

Markus Lægreid Haldorsen

HYPISO-1 Lunar Observations

Automated Hyperspectral Imaging: A Study on
Lunar Observations and Sensor Calibration

Master's thesis in Cybernetics and Robotics

Supervisor: Tor Arne Johansen

Co-supervisor: Sivert Bakken

June 2023

Markus Læg Reid Haldorsen

HYP SO-1 Lunar Observations

Automated Hyperspectral Imaging: A Study on Lunar Observations and Sensor Calibration

Master's thesis in Cybernetics and Robotics
Supervisor: Tor Arne Johansen
Co-supervisor: Sivert Bakken
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics



Norwegian University of
Science and Technology

HYP SO-1 Lunar Observations

Automated Hyperspectral Imaging: A Study on Lunar
Observations and Sensor Calibration

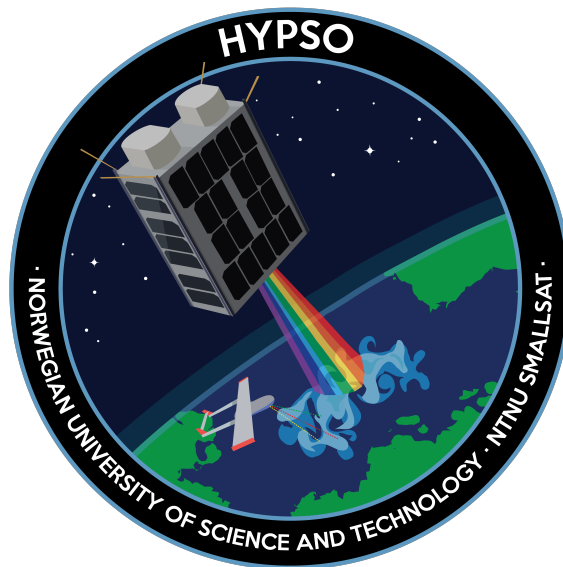
TTK4900 - 30.0 SP

Master's Thesis in Cybernetics and Robotics

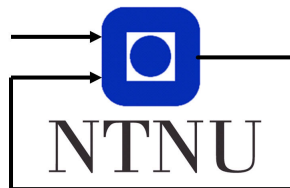
by

Markus Lægreid Haldorsen

June 2023



NTNU SmallSat Lab



Department of Engineering Cybernetics

Abstract

The importance of accurate hyperspectral imaging in remote sensing has given rise to the necessity of enhanced calibration techniques. This thesis investigates the automation of lunar observations as a method for calibrating hyperspectral imaging sensors, specifically onboard the HYPSONO-1 satellite. The study attempts to answer how hyperspectral imaging can be used to capture and analyze the full moon phases, the technical challenges in automating moon captures, and how the findings can be applied to other applications or systems.

The methodology involved developing and implementing algorithms and models to automate the lunar observation process, creating a system that minimizes the need for manual intervention. Python and Systems Tool Kit were the primary tools used for developing and testing the solution.

The work resulted in a streamlined process for capturing lunar data with the HYPSONO-1 satellite. While there were successful captures, the process also faced some failures, providing a balanced view of the performance and the areas for improvement.

This research offers an improved process for lunar observations and sensor calibration. It contributes a step towards automating the lunar observation process, offering potential benefits in terms of operational efficiency and accuracy. While its focus is the HYPSONO-1 satellite, the findings could have broader implications for other planetary observation missions or small satellite missions. It also highlights areas for future research, in refining the approach and expanding it to different contexts.

Abstrakt

Betydningen av nøyaktig hyperspektral bildebehandling i fjernmåling har ført til behovet for forbedrede kalibreringsteknikker. Denne avhandlingen undersøker automatisering av måneobservasjoner som en metode for kalibrering av hyperspektrale bildesensorer, spesifikt ombord på HYPPO-1-satellitten. Studien forsøker å svare på hvordan hyperspektral bildebehandling kan brukes til å fange og analysere fullmånefaser, de tekniske utfordringene med å automatisere månefangst, og hvordan funnene kan brukes til andre applikasjoner eller systemer.

Metodikken involverte utvikling og implementering av algoritmer og modeller for å automatisere måneobservasjonsprosessen, og skape et system som minimerer behovet for manuell intervensjon. Python og Systems Tool Kit var de primære verktøyene som ble brukt for å utvikle og teste løsningen.

Arbeidet resulterte i en strømlinjeformet prosess for å fange månedata med HYPPO-1-satellitten. Selv om det var vellykkede fangster, opplevde prosessen også noen feil, noe som gir et balansert syn på ytelsen og områdene for forbedring.

Denne forskningen tilbyr en forbedret prosess for måneobservasjoner og sensor kalibrering. Den bidrar til automatiseringen av måneobservasjonsprosessen, og tilbyr potensielle fordeler i form av operasjonell effektivitet og nøyaktighet. Selv om fokuset er på HYPPO-1-satellitten, kan funnene ha bredere implikasjoner for andre planetobservasjonsoppdrag eller små satellittoppdrag. Den fremhever også områder for fremtidig forskning, i raffinering av tilnærmingen og utvidelse til forskjellige kontekster.

Preface

This thesis marks the culmination of my Master's program in Cybernetics and Robotics at the Norwegian University of Science and Technology (NTNU), and it is a work I am particularly proud to present. The idea to explore the automation of lunar observations as a method for calibrating hyperspectral imaging sensors came as a natural progression from my interest in remote sensing and satellite technology, and it grew in scope as I delved deeper into this intricate and fascinating area.

The process of writing this thesis has been both challenging and rewarding. It has required me to employ and enhance my technical skills, and to engage in critical thinking and problem-solving in a complex, multi-faceted field. The journey towards its completion has provided me with a broader perspective and deeper understanding of hyperspectral imaging and lunar observations, as well as their significance in our quest to explore and understand the universe.

I am deeply grateful to the Norwegian University of Science and Technology for providing me with the resources and support necessary to complete this work. I wish to express my profound gratitude to my supervisor, Sivert Bakken, as well as the entire NTNU SmallSat Lab team, whose guidance, insights, and expertise have been invaluable to this study. Their support and dedication have helped me overcome many obstacles and enriched my learning experience, and the collaborative spirit and helpful feedback has greatly aided my work.

As I reflect on the journey of this thesis, I am filled with a sense of achievement, aware that it is a small but concrete step forward in the broad, intricate field of space technology. This thesis is a testament to the learning and growth I've experienced during my time at NTNU, and I'm looking forward to seeing its practical implications in the real world.

Markus Lægreid Haldorsen
Trondheim, June 2023

Contents

1	Introduction	2
1.1	Motivation	2
1.2	Problem Formulation	3
1.3	Objectives and Tasks	4
2	Background Theory	5
2.1	HYPISO-1	5
2.1.1	Mission Objectives	6
2.1.2	Instrumentation	6
2.1.3	Concept of Operations	8
2.2	Hyperspectral Remote Sensing	9
2.2.1	Spectral Information	9
2.2.2	Spectral Resolution	10
2.2.3	Data Analysis	10
2.3	Orbital Mechanics	11
2.3.1	Kepler’s Laws of Planetary Motion	11
2.3.2	Orbital Elements	12
2.3.3	Orbit Determination	13
2.4	Attitude Determination and Control System	14
2.4.1	Attitude Representation	14
2.4.2	Attitude Determination	15
2.4.3	Attitude Control	15
2.5	Radiometric Calibration	15
2.5.1	Lunar Calibration Using the LIME Model	16
3	Methods and Tools	18
3.1	Capturing the moon	18
3.1.1	Pixels needed across track	18
3.1.2	Frames needed along track	19
3.2	Python	21
3.2.1	Skyfield	21
3.3	Systems Tool Kit	22
3.3.1	Planning Moon Captures	22
3.3.2	Flight Scripts	23
3.3.3	Case Study: HYPISO-1 Calibration	25
3.4	Lunar Reflectance Models	25
3.4.1	Lunar Irradiance Model ESA (LIME)	25
3.4.2	Robotic Lunar Observatory (ROLO)	26
3.4.3	Other alternatives	26
4	Work and Results	27
4.1	Celestial Bodies	27
4.1.1	Distances	28
4.1.2	Distance from Object to Target	28

4.1.3	Get Minimum Distance	29
4.1.4	Angles	30
4.1.5	Get Off-Nadir Angle	30
4.1.6	Get Maximum Off Nadir Angle	32
4.1.7	Get Lunar Zenith Angle	33
4.2	Pointing	34
4.2.1	Get Quaternion	34
4.2.2	ECI to LVLH Transformation	35
4.2.3	Rodrigues Rotation	36
4.2.4	Rotation to Quaternion Conversion	37
4.3	Timing	38
4.3.1	Total Capture Time	38
4.3.2	Calculate Frames per Second	38
4.4	Execution	38
4.4.1	SatNav	39
4.4.2	Moon Capture Planning for HYPSON-1	40
4.5	Observing the moon	42
4.5.1	Planning captures	42
4.5.2	Obtaining captures	43
4.6	Captures	43
4.6.1	Successful captures	44
4.6.2	Unsuccessful captures	47
4.6.3	Analysis	52
5	Discussion	61
5.1	Celestial Bodies	61
5.2	Pointing	62
5.3	Timing	63
5.4	Execution	64
5.5	Observing the Moon and Results	65
5.6	Research Questions	66
5.6.1	Hyperspectral Imaging for Lunar Observation	66
5.6.2	Automation Challenges	66
5.6.3	Applications of Findings	67
5.7	Limitations	67
6	Conclusion and Future Work	69
7	Attachments	74

Acronyms and Abbreviations

ADCS Attitude Determination and Control System.

ASV Autonomous Surface Vehicles.

AU Astronomical Units.

AUV Autonomous Underwater Vehicles.

CAN Controller Area Network.

CONOPS Mission Concept Of Operations.

ECEF Earth-Centered, Earth-Fixed.

ECI Earth-Centered Inertial.

ESA European Space Agency.

ESTEC European Space Research and Technology Centre.

FC Flight Computer.

FoV Field of View.

FPGA Field Programmable Gate Array.

FPS Frames Per Second.

GCRS Geocentric Celestial Reference System.

GPS Global Positioning System.

HABs Harmful Algal Blooms.

HSI Hyperspectral Imaging.

Hypso-Cli Hypso Command Line Interface.

ICRS International Celestial Reference System.

LIME Lunar Irradiance Model ESA.

LVLH Local-Vertical-Local-Horizontal.

MCS Mission Control Software.

MPC Mission Performance Centre.

NAIF Navigation and Ancillary Information Facility.

NTNU the Norwegian University of Science and Technology.

OLCI Ocean Land Color Instrument.

OPU Onboard Processing Unit.

PBHSI Pushbroom Hyperspectral Imager.

PC Payload Controller.

RGB Red, Green and Blue.

ROLO Robotic Lunar Observatory.

RTC Real-Time Clock.

SLC stray-light correction.

SmallSat Lab SmallSat Lab.

SNR Signal-to-Noise Ratio.

SoC System-on-a-Chip.

STK Systems Tool Kit.

TLE Two-Line Elements set.

ToA Top of Atmosphere.

UAV Unmanned Aerial Vehicles.

UEMETSAT European Organisation for the Exploitation of Meteorological Satellites.

UTC Universal Time Coordinated.

VIS-NIR Visual and Near-Infrared.

VITO Flemish Institute for Technological Research.

VNIR Visible and Near-Infrared.

Chapter 1

Introduction

Contents

1.1	Motivation	2
1.2	Problem Formulation	3
1.3	Objectives and Tasks	4

On January 13th, 2022, a satellite with the name HYPSON-1 was launched on a Falcon 9 Block 5 vehicle from SLC-40 at Cape Canaveral Space Force Station in Florida. HYPSON-1 is the first research satellite made by a Norwegian university that has been successfully launched into orbit.

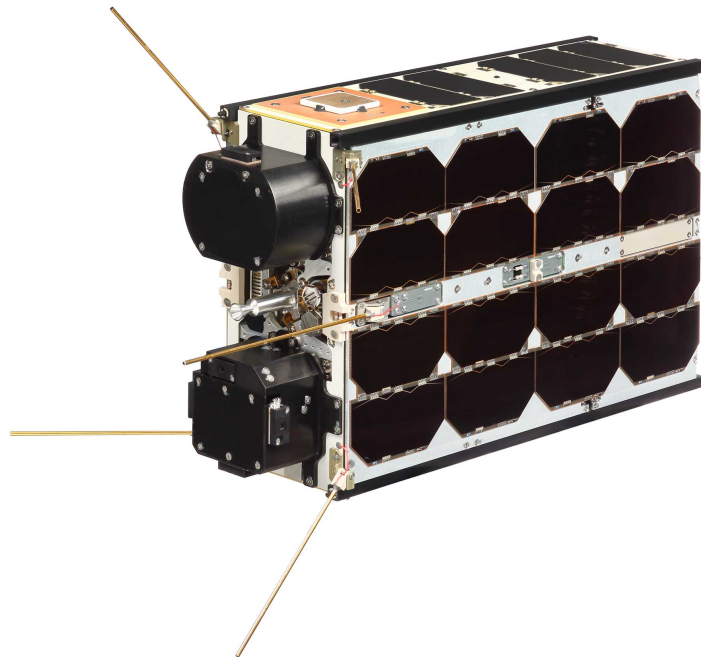


Figure 1.1: HYPSON-1 (HYPER-spectral Satellite for Ocean Observation, HYPSON-1), obtained from [1].

1.1 Motivation

The quest for more accurate and reliable data from earth observation instruments is an ongoing challenge in the field of remote sensing. Hyperspectral imaging sensors, due to their ability to capture a wide spectrum at many contiguous wavelengths, have revolutionized how we perceive and analyze the world. However, like all instruments,

their performance is influenced by numerous factors including noise, systematic bias, and environmental factors among others. As such, the process of calibration becomes imperative to ensure that the measurements made by these sensors are accurate, precise, and reliable.

Calibration, in essence, adjusts for these influences by establishing a known relationship between the observed values and the true values. This allows for more meaningful and confident data interpretation and application, such as detecting changes in ecosystems, studying the atmospheric composition, or analyzing the health of crops. Given the complexity of hyperspectral data, calibration becomes a non-trivial task that requires thoughtful design and execution.

The Moon, with its predictable cycles and stable spectral characteristics, provides a unique opportunity as a calibration source. By automating the process of lunar observations, we aim to enhance the calibration procedure of hyperspectral imagers. Yet, this endeavour is not without its challenges – challenges we aim to tackle within this thesis. This motivation sets the stage for an exciting journey into automating lunar observations for the calibration of hyperspectral imagers.

1.2 Problem Formulation

Hyperspectral imagers, with their ability to provide a wealth of spectral information, have the potential to revolutionize our understanding of the world. However, this potential can only be fully realized when the data they produce is reliable and accurate, which fundamentally depends on an effective calibration. One unique calibration method involves observing the Moon, a celestial body that presents us with a stable and predictable spectral source. Manual lunar observations for calibration purposes present challenges, including the timing of observations to coincide with the appropriate lunar phase, adjusting the orientation of the hyperspectral sensor to correctly target the Moon, and handling the vast amounts of data generated.

Creating an automated system to capture the lunar phases with the HYPSON-1's hyperspectral sensor would reduce the human workload and increase the efficiency of the calibration process. However, this automation involves understanding various technical aspects related to the HYPSON-1's operation and the specifics of hyperspectral imaging, as well as designing a system that successfully integrates these elements.

This thesis seeks to explore this intricate problem space, aiming to devise automated methods for capturing the lunar phases with the HYPSON-1 satellite's hyperspectral imager. This involves understanding the technical aspects related to the operation of the HYPSON-1 satellite and the specifics of hyperspectral imaging and, importantly, devising an automated process that successfully integrates these elements in the context of lunar observation. The challenges associated with this process lead us to the main research questions that this thesis seeks to answer:

- How can hyperspectral imaging be used to capture and analyze the full moon phases, and what benefits does this provide for lunar observation?

- What are the technical challenges involved in automating moon captures with hyperspectral imaging, and how can these be overcome?
- How can the findings from this research be applied to other applications or systems, such as small satellite missions or other planetary observation missions?

1.3 Objectives and Tasks

The primary objective of this thesis is to design, implement, and evaluate an automated system that uses HYPSON-1's hyperspectral imaging sensor for capturing lunar observations, thus facilitating the calibration process. This objective is divided into specific tasks that aim to:

- Understand the technical workings of the hyperspectral sensor aboard HYPSON-1 and its requirements for calibration.
- Explore the application of hyperspectral imaging in capturing and analyzing the lunar phases.
- Develop an automation solution that coordinates the sensor's data capture with the Moon's phases and positions.
- Assess the efficacy of the automated system through extensive testing and analysis.
- Discuss potential applications of the research findings in broader contexts, such as small satellite missions and planetary observation missions.

The anticipated outcome of this thesis is a reliable, automated system designed to simplify the calibration process of hyperspectral imaging sensors. By completing these tasks, the work seeks not only to contribute substantially to the existing body of knowledge but also to influence the design and operation of future hyperspectral imaging systems. Overall, the purpose is to offer valuable insights and tangible improvements to the ongoing efforts in automating lunar observations and the calibration of hyperspectral sensors.

Chapter 2

Background Theory

Contents

2.1	HYP SO-1	5
2.1.1	Mission Objectives	6
2.1.2	Instrumentation	6
2.1.3	Concept of Operations	8
2.2	Hyperspectral Remote Sensing	9
2.2.1	Spectral Information	9
2.2.2	Spectral Resolution	10
2.2.3	Data Analysis	10
2.3	Orbital Mechanics	11
2.3.1	Kepler’s Laws of Planetary Motion	11
2.3.2	Orbital Elements	12
2.3.3	Orbit Determination	13
2.4	Attitude Determination and Control System	14
2.4.1	Attitude Representation	14
2.4.2	Attitude Determination	15
2.4.3	Attitude Control	15
2.5	Radiometric Calibration	15
2.5.1	Lunar Calibration Using the LIME Model	16

In this chapter, we will explore the different elements that realize HYP SO-1’s mission, ranging from the satellite’s instrumentation and concept of operations to the concepts of hyperspectral remote sensing and orbital mechanics, as well as the determination and control of the satellite’s attitude. Finally, we will introduce the process of radiometric calibration through lunar observations and why this is necessary.

2.1 HYP SO-1

The HYP SO-1 satellite is a 6U CubeSat developed by the Norwegian University of Science and Technology (NTNU) with the mission objective of observing ocean colour along the coast of Norway. The satellite was launched into a sun-synchronous polar orbit at an altitude of 540 km on January 13th, 2022 [2]. To achieve its mission objectives, HYP SO-1 is equipped with a custom-built Pushbroom Hyperspectral Imager (PBHSI) that captures imagery in the spectral range of approximately 387-801 nm with a theoretical spectral bandpass of 3.33 nm and a swath width of 70 km [3].

2.1.1 Mission Objectives

The HYPSO-1 satellite has a set of mission objectives aimed at monitoring ocean colour events in the Visual and Near-Infrared (VIS-NIR) wavelengths between 400-800 nm, and inferring phytoplankton functional groups. To achieve these objectives, the mission has identified several key user needs that must be met, from [3]:

- Images should have spatial resolution better than 30 - 100 m per pixel.
- Raw hyperspectral data should have a spectral resolution of about 5 nm for VIS-NIR wavelengths.
- The imager's Signal-to-Noise Ratio (SNR) at Top of Atmosphere (ToA) should be greater than 400 in visual wavelengths for open ocean water, and the atmospherically corrected SNR of water-leaving signals should be between 40 - 100.
- Data latency should be less than 1 hour.
- Revisit times to dedicated areas of interest should be 3 - 72 hours.

The mission objectives of HYPSO-1 are primarily focused on identifying and monitoring Harmful Algal Blooms (HABs) or cyanobacteria in Norwegian coastal waters. These phytoplankton blooms have variable colouration and are often referred to as "red tides," "green tides," or "brown tides," with wavelengths between 400-700 nm that fall within the range of HYPSO-1's spectral capabilities. They sporadically appear worldwide with varying biomass concentrations, lasting from a few minutes to a few days, and cover areas ranging from tens to hundreds of square kilometres in size. Such blooms may cause sudden damage to the marine environment, ecosystems and sustainable food sources.

2.1.2 Instrumentation

Imager

The PBHSI employed by HYPSO-1 operates by scanning in the direction of the velocity vector, sequentially capturing several lines, N_x , each composed of instantaneous pixels, N_y and N_λ , to form a hyperspectral datacube [3], as shown in Figure 2.1. The number of spatial pixels perpendicular to the scan direction is denoted by N_y , while the number of pixels along the spectral dimension is represented by N_λ . The Field of View (FoV) of the PBHSI is characterized by the horizontal and vertical components, ϵ_w and ϵ_h , respectively. The integration time, Δt , represents the time elapsed between two consecutive lines, or frames, and can be expressed as:

$$\Delta t = \frac{1}{FPS} = \tau + \delta t \quad (2.1)$$

where

- FPS is the frame rate,

- τ is the camera exposure time,
- δt is the read-out time.

Figure 2.1 illustrates the collection process of N_x frames with N_λ and N_y pixels using the PBHSI [3]. The PBHSI is a valuable tool for capturing hyperspectral images, which are useful for a variety of remote sensing applications.

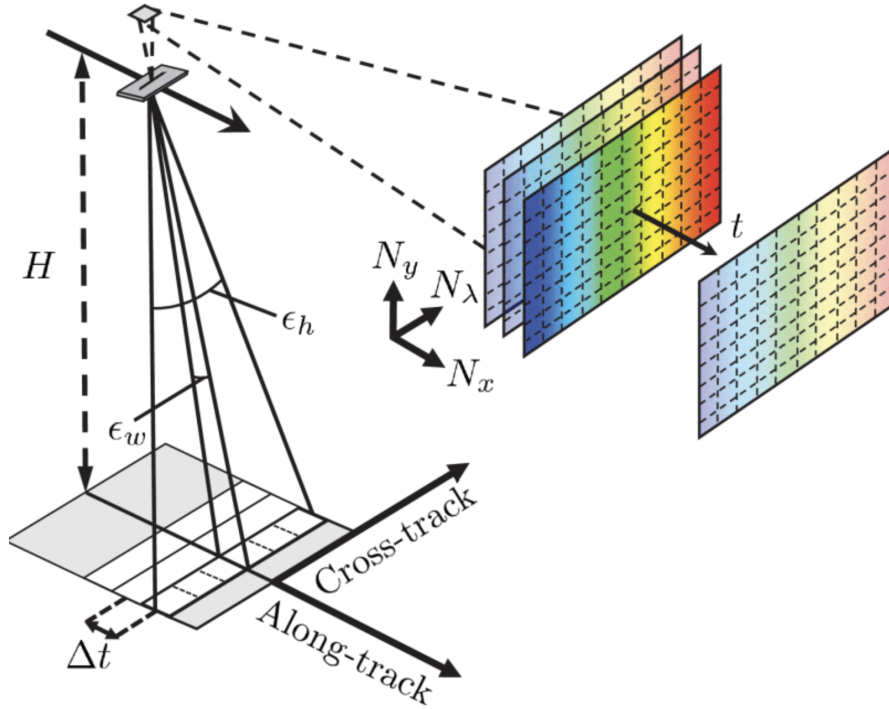


Figure 2.1: Illustration of a PBHSI collecting N_x frames with N_λ and N_y pixels, obtained from [3].

