around 0.5° from the reference attitude. The spacecraft remains at this pitch attitude until around 160 s, when it resumes convergence toward the reference pitch attitude. This unexpected delay in pitch convergence warrants further investigation. Supporting plots for the flexible model simulation are provided in Appendix C.2 for reference.

Upon closer examination, this behaviour can be attributed to the angular momentum storage limits and torque coupling effects of the reaction wheel system. Figure 6.6 zooms in on the time interval during which the reaction wheels reach their nominal momentum storage capacity. It can be observed that the +X reaction wheel reaches its limit earlier than the -X wheel, and a similar trend occurs between the +Z and -Z wheels. This asymmetry directly influences the pitch dynamics through coupled torques generated by the wheel configuration.

Each reaction wheel primarily produces torque in the roll or yaw direction but also contributes secondary components in the pitch axis due to geometric coupling. Given the initial condition on the pitch axis, all four reaction wheels collectively exert a net torque in the pitch direction during the initial phase of the manoeuvre. Consequently, the +X and +Z wheels accelerate slightly more (assuming the maximum torque is not yet saturated), while the -X and -Z wheels decelerate correspondingly. Once the +X wheel reaches its maximum momentum capacity, it ceases to generate torque, whereas the -X wheel continues producing torque in both roll and pitch directions.

The subsequent deceleration of the pitch motion can be explained through the characteristics of the

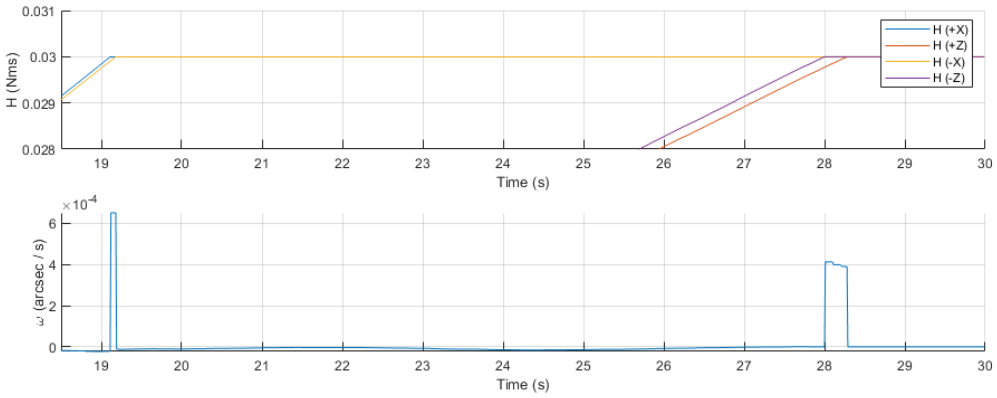


Figure 6.6: Reaction wheel system angular momentum capacity and torque produced in the pitch direction

LQR feedback gain matrix, $\mathbf{K}$. The controller places higher priority on correcting roll errors than pitch errors, resulting in a stronger control action in the roll axis. When a positive torque command is applied to the -X reaction wheel (accelerating the wheel), it produces the desired negative roll torque, but also induces a positive pitch torque due to coupling. A similar effect occurs between the +Z and -Z wheels, collectively acting to reduce the spacecraft's pitch motion.

This analysis shows that the pitch deceleration is a coupled by-product of both the reaction wheel momentum saturation and the axis interaction defined by the LQR gain structure. While the behaviour is non-ideal, it remains consistent with the system's dynamic characteristics and control prioritization strategy.

6.3.3. Variation in Initial State Conditions

Next to singular cases, it is important to obtain a broader view on how different variations in initial conditions changes the controlled states of Alticube+. In this case, the initial states are varied using a normal distribution $\mathcal{N}(\bar{\mu}, \mathbf{\Sigma})$. Here the normal distribution used for the initial attitude is $\mathcal{N}(\bar{0}, \sigma_\theta^2 \boldsymbol{I}_{3\times3})$ where $\sigma_\theta = 1\,deg$. The normal distribution used for the initial angular rates is $\mathcal{N}(\bar{0}, \sigma_\omega^2 \boldsymbol{I}_{3\times3})$ where $\sigma_\omega = 4\,arcsec$, where $3\sigma_\omega < 12.8\,arcsec$ is required because of reaction wheel angular momentum capacity.

Figures 6.7 present the roll, pitch, and yaw responses from the Monte-Carlo analysis. For each axis, a colour scheme is used to indicate whether the pointing control requirement **ADCS-PC-REQ-1** is satisfied at $t = 450\,s$. Red traces denote simulations in which the requirement is not met, orange traces indicate cases where the requirement is met but not to the desired level, and green traces reflect cases achieving the target pointing accuracy of Across all Monte-Carlo runs, approximately $14\,\%$ of the simulations fail to meet the requirement in at least one axis, while the remaining $86\,\%$ satisfy ADCS-PC-REQ-1. Importantly, in all simulations—successful or not—the LQR controller ultimately drives Alticube+ to the desired attitude. This confirms the closed-loop stability of the spacecraft's attitude control system.

The $14\,\%$ failure cases arise not from instability, but from the inability of the ADCS to bring the spacecraft into the required attitude within the available $450\,s$ window. The dominant limiting factor is reaction wheel saturation. In all unsuccessful cases, at least two of the four reaction wheels reach full angular-momentum capacity, and in some cases all four wheels saturate. When saturation occurs, the control system can no longer generate corrective torque, leaving Alticube+ in an uncontrolled rotational state until the wheel speeds naturally evolve to values where corrective torque becomes possible again.

Reaction wheel saturation is not limited to the unsuccessful runs. Indeed, Figure 6.8 shows the average angular momentum of the reaction wheel system across all simulations. Red traces correspond to runs that fail the requirement. Yellow traces represent simulations that ultimately succeed but experience at least one interval of full system saturation—this accounts for approximately $28\,\%$ of the successful $80\,\%$. Even cases meeting the requirement - $45\,\%$ of the total simulations - temporary reach saturation

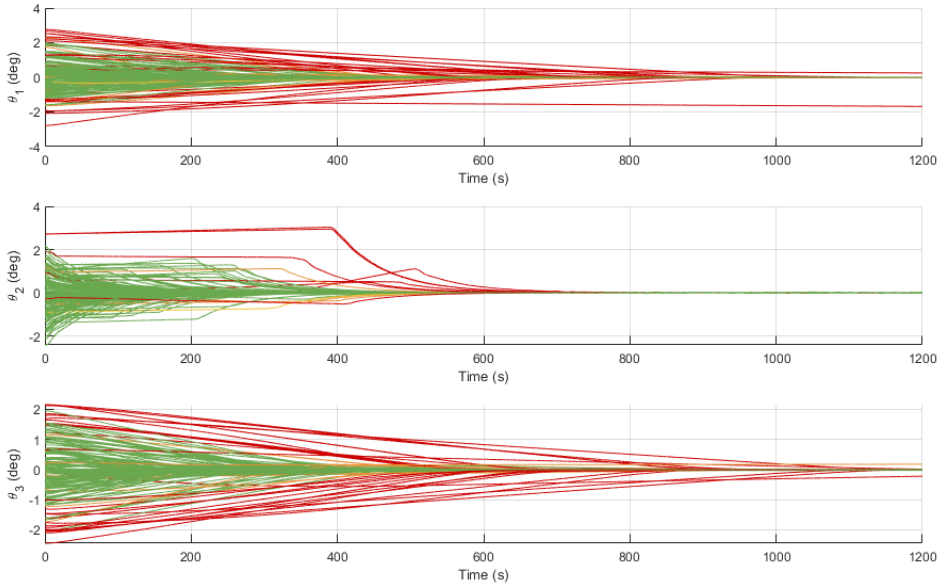


Figure 6.7: Alticube+ roll (θ_1), yaw (θ_2) and yaw (θ_3) states for $N = 200$ simulations

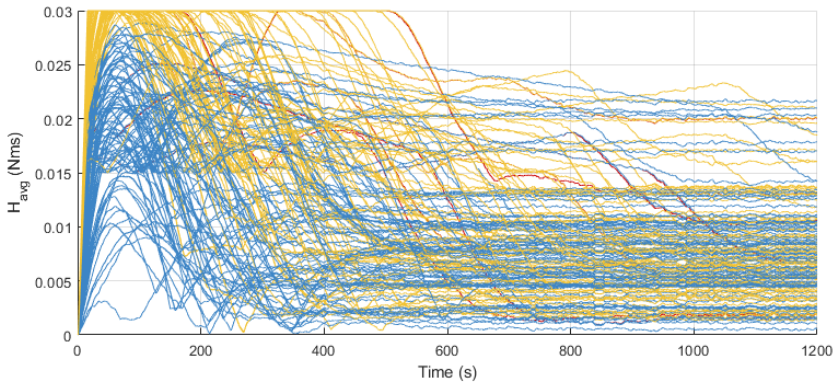


Figure 6.8: Average reaction wheel angular momentum of the entire reaction wheel system

limits and reduces the rate at which the spacecraft can reorient.

A saturated reaction wheel system poses two critical challenges:

1. Slower attitude transitions - the spacecraft cannot respond quickly to reference changes when torque authority is reduced or momentarily unavailable.
2. Increased vulnerability to disturbances - when fully saturated, the spacecraft cannot counter external torques, posing a risk to pointing performance and stability.

Mitigating these issues requires a design trade-off involving:

- increasing the total angular momentum storage of the reaction wheel assembly,
- increasing the allowed manoeuvre time, or
- reducing the allowable initial attitude offset.

One possible way to expand angular momentum capacity within the current system architecture is to employ multiple reaction-wheel assemblies. Besides providing additional stored momentum, this also increases the available control torque through simultaneous actuation. However, distributing reaction

wheels across multiple modules may introduce additional challenges, such as increased reaction wheel jitter transmitted through the flexible structure, possibly amplifying structural vibrations.

6.4. Jitter Induced Pointing Errors

In section 6.3, the results on the pointing control of Alticube+ using LQR and a Kalman filter was presented and discussed. Now in this section a close look is taken at the impact the reaction wheel system has on the pointing control and attitude stability of the baseline.

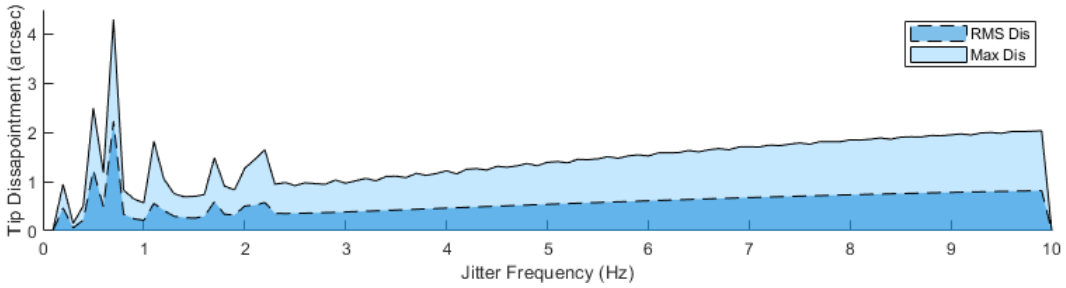


Figure 6.9: Jitter induced aggregated system tip disappointment.

Figure 6.9 shows the tip disappointment from the reaction wheel jitter at different modelled frequencies ($0 - 10\,Hz$). From this figure, the RMS value and the maximum value are covered. At the lower frequencies, multiple peaks can be observed at $0.1\,Hz$, $0.5\,Hz$ and $0.7\,Hz$. The largest spike vibrates at a frequency of $0.7\,Hz$ and has a maximum disappointment of 4 arcseconds. For frequencies $\geq 2.3\,Hz$, no spikes can be observed. This absence of structural resonance could be from the fact that the lumped parameter model is not able to model higher-order modes. Nevertheless, due to the increase in jitter torque, the tip disappointment does creep up slightly to around $2\,arcsec$ at $10\,Hz$. From this graph, it shows that the dominant flexible modes react to low frequency jitter inputs, even though the torque magnitude of the jitter is significantly smaller (2 orders or magnitude). These results stresses the importance of reaction wheel jitter mitigations for lower frequencies through angular momentum storage management.