RGB Camera

The satellite is equipped with an IDS UI-125x RGB camera with a 6mm F/1.4 Ci series fixed lens. This camera provides a footprint of 770 km x 540 km and a spatial resolution of approximately 500m [3]. Its main purpose is to support and validate hyperspectral images in the spatial domain.

Transceiver

For downlinking payload data, the satellite uses a 2.4 GHz IQ Spacecom S-band Transceiver. This transceiver provides a usable data rate of 1 Mbit/s [3].

Onboard Processing Unit

The satellite's Onboard Processing Unit (OPU) hosts a Zynq 7030 Xilinx PicoZed System-on-a-Chip (SoC) with flight heritage. It consists of two core ARM processors and a Field Programmable Gate Array (FPGA) dedicated for onboard image processing. The OPU allows for in-orbit updates of both software and FPGA hardware

reconfigurations for uploaded algorithms. Larger data sizes can be buffered from the OPU to the Payload Controller (PC) over Controller Area Network (CAN) before downlinking over S-band radio, or smaller amounts of data can be downloaded directly from the OPU. Buffering data to the PC enables full utilization of the S-band data rate and removes the need for keeping the OPU turned on for longer than necessary. Power and data-line distribution to the hyperspectral and RGB cameras are granted through a custom break-out board with PicoZed interfaces. Furthermore, the OPU hosts an SD-card with 8 GB storage capacity [3].

2.1.3 Concept of Operations

HYPSO-1's mission has four main capabilities as illustrated in Figure 2.2 that form the Mission Concept Of Operations (CONOPS). These capabilities are [3]:

1. **Hyperspectral imager orientation:** the satellite can orient its hyperspectral imager to start scanning a pre-defined area based on telecommands from a ground station.
2. **Single-axis slew manoeuvre:** the satellite can execute a single-axis slew manoeuvre to tilt the imager backwards as it translates forward, rotating the imager's footprint slowly backwards with respect to the scan direction. This enhances the spatial resolution along the scan direction.
3. **Onboard image processing:** images are processed immediately onboard to reduce their data size and speed up the download on the ground.
4. **Groundstation downlink:** the satellite downlinks data to ground stations located at NTNU Trondheim and KSAT Svalbard, Norway [4]. It may also (not yet integrated at the time of writing) downlink data to KSAT Puertollano, Spain, or to one of NTNU's Unmanned Aerial Vehicles (UAV)'s, Autonomous Surface Vehicles (ASV)'s and Autonomous Underwater Vehicles (AUV)'s that may collect in-situ data if within range of the observed area.

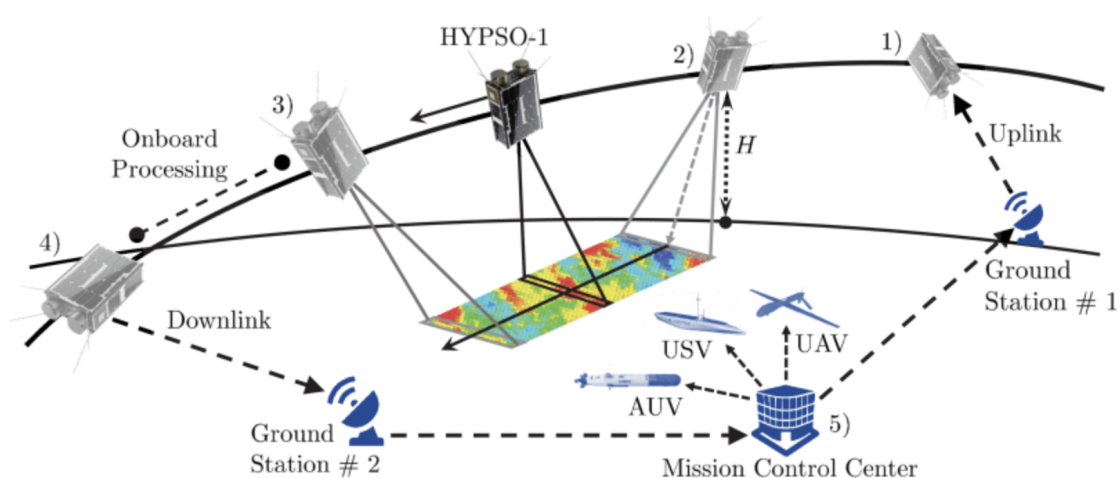


Figure 2.2: HYPSO-1 Concept of Operations, obtained from [3].

2.2 Hyperspectral Remote Sensing

Hyperspectral remote sensing, also known as Hyperspectral Imaging (HSI) or imaging spectroscopy, is an imaging technique used to collect and process information from across a broad range of wavelengths. Unlike the human eye and most conventional cameras, which can only see a small portion of the electromagnetic spectrum, namely visible light, HSI can detect individual absorption features that are due to specific bonds in materials. This makes it possible to identify different materials and their properties, such as the concentration of a particular pigment. This is particularly useful for detecting trace chemicals in the environment, such as chlorophyll-a concentration or harmful algal blooms.

2.2.1 Spectral Information

Much like a traditional image with three bands, Red, Green and Blue (RGB), hyperspectral images are composed similarly - only with many more bands. In the case of the HYPSONO-1 satellite, there are 120 bands with wavelengths between approximately 387-801 nm at a theoretical spectral bandpass of 3.33 nm [3]. The bands, or images, are stacked to create a hyperspectral image cube, with three dimensions: the spatial dimensions x and y, and the spectral dimension z. Figure 2.3 below illustrates the layers in a hyperspectral image.

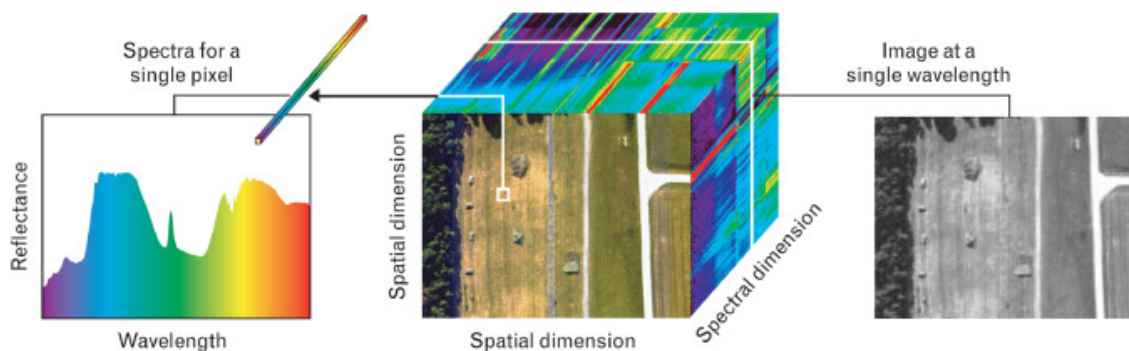


Figure 2.3: Illustration of a hyperspectral image cube, obtained from [5]. A hyperspectral image cube displays spatial and spectral dimensions at each pixel of an image. [6]

ã

The spatial dimensions represent the horizontal and vertical dimensions of the image, while the spectral dimension represents the wavelength dimension along which the image is collected. Each layer in the spectral dimension represents the spectral intensity at a specific wavelength at each pixel in the image. Passive hyperspectral sensors like the HSI in HYPSONO-1 use the sun as the primary source of radiation. The energy that reaches the materials on the earth's surface is reflected according to the material's spectral signature, and by taking out the solar and atmospheric contributions to the sensor measurements, only the ground reflectance is left. There are methods to correct the solar and atmospheric contributions, but this thesis will not discuss this further.

Figure 2.4 below shows a hyperspectral image captured by the HYPSONO-1 satellite,

displaying both an RGB representation and spectral intensity graphs of three chosen pixels marked by coloured dots. The RGB image is obtained by mapping the R, G, and B values to the 60th, 80th, and 90th bands, respectively. For the pixel marked with a green dot and graph, the corresponding values of the 60th, 80th, and 90th bands are also depicted as colour bars, which correspond to the visual colours of the image.

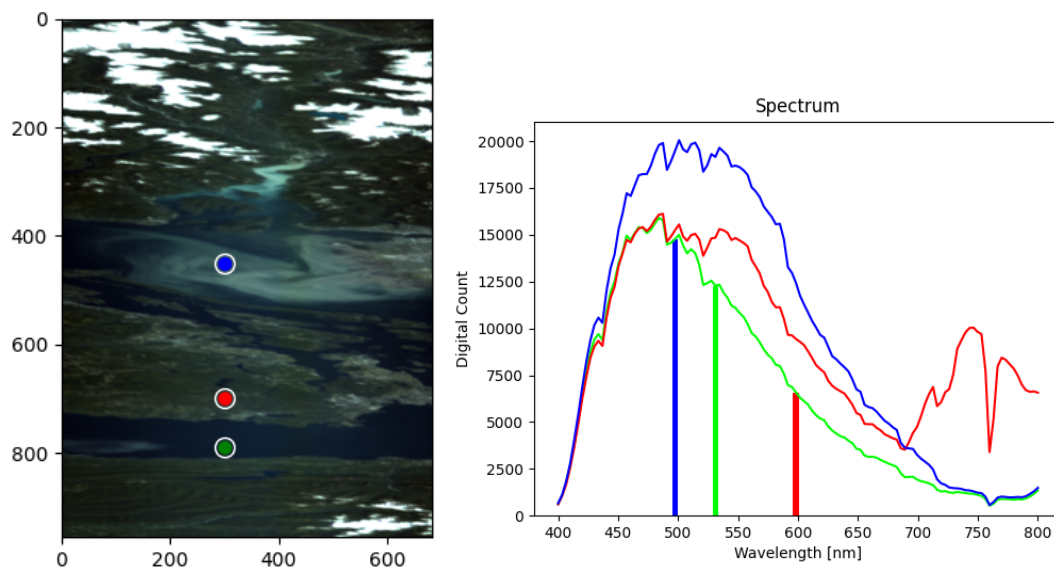


Figure 2.4: Left: Hyperspectral image obtained by HYPSO-1 satellite. Right: Graphs of the raw spectral intensity of three pixels, illustrated by coloured dots on the image. The bar chart shows the spectral intensity of the R, G and B values of the pixel marked with a green dot.

It is clear that the hyperspectral image contains much more information than a conventional RGB image, and that we can use the spectral intensity characteristics of different pixels to obtain information about the materials of which they are composed. For example, we see that the green and blue dots, which are located in water, have much lower spectral intensity in the wavelengths over 700 nm than that of the red dot, which is located on land. We also see a distinct difference between the green and blue dots because the blue dot is in an algal bloom.

2.2.2 Spectral Resolution

The spectral resolution of a hyperspectral image is the range of wavelengths that can be detected by the sensor. A high spectral resolution allows for the detection of subtle differences in spectral signatures, enabling the identification and classification of materials with greater accuracy. However, high spectral resolution can also result in large data volumes that require advanced data analysis techniques to extract meaningful information.

2.2.3 Data Analysis

Data analysis is a critical step in HSI, as large amounts of data are collected in the form of hyperspectral images. Various techniques can be used to extract useful in-

formation from hyperspectral images, such as hyperspectral target detection, which aims to identify specific targets based on their unique spectral signature [7]. Other methods include spectral unmixing [8], which decomposes the spectral signatures of pixels into the spectral signatures of constituent materials, classification algorithms [9], which group pixels with similar spectral signatures into classes, and anomaly detection [10], which identifies pixels with unusual spectral signatures.

In conclusion, HSI is a powerful technique that can detect and identify materials and their properties by analyzing the spectral signatures of pixels in a hyperspectral image. The technique's utility stems from its ability to collect information from a broad range of wavelengths and the spectral differences among materials. However, effective data analysis techniques are required to extract useful information from the large amounts of data collected.

2.3 Orbital Mechanics

Orbital mechanics is a fundamental field of study in astrophysics that deals with the motion of celestial bodies in space. All objects in orbit are governed by the laws of physics, which are subject to various factors such as gravity, mass, velocity, and the properties of the objects themselves. The understanding of these topics is important when working with satellites, for example in order to calculate pointing direction at specific times [11].

2.3.1 Kepler's Laws of Planetary Motion

Kepler's Laws, published by Johannes Kepler in the early 17th century, describe the motion of objects in orbit around a central body. These laws were developed based on observations made by Tycho Brahe [12].

First Law: Law of Ellipses

Kepler's first law states that the orbit of a planet around the sun is an ellipse with the sun at one focus. The distance between the sun and the planet changes throughout the orbit, with the closest point called perihelion and the farthest point called aphelion [12].

Mathematically, an ellipse can be represented by the formula:

$$r = \frac{\rho}{1 + \varepsilon \cdot \cos\theta} \quad (2.2)$$

where ρ is the semi-latus rectum, ε is the eccentricity of the ellipse, r is the distance from the Sun to the planet, and θ is the angle to the planet's current position from its closest approach, as seen from the Sun.

Second Law: Law of Equal Areas

Kepler's second law states that a planet in orbit around the sun sweeps out equal areas in equal times. This means that the planet moves faster when it is closer to the sun (at perihelion) and slower when it is farther away (at aphelion) [12].

This law is described mathematically by the following equation:

$$\frac{dA}{dt} = \frac{1}{2}r^2\frac{d\theta}{dt} \quad (2.3)$$

where A is the area swept out by the planet in a given time, r is the distance from the planet to the sun, and θ is the angle between the line connecting the planet and the sun and a reference line.

Third Law: Law of Harmonies

Kepler's third law states that the square of the period of a planet's orbit is proportional to the cube of the semi-major axis of the ellipse. In other words, planets that are farther from the sun take longer to complete an orbit than those that are closer [12].

This law is described mathematically by the following equation:

$$\frac{T^2}{a^3} = \frac{4\pi^2}{GM} \quad (2.4)$$

where T is the period of the planet's orbit, a is the semi-major axis of the ellipse, G is the gravitational constant, and M is the mass of the central body.

Overall, Kepler's laws of planetary motion are important for understanding the behaviour of objects in orbit around a central body and have been instrumental in the development of modern astronomy and space exploration.

2.3.2 Orbital Elements

[13] To accurately describe the position and motion of a spacecraft in orbit around a celestial body, six orbital elements are commonly used. These elements include the semi-major axis (a), eccentricity (e), inclination (i), right ascension of the ascending node (Ω), argument of periapsis (ω), and mean anomaly (M). A real orbit and its elements are influenced by the gravitational perturbations of other celestial bodies and the principles of general relativity. A Kepler orbit, as expressed by these six parameters, is an idealized, mathematical approximation of an orbit at a specific moment in time [14].

The semi-major axis, a , is the average distance between the centre of mass of the spacecraft and the centre of mass of the celestial body it is orbiting. This element is

a measure of the size of the orbit and is usually expressed in units of distance, such as kilometres or astronomical units.

The eccentricity is a measure of how elliptical the orbit is. An orbit with an eccentricity of zero is perfectly circular, while an orbit with an eccentricity greater than zero is elliptical. The eccentricity ranges from 0 (circular) to 1 (parabolic).

The inclination is the angle between the plane of the orbit and the plane of the celestial body's equator. This element is measured in degrees and can range from 0 (a perfectly equatorial orbit) to 180 (a retrograde orbit).

The right ascension of the ascending node is the angle between the celestial body's reference direction (such as the vernal equinox) and the point where the spacecraft's orbit crosses the celestial body's equatorial plane moving northward. This element is also measured in degrees.

The argument of periapsis is the angle between the ascending node and the point on the orbit closest to the centre of the celestial body being orbited (periapsis). This element is measured in degrees.

Finally, the mean anomaly is the angle between the spacecraft's position and the position it would occupy if it were moving at a constant speed in a circular orbit. This element is also measured in degrees and increases linearly with time.

Together, these six orbital elements provide a complete description of the position and motion of a spacecraft in orbit, and they are essential for calculating the spacecraft's position at any given time. In the context of this thesis, knowledge of these orbital elements will be crucial for accurately predicting the position of the spacecraft during lunar observations and for the calibration of the sensors.

2.3.3 Orbit Determination

Orbit determination is the process of determining the position and velocity of a satellite in its orbit at a given time. To accurately determine the orbit of a celestial body, a relatively advanced set of calculation based on observations from tracking stations on the ground are required. The United States Space Force [15] tracks all detectable objects in Earth orbit, creating publicly available Two-Line Elements set (TLE) data corresponding to each object.

Two Line Element set (TLE)

TLE is a standardized format for providing information about the orbit of a satellite. They consist of two lines of text that include information such as the inclination and eccentricity as described in 2.3.2, as well as the satellite's name, international designator, argument of perigee, mean anomaly, mean motion and epoch time.

There are many available tools that can be used to calculate a satellites position, velocity, azimuth and elevation and any other data required based on TLE sets, such as Skyfield [16], Pyorbital [17] and more, which are further discussed in Methods

and Tools.

While TLE data provide a convenient way to obtain information about a satellite's orbit, they have limitations. TLE's simplification ignores many of the complexities of the earth's gravity field, such as variations in the earth's shape, mass distribution, and the gravitational effects of other celestial bodies, to name a few. As a result, TLE data is only accurate for a short period of time, typically from a few days to a few weeks. It is therefore important to update TLE data often and to not rely on the calculated orbits if the time of interest is far in the future or past.

2.4 Attitude Determination and Control System

In order to point the satellite in the correct direction when capturing images, we need methods to estimate and represent the orientation of the satellite, as well as some way of controlling the orientation. The orientation of a satellite is also known as its attitude.

2.4.1 Attitude Representation

There are several ways to represent spacecraft orientation, including Euler angles and quaternions. Euler angles describe the orientation of a spacecraft in terms of three angles that represent rotations around three distinct axes [18]. These angles are typically denoted by the symbols ϕ (phi), θ (theta), and ψ (psi), and correspond to rotations around the spacecraft's X, Y, and Z axes, respectively.

Quaternions are another way of representing spacecraft orientation, as they are a set of four numbers that represent a rotation in three-dimensional space. The first three numbers correspond to the axis of rotation, and the fourth number represents the angle of rotation. Quaternions are a mathematical concept that extends the concept of complex numbers to four dimensions. In the context of spacecraft attitude representation, quaternions are a way of representing orientation that avoids some of the problems associated with Euler angles. One of the main advantages of quaternions is their ability to avoid gimbal lock, something which can occur when using Euler angles. Gimbal lock is the loss of one degree of freedom in a three-dimensional, three-gimbal mechanism that occurs when the axes of two of the three gimbals are aligned, "locking" the system into rotation in a degenerate two-dimensional space, ultimately causing the other two axes to become coincident. This can make it difficult or impossible to represent certain orientations using Euler angles.

In contrast, quaternions are able to represent all possible rotations without encountering gimbal lock, as quaternions represent orientation as a single rotation in three-dimensional space, rather than a series of rotations around distinct axes. This makes them particularly useful in applications that require high precision, such as satellite attitude control.

Another advantage of quaternions is their ease of use in numerical calculations. Quaternions can be multiplied and divided like regular numbers, which makes them particularly useful in computer simulations and other numerical applications. [19]

[20].

2.4.2 Attitude Determination

Attitude determination is a critical component of space missions that involve imaging or precise orientation requirements. To ensure precise attitude estimation during HYPSON-1's imaging process, the satellite uses a Nano Star Tracker ST-1 and Sensor STIM 210 Inertial Measurement Unit (IMU) [3]. These sensors rely on the detection and identification of known stars in the sky to accurately determine the spacecraft's position and orientation. By comparing the observed positions of the stars to known star positions, the star tracker can calculate the spacecraft's orientation in three-dimensional space. In cases where images will not be taken, the satellite uses six sun sensors, three magnetometers, and three gyroscopes instead. While these provide coarser attitude knowledge, they consume less power [3].

2.4.3 Attitude Control

Attitude control is critical to maintaining the stability and control of a spacecraft. HYPSON-1 uses four reaction wheels for attitude control, each providing up to 3.2 mNm of torque. Three of these wheels are placed orthogonally along the body axes, while the fourth is tilted at an angle of 54.7°. The wheels are used to generate the necessary torques to control the spacecraft's attitude and to maintain its orientation in space. Additionally, two magnetorquers are placed along each body axis for reaction wheel momentum dumping [3]. The combination of reaction wheels and magnetorquers allows the HYPSON-1 satellite to maintain precise control of its attitude, enabling it to execute its mission objectives.

2.5 Radiometric Calibration

Sensor calibration is a fundamental component of hyperspectral imaging, ensuring precise and consistent measurements of the spectral signatures reflected from Earth's surfaces. These spectral signatures play a significant role in accurately identifying and categorising different surface materials, which is crucial in a variety of applications, including environmental monitoring and precision agriculture.

Radiometric calibration is the process of converting raw sensor counts into calibrated radiance or reflectance values. It involves the measurement of a known reference target, the calculation of calibration coefficients based on the sensor's response to the reference target, and the subsequent application of these coefficients to the raw sensor data.

One challenge in radiometric calibration is the necessity to account for atmospheric conditions, such as the absorption and scattering of light. Lunar observations can be used as a calibration reference to mitigate these atmospheric effects. The Moon, with its lack of an atmosphere and relatively stable surface, provides a consistent source for calibrating sensor responses across a wide range of wavelengths. By comparing the sensor response to known lunar reflectance values, calibration coefficients can be derived and used to correct systematic errors in the sensor's measurements.

The Lunar Irradiance Model ESA (LIME) is an example of a tool used for radiometric calibration. Developed by the European Space Agency (ESA), this model simulates the spectral reflectance of the lunar surface. LIME takes into account several factors, including the geometry of the Sun-Moon-Earth system, the topography of the lunar surface, and the reflectance properties of lunar surface materials.

The utility of LIME lies in its capability to predict irradiance values at the sensor for any given lunar observation, thereby enabling precise calibration of the sensor response. The LIME instrument is scheduled to operate until 2024, with the model being updated annually based on accumulating lunar measurements. These continuous updates enhance the model's accuracy and broaden its application, reinforcing its value in the field of hyperspectral imaging.

2.5.1 Lunar Calibration Using the LIME Model

Lunar calibration employs the Moon as a radiometric reference for spaceborne and airborne sensors. The LIME model is of particular importance in this context, as it facilitates the derivation of accurate calibration coefficients for hyperspectral sensors based on lunar observations. The following sequence of steps elucidates the typical procedure adopted when implementing the LIME model for hyperspectral imager calibration:

1. *Collecting Lunar Observations:* The initial step involves amassing a series of lunar observations captured by the hyperspectral imager under diverse illumination conditions and viewing angles. This will be the main objective of this thesis, as automating the process of collecting lunar observations and successfully capturing enough diversity in the illumination conditions and viewing angles is an advanced and time-consuming task.
2. *Acquiring LIME Data:* The subsequent step involves accessing the LIME model data, which comprises lunar irradiance values for a plethora of conditions and geometries.
3. *Aligning Observations with LIME Data:* Each lunar observation needs to be corresponded with the appropriate data set in the LIME model. This process entails matching the illumination conditions and viewing angles of the Sun-Moon-Earth system with our observations.
4. *Computing Calibration Coefficients:* Once a match has been established between the observations and LIME data, calibration coefficients can be computed for each spectral band of the hyperspectral imager. The calculation of these coefficients generally involves minimizing the difference between the observed radiance values and the reference radiance values derived from the LIME model.
5. *Applying Calibration Coefficients:* The final phase involves applying the computed calibration coefficients to the raw sensor data to obtain calibrated radiance or reflectance values.

Time-dependent effects and changes in geometry need to be taken into account during the calibration process. An integral part of this procedure is also the validation of the calibration results, usually achieved by comparing the calibrated data with known reference data or verifying its applicability in a well-established application.

In conclusion, the LIME model presents a sophisticated tool for conducting lunar calibration of hyperspectral imagers, thereby bolstering the accuracy and reliability of remote sensing data. By employing such calibration strategies, we are better equipped to understand and analyze Earth's diverse environments and their changes over time.

Chapter 3

Methods and Tools

Contents

3.1	Capturing the moon	18
3.1.1	Pixels needed across track	18
3.1.2	Frames needed along track	19
3.2	Python	21
3.2.1	Skyfield	21
3.3	Systems Tool Kit	22
3.3.1	Planning Moon Captures	22
3.3.2	Flight Scripts	23
3.3.3	Case Study: HYPSON-1 Calibration	25
3.4	Lunar Reflectance Models	25
3.4.1	Lunar Irradiance Model ESA (LIME)	25
3.4.2	Robotic Lunar Observatory (ROLO)	26
3.4.3	Other alternatives	26

3.1 Capturing the moon

In order to capture the entire moon in one single image, we need to make sure that the HSI's FoV across track is wide enough to capture the diameter of the moon at the Moon's distance and calculate the maximum frame rate in order to give the HSI enough time to capture the entire moon along track. Lastly, we must also calculate the minimum frame rate in order to ensure that we do not get undersampling¹.

3.1.1 Pixels needed across track

The number of pixels occupied by the moon in the across-track field of view of the hyperspectral imager can be calculated using the following formulas:

$$\text{Pixel size} = \frac{FoV}{\sum \text{pixels}} \quad (3.1)$$

$$\alpha_{moon} = 2 \cdot \tan^{-1} \left(\frac{\varnothing_{moon}}{2 \cdot d_{sat_moon}} \right) \quad (3.2)$$

¹Undersampling refers to capturing too few frames during a rotation of the camera, resulting in incomplete coverage of the moon's (or other object's) surface. This can lead to missing information and gaps in the final image, potentially resulting in reduced image quality and loss of details.

where

- FoV is the HSI's field of view
- α_{moon} is the angular size of the moon in the HSI's FoV
- \mathcal{O}_{moon} is the moons diameter
- d_{sat_moon} is the distance from the satellite to the moon

From this, we can calculate the number of pixels occupied by the moon in the across-track field of view by:

$$\frac{\alpha_{moon}}{\text{Pixel size}} \quad (3.3)$$

Using the values $d_{sat_moon} = 360\,000$ km, $\mathcal{O}_{moon} = 3\,474.8$ km, FoV = 4.00415° and \sum pixels across track = 684 pixels, we get

$$\begin{aligned} \text{Pixels needed across track} &= \frac{2 \cdot \tan^{-1} \left(\frac{3474.8km}{2 \cdot 360000km} \right)}{\frac{4.00415^\circ}{684}} = \\ &= \frac{0.553027303 \text{ deg}}{0.005854020468} = 94.4696565417 \approx 95 \text{ pixels} \end{aligned} \quad (3.4)$$

As the non-binned images captured by the satellite have 684 pixels across track, and our calculations tells us we need 95 pixels to capture the entire moon, this means we are in the clear.

3.1.2 Frames needed along track

In the along-track direction, there are multiple factors at play. In the non-binned capture mode, which will be the main capture mode used to capture images of the moon, the satellite is in a constant pointing direction with regard to the earth's core. This means that in the span of one orbit, the satellite rotates 360° around its axis. With an orbit time of 95 minutes, this translates to a rotational speed of

$$\omega = \frac{360^\circ}{95 \text{ min} \cdot 60 \frac{s}{min}} = 0.063 \text{ }^\circ/s \quad (3.5)$$

The HSI is not always pointing in the direction which gives the extra rotational speed calculated in equation 3.5, as the off-nadir² angle of the Moon's position varies throughout the moon phase, and depending on where the satellite is in its

²Off-nadir describes the angle at which a satellite or airborne sensor deviates from directly pointing towards the nadir, or the zenith directly below the sensor - in other words, it refers to the angle at which the sensor is looking away from the vertical or straight down position.

orbit during the capture. The extra rotational speed of the HSI can therefore be calculated by

$$\omega_{hsi} = \left(\frac{360^\circ}{95 \text{ min} \cdot 60 \frac{s}{min}} \right) \cdot -\cos(\alpha_{\text{off_nadir}}) \quad (3.6)$$

The satellite also moves with a constant speed of 7.6 km/s along track. Ignoring the rotational speed of the satellite, the HSI would need to travel approximately the distance of the moon's diameter to capture the entire moon in one image. With the moons diameter $\varnothing_{moon} = 3474.8 \text{ km}$ this gives a total capture time = $\frac{3474.8 \text{ km}}{7.6 \text{ km/s}} = 457.2$ seconds.

As calculated in equation 3.4, the Moon occupies $\approx 0.55^\circ$ of the FoV. As the off-nadir angle varies between each capture, we must use this as a variable to calculate the maximum frame rate. The total capture time along-track can be calculated by

$$t = \frac{1}{\sqrt{\left(\frac{v_{sat}}{\varnothing_{moon}}\right)^2 + \left(\frac{\omega_{hsi}}{\alpha_{moon}}\right)^2}} \quad (3.7)$$

where

- v_{sat} is the linear speed of the satellite
- \varnothing_{moon} is the diameter of the moon
- ω_{hsi} is the rotational speed of the HSI
- α_{moon} is the angular size of the moon in the HSI's FoV

The shortest possible scan time in the along-track direction is when the off-nadir angle equals 180° . From this we get the HSI's rotational speed $\omega = 0.063^\circ/\text{sec}$, as shown in equation 3.5 , which tells us that it takes

$$t = \frac{1}{\sqrt{\left(\frac{v_{sat}}{\varnothing_{moon}}\right)^2 + \left(\frac{\omega_{hsi}}{\alpha_{moon}}\right)^2}} = \frac{1}{\sqrt{\left(\frac{7.6 \text{ km/s}}{3474.8 \text{ km}}\right)^2 + \left(\frac{0.063^\circ/\text{s}}{0.55^\circ}\right)^2}} = 8.73 \text{ s} \quad (3.8)$$

to scan the entire moon in the along-track direction. In the non-binned capture mode, the satellite captures 106 frames along track. Therefore, the maximum frame rate allowed to keep the entire moon in one capture is $\text{FPS}_{max} = \frac{\text{frames}}{\text{scan time}} = \frac{106}{8.73} = 12.14 \text{ FPS}$.

The longest possible scan time in the along-track direction is when the off-nadir angle equals 90° . From equation 3.6 we get the HSI's rotational speed

$$\omega_{hsi} = \left(\frac{360^\circ}{95 \text{ min} \cdot 60 \frac{s}{min}} \right) \cdot -\cos(90^\circ) = 0 \text{ }^\circ/\text{s} \quad (3.9)$$

which in turn gives the total capture time from equation 3.7

$$t = \frac{1}{\sqrt{\left(\frac{7.6 \text{ km/s}}{3474.8 \text{ km}}\right)^2 + \left(\frac{0^\circ/\text{s}}{0.55^\circ}\right)^2}} = 457.2s \quad (3.10)$$

There are however some limitations to the capture configurations in the HYPSON-1 satellite. One particular limitation that affects this case is that we can only set the frame rate to whole numbers between 1 and 22. As the number of frames in non-binned mode is 106, the longest possible capture time is 106 seconds, with FPS = 1. We, therefore, set a boundary on the total capture time when planning moon captures at an off-nadir angle of 95.5° and a rotational speed of $0.006 \text{ }^\circ/\text{s}$, which gives a total capture time of approximately 90 seconds for good measure.

To avoid missing any details, it is necessary to calculate the least FPS to ensure pixel overlap. Given the distance of 360 000 km and a vertex angle of 0.0573 degrees, which equates to the HSI's FoV in the along-track direction, the swath width calculates to 354.4 km. Assuming the frames are captured in real-time, we would need about 10 frames ($\varnothing_{moon}/\text{swath width} = 3474.8/354.4 = 9.8 \approx 10$ frames) to cover the entire moon without missing any information. When the satellite is at its highest effective capture speed, particularly when the off-nadir angle is 180 degrees and the rotational speed $\omega_{sat} = 0.063^\circ/\text{s}$, the total time required for capturing is around 8.73 seconds as shown in equation 3.8. Hence, to prevent undersampling, if the total capture time is under 10 seconds, we require at least 2 frames per second, and at least 1 frame per second if the total capture time extends over 10 seconds.

3.2 Python

In order to simplify the process of planning moon captures, we want to develop a tool that can do this for us. Python [21] offers several tools that we can use to do this, which we will explain in this section. Finally, we will discuss how we have combined these tools to create a program that automatically generates the necessary parameters for precise moon captures.

3.2.1 Skyfield

Skyfield is a powerful Python library that provides precise and comprehensive positional data for various celestial bodies, such as the Earth, Sun, Moon, earth-orbiting satellites, and other stars and planets. [16] Developed by programmer and amateur astronomer, Brandon Rhodes, Skyfield can be used in scientific research, astronomy, and astrophysics for calculating distances, angles, and other positional parameters.

Features and Functionality

Skyfield offers a rich set of features and functionality for working with celestial body positional data. It provides an extensive database of accurate ephemeris data, including precise positions, velocities, and accelerations for celestial bodies, based on high-quality observational data. Its results should agree with the positions generated by the United States Naval Observatory [22] and their Astronomical Almanac [23] to within 0.0005 arcseconds [16]³, or within approximately 0.000000139 degrees.

Skyfield can be used to calculate pointing vectors for captures based on satellite and moon positions. This can be done by retrieving the satellite and moon positions for a specified time, and then using mathematical operations to calculate the relative orientation or pointing direction in the three-dimensional space. Using TLE data gathered from reliable sources such as CelesTrak [24], Skyfield can accurately pinpoint the location of any registered satellite with its latest updated trajectory.

3.3 Systems Tool Kit

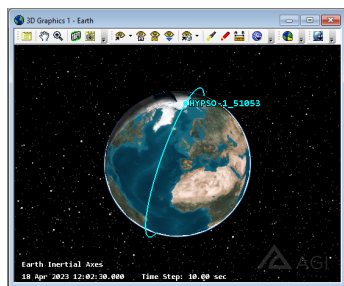
STK stands for "Systems Tool Kit" and is a suite of software tools for modelling, simulating, and analyzing a wide range of systems and environments. The STK software provides a comprehensive set of capabilities for modelling, visualizing, and analyzing the performance of complex systems over time. This can help organizations make informed decisions and optimize their systems' performance.

To achieve our goal of automating planning of precise pointing towards the moon, STK is a very useful tool to verify our results along the way. By inserting the HYPSON-1 satellite with its latest TLE we can simulate the orbit of the satellite around the earth, as well as the position of the sun, moon and other celestial bodies, in real-time. Using STK's built-in features we can visualize the HSI's FoV and the relative vectors between the satellite and the moon. We can also jump to a specific moment and observe the predicted positions at that point in time. Visualizing the direct vector between the satellite and the moon gives the desired pointing direction. By rotating the satellite using the quaternions we calculate, we can see how well the FoV matches with this vector.

3.3.1 Planning Moon Captures

Planning moon captures should be a fast and simple process. The current solution consists of visualizing body axes in STK, and manually picking a time when the moon will cross the y-z plane of HYPSON-1 at some time t . One must then manually set the azimuth and elevation angles of the HSI through trial and error until it is within an acceptable range of the moon vector. We then note the time and use a script to calculate the quaternions based on our azimuth and elevation angles. These can then be fed into the Mission Control Software (MCS) which transfers the capture scripts to the satellite.

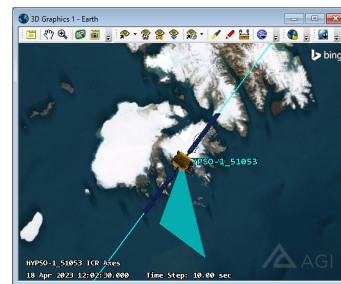
³An arcsecond is a unit of angular measurement, where one arcsecond is equal to 1/3600th of a degree.



(a) 3D Overview in STK



(b) 2D Overview in STK



(c) 3D Zoomed-in View in STK

Figure 3.1: Three Images showing the Satellite Orbit and Field of View (FoV) at April 18th 2023, 12:02:18. In image (c) we see the satellite above Svalbard, Norway with its HSI pointed at 0° off-nadir.

Our goal is to simplify this process. Preferably, the user should just have to specify what range of time they want a capture to be planned and how many captures they want. The program should then find the best fit(s) based on criteria such as off-nadir angle and distance to the moon, as well as ensure that the earth is not blocking the view and that the sun is not too close to the HSIs or startrackers FoV to keep from blinding the sensors. The program should then output all the necessary parameters in a simple format that can be fed directly into the next step of planning, which is the MCS process. The program should both be an executable that a user can manually run on their local computer, as well as importable from other scripts that may want to make use of the program's capabilities. In addition to the automation for HYPSON-1 Moon captures planning functionality, the program should accept arguments to plan pointing quaternions for other satellites to point towards other celestial bodies.

3.3.2 Flight Scripts

Flight Scripts is a collection of scripts developed by NTNU-SmallSat-Lab [25] that are helpful and/or necessary to generate scripts and control the Flight Computer (FC), Payload Controller (PC) and Hypso Command Line Interface (Hypso-Cli). In our case, we will mainly take advantage of the scripts `script_generator.py` and `script_series.py`.

The `script_generator.py` is a script developed by NTNU-SmallSat-Lab that is used to generate PC and FC scripts based on the provided script arguments. The mandatory options that need to be provided are:

- **-u/--unix**: Capture start time in Unix time format. This option needs to be provided at least once and can be provided twice with **-u** and **-u2** in case of dual capture.
- **-p/--pointing**: Pointing mode for the script. It can take one of the following values: generic, quaternion, nadir, slew, generic_slew, nonbinned, fullframe, maxspatial, wide.

- **-a/--append**: Whether to append to an existing script.

Additionally, several optional arguments can be provided to customize the generated scripts:

- **-e**: Exposure time in milliseconds.
- **--se**: Selection of which PC script engine to run.
- **-n/--name**: Name of the hsi-folder.
- **-b/--buff_file**: File ID of buffer file.
- **-r/--r**: Specifies the quaternion component 'r' for the primary target pointing direction.
- **-l/--l**: Specifies the quaternion component 'l' for the primary target pointing direction.
- **-j/--j**: Specifies the quaternion component 'j' for the primary target pointing direction.
- **-k/--k**: Specifies the quaternion component 'k' for the secondary target pointing direction in case of dual capture mode.
- **-d**: Delaying buffering of metazip and hsi-cube by 20min if capture is in a pass.
- **-s**: Flag for enabling the star tracker.
- **-fps**: Frame rate in frames per second.
- **-lat/--latitude**: Latitude of the capture location to be included in the metadata.
- **-lon/--longitude**: Longitude of the capture location to be included in the metadata.
- **-fr/--frames**: Number of frames to capture.

The generated scripts will depend on the provided arguments and can be used to control the PC and FC in the desired pointing mode and capture start time.

The generated scripts must then be updated on the MCS, and a new upload sequence must be manually enabled. The MCS then ensures that the new scripts are uploaded on the next satellite pass.

3.3.3 Case Study: HYPSON-1 Calibration

In this study, we aim to apply similar techniques and methods used in the commissioning of the Sentinel-3B satellite during a single lunar observation [26]. The lunar observation allowed the assessment of the possible use of the moon for characterizing and validating onboard instruments. The stable orientation of the satellite after a roll manoeuvre enabled imaging of the moon by the Earth view of instruments. The data acquired by the Ocean Land Color Instrument (OLCI) facilitated the verification of stray-light correction (SLC) performed by the Mission Performance Centre (MPC) and assessment of radiometric behaviour of the instrument in comparison with lunar irradiance models developed in collaboration between European Space Research and Technology Centre (ESTEC) and European Organisation for the Exploitation of Meteorological Satellites (UEMETSAT).

In our experiments, we plan to replicate the approach used in this paper by performing similar lunar observations and utilizing the acquired data to assess the performance of our onboard instruments. We will use the proposed update of stray-light correction developed with the use of lunar data as a reference to evaluate the accuracy and effectiveness of our own stray-light correction techniques.

3.4 Lunar Reflectance Models

Lunar reflectance models represent our understanding of how the Moon reflects sunlight and other incident light. They are created using a combination of direct observations from lunar missions, earth-based telescopic measurements, and theoretical calculations of reflectance properties. These models attempt to capture the reflectance properties across the Moon's surface, taking into account factors such as the lunar phase, lunar libration, and the position of the observer.

The use of these models is particularly beneficial for the calibration of satellite sensors. Satellite sensors, especially those operating in deep space, experience a different environment than the one they were built and tested in, causing discrepancies between the observed and actual values. By observing a known target, such as the Moon, it's possible to adjust sensor readings to be more accurate. The Moon, as an extraterrestrial reference source, is a constant and predictable target whose reflectance characteristics are well-studied and understood.

Stray light, i.e., unwanted or scattered light in an optical system, is a significant concern in satellite sensor calibration. Although it can distort sensor readings, the stable and predictable nature of lunar reflectance allows us to separate the signal of interest from the stray light, thus providing a correction factor that can be used for more accurate measurements.

3.4.1 Lunar Irradiance Model ESA (LIME)

The European Space Agency's Lunar Irradiance Model, also known as LIME, represents a modern, high-accuracy approach to lunar reflectance modelling. More detailed information can be found on the dedicated CEOS portal and associated

documentation [27, 28].

LIME is based on extensive sets of high-resolution images and radiometric measurements of the Moon, capturing a wide range of lunar phases, libration states, and solar illumination angles. The result is a model that can generate realistic and highly accurate spectral radiance maps of the Moon for any given observation geometry.

LIME is built upon several models including the empirical model from the U.S. Geological Survey’s Robotic Lunar Observatory (ROLO), but it further enriches this with data from other lunar observation campaigns and advanced modelling techniques.

The calibration process using LIME involves comparing the observed lunar irradiance with the one predicted by the model. The ratio of these two quantities provides the correction factor that needs to be applied to the sensor readings. Regular comparisons between sensor observations and the model allow for continuous calibration and performance monitoring over the sensor’s lifetime.

3.4.2 Robotic Lunar Observatory (ROLO)

The U.S. Geological Survey’s ROLO is another significant player in lunar reflectance modelling. In over 6 years of operations, ROLO has acquired over 85 000 images of the Moon [29], generating an extensive database of lunar images and spectra that can be used for sensor calibration. ROLO’s model focuses on accounting for variations in lunar reflectance caused by changing viewing geometry, lunar phase, and lunar libration.

3.4.3 Other alternatives

There are several other methods for sensor calibration, including the use of stars, asteroids, and other celestial bodies. However, these sources lack the constant and predictable nature of the Moon, making them less suitable for routine calibration.

Artificial light sources on Earth can also be used for calibration, but they suffer from atmospheric distortion and are not as consistent or predictable as the Moon.

In conclusion, lunar reflectance models, including LIME and ROLO, are valuable tools for the calibration of satellite sensors. The constant and predictable nature of lunar reflectance, combined with sophisticated models that accurately capture these properties, provides a powerful method for sensor calibration and ongoing performance monitoring.

Chapter 4

Work and Results

Contents

4.1	Celestial Bodies	27
4.1.1	Distances	28
4.1.2	Distance from Object to Target	28
4.1.3	Get Minimum Distance	29
4.1.4	Angles	30
4.1.5	Get Off-Nadir Angle	30
4.1.6	Get Maximum Off Nadir Angle	32
4.1.7	Get Lunar Zenith Angle	33
4.2	Pointing	34
4.2.1	Get Quaternion	34
4.2.2	ECI to LVLH Transformation	35
4.2.3	Rodrigues Rotation	36
4.2.4	Rotation to Quaternion Conversion	37
4.3	Timing	38
4.3.1	Total Capture Time	38
4.3.2	Calculate Frames per Second	38
4.4	Execution	38
4.4.1	SatNav	39
4.4.2	Moon Capture Planning for HYPSON-1	40
4.5	Observing the moon	42
4.5.1	Planning captures	42
4.5.2	Obtaining captures	43
4.6	Captures	43
4.6.1	Successful captures	44
4.6.2	Unsuccessful captures	47
4.6.3	Analysis	52

4.1 Celestial Bodies

In order to efficiently and accurately calculate the distances, angles and velocities between the different celestial bodies, we need to develop scripts that can do this for us given a few simple inputs. To do this, we mainly take advantage of the Python library Skyfield [16] as discussed in chapter 3.2.