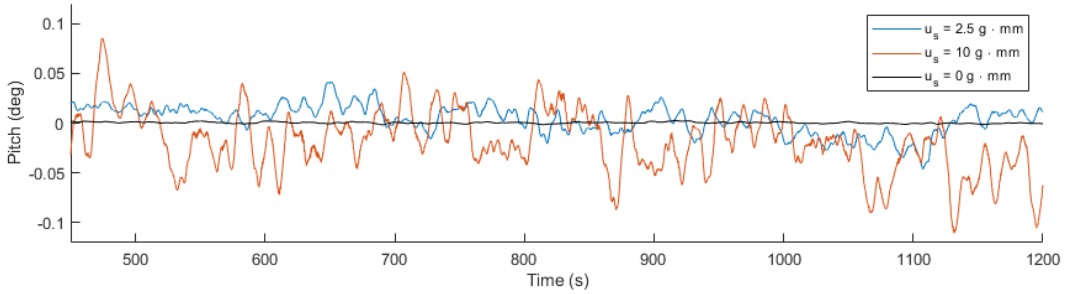


Figure 6.10: Alticube+ pitch without jitter (black line), average expected jitter (blue line) and maximum expected jitter (orange line).

Figure 6.10 presents the pitch of Alticube+ during the payload operation phase (i.e. after $t = 450\,s$). The black line represents the pitch for the flexible model without jitter. The blue line is the pitch for the flexible model using the average static imbalance as stated in section 4.4. The orange line represents the pitch where the reaction wheels has the maximum expected static imbalance. From this figure, it can be observed that the reaction wheels affect pitch through the flexible modes. The rigid model does feel the influence by the reaction wheel system, but can be easily compensated by the the controller. next to this, the reaction wheel jitter in the rigid body case needs to rotate all five satellites - with a larger mass moment of inertia in the pitch axis than for a single spacecraft - instead of just the central node. For the flexible model, the central node is able to rotate more freely than for the rigid spacecraft model. This can be observed as the pitch deviates from the reference with a mean value of $0.01\,deg$ RMS. Although this is just $10\,\%$ of the desired $0.1\,deg$ pointing control bandwidth, Alticube+ does reach a maximum

deviation of $0.05\,deg$, which leaves a $50\,\%$ margin. Increasing the static imbalance also increases the pitch instability as now Alticube+ deviates from the reference with a mean value of $0.04\,deg$ RMS and a maximum of $0.11\,deg$.

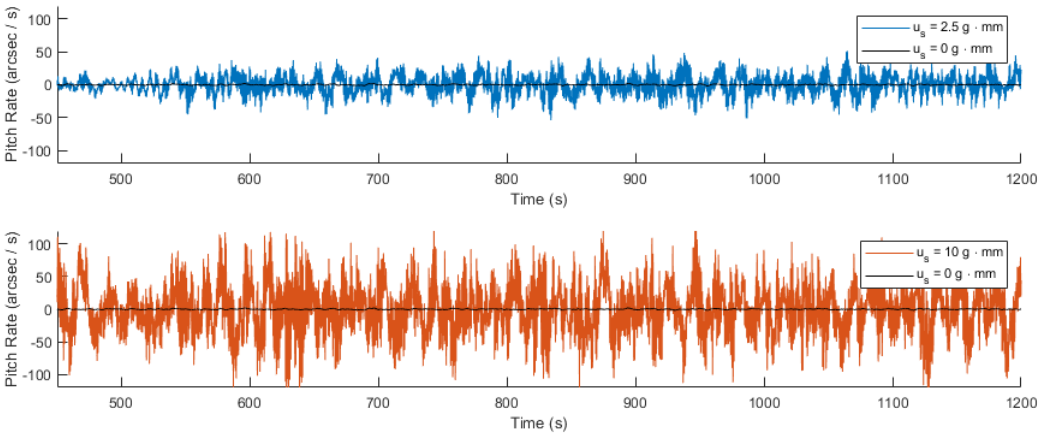


Figure 6.11: Alticube+ pitch rate for average jitter (blue line) and maximum jitter (orange line).

Figure 6.11 shows the pitch rate for the same three cases presented in Figure 6.10. The top figure shows how the pitch rate of the flexible model with an average static imbalance compares with the rigid body. For this case, it can be observed that the pitch rate is significantly increased, which is expected from the results in Figure 6.10. The mean pitch rate for this case is $14.1\,\frac{arcsec}{sec}$ RMS with a maximum of $54.0\,\frac{arcsec}{s}$. These values are still within requirement **ADCS-PC-REQ-4**. The case with the maximum static instability gives also increased pitch rate instability, resulting in a $37.8\,\frac{arcsec}{sec}$ RMS pitch rate with a maximum of $120.4\,\frac{arcsec}{s}$. Comparing the two flexible cases, increasing the static imbalance by four times causes the pitch instability to increase by approximately two times. Although the mean pitch rate still falls within requirement **ADCS-PC-REQ-4**, the maximum does exceed this requirement. Looking at the applied torque by the LQR controller, it can be said that jitter is not the sole contributor to the instability. The controller reacts to the reaction wheel jitter by compensating wit a counter torque. But as the jitter torque is oscillatory, the LQR couples with the jitter torque to increase the pitch instability.

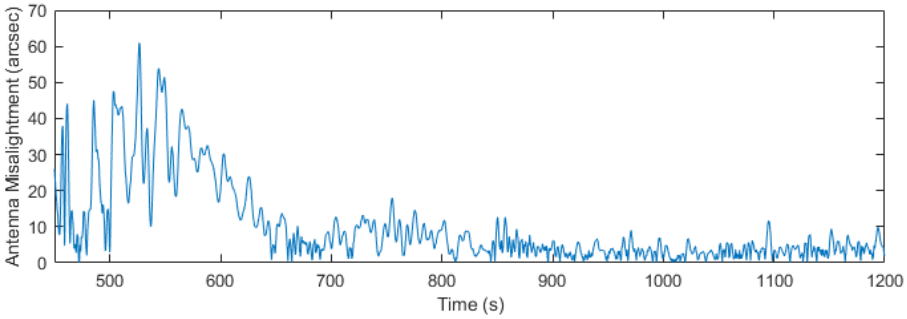


Figure 6.12: Antenna dynamic pointing misalignment during payload operation.

Figure 6.12 presents how jitter and the control torques together influence the alignment between two antennas that make up a baseline. The misalignment seen in the graph is a combination of the misalignments in the AT-ND plane and the CT-ND plane. From the graph, the antenna misalignment peaks at around $60\,arcsec$ and stabilises to around $10\,arcsec$. This misalignment falls below the requirements of $150\,arcsec$ in both planes (see requirements **ADCS-PC-REQ-2** and **ADCS-PC-REQ-3**).

Lastly, information on the baseline AKE is analysed to verify wether the Kalman filter is able to satisfy requirement **ADCS-PK-REQ-1**. Figure 6.13 shows the baseline AKE of Alticube+ between CubeSat node $\#1$ and $\#4$. From this graphs, the maximum AKE remains below $45\,arcsec$ and computing the

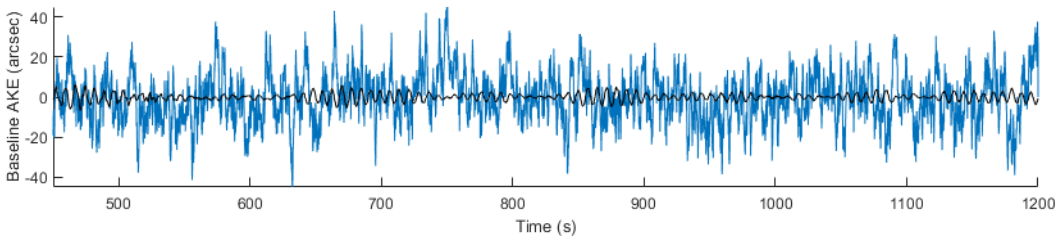


Figure 6.13: Baseline pointing knowledge error between node $\#1$ and $\#4$.

mean knowledge error over the measurement track is equal to $MKE = 12.7\,arcsec$, satisfying the requirement.

6.4.1. Worst-case: Jitter and control

The worst jitter case is a follow up from section 6.3, where jitter from multiple reaction wheels could amplify structural vibrations. Researching this explores the worst case possibility that Alticube+ can experience during payload operations. Here, all five reaction wheel systems are active and the maximum expected static imbalance is used. Then remaining setup is similar to section 6.3 with identical initial conditions compared to the full case manoeuvrer, enabling quantitive comparison between the two cases.

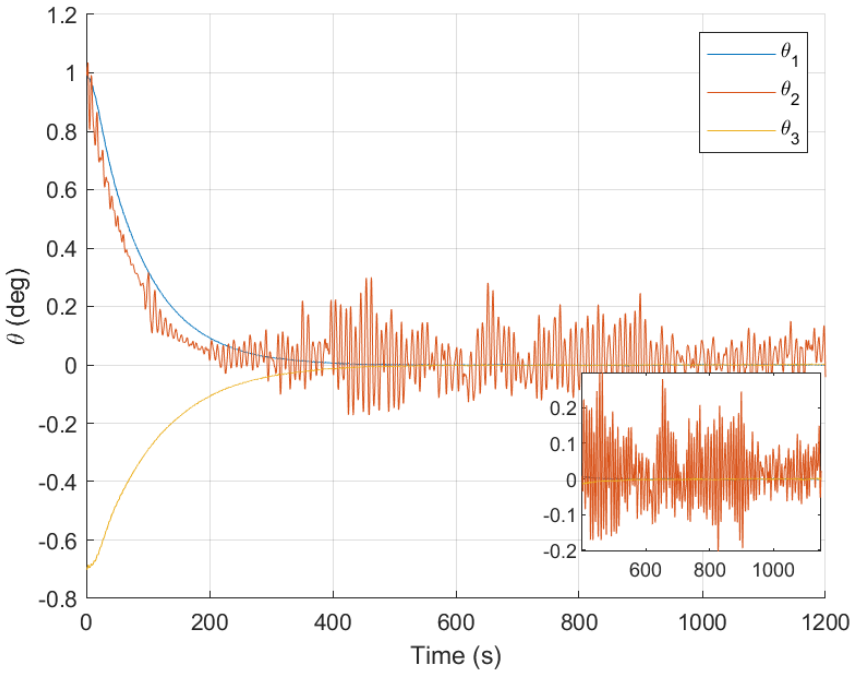


Figure 6.14: Alticube+ attitude states for worst-case jitter.

Figure 6.14 shows Alticube+'s attitude. It can be observed that the roll and yaw are not much affected by the increase in jitter caused by the five reaction wheel systems. however, pitch is majorly affected by this increase in jitter as the spacecraft is oscillating with peaks greater than $0.2\,deg$ from the reference attitude. This pitch instability does not satisfy multiple pointing control requirements on APE and PDE for the pitch direction. With the instability in pitch, an increase in antenna misalignment can be seen with low order oscillating peaks between $30\,arcsec$ and $60\,arcsec$. The baseline roll knowledge error has not increased, as the baseline oscillation amplitude does not increase with the increase in jitter across the five reaction wheel systems. Similar to the normal case, as the knowledge error is driven by the sensor accuracy and Kalman Filter effectiveness, the error on the baseline remains the same. Overall,

jitter does influence the pointing stability of the antenna's and will introduce relative pointing errors. The threat to Alticube+ lies in the pitch stability as jitter together with the LQR controller significantly influences the pitch stability.

6.5. Alticube+ Model Uncertainties Simulation

The last simulation done is regarding the model uncertainties, specifically, uncertainty in the inertia matrix. To simulate this effect, the internal controller assumed intertia matrix is changed according to the normal distribution $\mathcal{N}(\mu, \sigma^2)$. Here, the mean is taken around the true inertia used by the spacecraft model framework ($\mu_i = \boldsymbol{J}_{ii}$) where i is either x, y or z). The variance can be expressed as a standard deviation from the mean, which is taken to be $15\,\%$ for all axis ($\sigma_i = 0.15\mu_i$).