4.1.1 Distances

Distances between objects in space are a critical aspect of space missions and astronomical observations. However, determining distances in space is not as straightforward as it may seem, as they are relative to the reference point or system in which they are measured. The International Celestial Reference System (ICRS) and J2000 epoch are commonly used as reference systems in astronomy and space exploration. Additionally, there are different types of distances to consider, including barycentric, astrometric, and apparent distances. Furthermore, different coordinate systems are used to represent the position and motion of objects in space, including the Earth-Centered Inertial (ECI) and Earth-Centered, Earth-Fixed (ECEF) coordinates. Understanding these concepts is crucial for accurately measuring distances and predicting the motion of objects in space.

International Celestial Reference System (ICRS)

The International Celestial Reference System (ICRS) is the current standard celestial reference system, with its centre at the barycenter¹ of the solarsystem. Using `Skyfields at()` method on celestial bodies we get the barycentric position of the body with reference to the solar systems barycenter in either Astronomical Units (AU), kilometres or meters.

EarthSatellite and Observers

Using TLE files and `Skyfields api.load` package [31], we can import an earth-orbiting satellite as an `EarthSatellite` object [32]. The position returned from `Skyfields at()` method on satellites is an XYZ (3D Cartesian) system relative to the Earth's centre in the Geocentric Celestial Reference System (GCRS).

Using `Skyfields observe()` method, we are able to get the relative distance from an observer, i.e. a celestial body, to a target, i.e. another celestial body. Using the Earth as the observer, we get the distance to the target in GCRS coordinates. This way the observer, target and orbiting object are all in the same coordinate system with the earth's core at its centre. From this, we can easily calculate distances, angles and relative velocities over time.

4.1.2 Distance from Object to Target

The `distance_obj_to_target()` function is used to compute the linear distance between an orbiting object and a target at a given time `t`. The function takes four arguments, namely `time`, `satellite`, `target` and `observer`. The function returns the distance between the satellite and the target in kilometres as a floating-point number.

To compute the distance between the objects, the function first calculates the position of the `target` as observed from the `observer` at the given time `t`. This is done using the `observe()` method of the observer object from the `Skyfield` library. Next,

¹Barycenter is the centre of mass of two or more bodies that orbit one another and is the point about which the bodies orbit. [30]

the function computes the position of the `satellite` at the same time `t` using the `at()` method of the satellite object. Once the positions of the target and the satellite are known, the function calculates the distance between the two objects using the `np.linalg.norm()` function. The `np.linalg.norm()` function calculates the Euclidean distance between the two position vectors, which gives the linear distance between the orbiting object and the target.

This function is particularly useful when combined with the `get_minimum_distance()` function, which searches for the time when the distance between two celestial bodies is at its minimum within a specified time frame.

4.1.3 Get Minimum Distance

The `get_minimum_distance()` function is used to find the time when the distance between an object and a target is at its minimum within a specified timeframe. This function takes five arguments: `t_start`, `t_end`, `satellite`, `target`, and `observer`. An optional argument, `search_interval`, can also be specified to set the time step in minutes. The function returns two values, the minimum distance in kilometres as a float, and the time of the minimum distance as a `DateTime` object.

The initial minimum distance is calculated at `t_start` using the `distance_obj_to_target()` function as explained above and stores it as `min_d`. The variable `min_t` is set to `t_start` to store the time at which the minimum distance occurs.

The total number of iterations is computed based on the specified timeframe and search interval. Then, the function enters a loop that iterates through each time step between `t_start` and `t_end` with the specified search interval.

Inside the loop, the function calculates the distance at the current time step using the `distance_obj_to_target()` function. If the calculated distance is smaller than the current minimum distance, the minimum distance and its corresponding time are updated.

After the loop completes, the variable `max_t` is converted to a UTC `DateTime` object. Finally, the function returns the minimum distance in kilometres and the time of its occurrence as a `DateTime` object.

In figure 4.1 we see the distances between the HYPSON-1 satellite and the moon, at one-minute intervals, over 4 hours on May 26th 2023. The minimum distance per orbit is marked in red.

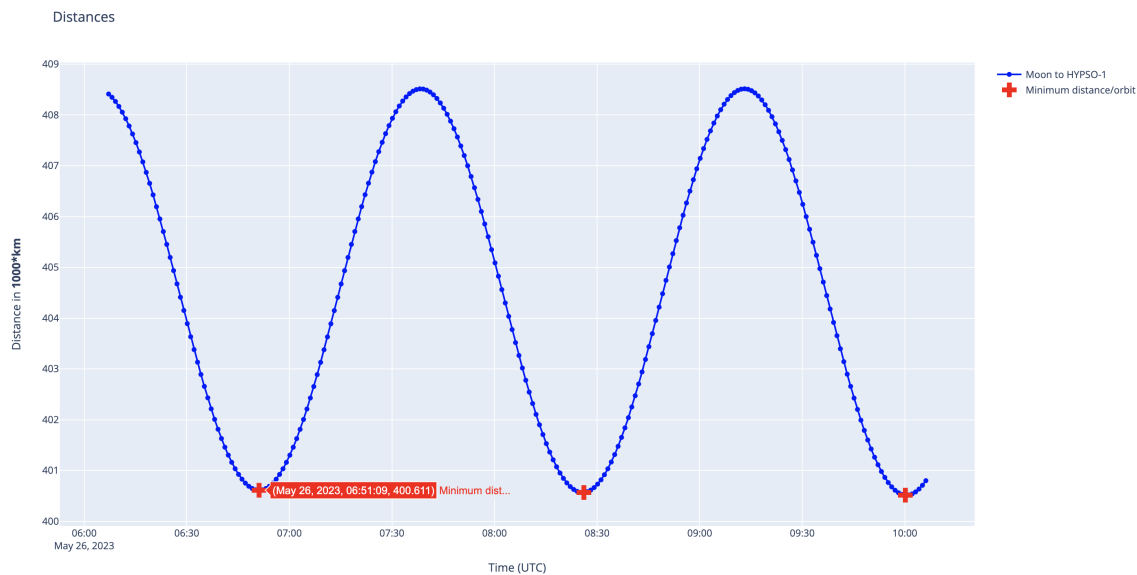


Figure 4.1: Hypso-moon distance four hours on May 26th 2023. The minimum distances are marked in red.

4.1.4 Angles

As important as the distance between the orbiting object, target and observer are, the angles between them are equally important. In the span of one orbit, the satellite should have its maximum off-nadir angle at the same point in time as when the distance between the satellite and the target is at its minimum. Calculating the off-nadir angle is therefore an efficient way to ensure that the minimum distance calculated occurs at the correct time. The lunar zenith angle is also important to calculate, as it is an important input in the calibration process.

4.1.5 Get Off-Nadir Angle

The `get_off_nadir_angle()` function calculates the off-nadir angle between a satellite and a target at a given time. It takes four arguments: `time`, `earth`, `target`, and `sat`.

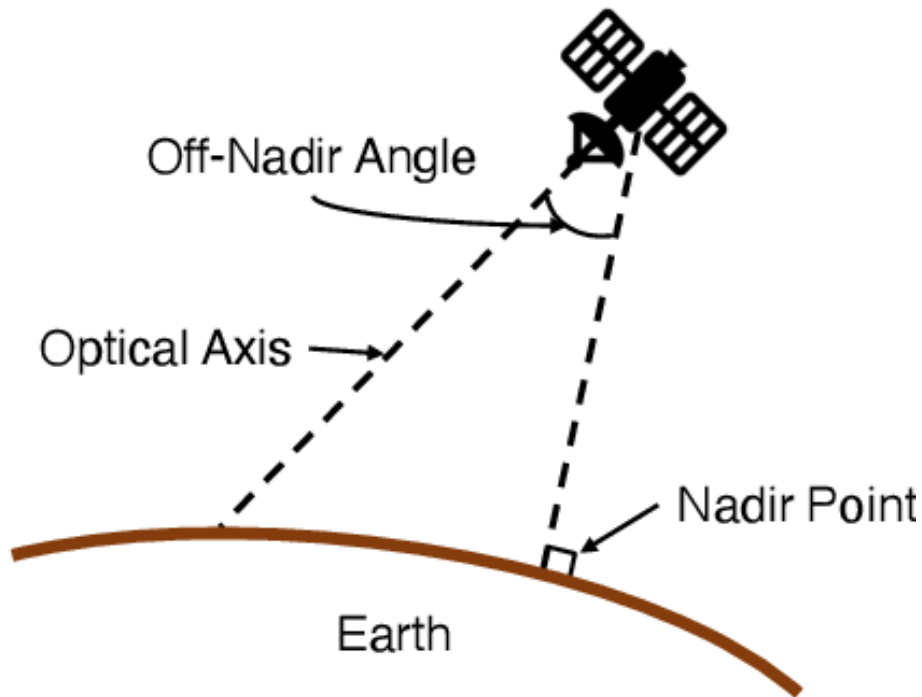


Figure 4.2: Visualization of the off-nadir angle. 0° off-nadir is the vector pointing at the earth's core (nadir point), perpendicular to the earth's surface. The higher the off-nadir angle, the more away from the earth's core the optical axis direction. From [33].

First, the function obtains the positions and velocities of the satellite and the target at the specified time using the `get_positions()` and `get_velocity()` functions.

Next, the function performs coordinate transformation from the Earth-Centered Inertial (ECI) frame to the Local-Vertical-Local-Horizontal (LVLH) frame. It uses the `eci2LVLH()` function to calculate the LVLH frame's position vector (\mathbf{r}_o), velocity vector (\mathbf{v}_o), and transformation matrix (\mathbf{R}_{io}).

The relative position vector between the target and the satellite is calculated by subtracting the satellite's position from the target's position. This relative position vector is then transformed into the orbit frame using the transformation matrix \mathbf{R}_{io} .

To compute the off-nadir angle, the function normalizes the relative position vector in the orbit frame by dividing it by its magnitude using the `np.linalg.norm()` function. The nadir vector in the orbit frame ($\mathbf{z}_{o_hat_o}$) is defined as $[0, 0, 1]$. The cosine of the off-nadir angle is calculated by taking the dot product between the target unit vector and the nadir vector. Finally, the off-nadir angle is obtained by taking the arccosine of the cosine and converting the result from radians to degrees.

The function returns the off-nadir angle in degrees.

4.1.6 Get Maximum Off Nadir Angle

Similarly as for the `get_minimum_distance()` function explained above, the `get_maximum_off_nadir_angle()` function is used to find the time when the off-nadir angle between a satellite and a target is at its maximum within a specified timeframe. It takes five arguments: `t_start`, `t_end`, `obj`, `target`, and `observer`. Additionally, it has an optional argument `search_interval` to specify the time step in minutes. The function returns two values, the maximum off-nadir angle in degrees, and the time of the maximum off-nadir angle as a `DateTime` object.

The function performs similarly to the `get_minimum_distance()` function, by iteratively calling the `get_off_nadir_angle()` for each time step between `t_start` and `t_end` with the given time step specified by `search interval`. At each step, the function checks if the off-nadir angle is greater than the previously computed maximum off-nadir angle, and if so, updates the maximum off-nadir angle and the corresponding time.

In figure 4.3 we see the off-nadir angles between the HYPSON-1 satellite and the earth, at one-minute intervals, over four hours on May 26th 2023. The maximum off-nadir angle per orbit is marked in red.

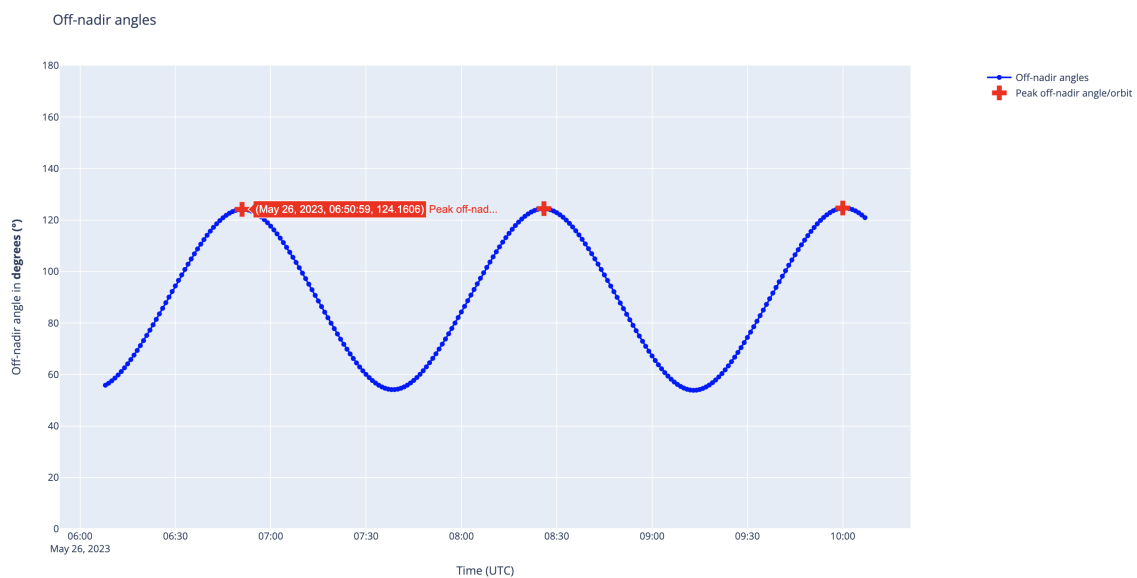


Figure 4.3: Off-nadir angles over four hours on May 26th 2023. Maximum off-nadir angles are marked in red.

In figure 4.4 we see both the distances between the HYPSON-1 satellite and the moon and the off-nadir angles between the HYPSON-1 satellite and the earth, at one-minute intervals on May 26th 2023. As expected, the peak off-nadir angles occur at the same point in time as the minimum distance between the satellite and the moon.

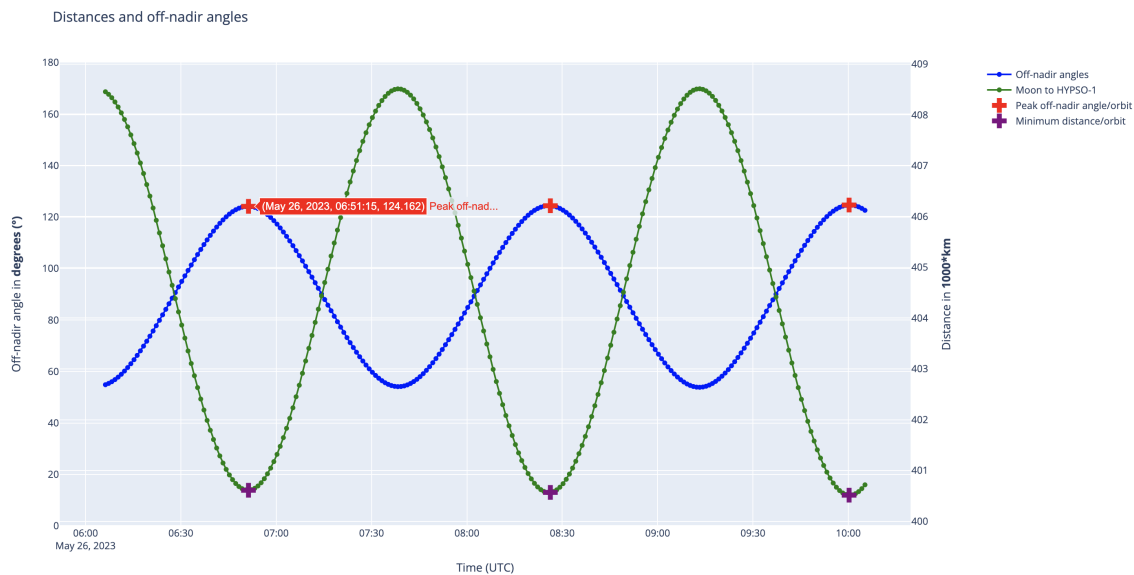


Figure 4.4: Hypso-moon distance and off-nadir angles over two orbits. Maximum off-nadir angles are marked in red, and minimum distances are marked in purple.

4.1.7 Get Lunar Zenith Angle

The lunar zenith angle is important to note for each capture, as it is an important aspect of the calibration process. The lunar radiance varies greatly depending on the lunar phase, and to determine the expected radiance versus the captured radiance, we need to classify the lunar phase at capture.

Using the position of the moon, earth and sun, acquired by the same methods as previously discussed in this chapter, we can calculate the lunar zenith angle. The angle is simply found by calculating the dot product between the sun-earth and sun-moon vectors, calculating the magnitudes of these vectors, and finally calculating the angle using the arccosine function. The angle for each capture is thus readily available for each capture at any point in time.

In figure 4.5 we see the variety of lunar reflectances measured at 440 nm from the 1088 and 933 instruments used to develop the LIME model.

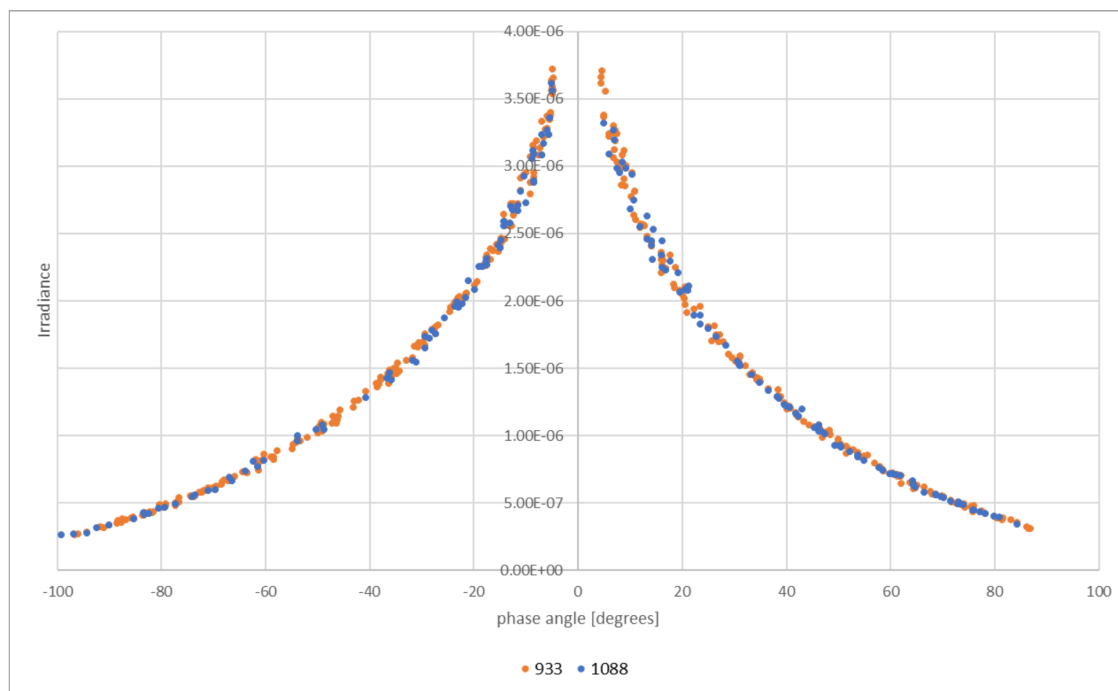


Figure 4.5: Lunar irradiance measurements for 440 nm from the 1088 and 933 instruments [34].

4.2 Pointing

For HYPSON-1, we employ quaternion-based pointing methodology due to its advantages in representing and manipulating orientation in three-dimensional space, as discussed in section 2.4.1. Quaternions provide a compact and computationally efficient representation, avoiding common issues such as singularities that can occur with other orientation representations such as Euler angles. Quaternion-based pointing allows for smooth and continuous rotations, enabling precise control and flexibility in aligning the satellite with desired targets.

In this section, we delve into the functions developed to calculate the quaternions required for our satellite’s pointing. These functions involve various mathematical transformations and calculations, enabling us to determine the optimal orientation of the satellite relative to the Earth, target objects, and reference frames. By understanding and implementing these functions, we gain the ability to accurately determine the necessary quaternion parameters for achieving desired pointing configurations.

4.2.1 Get Quaternion

The `get_quaternion()` function is designed to calculate the rotation quaternion that represents the desired orientation of a satellite with respect to a target at a given time. The function takes four arguments: `time`, `earth`, `target`, and `sat`.

To begin, the function obtains the positions of the satellite (`sat_pos`) and the target

(`target_pos`) using the `get_positions()` function. The velocity of the satellite at the specified time is also acquired (`sat_vel`).

Next, the function computes the relative position vector from the satellite to the target by subtracting the satellite's position from the target's position. This relative position vector is then transformed into the orbit frame using the transformation matrix `R_io` calculated from the satellite's position and velocity, using the `eci2LVLH` function described below.

The function normalizes the relative position vector to obtain the target unit vector. It then calculates the cosine of the off-nadir angle by taking the dot product of the target unit vector and the nadir vector in the orbit frame (`z_o_hat_o`). The off-nadir angle is computed in degrees using the inverse cosine function.

Finally, the function calculates the rotation quaternion by constructing the rotation matrix `R_ob` that represents the rotation from the orbit frame to the body frame based on the off-nadir angle. This rotation matrix is then converted to a quaternion (`q_ob`) using the `rot2q()` function.

The function returns the rotation quaternion (`q_ob`) as the desired orientation of the satellite with respect to the target.

4.2.2 ECI to LVLH Transformation

The `eci2LVLH()` function is responsible for converting the position and velocity vectors from the Earth-Centered Inertial (ECI) frame to the Local-Vertical-Local-Horizontal (LVLH) frame. The function takes two arguments: `r_i` (position vector in ECI frame) and `v_i` (velocity vector in ECI frame).