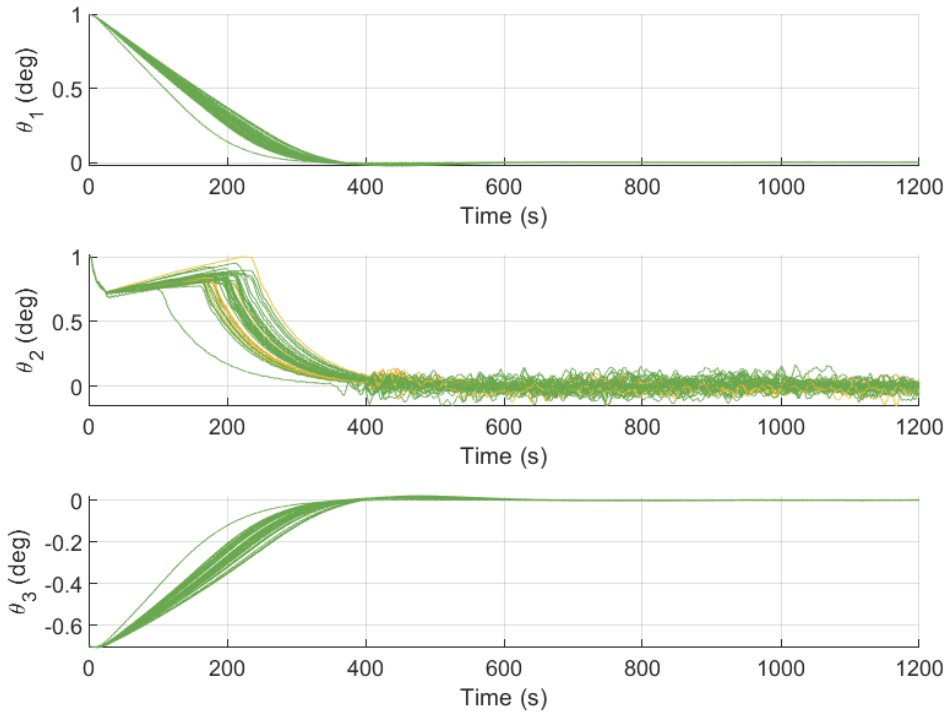


Figure 6.15: Alticube+ roll (θ_1), pitch (θ_2) ans yaw (θ_3) with uncertain inertia matrices for $N = 40$ simulations.

Figure 6.15 presents the attitude response of Alticube+ for the 40 Monte-Carlo simulations conducted with an uncertain inertia matrix. The results show that inertia uncertainty does influence the spacecraft attitude during the manoeuvre, but the magnitude and nature of this influence vary across the three axes.

For both roll and yaw, the attitude trajectories exhibit noticeable deviations when compared to the nominal case shown in Figure 6.5. However, these deviations remain small enough that the pointing control requirements are still satisfied across all simulations. The relatively modest impact on roll and yaw is primarily due to the fact that uncertainty in the mode masses—the primary contributors to roll and yaw inertia—dominates the variation in these axes. Since roll and yaw have comparatively large inertia moments, the spacecraft responds more slowly to external torques and dynamic disturbances. This slower response gives the Kalman filter sufficient time to correct prediction errors and maintain stable control performance.

In contrast, the pitch axis shows a broader spread in the attitude response and exhibits a higher level of dynamic instability. During the operational window, the pitch deviation reaches a maximum of approximately $0.15\,deg$, which is roughly $50\,\%$ larger than the pitch instability observed under maximum reaction-wheel jitter conditions. This increased deviation is driven by uncertainty in the central node

inertia, which is the dominant contributor to the pitch inertia of Alticube+. Because the pitch axis has a significantly smaller inertia moment than roll and yaw, the spacecraft responds much more quickly to reaction-wheel torques and jitter. As a result, even relatively small variations in the pitch inertia cause noticeable differences in the dynamics, leading to a wider spread in the pitch attitude.

Overall, the simulation results indicate that while inertia uncertainty introduces measurable deviations in all three axes, the system remains within the pointing control requirements. The pitch axis is the most sensitive to these uncertainties due to its low inertia moment, whereas roll and yaw remain comparatively robust due to their higher inertia and slower dynamic response.

7
Conclusion

The objective of this research is to design a centralised attitude controller for Alticube+ during its science operation phase and analyse the impact of reaction wheel jitter on pointing stability for a flexible structure. Guided by this objective, four supporting research questions are formulated to support this research objective. Conclusions based on the research and simulations results are presented here.

7.1. RQ-1: ADCS Requirements

The first research question - **What are the attitude determination and control system requirements that meet Alticube+'s 2024 concept proposal scientific mission objectives during the scientific operation phase?** - aims to create a foundation where subsequent research questions can build upon by identifying and deriving the ADCS requirements. Recent literature on the Alticube+ 2024 concept proposal has been consulted, from which scientific mission objectives, pointing control requirements and pointing knowledge requirements are derived. The key pointing control requirements derived are that Alticube+ requires a desired $APE \leq 0.1\,deg$ with a maximum of $APE \leq 0.2\,deg$ for all three axis. Further pointing control requirements on the maximum relative error between two antennas needs to be less than $150\,arcsec$ to prevent data loss during payload operation. Pointing knowledge requirements on the Alticube+'s attitude should be less than $100\,arcsec$ to enable calibration techniques for height measurement data extraction within the mission height error budget of $2\,cm$. Additional hardware requirements have been added, tailored to the scope of this research to ensure tractability and transparency. The key requirements limited the ADCS system to only use the central node sensors and reaction wheel system for pointing control and pointing knowledge.

7.2. RQ-2: Controller Design

The second research question - **How can a centralised LQR controller and Kalman filter be designed and tuned for the rigid-body model of Alticube+, and how does this impact their performance to satisfy the ADCS pointing control and knowledge requirements?** - aimed to design a controller for Alticube+. A centralised LQR controller combined with a Kalman filter was designed for the rigid-body spacecraft model to address the second research question. The LQR formulation provides a systematic multi-axis control law capable of achieving stable attitude control. Trade-offs between pointing performance and actuator effort is tuned through the cost matrices. After Formulation of the LQR controller, an integral cost matrix tuning is performed. This research uses Bode plots to analyse Controller stability over a wide spectrum of tuning parameter configurations. Analysis shows that values for $\boldsymbol{Q} < 1$ result in a stable control system with positive margins. To determine a cost configuration that ensures a stable convergence to the reference attitude within the $450\,s$ manoeuvrer window, a deeper refinement on the range of possible cost configurations is needed. Separating the cost matrix $\boldsymbol{Q}$ into attitude and angular rate cost weighting parameters shows that lowering weights of the attitude cost weights leaves no margin for the control system to work with. It is found that the weighting values of $10^{-1.8} - 0.15$ for the attitude cost weights provides a phase margins between $30\,deg - 60\,deg$. Pitch however shows marginal stability which can results in oscillatory behaviour of the system in the

pitch axis. Lowering the cost weight for pitch to a value between $10^{-3} - 10^{-2}$ results in a stable pitch control with sufficient margin. The tuning configuration was then validated using the rigid body model in a simulation environment, where it showed that the LQR control system together with the Kalman filter is able to converge to the desired attitude within the required $450\,s$ operation window for attitude manoeuvrers of $2\,deg$ in any axis. However, analysing the control loop shows that the reaction wheels quickly saturate for manoeuvrers larger than $1\,deg$ in both roll and yaw simultaneously. Although the spacecraft satisfies pointing control requirements below $0.2\,deg$ for combined roll and yaw manoeuvrers of $1.5\,deg$, larger initial offsets result in a delayed convergence to the reference attitude, not complying with ADCS pointing control requirement of $0.2\,deg$ within the manoeuvrer time window. In addition to this, sudden impulse disturbance torques with relative large magnitude could threaten the stability of Alticube+. Preventing this issue would require either an increase in angular momentum storage or possibly a form of distributed control.

7.3. RQ-3: Reaction Wheel Jitter Interaction

With the design of the attitude controller finished, the third research question - **How do the dominant flexible modes of Alticube+ interact with reaction wheel jitter to affect pointing stability?** - can be addressed by expanding the rigid body model to a flexible model with reaction wheel jitter. The flexible model is developed using Kane's equations method for describing dynamics. Implementing Kane's equations allows for a customized model together with the lumped parameter method to describe five rigid body CubeSats, each linked through a torsional joint, configured to model Alticube+'s flexible baseline structure. The jitter of the reaction wheel system is modelled using the static imbalance of the reaction wheel, which generates a jitter force that rotates around the spinning axis in the local reaction wheel reference frame. The interaction between the reaction wheel jitter and the flexible structure is analysed through varying initial conditions and changing static imbalance values within the expected range. Simulation results show that the presence of flexible joints introduces dynamic lag in attitude tracking, particularly visible in roll and yaw, and small oscillatory behaviour. Despite this, the LQR controller maintains stable convergence within the required $450\,s$ operational window. However, the analysis also reveals the sensitivity of the flexible structure to reaction wheel jitter. Pitch stability is most affected by the reaction wheel jitter. Pointing errors up to $0.11\,deg$ in the pitch direction and angular rates up to $120\,arcsec/s$ are observed as static imbalance increases. In addition to this, a worst-case jitter simulation where all five reaction wheel systems produced reaction wheel jitter is explored to understand the pointing control under maximum jitter possible by Alticube+. This simulation showed pitch pointing errors of $0.3\,deg$, going above the maximum pointing error of $0.2\,deg$.
Internal antenna pointing misalignment due to reaction wheel jitter stays within the requirements throughout the payload operation window. Pointing knowledge on the baseline roll is mainly driven by the sensor noise and the Kalman filter's performance to estimate the spacecraft attitude. Results show that the knowledge error also is within the requirement.

7.4. RQ-4: Pointing Performance under Inertia Uncertainty

Lastly, the fourth research question - **What is the achievable envelope of pointing stability performance for Alticube+ under inertia uncertainties?** - explores how uncertainty in the known inertia matrix affects the pointing performance in combination with reaction wheel jitter. Through variations in the inertia matrix, the simulations revealed axis-dependent sensitivity. Roll and yaw matrix uncertainty is mainly driven by the uncertainty in the mass of each CubeSat platform, due to the long baseline structure. Root cause for uncertainty in mass can originate from the expenditure or RCS propellant. If CubeSats expend different amounts of propellant during the rendezvous and docking procedure, imbalance in mass between the CubeSat nodes can be observed which deviate from the predicted mass. Effects of the this uncertainty remained limited due to their large inertia moments, allowing the Kalman filter adequate time to correct state prediction errors. In contrast, pitch deviations are governed primarily by the inertia uncertainty of CubeSat platforms. This uncertainty has an effect on the pitch error which reached magnitudes up to $0.15\,deg$, approximately $50\,\%$ greater than the instability observed under maximum static imbalance conditions. This underlines the amplified impact of uncertainties on pitch stability.

7.5. Overall Conclusions

In conclusion, this research successfully achieved its objective of designing a centralised attitude controller for Alticube+ using LQR during its science operation phase and assessing the impact of reaction wheel jitter on pointing stability for a flexible structure. By systematically addressing the four supporting research questions, a complete and traceable analysis path was established—from defining ADCS requirements, to designing and tuning a centralised LQR-KF architecture, to extending the dynamics to a flexible multi-body system, and finally evaluating pointing stability under mass/inertia uncertainties. The results show that a centralised LQR controller, supported by a Kalman filter for state estimation, is capable of regulating the attitude of Alticube+ within the operational constraints and achieving convergence within the required manoeuvre window for both rigid and flexible configurations. In addition to this, antenna pointing misalignment and baseline knowledge errors caused by reaction wheel jitter are within the allowable range, enabling Alticube+ to achieve its scientific mission objectives. However, the research also highlights several important limitations. Firstly, the reaction wheel saturation fundamentally constrains the range of feasible initial conditions where Alticube has full control over all axis during this phase. Future research mitigating these issues will require a trade-off between increasing angular momentum storage for the central node, increasing allowed time for the manoeuvrer, limiting manoeuvrer range or addressing the possibility for distributed control. Secondly, reaction wheel jitter from static imbalances interacts proportionally the with flexible modes, mainly affecting pitch stability. Future research can investigate more advanced controller strategies that limits jitter-controller resonance with the flexible modes. Lastly, mass and inertia uncertainties further amplify pitch pointing errors, emphasising the pitch sensitivity of the spacecraft. Future research expanding the inertia matrix fidelity could allow for better assessment of the inertia matrix, creating more accurate understanding of the pointing control impact.

8

Recommendations

In Chapter 7, conclusions on the research objective and the supporting research question have been presented. From these conclusions, several recommendations can be highlighted for future research. The recommendations are divided into two parts. The first part discusses improvements on the modelling approach. The second part expands on the controller design for Alticube+ and proposes further recommendations on possible control allocation problem for long-baseline spacecraft.

8.1. Model-Related Recommendations

- **Refine structural flexibility modelling.** Although the flexible spacecraft model presented in this thesis captures the primary dynamic coupling between the central node and connected nodes, first order mode shapes resulted in an overestimation of the tip pointing error, compared to a high order FEM model. Future work should consider including higher-order mode shapes, comparing the impact on pointing errors against the findings of this research.
- **Improve CubeSat inertia matrix.** This research captured the largest dynamics of the spacecraft using a simplified inerti matrix model of Alticube+ in the stowed configurations. But in reality, its deployed configurations alters the inertia matrices through shifts in mass distribution, impacting dynamics. Improving the inertia matrix allows for a more detailed understanding on Alticube+'s dynamics, leading to better controller designs.
- **Expand on boom joint flexibility and thermal effects.** Additional refinement to flexibility of the structure, specifically for multi-linked CubeSat structures like Alticube+, includes boom joint flexibility and thermal effects. These effects could have static and dynamics impact on the baseline pointing of the spacecraft. This research has not included these effects, which could lead into not fully capturing the antenna pointing error during the payload operations. Including these effects will create better understanding on pointing errors for general long-baseline structures.
- **Integrate environmental disturbance models.** While disturbances such as gravity-gradient torque, aerodynamic drag, and solar radiation pressure were considered at a high level, mission-specific disturbance characterisation may improve accuracy, particularly for low-altitude missions.
- **Addition of reaction wheel system structural transfer function.** To research the effects of reaction wheel jitter on the flexible structure, a reaction wheel model that included static imbalance was created. This research, however, only looked at direct pure jitter torques to the center of mass of the central node. In reality, structural interfaces with the reaction wheel to the spacecraft introduce transfer functions, which can resonate or dampen certain frequencies. The addition of a structural transfer function between the jitter torques and the experienced vibrations by the central node could further analyse the effects of reaction wheel jitter on Alticube+ pointing performance.

8.2. Controller Design Recommendations

- **Investigate LQR improvements for pitch-axis sensitivity.** The results show that pitch control performance is more sensitive to inertia uncertainty and reaction wheel jitter due to the axis' s smaller inertia moment. Future designs should consider adjusting control gains, implementing adaptive filtering or add additional control constraints resonance with the reaction wheel jitter.

- **Analyse operational constraints related to wheel jitter.** Since reaction wheel jitter plays a significant role in pointing stability, especially during fine-pointing operations, it is recommended to examine potential wheel desaturation and scheduling strategies to optimally avoid jitter.
- **Investigation of distributed control methods.** Although LQR is able to control Alticube+ within the requirements, pitch axis sensitivity to reaction wheel jitter does lead to small margins in spacecraft pointing control. Distributed control could utilise multiple sets of reaction wheels in the five CubeSat nodes. Researching and investigating these control methods could allow for a more effective utilisation of control actuators and reduce the pitch sensitivity.
- **Investigation of post-sensor data state estimation** In this research, the central node was used to determine the spacecraft's state. For in-orbit pointing control of Alticube+, using a Kalman filter is sufficient for this task. However, four sets of sensors are not utilised. Future research on post-sensor data estimation strategies is of interest to further understand the pointing knowledge capabilities of Alticube+.

References

[1] Margaret Srinivasan and Vardis Tsontos. *Satellite Altimetry for Ocean and Coastal Applications: A Review*. Aug. 2023. DOI: 10.3390/rs15163939.

[2] Jian Guo et al. *AltiCube+: A Low-Cost Long Fixed-Baseline Radar Altimeter Solution Based On CubeSats On-Orbit Assembly*. Tech. rep. 2024, pp. 14–18.

[3] R Fjørtoft et al. *KARIN-THE KA-BAND RADAR INTERFEROMETER ON SWOT: MEASUREMENT PRINCIPLE, PROCESSING AND DATA SPECIFICITIES*. Tech. rep. 2010.

[4] Sasha V Weston et al. *State-of-the-Art Small Spacecraft Technology*. Tech. rep. Feb. 2025. URL: http://www.sti.nasa.gov.

[5] Eva Peral et al. "RainCube: the first ever radar measurements from a CubeSat in space". In: *Journal of Applied Remote Sensing* 13 (3 2019), p. 32504.

[6] Yuanhao Li et al. "CubeSat Altimeter Constellation Systems: Performance Analysis and Methodology". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–19. DOI: 10.1109/TGRS.2021.3100850.

[7] Jian Guo et al. *AltiCube+-An Aggregated CubeSats Swarm for Long Fixed-baseline Radar Altimetry Final report Activity Type: Study*. Tech. rep. Apr. 2024. URL: www.esa.int/discovery.

[8] Alberto Moreira. *Synthetic Aperture Radar (SAR): Principles and Applications Remote Sensing*. Tech. rep. German Aerospace Centre (DRL), July 2013. URL: www.dlr.de/HR.

[9] Roger Fjørtoft et al. "KaRIn-the Ka-band radar interferometer on SWOT: Measurement principle, processing and data specificities". In: *2010 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2010, pp. 4823–4826.

[10] Angelica Rodriguez. *SWOT Surface Water and Ocean Topography*. 2025. URL: https://swot.jpl.nasa.gov/.

[11] Gérald Dibarboure et al. "Data-driven calibration algorithm and pre-launch performance simulations for the SWOT mission". In: *Remote Sensing* 14.23 (2022), p. 6070.

[12] L. R. Wyatt. "Ocean Wave Directional Spectrum". In: *Oceanography* 10.2 (1997), pp. 85–89.

[13] Sodern. 2025. URL: https://sodern.com/wp-content/uploads/2025/02/AURIGA_CP.pdf.

[14] Lens R&D B.V. *Maus*. URL: https://lens-rnd.com/products/maus.

[15] Safran. Apr. 2025. URL: https://safran-navigation-timing.com/product/stim277h-mems-gyroscope/.

[16] ISISpace. Mar. 2025. URL: https://www.isispace.nl/product/isis-magnetorquer-board/.

[17] Nadia A Saad and MN Ismail. "Influence of geomagnetic field on the motion of low satellites". In: *Astrophysics and Space Science* 325 (2010), pp. 177–184.

[18] Astrofein. Apr. 2023. URL: https://www.astrofein.com/reaktionsraeder/rw-25/.

[19] Kai Cao et al. *Dynamic Modeling and Distributed Control of Asymmetric Flexible Spacecraft*. Tech. rep. URL: https://www.researchgate.net/publication/363565178.

[20] G ; Song et al. *Downloaded from NPS Archive*. Tech. rep. 2001, pp. 73–83. URL: www.elsevier.com/locate/actaastro.

[21] Qiufan Yuan, Yanfang Liu, and Naiming Qi. "Active vibration suppression for maneuvering spacecraft with high flexible appendages". In: *Acta Astronautica* 139 (Oct. 2017), pp. 512–520. ISSN: 00945765. DOI: 10.1016/j.actaastro.2017.07.036.

[22] Arghya Chakravarty and Chitralekha Mahanta. "Compensating actuator failures in near space vehicles using adaptive finite time disturbance observer based backstepping controller". In: *2016 European Control Conference (ECC)*. IEEE. 2016, pp. 98–103.

[23] Shunan Wu et al. “Multi-objective integrated robust H∞ control for attitude tracking of a flexible spacecraft”. In: *Acta Astronautica* 151 (2018), pp. 80–87.

[24] Hyochoong Bang, Cheol-Keun Ha, and Jin Hyoung Kim. “Flexible spacecraft attitude maneuver by application of sliding mode control”. In: *Acta Astronautica* 57.11 (2005), pp. 841–850.

[25] Tor A Johansen and Thor I Fossen. “Control allocation—A survey”. In: *Automatica* 49.5 (2013), pp. 1087–1103.

[26] Haitao Chang et al. “Distributed control allocation for spacecraft attitude takeover control via cellular space robot”. In: *Journal of Guidance, Control, and Dynamics* 41.11 (2018), pp. 2499–2506.

[27] Weixing Liu et al. “Distributed constrained control allocation for cellularized spacecraft attitude control system”. In: *Journal of Guidance, Control, and Dynamics* 45.2 (2022), pp. 385–393.

[28] J Russell Carpenter. “Decentralized control of satellite formations”. In: *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* 12.2-3 (2002), pp. 141–161.

[29] J Russell Carpenter. “A preliminary investigation of decentralized control far satellite formations”. In: *2000 IEEE Aerospace Conference. Proceedings (Cat. No. 00TH8484)*. Vol. 7. IEEE. 2000, pp. 63–74.

[30] Jiakang Zhou, Qinglei Hu, and Michael I Friswell. “Decentralized finite time attitude synchronization control of satellite formation flying”. In: *Journal of Guidance, Control, and Dynamics* 36.1 (2013), pp. 185–195.

[31] Baolin Wu, Danwei Wang, and Eng Kee Poh. “Decentralized control for satellite formation using local relative measurements only”. In: *IEEE ICCA 2010*. IEEE. 2010, pp. 661–666.

[32] Chen Bifa et al. “A multi-point decentralized control for mitigating vibration of flexible space structures using reaction wheel actuators”. In: *Scientific Reports* 14.1 (2024), p. 10326.

[33] Zhaohui Wang et al. “Vibration suppression-based attitude control for flexible spacecraft”. In: *Aerospace Science and Technology* 70 (2017), pp. 487–496. ISSN: 1270-9638. DOI: https://doi.org/10.1016/j.ast.2017.08.014. URL: https://www.sciencedirect.com/science/article/pii/S127096381730038X.

[34] Ran Zhang, Tao Li, and Lei Guo. “Disturbance observer based H∞ control for flexible spacecraft with time-varying input delay”. In: *Advances in Difference Equations* 2013.1 (2013), p. 142.

[35] Jalal Eddine Benmansour, Rima Roubache, and Akram Adnane. “LQR Control of a Flexible Satellite with Movable Mass Actuation”. In: *Algerian Journal of Engineering and Technology* 10.1 (2025), pp. 48–59.

[36] Guiqin He and Dengqing Cao. “Dynamic modeling and attitude–vibration cooperative control for a large-scale flexible spacecraft”. In: *Actuators*. Vol. 12. 4. MDPI. 2023, p. 167.

[37] B Wie. “Space Vehicle Dynamics and Control, AIAA Education Series, AIAA”. In: *Inc., Reston* (1998), pp. 233–240.

[38] James R. Wertz, ed. *Spacecraft Attitude Determination and Control*. Dordrecht: Springer, 1978.

[39] Arthur Earl Bryson. *Applied optimal control: optimization, estimation and control*. Routledge, 2018.

[40] John L. Crassidis and John L. Junkins. *Optimal Estimation of Dynamic Systems*. Boca Raton, FL: CRC Press, 2011.

[41] Leonard Meirovitch. *Dynamics and Control of Structures*. New York: Wiley, 1990.