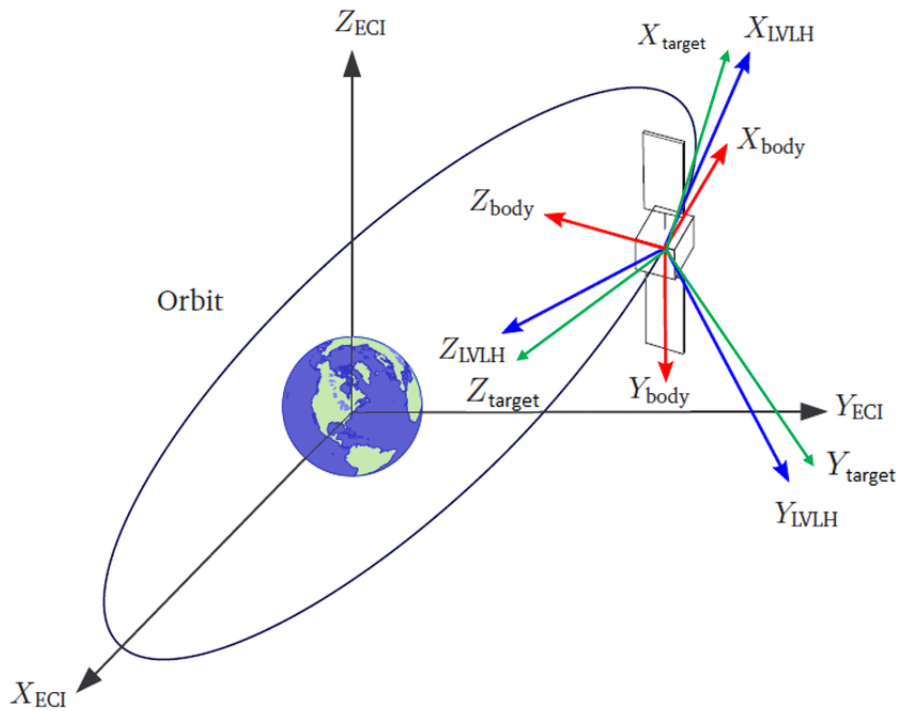


Figure 4.6: ECI and LVLH reference frames shown in black and blue, respectively. From [35].

To begin, the function calculates the unit vector along the negative radial direction (\mathbf{z}_o) by normalizing the position vector (\mathbf{r}_i). The unit vector along the negative cross-track direction (\mathbf{y}_o) is computed by taking the cross product of the position vector (\mathbf{r}_i) and the velocity vector (\mathbf{v}_i), and then normalizing it. The unit vector along the in-track direction (\mathbf{x}_o) is obtained by taking the cross product of the cross-track vector (\mathbf{y}_o) and the radial vector (\mathbf{z}_o).

Using these three orthogonal unit vectors, the function constructs the rotation matrix \mathbf{R}_{o_i} that represents the transformation from the ECI frame to the LVLH frame. The transpose of this rotation matrix (\mathbf{R}_{i_o}) is also computed for the reverse transformation.

The function applies the transformation to the position vector (\mathbf{r}_i) and the velocity vector (\mathbf{v}_i) using matrix multiplication. This results in the position vector (\mathbf{r}_o) and the velocity vector (\mathbf{v}_o) represented in the LVLH frame.

Finally, the function returns the transformed position vector (\mathbf{r}_o), velocity vector (\mathbf{v}_o), and the rotation matrix (\mathbf{R}_{o_i}) as the outputs.

4.2.3 Rodrigues Rotation

The `rot_rodrigues()` function implements the Rodrigues rotation formula to compute a rotation matrix that rotates a vector \mathbf{a} towards another vector \mathbf{b} by an angle

`theta`. This formula is commonly used in various geometric and computational applications. The function takes three arguments: `a` (initial vector), `b` (target vector), and `theta` (rotation angle in radians).

To start, the function normalizes the initial vector `a` and the target vector `b` by dividing them by their respective norms. Then, it calculates the cross product of the normalized vectors, resulting in the vector `lambda`. The norm of `lambda` is also computed.

Next, the function checks if the norm of `lambda` is below a certain threshold value (in this case, 1×10^{-12}). If it is, the function assigns `lambda` to `lambda_hat`. Otherwise, it divides `lambda` by its norm to obtain `lambda_hat`, the unit vector in the direction of `lambda`.

The function then computes the skew-symmetric matrix `skew` by multiplying `lambda_hat` with `theta` and passing it to the `skew_sym()` function. The `skew_sym()` function constructs the skew-symmetric matrix based on the given input.

Finally, the function calculates the rotation matrix `R` using the exponential matrix function (`expm()`) from the SciPy library applied to the `skew` matrix.

The resulting rotation matrix `R` represents the transformation required to rotate the initial vector `a` towards the target vector `b` by the specified angle `theta`.

4.2.4 Rotation to Quaternion Conversion

The `rot2q()` function is responsible for converting a rotation matrix `R` into a quaternion representation. Quaternions are commonly used to represent rotations in 3D space due to their compactness and computational advantages. This function takes a 3x3 rotation matrix `R` as its input.

The conversion process starts by computing the rotation angle `theta` using the arccosine function applied to the trace of the rotation matrix, subtracted by one, and divided by two. The trace of the matrix represents the sum of its diagonal elements. If the computed `theta` is very close to zero (using the `isclose()` function), the function sets the axis of rotation `e_hat` as the zero vector `[0, 0, 0]`. Otherwise, it calculates `e_hat` as the normalized vector `[R[1,2]-R[2,1], R[2,0]-R[0,2], R[0,1]-R[1,0]]` divided by twice the sine of `theta`.

Next, the function computes the quaternion components: `q_0` is set to $\cos(\theta/2)$, and `q_1`, `q_2`, and `q_3` are calculated by multiplying the respective components of `e_hat` with $\sin(\theta/2)$. The resulting quaternion `q` is represented as an array `[q_0, q_1, q_2, q_3]`.

Finally, to ensure the quaternion is normalized, the function divides `q` by its Euclidean norm using the `linalg.norm()` function.

4.3 Timing

Timing plays a crucial role in satellite imaging, particularly when targeting celestial bodies like the Moon. To optimize the capture of high-quality images and ensure efficient utilization of satellite resources, it is essential to consider the total capture time and the maximum Frames Per Second (FPS) achievable during imaging operations. In this section, we will explore the concepts of total capture time and calculate the frames per second required for effective imaging.

4.3.1 Total Capture Time

The total capture time refers to the duration required to capture a specific target or area of interest during satellite imaging. It is influenced by various factors such as the off-nadir angle, which represents the deviation from the satellite's nadir pointing direction. The off-nadir angle determines the extent of rotation needed to align the imaging sensor with the target, affecting the overall imaging time.

To calculate the total capture time for a given off-nadir angle, the function `calculate_total_capture_time(off_nadir_angle)` is utilized. This function takes the off-nadir angle as input and returns the total capture time in seconds. It considers parameters such as the satellite's linear speed, the Moon's field of view, and the Moon's diameter to estimate the time required for complete imaging coverage.

4.3.2 Calculate Frames per Second

In satellite imaging, the number of frames captured per second (FPS) is a crucial parameter that affects the temporal resolution of the acquired imagery. The higher the FPS, the more frames can be captured within a given time.

To determine the optimal frames per second for capturing the target within the calculated total capture time, the function `calculate_frames(total_time)` is employed. This function takes the total capture time as input and calculates the number of frames and the required frames per second. It considers a target number of frames and a target FPS as reference values and adjusts the FPS based on the total capture time to ensure an appropriate balance between temporal resolution and data volume.

By understanding the total capture time and calculating the frames per second, satellite imaging operations can be carefully planned and optimized to achieve desired results within the allocated resources and time constraints.

4.4 Execution

The execution of the program has been designed to be user-friendly and straightforward, requiring minimal input from the user. By providing the necessary parameters specific to their case, users can easily run the program and leverage the developed methods. To enhance usability, two executable scripts have been developed, each tailored to different functionalities. The first script, **SatNav**, is a general program

capable of searching for optimal capture times and calculating pointing quaternions for any registered satellite on a selection of celestial bodies. The second script focuses specifically on planning moon captures for the HYPSON-1 satellite. In this section, we will explore both programs, providing instructions on their usage and discussing their capabilities.

4.4.1 SatNav

The general program offers a versatile solution for users seeking to determine the best possible capture time(s) and calculate the corresponding pointing quaternions for their registered satellite. This program supports a range of selectable celestial bodies, allowing users to tailor the imaging operations according to their specific requirements. By inputting the relevant parameters, users can initiate the program and obtain optimized capture times and corresponding quaternions for precise satellite pointing.

```

StarNav -- zsh -- 94x53
% python3 src/main.py

SatNav

-----Satellite Targeting Tool-----
Enter the following information to configure the tool. Press enter to use default value.
Enter 1 to run in single planner mode, or 2 to run in multi planner mode (default is 1): 2
Enter satellite catalog number (default is 51053 (HYPSO-1)):
Enter target segment number (default is 301 (the moon). See README for supported bodies):
Enter hours in the future for start time of search (default is 0 (now)): 4
Enter hours in the future for end time of search (default is 24 (1 day from now)): 28
Enter number of intervals to search (default is end_time_delta/24 (one capture per day)): 2
Enter search_interval for minimum distance search (default is 1 (1 minute)):
Enter true to force update TLE data, or press Enter to skip:
2023-05-27 14:16:41 - INFO - Using satellite catalog number: 51053
2023-05-27 14:16:41 - INFO - Using start time: 2023-05-27 16:17:39 UTC
2023-05-27 14:16:41 - INFO - Using end time: 2023-05-28 16:17:39 UTC
2023-05-27 14:16:41 - INFO - Loading TLE file from local file...
301
2023-05-27 14:16:41 - INFO - Config: {
  "configured_at": "2023-05-27 14:16:41.484172",
  "last_pulled_tle": "2023-05-27 14:14:14.246325",
  "configured_by": "admin",
  "log_level": "INFO",
  "name": "HYPSO-1",
  "catnr": 51053
}
2023-05-27 14:16:41 - INFO - Epoch: HYPSO-1 catalog #51053 epoch 2023-05-27 03:12:11 UTC
2023-05-27 14:16:41 - INFO - Multi planner
2023-05-27 14:16:41 - INFO - Completed 0/2
2023-05-27 14:16:43 - INFO - Completed 1/2
2023-05-27 14:16:44 - INFO - -----
2023-05-27 14:16:44 - INFO - Capture nr. 1
2023-05-27 14:16:44 - INFO - Time = 2023-05-28 03:02:29.404865+00:00
2023-05-27 14:16:44 - INFO - Qx = 0.9223893944
2023-05-27 14:16:44 - INFO - Qy = 0.0110553420
2023-05-27 14:16:44 - INFO - Qz = 0.0000000000
2023-05-27 14:16:44 - INFO - Qs = 0.3861030750
2023-05-27 14:16:44 - INFO - Off-nadir angle = 134.5765248189 degrees

2023-05-27 14:16:44 - INFO - Capture nr. 2
2023-05-27 14:16:44 - INFO - Time = 2023-05-28 15:40:29.404865+00:00
2023-05-27 14:16:44 - INFO - Qx = 0.9350921796
2023-05-27 14:16:44 - INFO - Qy = 0.0179487505
2023-05-27 14:16:44 - INFO - Qz = -0.0000000000
2023-05-27 14:16:44 - INFO - Qs = 0.3539497960
2023-05-27 14:16:44 - INFO - Off-nadir angle = 138.5418129424 degrees

2023-05-27 14:16:44 - INFO - -----

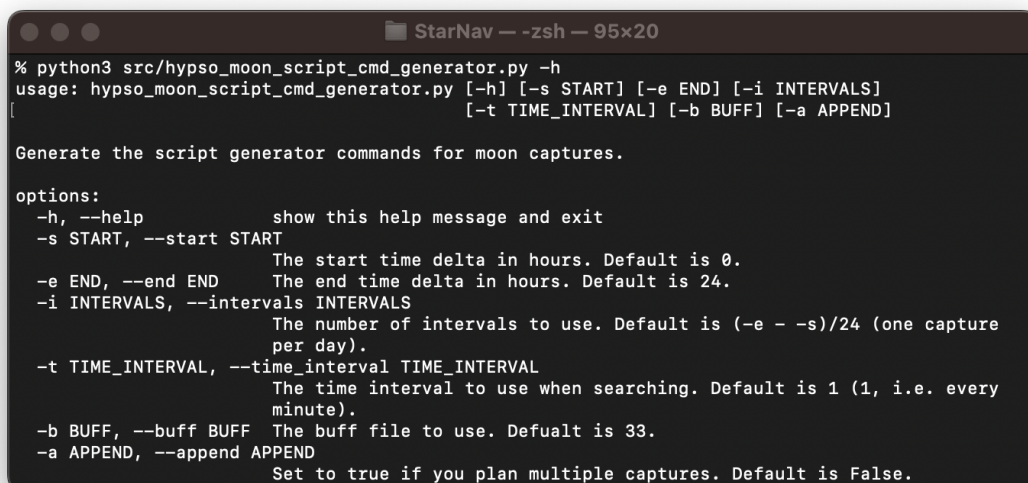
```

Figure 4.7: SatNav execution, specifically for two captures of the moon from the HYPSO-1 satellite, sometime between the next 4 and 28 hours, executed on May 27th 2023 at 14:16:41 UTC+2.

As shown in figure 4.7 above, the execution of the SatNav script provided two captures on May 28th, at 03:02:29.40 UTC and 15:40:29.40 UTC, respectively. The off-nadir angles of these captures are calculated to be at 134.58° and 138.54°, and the required pointing quaternions are also provided.

4.4.2 Moon Capture Planning for HYPSO-1

The second executable script is dedicated to planning moon captures specifically for the HYPSO-1 satellite. With a specialized focus on lunar imaging, this program streamlines the process of determining optimal capture times and generating pointing quaternions tailored to the unique characteristics of the satellite and the Moon. By utilizing this program, users can efficiently plan and execute imaging operations targeting the Moon with the HYPSO-1 satellite.



```

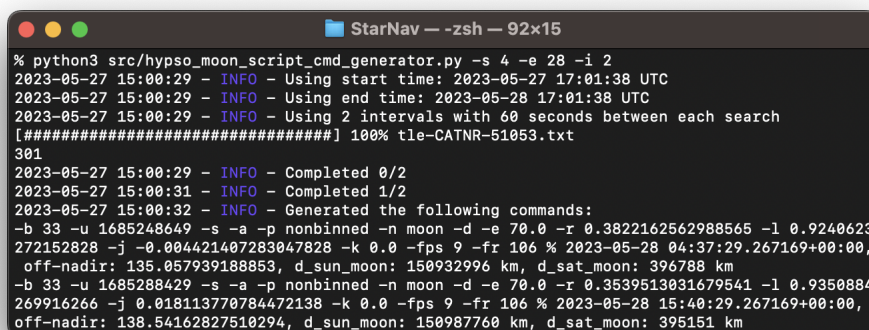
StarNav --zsh -- 95x20
% python3 src/hypso_moon_script_cmd_generator.py -h
usage: hypso_moon_script_cmd_generator.py [-h] [-s START] [-e END] [-i INTERVALS]
[-t TIME_INTERVAL] [-b BUFF] [-a APPEND]

Generate the script generator commands for moon captures.

options:
  -h, --help            show this help message and exit
  -s START, --start START
                        The start time delta in hours. Default is 0.
  -e END, --end END     The end time delta in hours. Default is 24.
  -i INTERVALS, --intervals INTERVALS
                        The number of intervals to use. Default is (-e - -s)/24 (one capture
                        per day).
  -t TIME_INTERVAL, --time_interval TIME_INTERVAL
                        The time interval to use when searching. Default is 1 (1, i.e. every
                        minute).
  -b BUFF, --buff BUFF  The buff file to use. Default is 33.
  -a APPEND, --append APPEND
                        Set to true if you plan multiple captures. Default is False.

```

Figure 4.8: Hypso Moon Script Command Generator help option, providing the necessary information about the usage of the script.



```

StarNav --zsh -- 92x15
% python3 src/hypso_moon_script_cmd_generator.py -s 4 -e 28 -i 2
2023-05-27 15:00:29 - INFO - Using start time: 2023-05-27 17:01:38 UTC
2023-05-27 15:00:29 - INFO - Using end time: 2023-05-28 17:01:38 UTC
2023-05-27 15:00:29 - INFO - Using 2 intervals with 60 seconds between each search
[#####] 100% tle-CATNR-51053.txt
301
2023-05-27 15:00:29 - INFO - Completed 0/2
2023-05-27 15:00:31 - INFO - Completed 1/2
2023-05-27 15:00:32 - INFO - Generated the following commands:
-b 33 -u 1685248649 -s -a -p nonbinned -n moon -d -e 70.0 -r 0.3822162562988565 -l 0.9240623
272152828 -j -0.004421407283047828 -k 0.0 -fps 9 -fr 106 % 2023-05-28 04:37:29.267169+00:00,
off-nadir: 135.057939188853, d_sun_moon: 150932996 km, d_sat_moon: 396788 km
-b 33 -u 1685288429 -s -a -p nonbinned -n moon -d -e 70.0 -r 0.3539513031679541 -l 0.9350884
269916266 -j 0.018113770784472138 -k 0.0 -fps 9 -fr 106 % 2023-05-28 15:40:29.267169+00:00,
off-nadir: 138.54162827510294, d_sun_moon: 150987760 km, d_sat_moon: 395151 km

```

Figure 4.9: Hypso Moon Script Command Generator, specifically for two captures of the moon from the HYPISO-1 satellite, sometime between the next 4 and 28 hours, executed on May 27th 2023 at 15:00:29 UTC+2.

As shown in figure 4.9 above, the execution of the Hypso Moon Script Command Generator provided two commands, each representing a capture on May 28th 2023, at 04:37:29.28 UTC and 15:40:29.27 UTC, respectively. The off-nadir angles of these captures are calculated to be 135.06° and 138.54° . The command is specifically formatted to suit the NTNU SmallSat Lab's tools for generating Payload Controller (PC) and Flight Computer (FC) scripts that are fed to the satellite. As discussed in 3.3.2, the command includes:

- *-b*: the file ID buffer file for the capture, here set to *33*.
- *-u*: Capture start time in Unix time format, here set to *1685248649* and

1685288429 respectively.

- *-s*: flag for enabling the star tracker.
- *-a*: Whether to append an existing script.
- *-p*: Pointing mode, here set to *nonbinned*.
- *-n*: Name of the hsi-folder, here set to *moon*.
- *-d*: Delaying buffering of metazip and hsi-cube by 20min if capture is in a pass.
- *-e*: Exposure time in milliseconds, here set to *70 milliseconds*.
- *-r, -l, -j and -k*: Specifies the pointing quaternion components.
- *-fps*: Frame rate in frames per second, here set to *9*.
- *-fr*: Number of frames to capture, here set to *106*.

Additionally, the timestamp, off-nadir angle, sun-moon distance and satellite-moon distance are printed along with the command for readability and easier analysis later in time.

4.5 Observing the moon

4.5.1 Planning captures

The process of planning captures is not as simple as generating commands from the above scripts and uploading them to the satellite. Multiple factors at play need to be accounted for. Among many others, some of these are:

- Two captures should not be planned within half an hour of each other, as the payload needs time both before and after a capture to boot, prepare, capture, buffer etc. If these processes are interrupted, the payload can reboot causing the images to be lost or corrupted. The payload is continuously booked to capture several locations on Earth, and we must therefore find capture times for lunar observations that match with these.
- There is a set number of buffer files available, and these need time to downlink before a new capture can be buffered to the same file.
- In case of bad space weather, or other significant events that affect the performance of the satellite, we might be forced to shut the payload down. In this case, we must plan the captures around these events.

4.5.2 Obtaining captures

Even if the planning procedure works out and the necessary scripts are uploaded to the satellite, we are not assured that the capture will be successful. Again, multiple factors are at play, some of which we are not yet able to predict or adjust for. Among many others, some of these are:

- *Time sync:* The Onboard Processing Unit (OPU) syncs time from PC, which syncs time from FC Real-Time Clock (RTC). The FC RTC is synced from Global Positioning System (GPS) time but is still running UTC/UNIX, so it does sync to the GPS to correct drift but is not synced to GPS time, which is approximately 18 s offset from UTC/UNIX time.
- *TLE offset:* The satellite updates its TLE lines every day. The TLE lines used for planning are not necessarily from the same day, as we plan multiple days ahead. This can cause an offset.
- *Script engine rounding error:* The script engine interprets UNIX time, but can only use integers, not floats. This could cause a rounding error to the nearest second.
- *Capture timing uncertainty:* The capture itself is run by a wrapper function in the HSI Capture software. There are many underlying functions in this software, as well as initialization delays etc., which can affect when a capture begins. The exact uncertainty of this is still unknown.
- *Attitude Determination and Control System (ADCS)* The actual pointing accuracy is also subject to the satellite platform ADCS functionality and performance, on which a further discussion is outside the scope of this thesis.

Ultimately the combination of all of these effects causes the pointing and timing to be somewhat uncertain. As the moon only obtains roughly 0.52° of the HSI's FoV, it is expected that the combined uncertainty will result in occasional misses or deviations in the lunar observations.

4.6 Captures

In this chapter, we present the captures obtained during our study, which aimed to automate the process of capturing images of the Moon using the hyperspectral imager on HYPSON-1. We provide a comprehensive analysis of both successful and unsuccessful captures, discussing the configurations employed for each capture and highlighting the factors that contributed to the outcomes. Additionally, we delve into the analysis of wavelength radiance and other relevant metrics to gain further insights into the captured data.