[42] Yong-chun Xie et al. “Applications of advanced control methods in spacecrafts: progress, challenges, and future prospects”. In: *Frontiers of Information Technology & Electronic Engineering* 17.9 (2016), pp. 841–861.

[43] Alberto Bemporad and Manfred Morari. “Control of systems integrating logic, dynamics, and constraints”. In: *Automatica* 35.3 (1999), pp. 407–427.

[44] Andrew Richards and Jonathan P. How. “Robust constrained model predictive control”. In: *IEEE Transactions on Automatic Control* 50.6 (2005), pp. 850–862.

[45] Marco Lovera. “Model predictive control for spacecraft attitude control”. In: *Aerospace Science and Technology* 46 (2015), pp. 176–190.

[46] Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. Chichester: Wiley, 2005.

[47] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice Hall, 1991.

[48] Mykel J. Kochenderfer and Tim A. Wheeler. *Algorithms for Decision Making*. Cambridge, MA: MIT Press, 2022.

[49] Mei Liu et al. “Dynamic modeling and vibration control of a large flexible space truss”. In: *Meccanica* 57.5 (2022), pp. 1017–1033.

[50] ESA Requirements and Standards Division. “Space engineering control performance handbook”. In: *European Cooperation for Space Standardization*. 2008.

[51] T Ott et al. “ESA pointing error engineering handbook”. In: *8th International ESA Conference on Guidance, Navigation & Control Systems*. Vol. 17. 2011.

[52] Michael Paluszek. “Chapter 8 - Disturbances”. In: *ADCS - Spacecraft Attitude Determination and Control*. Ed. by Michael Paluszek. Elsevier, 2023, pp. 117–145. ISBN: 978-0-323-99915-1. DOI: https://doi.org/10.1016/B978-0-32-399915-1.00019-X. URL: https://www.sciencedirect.com/science/article/pii/B978032399915100019X.

[53] *ATMOSPHERE PROPERTIES*. URL: http://www.braeunig.us/space/atmos.htm.

[54] James Armstrong et al. “Pointing control for low altitude triple CubeSat space darts”. In: (2009).

[55] Anton H de Ruiter, Christopher Damaren, and James R Forbes. *Spacecraft Dynamics and Control: An Introduction*. Wiley, 2013.

[56] Eric Stoneking. “Implementation of Kane’s method for a spacecraft composed of multiple rigid bodies”. In: *AIAA Guidance, Navigation, and Control (GNC) Conference*. 2013, p. 4649.

[57] Takaya Inamori et al. “Jitter reduction of a reaction wheel by management of angular momentum using magnetic torquers in nano-and micro-satellites”. In: *Advances in Space Research* 52.1 (2013), pp. 222–231.

[58] *Reaction Wheel Data Sheet*. 3.1. Rocket Lab. Mar. 2025.

[59] E. Van Kampen. *Spacecraft Attitude Control and Dynamics*. Faculty of Aerospace Engineering, Technical University of Delft. [PowerPoint slides]. 2024.

A

Mathamatical Derivations of Quaternion Kinematics Inversion

Let the following relationship of the quaternion kinematics be described by equation (A2.1)

$$\begin{bmatrix} \dot{\bar{q}} \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_4 I_{3\times 3} + \bar{q}_\times \\ -\bar{q}^T \end{bmatrix} \bar{\omega} = \mathbf{\Lambda}_q \bar{\omega} \tag{A2.1}$$

The inverse of this equation will result in

$$\bar{\omega} = (\mathbf{\Lambda}_q^T \mathbf{\Lambda}_q)^{-1} \mathbf{\Lambda}_q^T \begin{bmatrix} \dot{\bar{q}} \\ \dot{q}_4 \end{bmatrix} \tag{A2.2}$$

Because all quaternions are unit vectors:

$$(\mathbf{\Lambda}_q^T \mathbf{\Lambda}_q)^{-1} = 4 I_{3\times 3}$$

This will then result in the following relationship seen in equation (A2.3).

$$\bar{\omega} = 2 \begin{bmatrix} q_4 I_{3\times 3} + \bar{q}_\times \\ -\bar{q}^T \end{bmatrix}^T \begin{bmatrix} \dot{\bar{q}} \\ \dot{q}_4 \end{bmatrix} \tag{A2.3}$$

The following identities can be used:

$$\mathbf{\Lambda}_q = \frac{1}{2} \begin{bmatrix} q_4 I_{3\times 3} + \bar{q}_\times \\ -\bar{q}^T \end{bmatrix}, \qquad (\mathbf{\Lambda}_q^T \mathbf{\Lambda}_q)^{-1} \mathbf{\Lambda}_q^T = 2 \begin{bmatrix} q_4 I_{3\times 3} + \bar{q}_\times \\ -\bar{q}^T \end{bmatrix}^T$$

$$\begin{bmatrix} q_4 I_{3\times 3} + \bar{q}_\times \\ -\bar{q}^T \end{bmatrix} = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}$$

$$\begin{bmatrix} q_4 I_{3\times 3} + \bar{q}_\times \\ -\bar{q}^T \end{bmatrix}^T = \begin{bmatrix} q_4 & q_3 & -q_2 \\ -q_3 & q_4 & q_1 \\ q_2 & -q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}$$

Now the inversion of equation (A2.3) can be completed.

$$\bar{\omega} = \mathbf{\Lambda}_q^* \begin{bmatrix} \dot{\bar{q}} \\ \dot{q}_4 \end{bmatrix} - n \begin{bmatrix} 2(q_1 q_2 + q_3 q_4) \\ 1 - 2(q_1^2 + q_3^2) \\ 2(q_3 q_2 - q_1 q_4) \end{bmatrix} \tag{A2.4}$$

$$\mathbf{\Lambda}_q^* = 2 \begin{bmatrix} q_4 I_{3\times 3} + \bar{q}_\times \\ -\bar{q}^T \end{bmatrix}^T = (\mathbf{\Lambda}_q^T \mathbf{\Lambda}_q)^{-1} \mathbf{\Lambda}_q^T$$

$$\Rightarrow \bar{\omega} + n \begin{bmatrix} 2(q_1 q_2 + q_3 q_4) \\ 1 - 2(q_1^2 + q_3^2) \\ 2(q_3 q_2 - q_1 q_4) \end{bmatrix} = \mathbf{\Lambda}_q^* \begin{bmatrix} \dot{\bar{q}} \\ \dot{q}_4 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \dot{\bar{q}} \\ \dot{q}_4 \end{bmatrix} = (\mathbf{\Lambda}_q^{*T} \mathbf{\Lambda}_q^*)^{-1} \mathbf{\Lambda}_q^* \left(\bar{\omega} + n \begin{bmatrix} 2(q_1 q_2 + q_3 q_4) \\ 1 - 2(q_1^2 + q_3^2) \\ 2(q_3 q_2 - q_1 q_4) \end{bmatrix} \right)$$

From the identity of $\mathbf{\Lambda}_q^*$ can be seen that $(\mathbf{\Lambda}_q^{*T} \mathbf{\Lambda}_q^*)^{-1} \mathbf{\Lambda}_q^* = \mathbf{\Lambda}_q$. Further, the relationship can be fully derived.

$$\begin{bmatrix} \dot{\bar{q}} \\ \dot{q}_4 \end{bmatrix} = \mathbf{\Lambda}_q \bar{\omega} + n \mathbf{\Lambda}_q \begin{bmatrix} 2(q_1 q_2 + q_3 q_4) \\ 1 - 2(q_1^2 + q_3^2) \\ 2(q_3 q_2 - q_1 q_4) \end{bmatrix} \tag{A2.5}$$

$$\begin{bmatrix} \dot{\bar{q}} \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_4 I_{3\times 3} + \bar{q}_\times \\ -\bar{q}^T \end{bmatrix} \bar{\omega} + n \begin{bmatrix} q_4(q_1 q_2 + q_3 q_4) - q_3 \left(\frac{1}{2} - (q_1^2 + q_3^2)\right) + q_2(q_3 q_2 - q_1 q_4) \\ q_3(q_1 q_2 + q_3 q_4) + q_4 \left(\frac{1}{2} - (q_1^2 + q_3^2)\right) - q_1(q_3 q_2 - q_1 q_4) \\ -q_2(q_1 q_2 + q_3 q_4) + q_1 \left(\frac{1}{2} - (q_1^2 + q_3^2)\right) + q_4(q_3 q_2 - q_1 q_4) \\ -q_1(q_1 q_2 + q_3 q_4) - q_2 \left(\frac{1}{2} - (q_1^2 + q_3^2)\right) - q_3(q_3 q_2 - q_1 q_4) \end{bmatrix} \tag{A2.6}$$

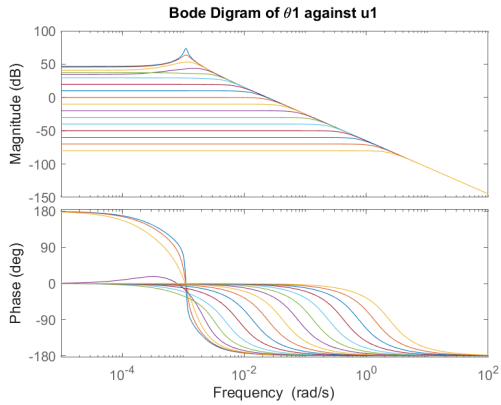
Using the relationship $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$, the vector can be reduced further.

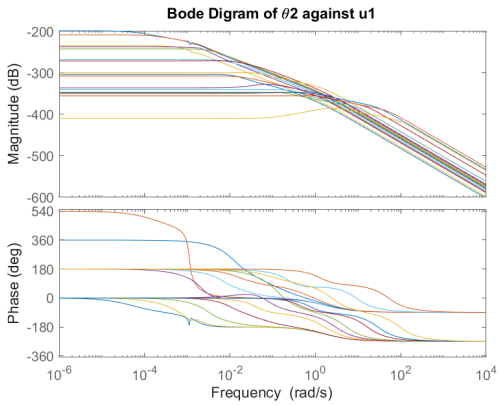
$$\begin{bmatrix} \dot{\bar{q}} \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \mathbf{\Lambda}_q \bar{\omega} + \frac{n}{2} \begin{bmatrix} 0_{2\times 2} & I_{2\times 2} \\ I_{2\times 2} & 0_{2\times 2} \end{bmatrix} \begin{bmatrix} \bar{q} \\ q_4 \end{bmatrix} = \frac{1}{2} \mathbf{\Lambda}_q \bar{\omega} + \boldsymbol{N}_q \begin{bmatrix} \bar{q} \\ q_4 \end{bmatrix} \tag{A2.7}$$

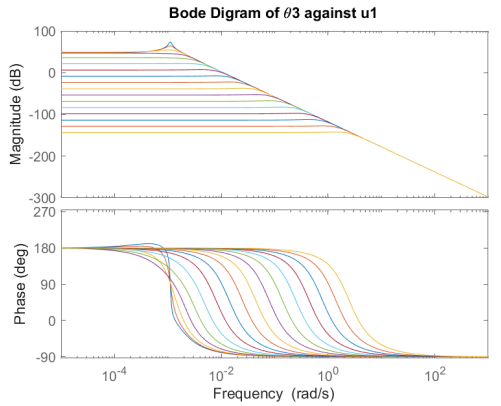
B

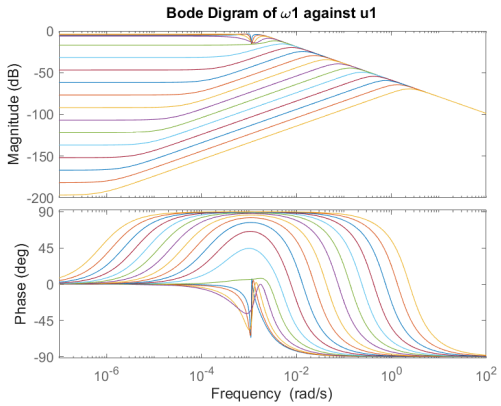
Bode diagrams

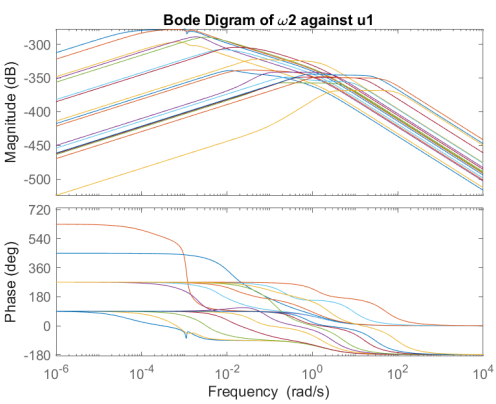
B.1. Bode Plots Tuning Q_q

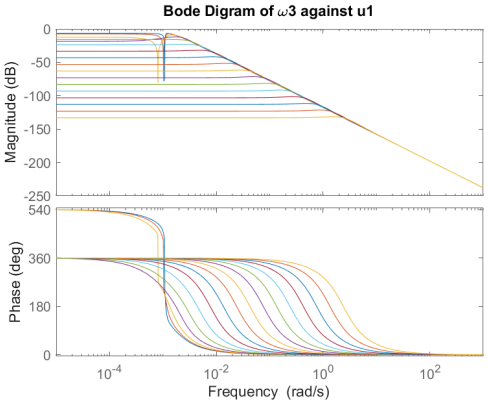


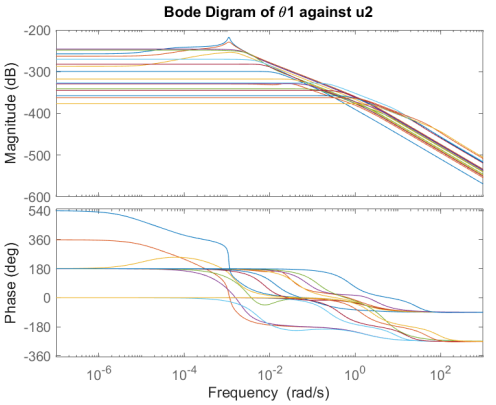


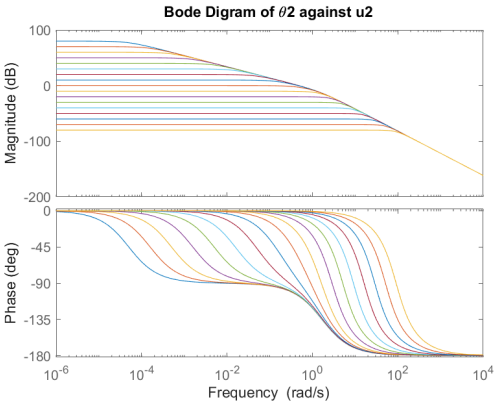


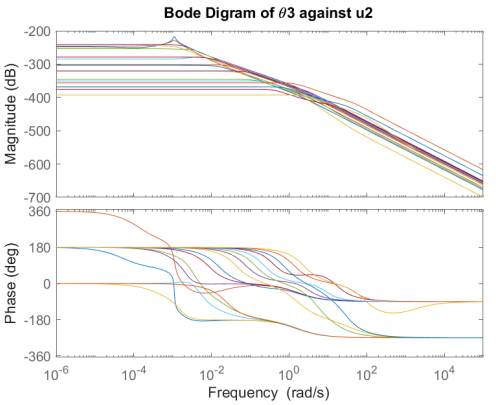


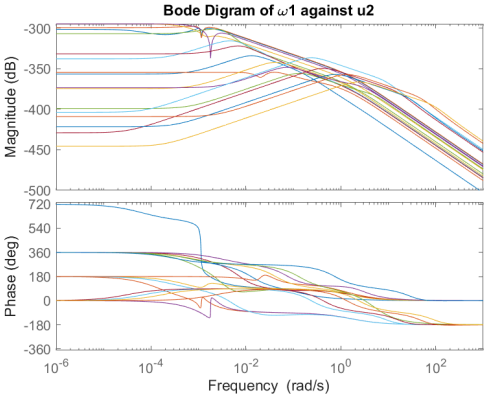

Bode Digram of ω2 against u1
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

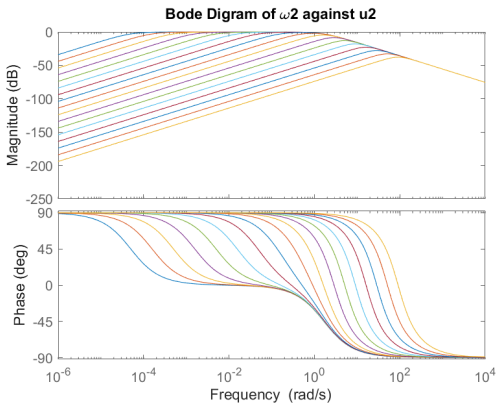

Bode Digram of ω3 against u1
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

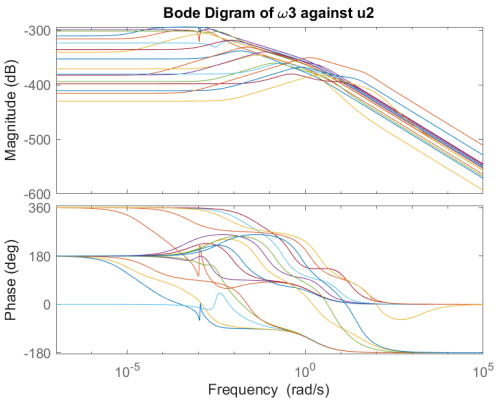

Bode Digram of θ1 against u2
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

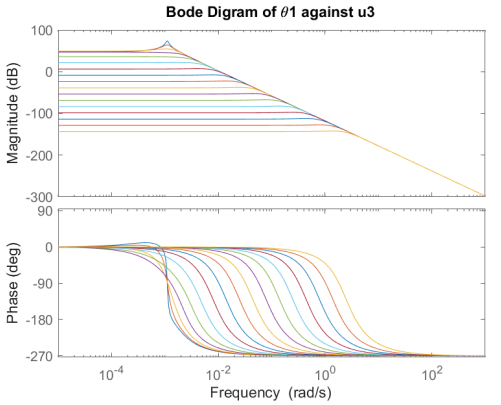

Bode Digram of θ2 against u2
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

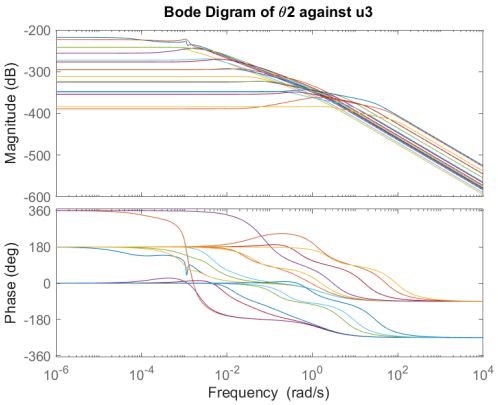

Bode Digram of θ3 against u2
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

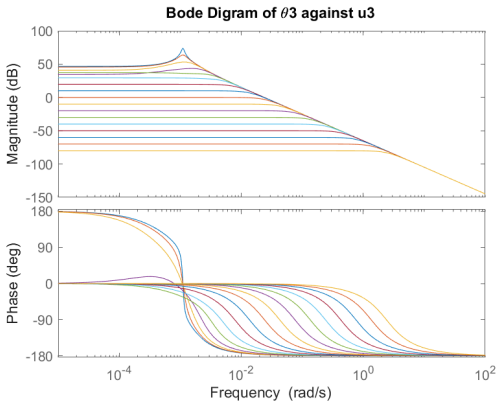

Bode Digram of ω1 against u2
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

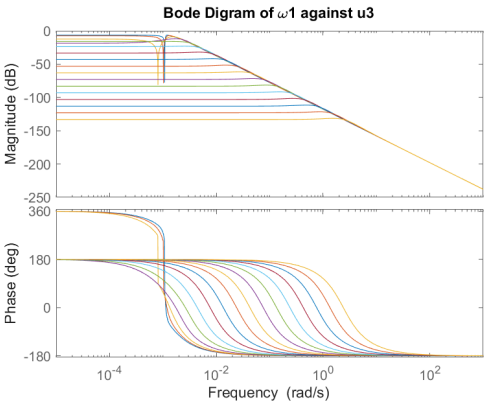

Bode Digram of ω2 against u2
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

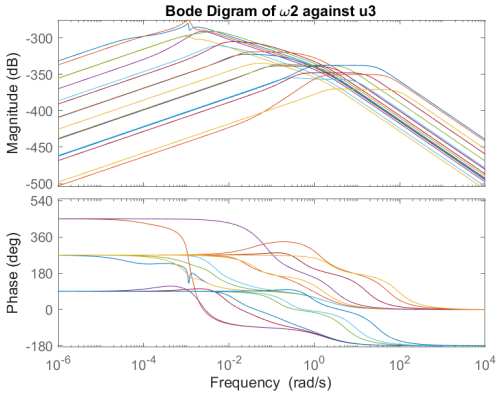

Bode Digram of ω3 against u2
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

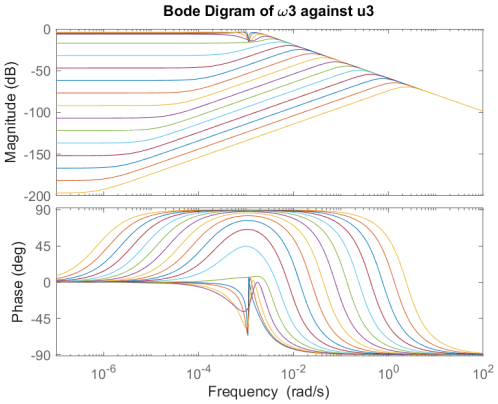

Bode Digram of θ1 against u3
Magnitude (dB)
Phase (deg)
Frequency (rad/s)


Bode Digram of θ2 against u3
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

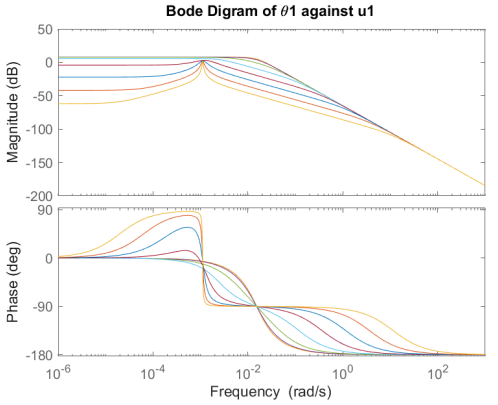

Bode Digram of θ3 against u3
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

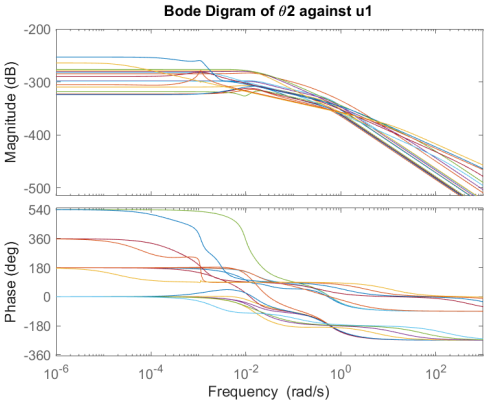

Bode Digram of ω1 against u3
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