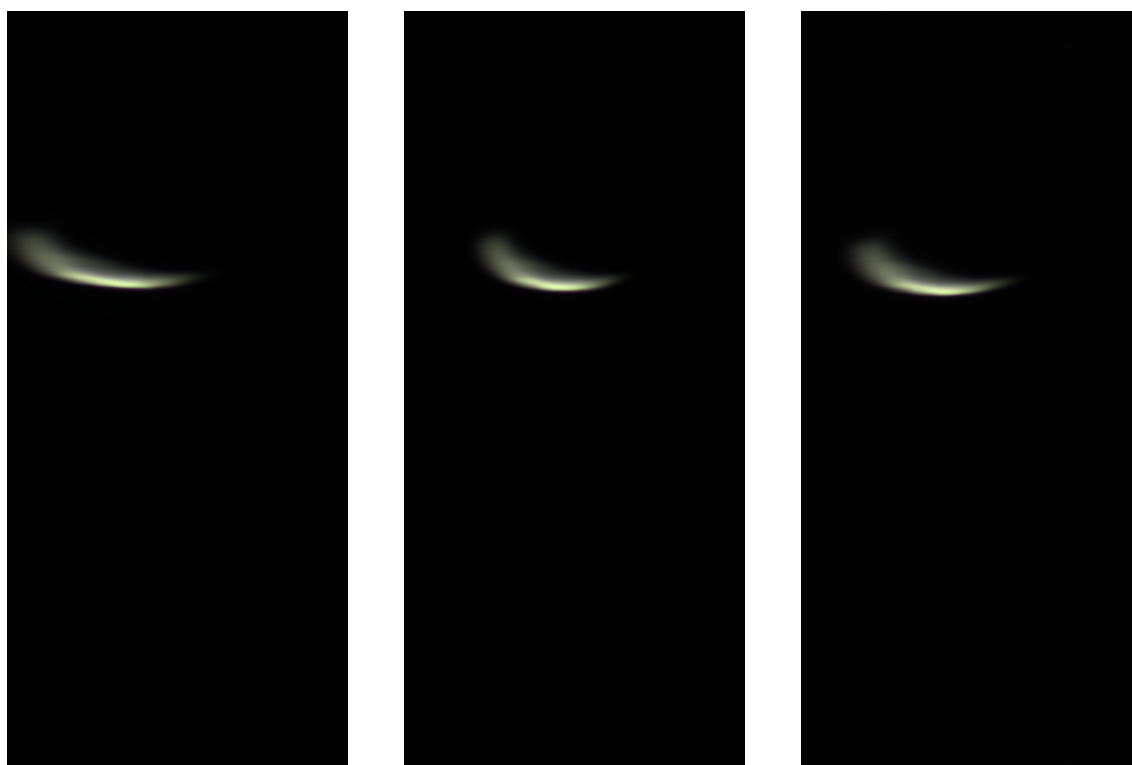
The section is organized as follows: In Subsection 4.6.1, we showcase the successful captures. This will be followed by Subsection 4.6.2, where we discuss the captures that did not yield the desired results, identifying the limitations and challenges

encountered. Finally, in Subsection 4.6.3, we perform a detailed analysis of the successfully captured data, focusing on wavelength radiance measurements, and any notable observations that arise from our findings.

4.6.1 Successful captures

The first successful captures were obtained on March 16th at 23:50 UTC+00, March 17th at 01:25 UTC+00 and 03:00 UTC+00. These are shown in figure 4.10 below.

During these captures, the moon-sun-earth angle (moon zenith angle) was -113.84° , -114.73° and -115.68° , respectively. The moon was at this point in its *Wanin Crescent* phase, meaning it is less than 50% illuminated, but has not yet reached 0% (*New Moon*).



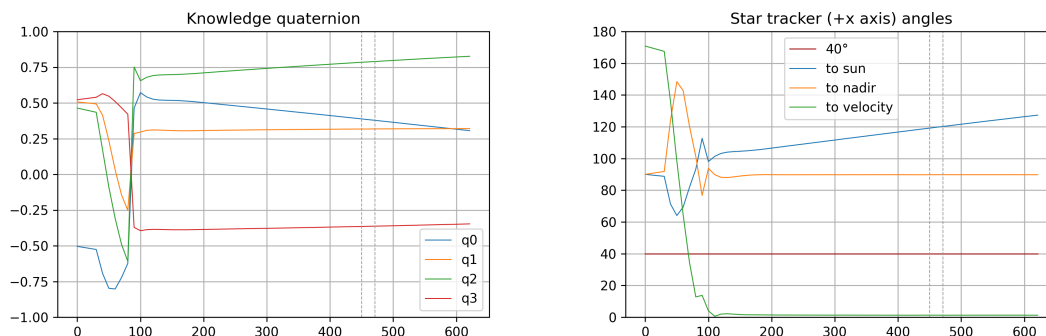
(a) March 16th, 23:50 UTC+00. Capture mode: nonbinned, exposure: 50 ms, fps: 7.

(b) March 17th, 01:25 UTC+00. Capture mode: nonbinned, exposure: 70 ms, fps: 5.

(c) March 17th, 03:00 UTC+00. Capture mode: nonbinned, exposure: 40, fps: 6.

Figure 4.10: Three successful captures between March 16th and March 17th.

In figure 4.11 are the graphs corresponding to the knowledge quaternions and star tracker angles at the time of capture of figure 4.10b, as calculated by the satellite payload.

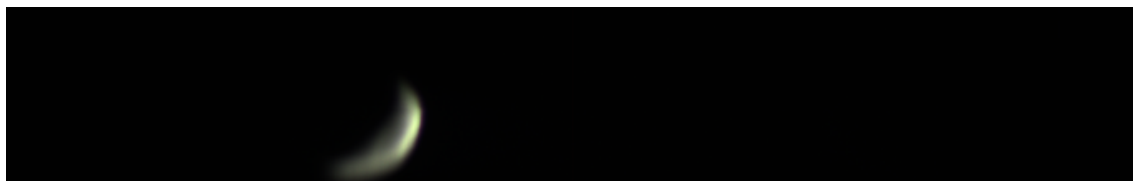


(a) Knowledge quaternions as calculated by the satellite payload. (b) Star tracker angles as calculated by the satellite payload.

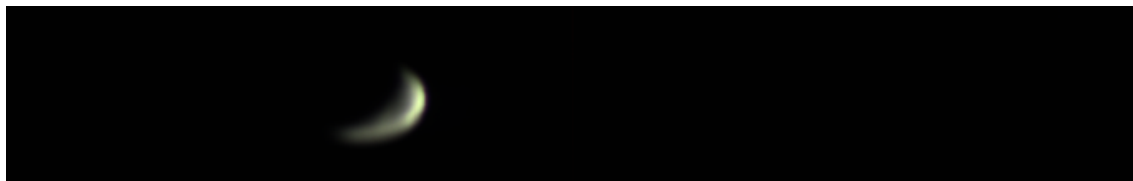
Figure 4.11: Knowledge quaternions and star tracker angles from capture at March 17th, 01:25 UTC+00.

As we see in figure 4.11b, the star tracker angle to the sun is approximately 90° at the time when the satellite aligns with its desired orientation, meaning that the star tracker should not be blinded by the sun. Startracker blinding is further discussed in 4.6.2. Further, we see that the star tracker is pointing at 0° towards the velocity vector and approximately 90° towards the nadir.

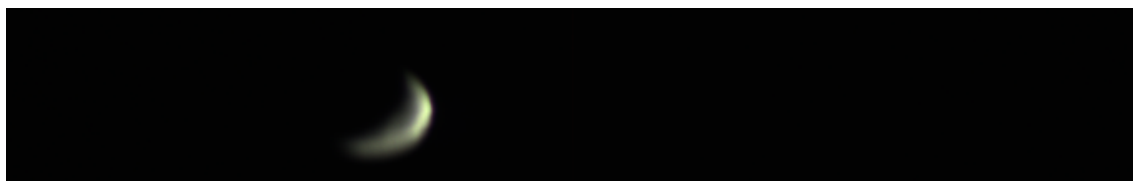
At the time of this thesis's completion, we have 10 successful captures. These are all shown in figure 4.12 below.



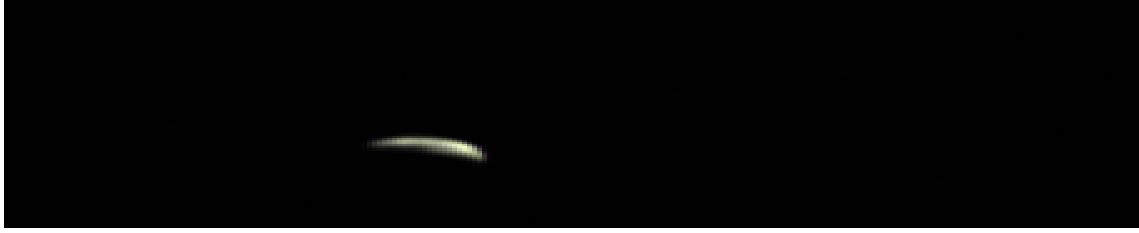
(a) Successful capture from March 16th, 23:50 UTC+00. Capture mode: nonbinned, exposure: 50, fps: 7, lunar phase = -113.84°



(b) Successful capture from March 17th, 01:25 UTC+00. Capture mode: nonbinned, exposure: 70, fps: 5, lunar phase = -114.73°



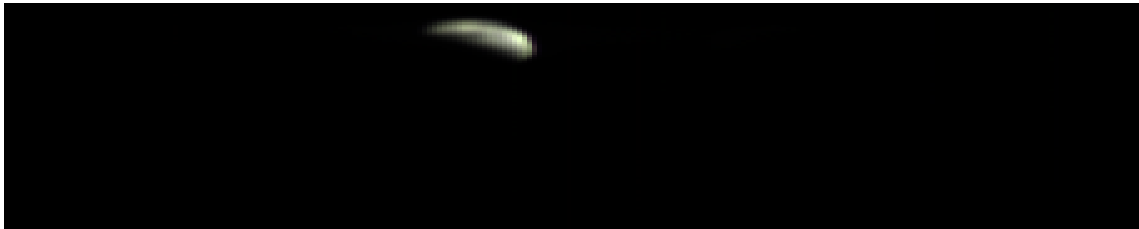
(c) Successful capture from March 17th, 03:00 UTC+00. Capture mode: nonbinned, exposure: 40, fps: 6, lunar phase = -115.68° .



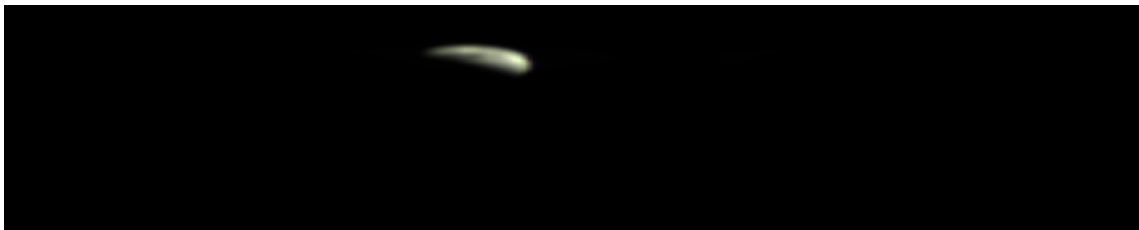
(d) Successful capture from April 26th, 17:45 UTC+00. Capture mode: nonbinned, exposure: 70, fps: 2, lunar phase = 103.43° .



(e) Successful capture from April 26th, 19:19 UTC+00. Capture mode: quaternion, exposure: 70, fps: 14, lunar phase = 102.73° .



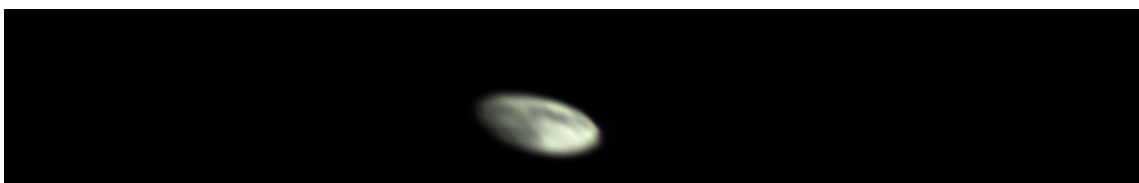
(f) Successful capture from April 27th, 17:27 UTC+00. Capture mode: nonbinned, exposure: 70, fps: 2, lunar phase = 92.64° .



(g) Successful capture from April 27th, 20:35 UTC+00. Capture mode: quaternion, exposure: 70, fps: 14, lunar phase = 91.23° .



(h) Successful capture from May 31st, 21:06 UTC+00. Capture mode: nonbinned, exposure: 30, fps: 4, lunar phase = 42.57° .



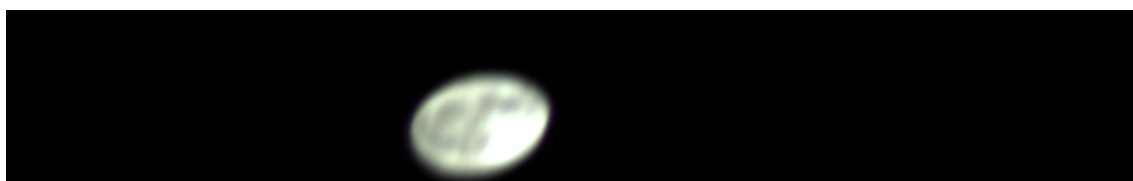
(i) Successful capture from June 1st, 01:50 UTC+00. Capture mode: nonbinned, exposure: 30, fps: 4, lunar phase = 40.18° .



(j) Successful capture from June 1st, 08:10 UTC+00. Capture mode: nonbinned, exposure: 20, fps: 4, lunar phase = 36.97° .



(k) Successful capture from June 3rd, 17:04 UTC+00. Capture mode: nonbinned, exposure: 40, fps: 6, lunar phase = 7.19° .



(l) Successful capture from June 3rd, 23:22 UTC+00. Capture mode: nonbinned, exposure: 40, fps: 6, lunar phase = 4.63° .

Figure 4.12: All successful captures as of this thesis's completion.

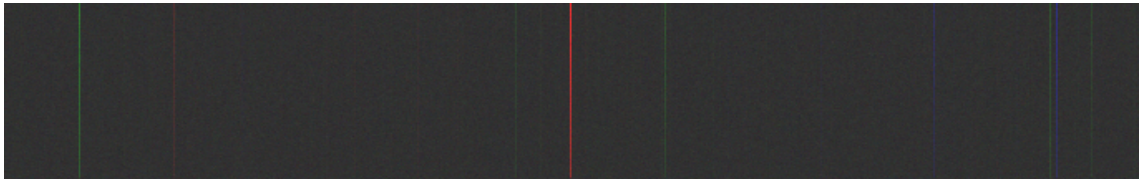
Further analysis of the successful captures is discussed in section 4.6.3 below.

4.6.2 Unsuccessful captures

As discussed in section 4.5, there are many reasons as to why a successful capture may be hard to obtain. The research conducted in this thesis is performed on a 'real' system, i.e. the satellite HYPSON-1, orbiting the earth at great speed. This includes many unpredictable errors in its trajectory calculations, time synchronization, rounding errors and so on, as well as varying accuracy in positional calculations for the involved celestial bodies. Therefore, as expected, the HSI missed its target, the Moon, on multiple occasions. In addition to this, we mentioned issues that can occur in case of unplanned reboots of the payload and other causes for interrupted data downlinking.

Startracker blinding

In figure 4.13a below we see an example of a capture from April 5th, 22:43 UTC+00, where the HSI was pointing in the wrong direction at the time of capture. In figure 4.13b we see a successful capture from May 31st, 21:06 UTC+00, which we will use for comparison in this section.



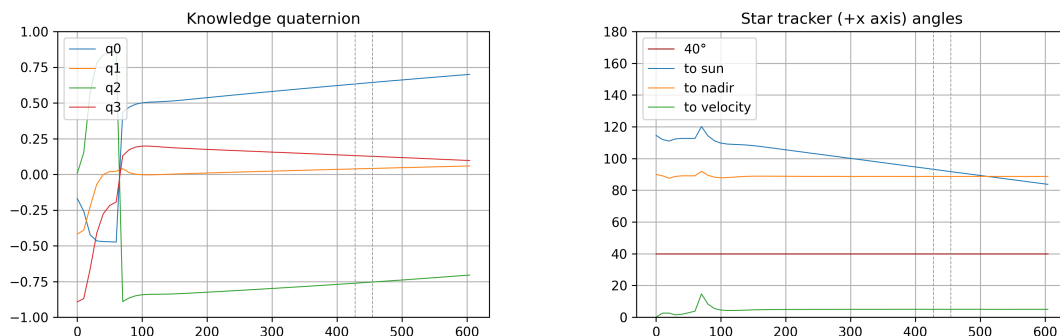
(a) Unsuccessful capture from April 5th, 22:43 UTC+00. Capture mode: nonbinned, exposure: 70 ms, fps: 4.



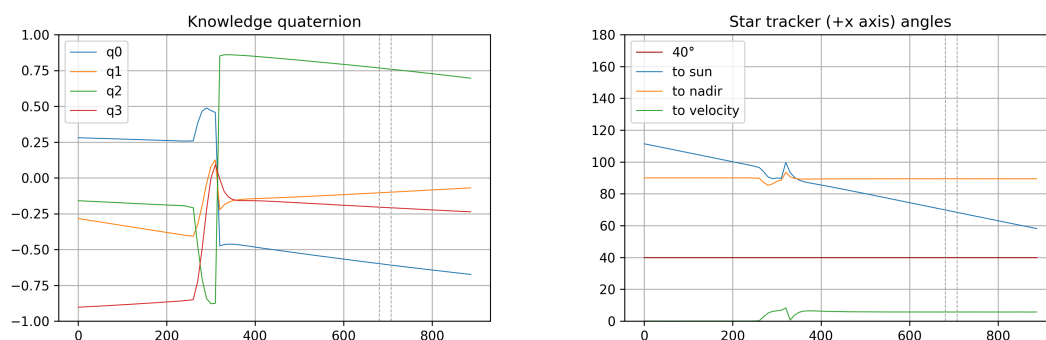
(b) Successful capture from May 31st, 21:06 UTC+00. Capture mode: nonbinned, exposure: 30 ms, fps: 4.

Figure 4.13: Unsuccessful and successful captures.

We will now take a deeper look at the captures from April 5th, 22:43 UTC+00 and May 31st, 21:06 UTC+00. In figure 4.14 are the graphs corresponding to the knowledge quaternions and star tracker angles at the time of capture, as calculated by the satellite payload.



(a) Knowledge quaternions as calculated by the satellite payload, from capture on April 5th. (b) Star tracker angles as calculated by the satellite payload, from capture on April 5th.



(c) Knowledge quaternions as calculated by the satellite payload, from capture on May 31st. (d) Star tracker angles as calculated by the satellite payload, from capture on May 31st.

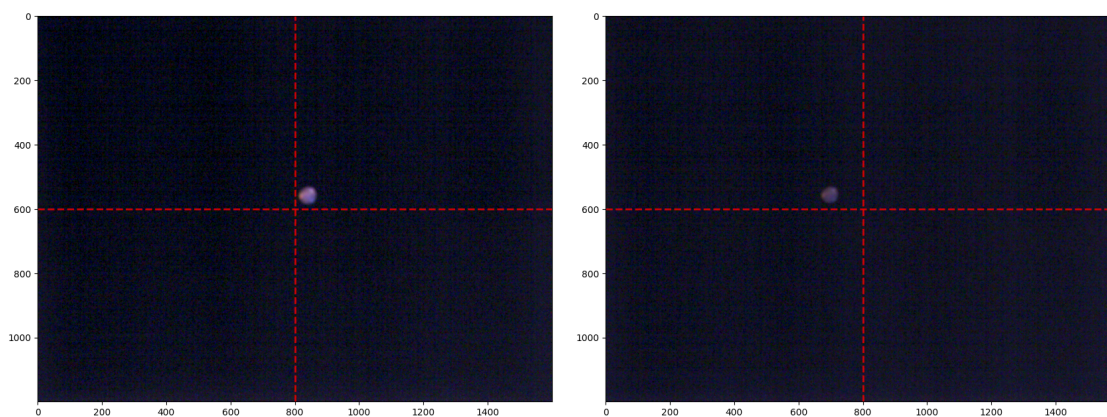
Figure 4.14: Knowledge quaternions and star tracker angles from capture at April 5th, 22:43 UTC+00.

As we see in figure 4.14b, the star tracker angle to the sun is approximately 110° and the angle to the nadir is approximately 90° at the time when the satellite aligns with its desired orientation. Similarly, in figure 4.14d, the star tracker angle to the sun is approximately 90° and the angle to the nadir is approximately 90° at the time when the satellite aligns with its desired orientation.

Regarding the star tracker's susceptibility to being blinded, it is generally expected that as long as the angle to the sun and earth is greater than 90 degrees, the star tracker should not be affected. However, our observations have indicated that the Earth Inhibition angle² may have a greater impact than initially assumed. It is suspected that if the satellite is on the daytime side of the Earth, it can still be potentially blinded by the Earth, even if the angle to the Earth is greater than 90 degrees. Conversely, if the satellite is on the nighttime side of the Earth, it is less likely to be affected by such blinding. Further investigation and analysis are required to better understand and mitigate this issue. It is suspected that this is the cause of failure for this capture and other unsuccessful captures with similar results.

²The minimum angle required between the line of sight from the satellite to the Sun and the Earth's limb. It represents the angular distance necessary to prevent the satellite from being obstructed or blinded by the Earth or Sun.

In addition to the hyperspectral image captured by the HSI, the satellite payload also captures an RGB image with a separate camera. In figure 4.15 below, we see the RGB image of the unsuccessful capture on April 5th, 22:43 UTC+00 (left), as well as the RGB image of the successful capture from May 31st, 21:06 UTC+00 (right). A line has been added in the middle of the images, in both the horizontal and vertical directions.



(a) RGB image of capture from April 5th, 22:43 UTC+00. The Moon is visible on the upper right side of the centre. The capture corresponding to this RGB image was unsuccessful. (b) RGB image of capture from May 31st, 21:06 UTC+00. The Moon is visible on the upper left side of the centre. The capture corresponding to this RGB image was successful.

Figure 4.15: RGB images of one unsuccessful and one successful capture. The horizontal and vertical centre lines are marked in red.

As seen in the figure, there is a noticeable difference in the position of the moon in the RGB images. The satellite payload is configured to capture the RGB image once it has finished capturing the hyperspectral image. The total number of frames and the number of frames captured per second therefore plays a significant role in when the RGB image will be triggered, with regards to the total duration of the hyperspectral capture. The total number of frames and number of frames per second configured for these captures are both set to 106 and 4 respectively, so the total capture time is the same. In theory, the Moon should therefore be in the same position in the RGB images.

In figure 4.14 it can be observed that the star tracker angles to the velocity vector are the same for both captures, i.e. the orientation of the RGB images in figure 4.15 is the same. In figure 4.15b, the successful capture, we see that the Moon is positioned on the left-hand side of the centre line. This is expected, as the Moon should be in the centre of both cameras FoV's in the middle of the hyperspectral imaging process, i.e. on either side of the centre line when the RGB camera is triggered. Given that the Moon appears on the left side in the successful capture's centre line, it's clear that the Moon was moving left relative to the camera's field of view during this pass. In figure 4.15a, the unsuccessful capture, we see that the Moon is positioned slightly on the right-hand side of the centre line. Considering that the RGB images are oriented in the same direction, this implies that the hyperspectral imaging process started too early, with the Moon positioned ahead of the cameras FoV centre.

In conclusion, this section analyzed the influence of factors like the star tracker's susceptibility to blinding, the role of the Earth Inhibition angle, and the timing of the hyperspectral capture process on the success of Moon captures. This provides a deeper understanding of the challenges faced in automating lunar capture and serves as a foundation for improving future attempts. Nonetheless, additional research is needed to further validate these findings and to determine effective mitigation strategies.

Downlink-Related Failures in Image Capturing

Successful image acquisition via a hyperspectral imager on a satellite involves not only capturing the target accurately but also ensuring that the acquired image data is downlinked effectively to a ground station. In this section, we delve into scenarios that have caused failures due to complications in the downlinking process.

An essential aspect of a successful downlink is the proper synchronization between the image buffering and downlinking processes. This critical sequence begins when the ground station recognizes a buffered file's presence and size on the satellite. It then initiates a download process based on this data. However, complications arise if a new capture begins and the corresponding data are buffered to the same file before the download has completed. This overlapping can create file conflicts, resulting in unsuccessful data retrieval despite the capture process being error-free.

The stability of the Payload Controller (PC) is another crucial factor for successful image capture and downlink. The PC, which oversees payload operations including the hyperspectral imager and data handling, must maintain uninterrupted function during the buffering process. An unexpected reboot of the PC during this period can lead to data loss or corruption, thereby disrupting the image capture and transmission despite the capture being correctly executed.

Figure 4.16 presents an image disrupted by a PC reboot. It is evident that the PC reboot causes substantial data loss, resulting in fragmented and incomplete captures. The image depicts stark discontinuities and data corruption, signifying the loss of crucial information during the image acquisition process.

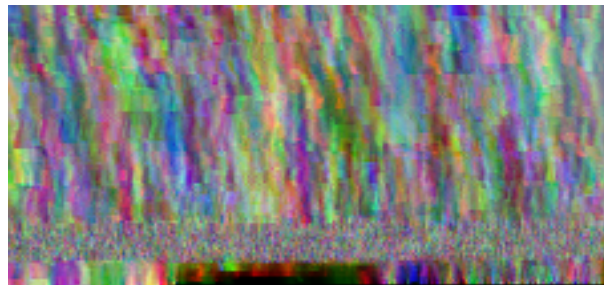


Figure 4.16: Capture from May 5th disrupted by a Payload Controller (PC) reboot. The resulting data loss and corruption underline the potential impacts of system reboots during the image acquisition process.

Even with successful image capture and buffering, an image downlink process may

not always reach completion due to a variety of reasons. These factors can range from signal disruptions and hardware malfunctions on the satellite to exceeding the allocated bandwidth for data transfer. In these scenarios, the unavailability of full image data at the ground station results in unsuccessful image retrieval.

Buffering errors are another significant concern that can lead to unsuccessful captures. These errors encompass a wide range of issues, including software glitches, memory allocation problems, and hardware faults on the satellite. Such problems can prevent the proper buffering of the captured image data, thereby hindering the downlink process.

Figure 4.17 below provides an illustrative example of a capture affected by buffering issues. As can be seen, these issues can significantly compromise the integrity of the image data, resulting in incomplete or corrupted image output. In this particular capture, the erratic patterning and missing data sections are indicative of buffering errors that occurred during the image acquisition process.

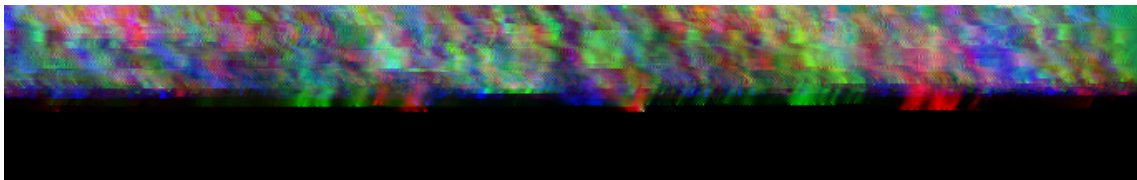
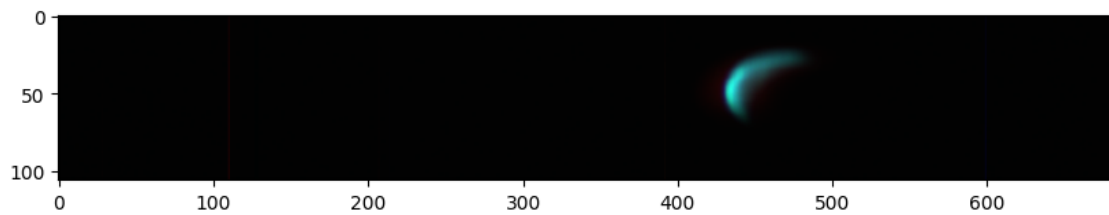


Figure 4.17: Capture from May 25th, illustrating a capture affected by a buffer bug. The significant degradation in image quality and data integrity highlights the influence of buffering issues on successful image acquisition.

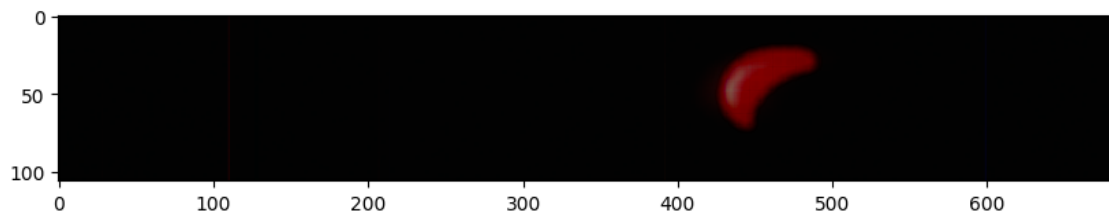
In sum, these cases reflect the complexity and the delicate balance required in capturing images using a hyperspectral imager on a satellite. Any disruption or anomaly in the intricate interplay between the various steps - image capturing, buffering, and downlinking - can lead to unsuccessful image retrieval despite a potentially successful capture. Addressing these concerns and enhancing the robustness and reliability of this process is out of the scope of this thesis, but clearly affects the results.

4.6.3 Analysis

To analyse the image of the moon, a simple script has been developed to identify which pixels are of the moon and which pixels are of deep space. This is achieved by systematically examining each pixel in the image and determining the average radiance based on a specific set of wavelengths. These wavelengths are recognized for exhibiting elevated values if the pixel represents the moon, and diminished values if the pixel corresponds to deep space or a shadowed area on the moon. A representation of the pixels of interest on a capture from March 17th is shown in figure 4.18 below.



(a) Capture from March 17th, 01:25 UTC+00.



(b) Capture from March 17th, 01:25 UTC+00, with pixels of interest marked in red.

Figure 4.18: Raw RGB image and image with pixels of interest (marked in red) from capture on March 17th, 01:25 UTC+00.

To further use the moon capture for analysis of the HSI's wavelengths, we calculate the average radiance of the pixels of interest. The radiance is determined by conducting radiometric calibration on the image cube, using the calibration pipeline as prescribed by NTNU's SmallSat Lab Cal-Char-Corr scripts [36]. The complete cube with radiometric calibration applied is shown in figure 4.19 below.

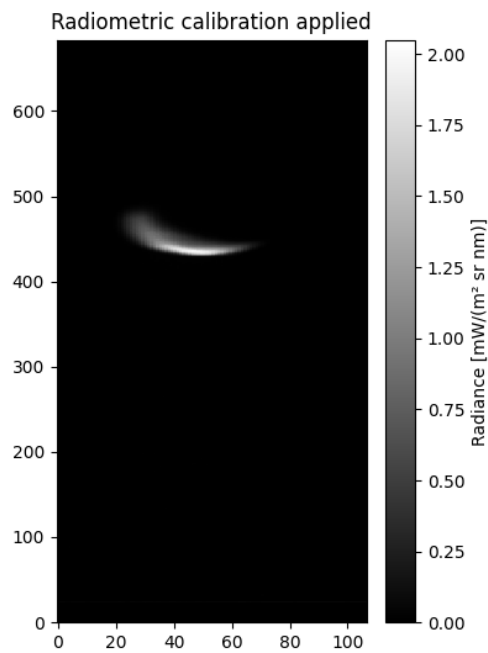


Figure 4.19: Radiometric calibration applied to capture at March 17th, 01:25 UTC+00, illustrated at 550 nm wavelength. The radiance values are shown in the grayscale bar on the right.

The average radiance of the moon at each wavelength is shown in figure 4.20 below.

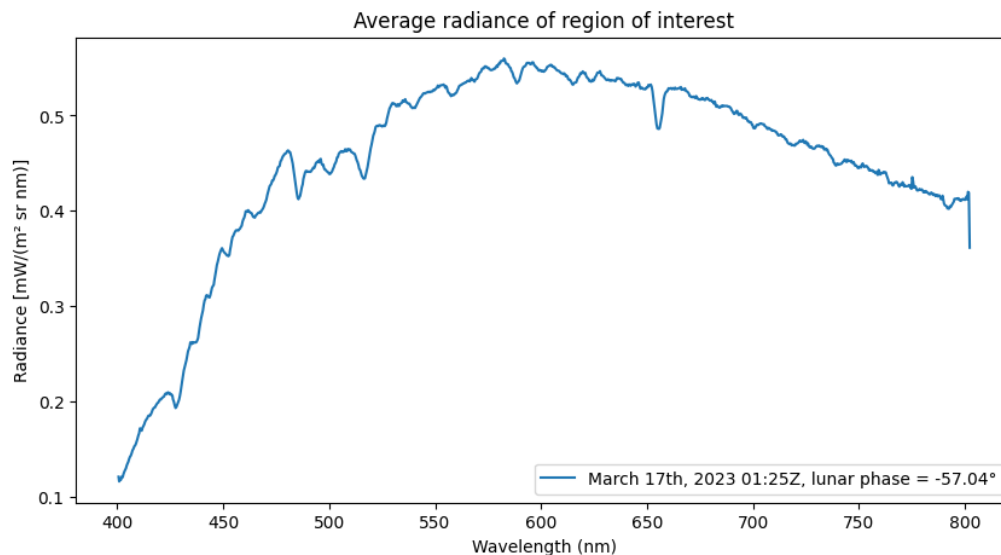
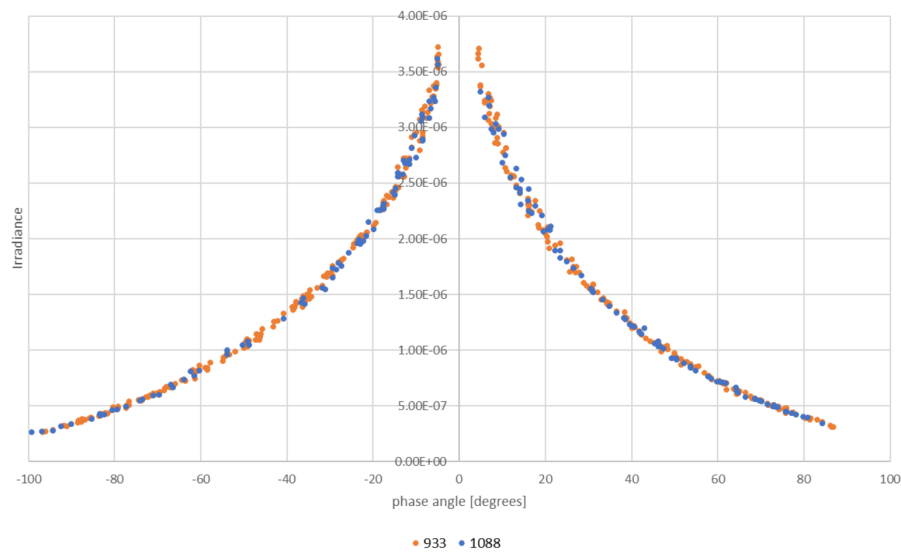


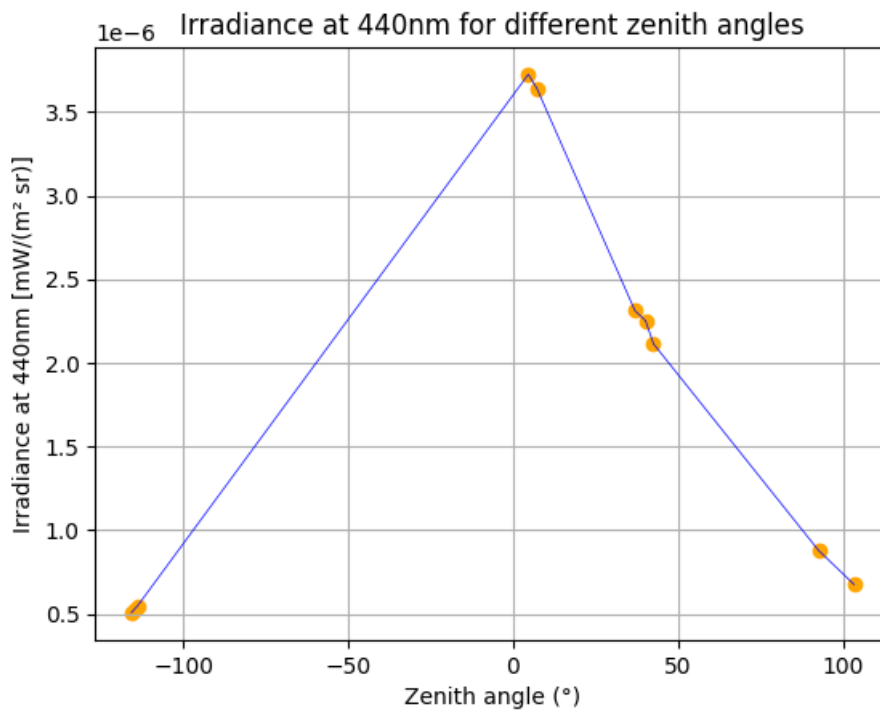
Figure 4.20: The average radiance of the pixels of interest at each wavelength, calculated from the capture on March 17th, 01:25 UTC+00.

Doing this for all successful captures, we can visualize how the lunar phase affects the average radiance of the moon throughout the lunar phase. In figure 4.21 below,

the lunar irradiance of the moon at the corresponding lunar phases is visualized for both the HYPSON-1 measurements and the LIME model measurements, as obtained from the 1088 and 933 instruments as discussed in 4.1.7.



(a) Irradiance values of 1088 and 933 instruments.



(b) Irradiance values of successful captures at 440 nm.

Figure 4.21: Irradiance values of HYPSON-1 captures vs 1088 and 933 instruments at different wavelengths

It can be observed that the irradiance values for the corresponding lunar phases form a similar curve in the LIME model and the HYPSON-1 measurements, although there are deviations in the measured irradiance. These differences are what shall be used further to calculate the difference between the measured values and the modelled

values, to finally calculate the calibration table for our system. In figure 4.22 below, we see the spectral signature across all wavelengths for each of the captures. The lunar phase of the corresponding capture is denoted in the legends to the right of the graph. Similarly to what we see in figure 4.21, it is clear that a lower lunar zenith angle results in a higher overall radiance value.

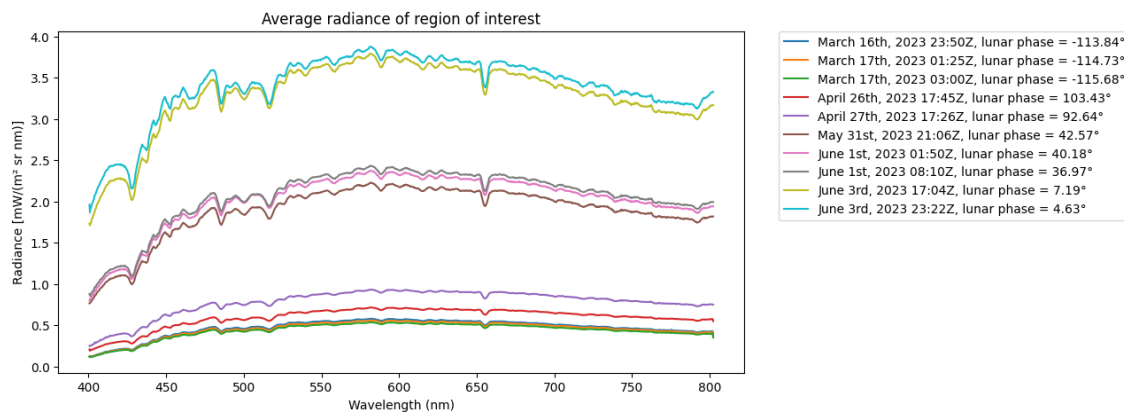


Figure 4.22: Radiance across all wavelengths for each successful capture, with lunar phase angle.

To get the calibration table for our system, the Measurement and Model comparison procedure from LIME needs to be used. Despite multiple attempts to get in contact with Flemish Institute for Technological Research (VITO), the institute responsible for the LIME model development, the pipeline for Measurement and Model comparison procedure has not yet been obtained at the time of completing this thesis. The Pleiades-1B HR instrument, also called PHR1B, has been used to verify the comparison procedure of LIME [34]. We will therefore introduce the procedure of this instrument instead, with the aim of applying the same procedure to our measurements in the future.

PLEIADES instrument

The PHR1B instrument is a high-resolution multi-spectral imager, with five spectral bands in the Visible and Near-Infrared (VNIR) region, with the fifth band being a pan-chromatic band. The PHR1B instrument has in the period between February 18th 2023 and April 7th 2017 acquired 68 lunar observations. In figure 4.23 we can observe the sparsity of the measurements with respect to the lunar phase angle, similarly as shown in figure 4.21.

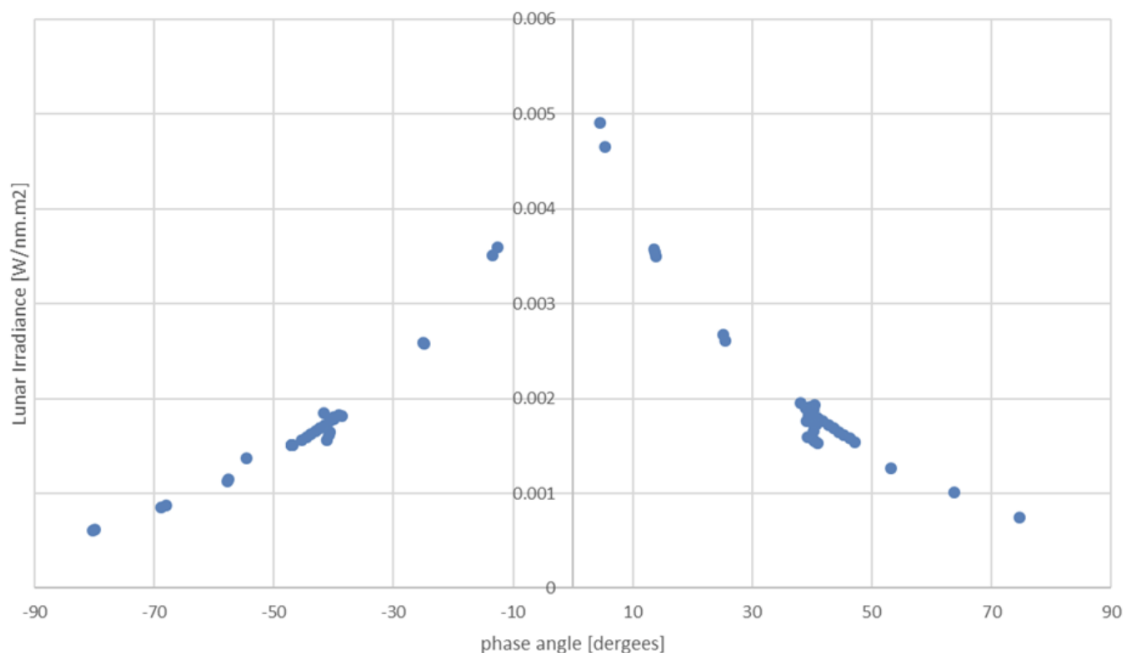


Figure 4.23: Pleiades lunar irradiance acquisitions.