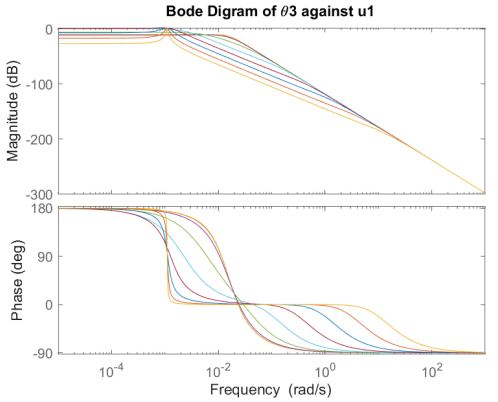


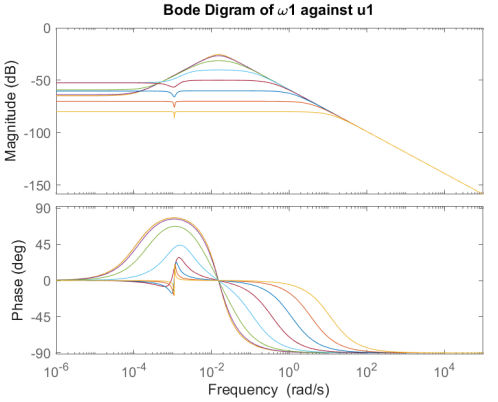


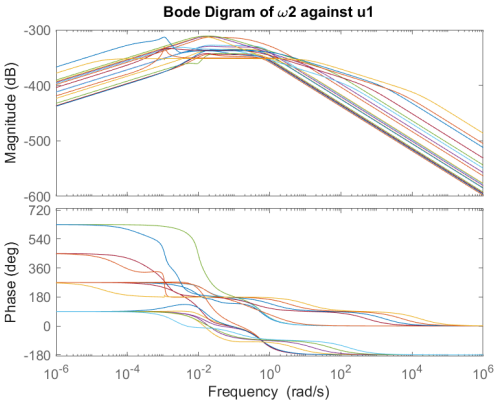
B.2. Bode Plots Tuning Q_ω

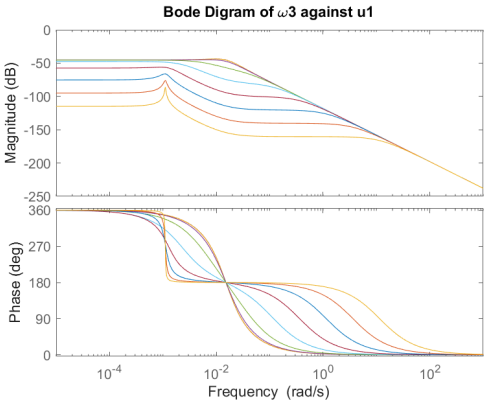


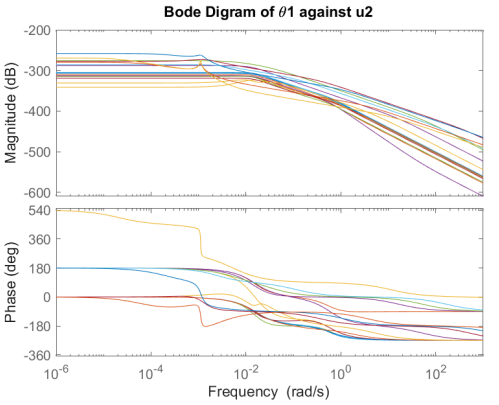


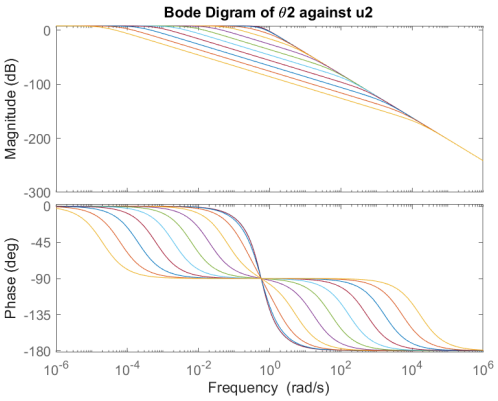


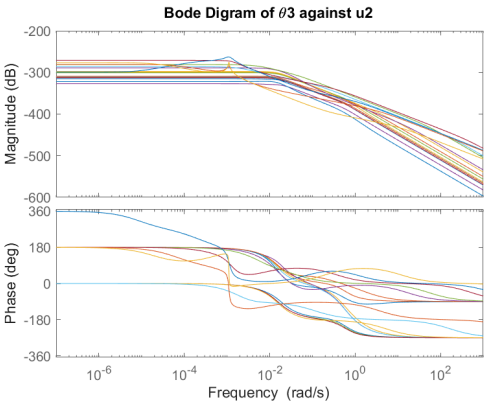


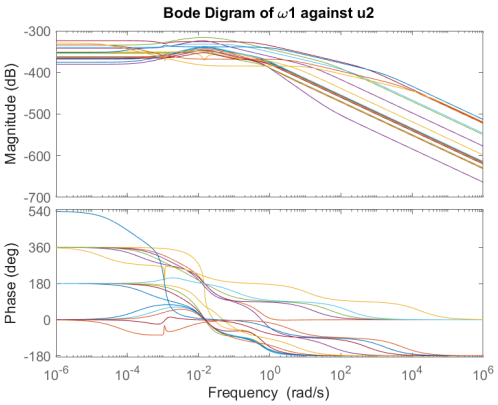

Bode Digram of ω2 against u1
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

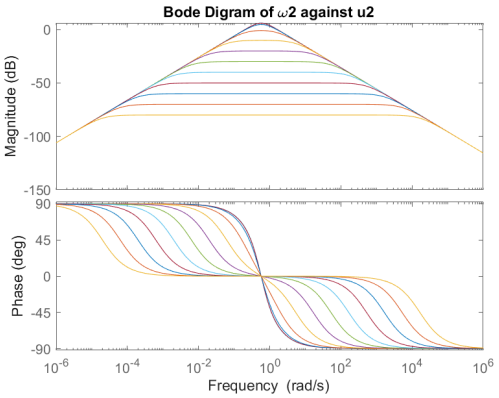

Bode Digram of ω3 against u1
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

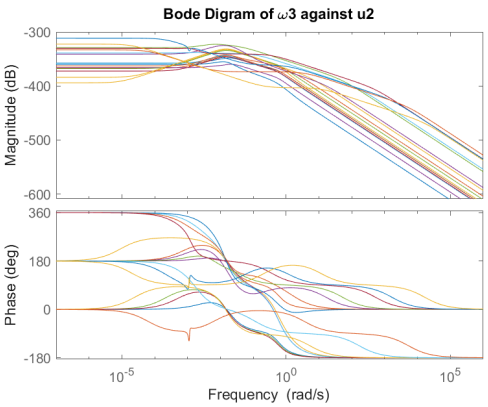

Bode Digram of θ1 against u2
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

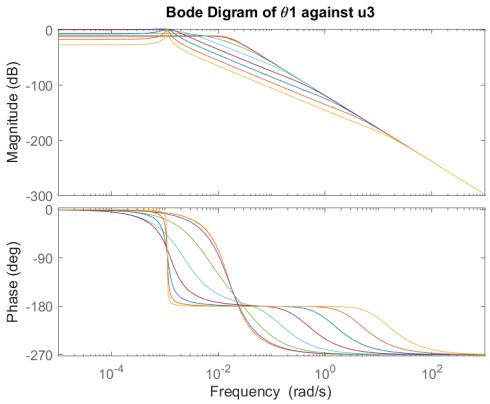

Bode Digram of θ2 against u2
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

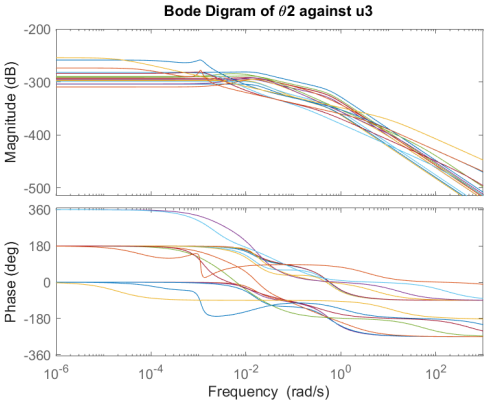

Bode Digram of θ3 against u2
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

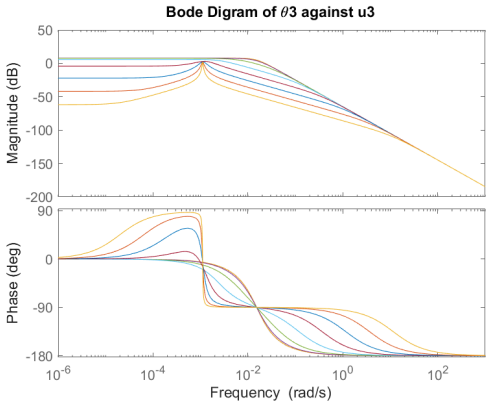

Bode Digram of ω1 against u2
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

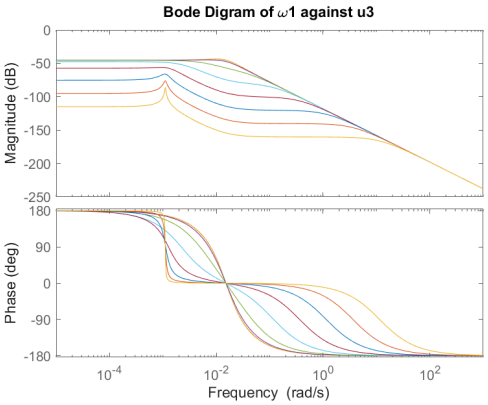

Bode Digram of ω2 against u2
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

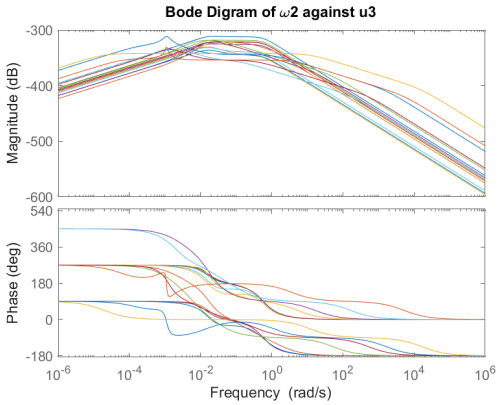

Bode Digram of ω3 against u2
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

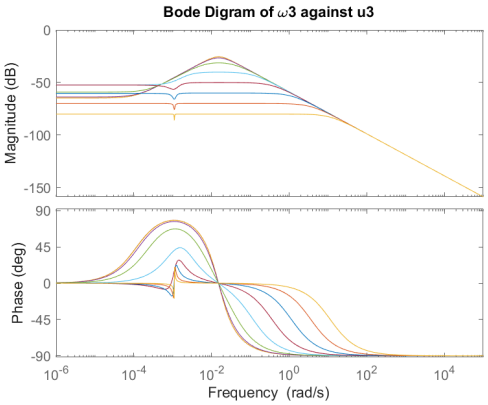

Bode Digram of θ1 against u3
Magnitude (dB)
Phase (deg)
Frequency (rad/s)


Bode Digram of θ2 against u3
Magnitude (dB)
Phase (deg)
Frequency (rad/s)


Bode Digram of θ3 against u3
Magnitude (dB)
Phase (deg)
Frequency (rad/s)


Bode Digram of ω1 against u3
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

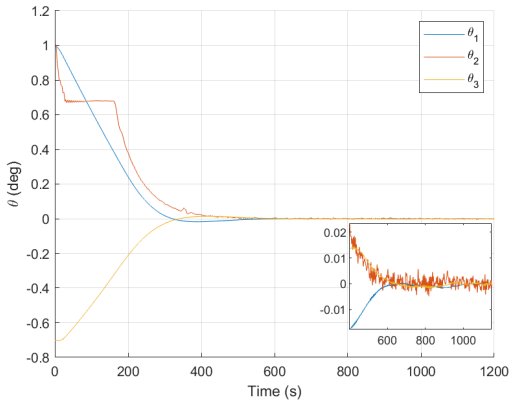

Bode Digram of ω2 against u3
Magnitude (dB)
Phase (deg)
Frequency (rad/s)


Bode Digram of ω3 against u3
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

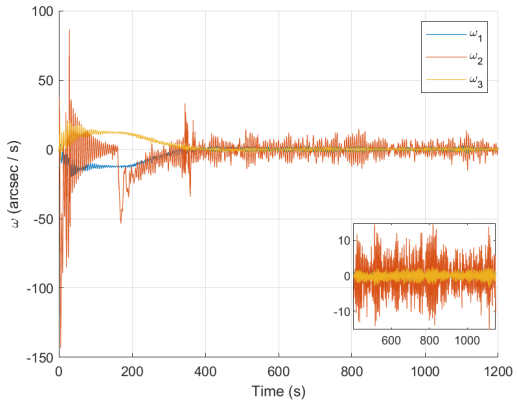
C

Simulation Results; Support Figures

C.1. Simulation Figures Case: Full Manoeuvrer

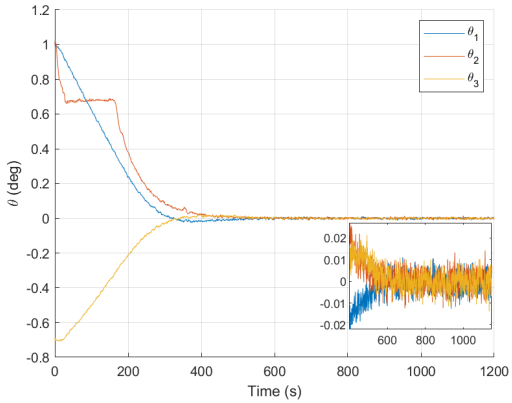


(a) Attitude angles

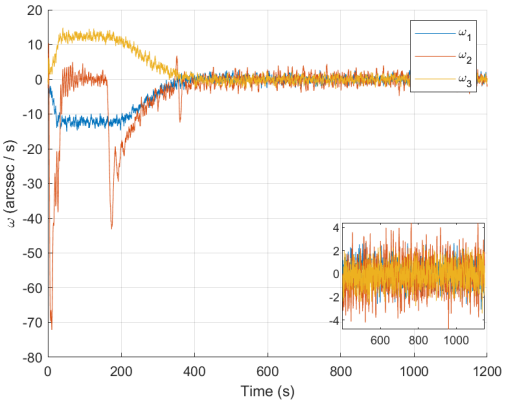


(b) Angular rates

Figure C.1: Spacecraft state of Alticube+'s central node

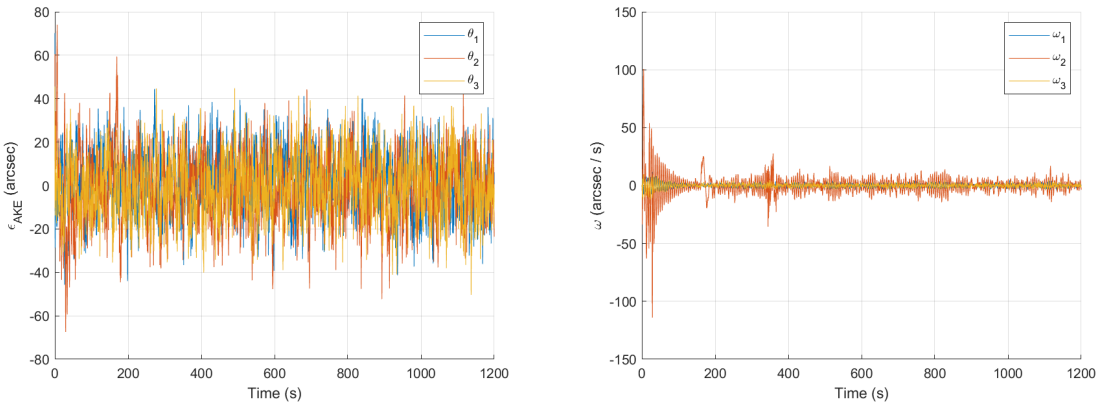


(a) Attitude angles



(b) Angular rates

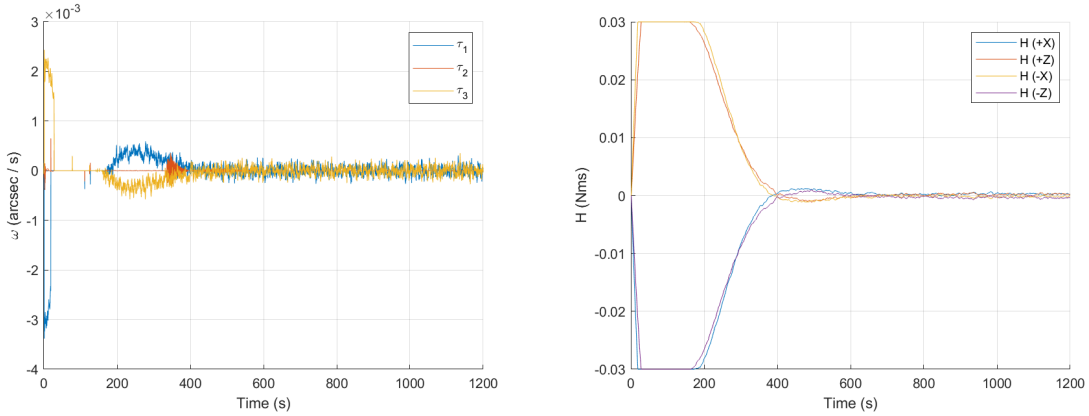
Figure C.2: Estimated state of Alticube+'s central node



(a) State error

(b) Angular rate error

Figure C.3: Estimated state error ϵ_{AKE} of Alticube+'s central node

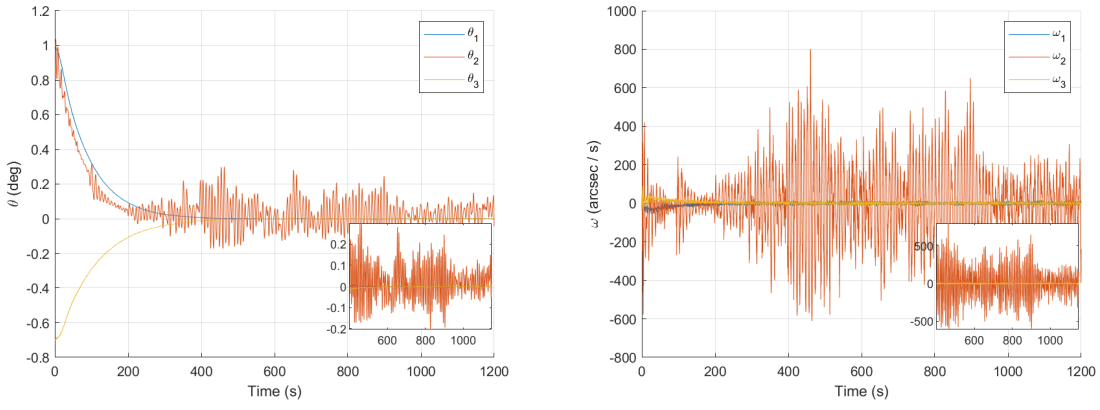


(a) Applied torque in the roll/pitch/yaw direction

(b) Angular momentum storage of central node reaction wheel system

Figure C.4: Reaction wheel system torque output and angular momentum storage

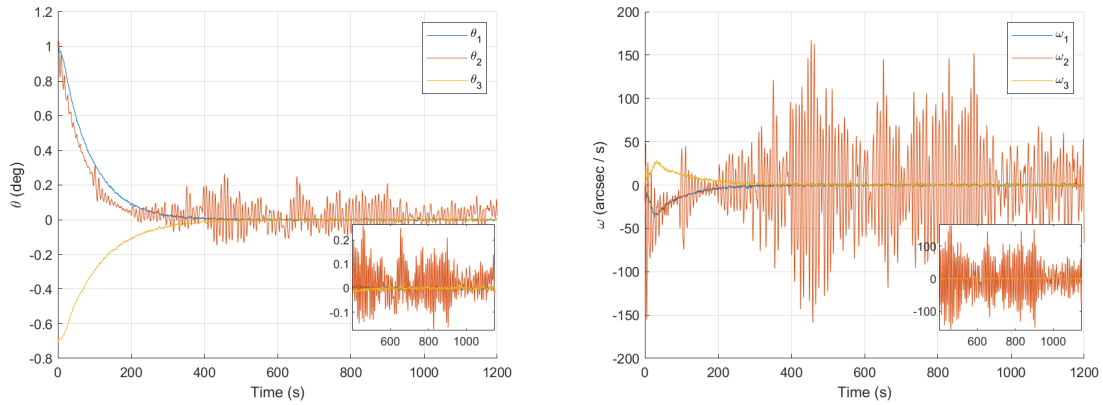
C.2. Simulation Figures Case: Worst Case



(a) Attitude angles

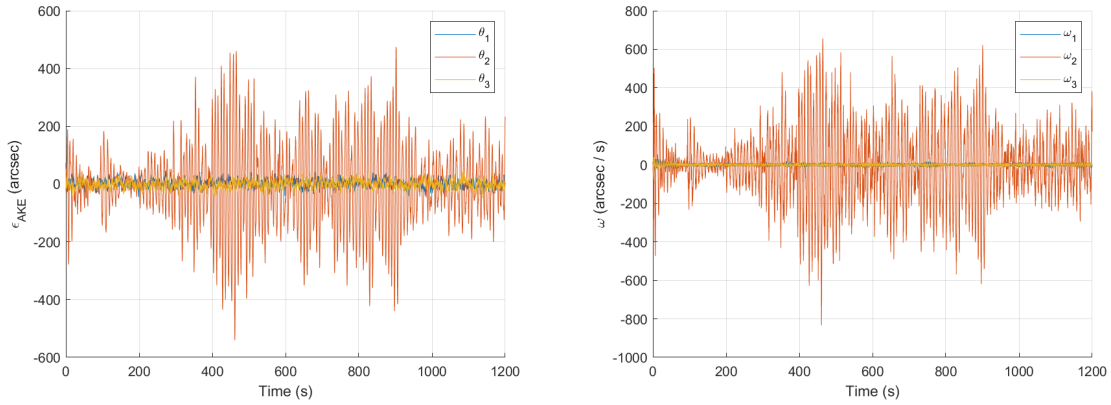
(b) Angular rates

Figure C.5: Spacecraft state of Alticube+'s central node



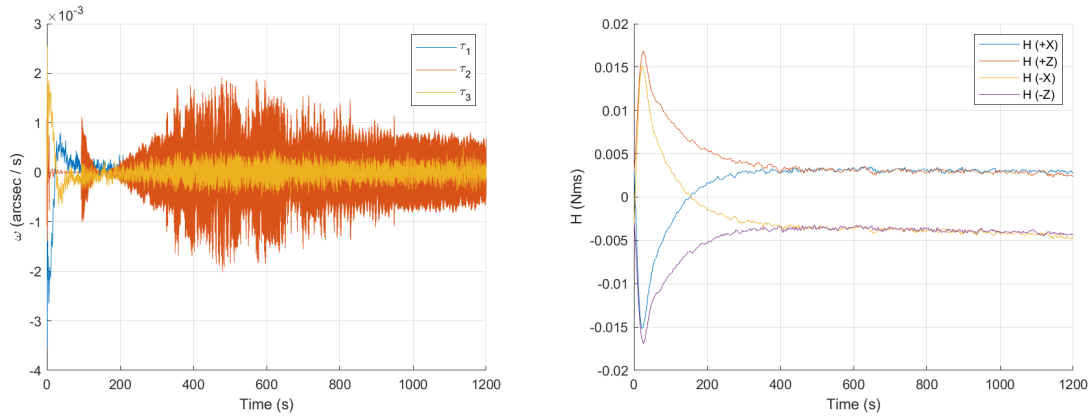
(a) Attitude angles **(b)** Angular rates

Figure C.6: Estimated state of Alticube+'s central node



(a) State error **(b)** Angular rate error

Figure C.7: Estimated state error ϵ_{AKE} of Alticube+'s central node



(a) Applied torque in the roll/pitch/yaw direction **(b)** Angular momentum storage of central node reaction wheel system

Figure C.8: Reaction wheel system torque output and angular momentum storage