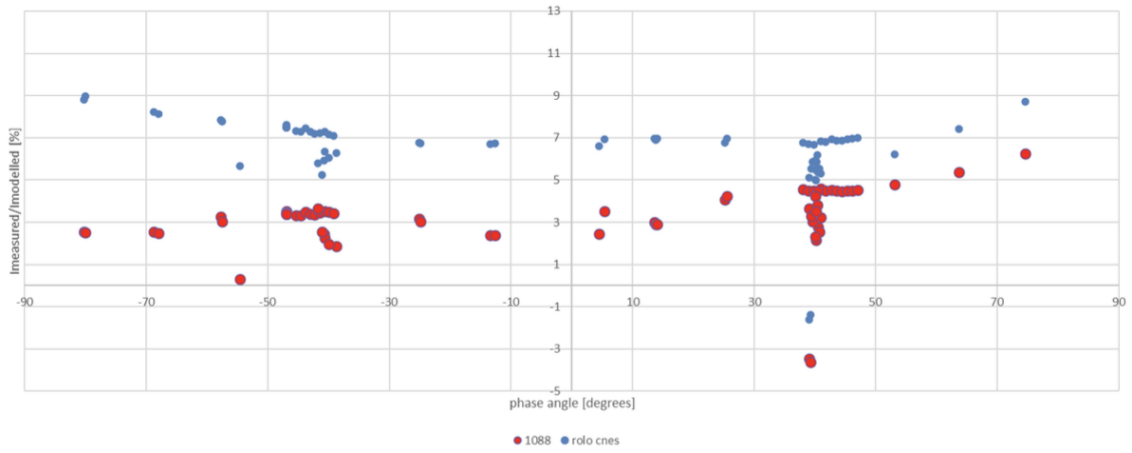
The official calibration table for the PHR1B instrument is as shown in table 4.1 below:

Date	B0	B1	B2	B3	PAN
01/12/2012	1.117	1.085	1.075	1.015	1.034
01/09/2013	1.112	1.079	1.071	1.013	1.034
01/12/2013	1.110	1.078	1.070	1.012	1.034
01/03/2014	1.108	1.076	1.069	1.011	1.034
01/06/2014	1.106	1.074	1.067	1.011	1.034
01/09/2014	1.104	1.072	1.066	1.010	1.034
01/12/2014	1.103	1.070	1.065	1.009	1.034
01/03/2015	1.100	1.068	1.064	1.008	1.034
01/06/2015	1.099	1.066	1.062	1.008	1.034
01/09/2015	1.097	1.064	1.062	1.007	1.034
01/12/2015	1.095	1.061	1.062	1.006	1.034
01/03/2016	1.093	1.061	1.062	1.006	1.034
01/06/2016	1.090	1.056	1.054	1.003	1.032
01/09/2016	1.089	1.055	1.053	1.003	1.031
01/01/2017	1.087	1.053	1.050	1.001	1.029
01/03/2017	1.085	1.051	1.048	1.000	1.028

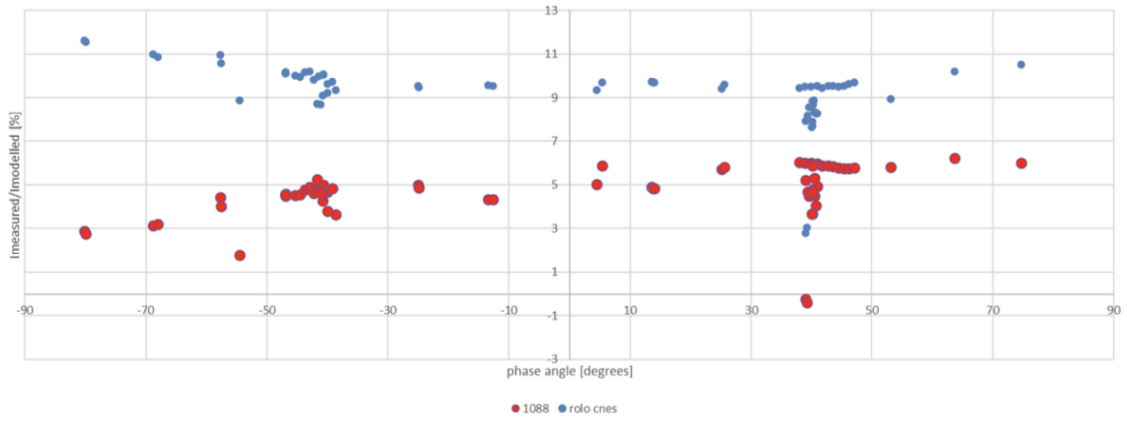
Table 4.1: PHR1B calibration table, given by [34].

For all Pleiades observations and spectral bands, a model output is generated. The difference, in %, is calculated and plotted against the phase angle in figure 4.24 below. With the Pleiades data, the results of the CNES implementation of the ROLO model are delivered as well, and the comparison with these results is plotted

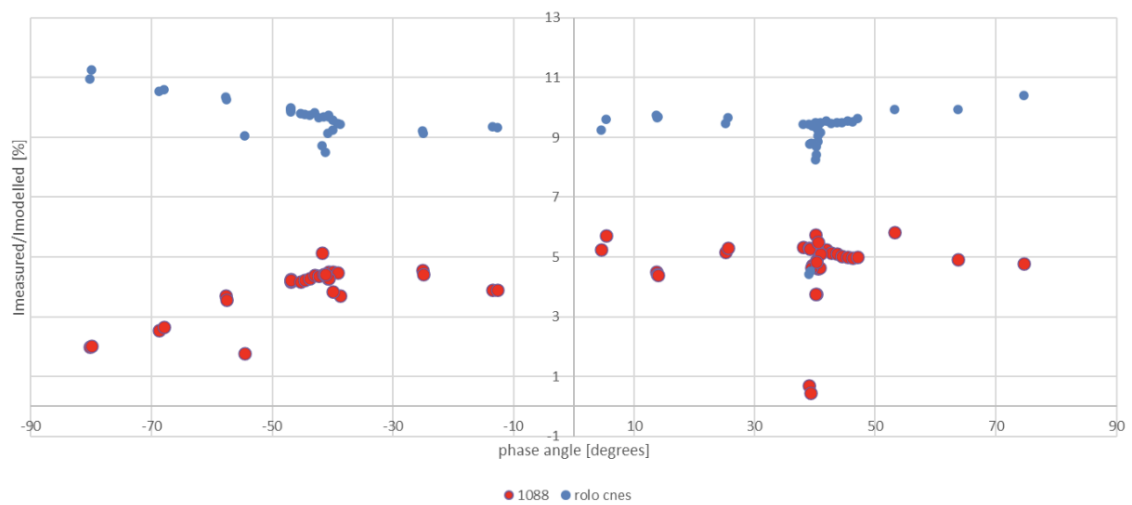
as well (in blue) as an extra reference.



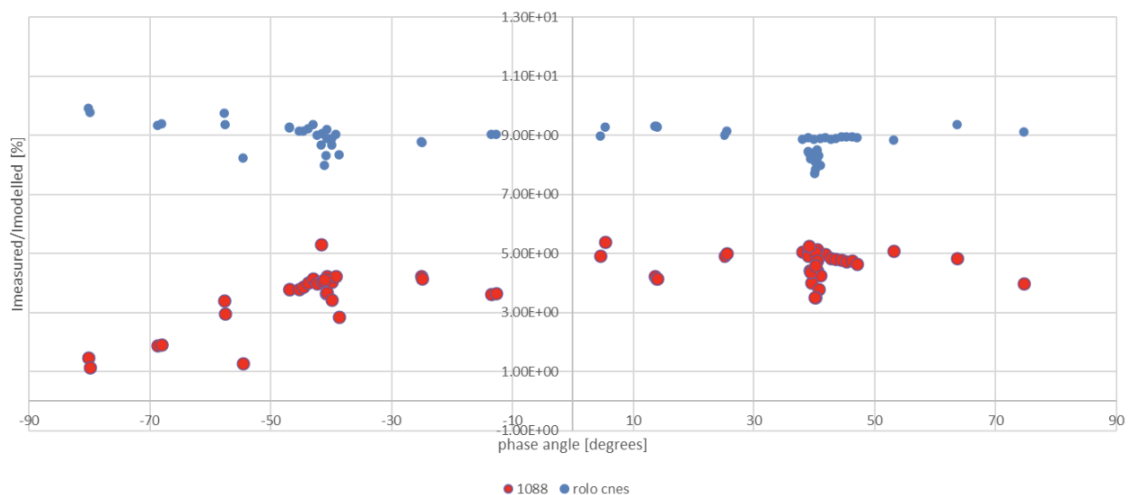
(a) PHR1B band 1 (blue) result – 1088+933 model



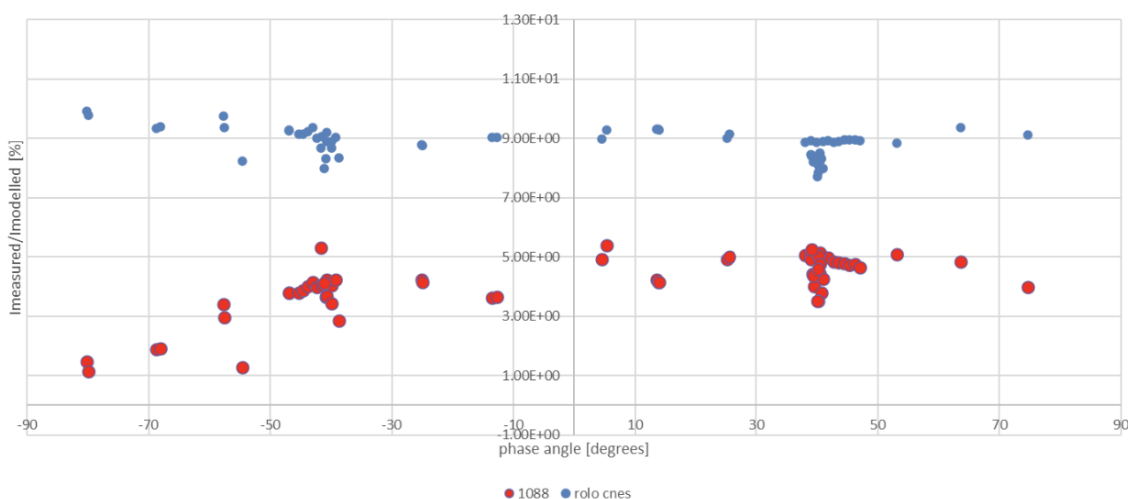
(b) PHR1B band 2 (green) result – 1088+933 model



(c) PHR1B band 3 (red) result – 1088+933 model



(d) PHR1B band 4 (nir) result – 1088+933 model



(e) PHR1B band 5 (pan) result – 1088+933 model

Figure 4.24: Difference in % between Pleiades observations and model output for each band. From [34].

The table of averages and standard deviations of Pleiades data against the 1088+933 model is shown in table 4.2.

%	Blue	Green	Red	NIR	PAN
AVG	3.099	4.614	4.342	6.630	5.773
STDEV	1.414	1.138	0.940	1.173	9.505

Table 4.2: Average and Stdev of Pleiades data against the 1088+933 model, given by [34].

In conclusion, this section has elucidated the process of using lunar observations for hyperspectral imaging calibration. This radiance information, coupled with the Cal-Char-Corr scripts and LIME model, provided useful calibration data. The Pleiades

instrument serves as a valuable reference for the further application of our procedure. Although we faced challenges in obtaining the complete LIME measurement and model comparison procedure, we expect it to further improve our calibration process. Future work will be aimed at obtaining this procedure and applying it to our measurements.

Chapter 5

Discussion

Contents

5.1	Celestial Bodies	61
5.2	Pointing	62
5.3	Timing	63
5.4	Execution	64
5.5	Observing the Moon and Results	65
5.6	Research Questions	66
5.6.1	Hyperspectral Imaging for Lunar Observation	66
5.6.2	Automation Challenges	66
5.6.3	Applications of Findings	67
5.7	Limitations	67

This chapter is intended to interpret and discuss the findings of this thesis, with a particular focus on the research questions that guided the study. By considering the results in relation to the initial hypotheses and the body of existing literature, we hope to offer a comprehensive understanding of the implications and potential applications of the research.

5.1 Celestial Bodies

The choice of Python and the Skyfield library [16] for the calculations involving celestial bodies was driven by several factors. First and foremost, Python is a high-level programming language that has wide acceptance in the scientific community. This is partly due to its readability, flexibility and a vast array of scientific libraries that make it a perfect fit for such complex calculations.

Alternatives to Python include other programming languages such as MATLAB or Java. MATLAB, for example, is frequently used in academic and research environments for complex mathematical operations and simulations. It provides a broad range of toolboxes for various applications, including astronomy and space exploration, and is noted for its robust performance in matrix operations. However, Python was chosen over MATLAB due to its open-source nature, the availability of multiple free astronomical libraries, and its general-purpose application which ensures the study's broader accessibility beyond the MATLAB community.

The Skyfield library in Python was selected as it provides a simple yet powerful API for astronomical calculations. Skyfield's ability to handle time and spatial

computations using different frames of reference, such as the International Celestial Reference System (ICRS) and Geocentric Celestial Reference System (GCRS), is critical to our research. Skyfield also seamlessly integrates with the TLE datasets, which is an essential aspect of satellite tracking and monitoring.

Possible alternatives to Skyfield include the Astropy library [37], another open-source Python library for astronomy, or the SPICE toolkit [?] by NASA's Navigation and Ancillary Information Facility (NAIF). Astropy has a similar scope to Skyfield but is considered more complex and might be more suitable for projects requiring advanced features that are beyond the scope of this study. The SPICE toolkit, on the other hand, provides a robust system for handling space mission geometry but requires more effort to set up and is less intuitive for those unfamiliar with its working principles.

The custom functions developed for this research, including the `distance_obj_to_target()`, `get_minimum_distance()`, `get_off_nadir_angle()`, and `get_maximum_off_nadir_angle()` functions, represent a methodology designed to leverage the features of Python and Skyfield to achieve the research objectives. These functions were developed to automate and streamline the complex calculations required to study the distances and angles between celestial bodies. While these calculations could have been performed manually or using less automated methods, the custom functions enhance the efficiency, accuracy, and repeatability of the calculations. They also offer flexibility as they can be easily adjusted to suit different celestial bodies, targets, and observation times.

In conclusion, while there are alternative programming languages and libraries that could have been used for this research, the selection of Python and the Skyfield library was a strategic decision that balanced the need for precision, efficiency, flexibility, and accessibility. The custom functions developed for this study further capitalized on these features to facilitate the study of the relationship between celestial bodies in the context of our research.

5.2 Pointing

The section on pointing represents a critical aspect of the study, with quaternion-based pointing methodology chosen for HYPSON-1 due to its advantages in handling orientation in three-dimensional space. Using quaternions over alternatives like Euler angles can help evade certain problems, such as gimbal lock, and results in smoother and continuous rotations. This enhances the precision and flexibility of satellite alignment with intended targets, an essential attribute when performing high-precision operations like monitoring and imaging.

A significant portion of the pointing section revolves around functions developed for quaternion calculations, which underpin the satellite's pointing strategy. These functions enable the determination of the optimal orientation relative to the Earth, target objects, and other frames of reference. For instance, the `get_quaternion()` function is central to achieving the desired orientation. This function employs numerous mathematical transformations and calculations, highlighting the importance

of understanding quaternion mathematics and the complex interplay between different coordinate systems in satellite pointing operations.

In support of this purpose, the `eci2LVLH()` function plays a vital role in transforming the position and velocity vectors from the Earth-Centered Inertial (ECI) frame to the Local Vertical Local Horizontal (LVLH) frame. This transformation allows for better comprehension of satellite positioning in relation to Earth, which is crucial when planning the satellite's orientation towards any given target.

To ensure a smooth rotation from one vector to another, the `rot_rodrigues()` function uses the Rodrigues rotation formula. This approach leverages the concept of rotation around an axis, which is more intuitive in certain applications compared to traditional rotation matrices. Rodrigues' rotation formula, by its direct application of the rotation axis and angle, enhances the interpretability of the satellite's orientation.

Lastly, the `rot2q()` function emphasizes the relevance of quaternion representations in the context of satellite orientation. The use of quaternions in rotation applications provides several computational advantages, including but not limited to numerical stability, ease of interpolation, and simpler calculations for compounded rotations. Quaternions also circumvent the issue of gimbal lock, which is inherent in Euler angle rotations, and are thus more suitable for a three-dimensional and continuous rotation scenario like satellite pointing.

Overall, the methodologies and functions developed in this study highlight the importance and complexity of satellite pointing, emphasizing the robust mathematical techniques required to handle this issue. While the methods chosen - quaternion-based pointing and the transformation of coordinates to LVLH frames - offer significant benefits, they also highlight the intricate nature of spatial orientation in satellite navigation and control. The development and usage of functions like `get_quaternion()`, `eci2LVLH()`, `rot_rodrigues()`, and `rot2q()` provide a practical means of managing these complexities, giving us the tools necessary to accurately determine and control HYPSO-1's orientation.

5.3 Timing

The exploration of timing in satellite imaging, as outlined in section 4.3, presents important insights into the operational efficiency of these processes. The interplay between off-nadir angles, total capture time, and FPS offers a detailed understanding that is essential for optimizing imaging operations.

Beginning with the off-nadir angles and the total capture time, this analysis unveils a somewhat counterintuitive relationship. Larger off-nadir angles, contrary to what one might initially anticipate, result in shorter total capture times. This is due to an increase in the rotational speed of the satellite imaging sensor as the off-nadir angle increases, as the satellite has a constant orientation wrt. the earth, consequently reducing the total capture time. This understanding prompts a reassessment of how satellite imaging, particularly when targeting celestial bodies like the moon, can be

more efficiently planned and conducted.

Likewise, the approach to calculating FPS within this analysis presents an important consideration for satellite imaging operations. While the instinct might be to maximize FPS for the highest possible detail, this study highlights the necessity of a balance to ensure the full coverage of the moon within each frame. This knowledge reinforces the requirement for thorough planning and informed decisions in satellite imaging operations.

The results obtained align with the existing literature on satellite imaging optimization, which also emphasizes efficient timing and careful FPS calculations for achieving optimal image quality. The understanding of the impact of off-nadir angles on total capture time and the trade-offs required for determining optimal FPS adds to the current body of knowledge in this area.

In conclusion, the discussion on timing in chapter 4 is not only a technical exploration but also a practical guide for satellite imaging. It emphasizes the need for careful planning and calculation to enhance the efficiency and quality of satellite imaging operations, reinforcing the thesis's overall argument.

5.4 Execution

The 'Execution' section of chapter 4 outlines two main executable scripts: SatNav and the Hypso Moon Script Command Generator. These scripts signify a significant contribution to the field of satellite imaging, with a focus on ease of use and adaptability.

The SatNav script introduces a flexible solution for a variety of satellite imaging requirements. Its capability to calculate optimal capture times and pointing quaternions for a range of celestial bodies is a key feature that adds value to its users. This function responds to the need for a tailored approach in satellite imaging operations, accommodating different requirements based on the celestial body under study. The application of this script, as evidenced by the execution demonstrated, efficiently produces the desired outcomes, making it a practical tool for satellite imaging.

The Hypso Moon Script Command Generator provides a more specialized solution, focusing on the HYPSO-1 satellite's moon capture operations. This specialized script indicates the appreciation of the unique requirements of lunar imaging operations for HYPSO-1. As shown in the execution, the script not only calculates the necessary parameters but also presents them in a format that is directly usable for planning and controlling HYPSO-1 satellite operations. This reduces the steps necessary for satellite operators, thus simplifying their work.

The significance of these tools lies in their user-friendly design and efficient execution, reducing the complexity of the satellite imaging planning process. The scripts deliver a user-oriented solution that directly impacts the effectiveness of satellite imaging operations, presenting a unique contribution to the field. Moreover, the capacity of

these scripts to automate complex calculations and processes, specifically tuned to the needs of the user, demonstrates a clear understanding of the real-world challenges of satellite imaging. This reflects the broader aim of this thesis: to deliver practical solutions that enhance operational efficiency in satellite imaging.

On a more critical note, although these scripts provide a significant step forward in simplifying satellite imaging planning and execution, the possibility of further refinements should not be overlooked. Future work may focus on expanding the range of supported satellites and celestial bodies and integrating these scripts into larger, more comprehensive satellite operation planning systems. Such enhancements would broaden the applicability and usefulness of the tools, cementing their contribution to the field of satellite imaging.

5.5 Observing the Moon and Results

In sections 4.5 and 4.6 of the 'Work and Results' chapter, the complexity and delicacy of image capture and retrieval were addressed. The hyperspectral imager onboard the satellite had to accurately capture images of the moon, which were then successfully downlinked to a ground station for analysis. Failures in this process could occur due to improper synchronization between the image buffering and downlinking processes, an unexpected Payload Controller (PC) reboot, or disruptions in the downlinking process, among other reasons. Each of these components, if not properly executed, could potentially hinder the successful retrieval of images, despite a successful capture.

During the image-capturing process, the issues encountered, as depicted in figures 4.16 and 4.17, underlined the importance of stable operations and error-free processing. The images clearly showcased how a simple malfunction or disruption in any part of the process could lead to significant data loss and image degradation. These findings suggest that a more robust system, designed to handle such complications, could potentially improve the accuracy and reliability of the hyperspectral imager. However, the detailed development of such a system lies outside the scope of this thesis but forms a compelling avenue for further research and investigation.

Section 4.6.3 detailed the process of pixel identification and radiance calculation for the lunar images captured by the hyperspectral imager. A script systematically examined each pixel to distinguish between those representing the moon and those representing deep space and calculated the average radiance for the pixels of interest. This process allowed for a clear visualization of how the lunar phase affected the radiance, as shown in figure 4.21b and 4.22.

The data obtained from these image analyses were compared with the LIME model measurements, with discrepancies between the two being used to derive a calibration table for the hyperspectral imager. The detailed procedure for this comparison was unfortunately not obtained at the time of completing the thesis, but the PHR1B instrument's procedure was instead reviewed for reference. Figures and tables showing the Pleiades lunar irradiance acquisitions and the PHR1B instrument calibration table, along with a comparison of Pleiades data against the 1088+933 model, all pro-

vided critical information for the eventual calibration process.

Despite the challenges faced in the image capturing and downlinking processes and the lack of access to the complete LIME comparison procedure, the analyses conducted in this thesis were robust and provided valuable insights for the hyperspectral imaging process. Future work should focus on the development of more reliable image-capturing and downlinking processes, and on gaining access to and implementing the LIME comparison procedure to improve calibration methods. This would enable a more accurate and reliable analysis of hyperspectral images captured by the satellite, thereby advancing our understanding of lunar observations and their implications.

5.6 Research Questions

5.6.1 Hyperspectral Imaging for Lunar Observation

The first research question posed was: How can hyperspectral imaging be used to capture and analyze the full moon phases, and what benefits does this provide for lunar observation?

Hyperspectral imaging (HSI) offers the ability to capture a much broader range of the electromagnetic spectrum than traditional imaging techniques. This allows for a highly detailed analysis of lunar surface features and compositions. However, a key insight of this study is that hyperspectral imaging of full moon phases can be particularly beneficial for calibrating hyperspectral imaging sensors. Given the constant radiance of the moon, these data can be used as a consistent reference point to adjust and fine-tune satellite sensors, thereby improving the quality and accuracy of readings for their primary missions.

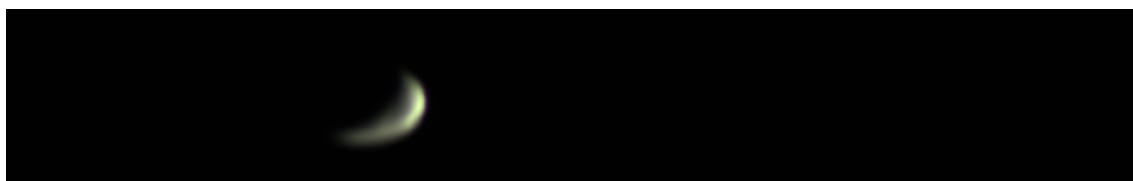


Figure 5.1: Example of a hyperspectral image of the moon.

For example, in the context of HYPSON-1, the primary mission objective is to observe ocean colour along the coast of Norway. By calibrating the HSI, we can ensure more accurate data collection for this task. Furthermore, the benefits extend beyond this specific mission; other satellites with different mission objectives could also employ similar calibration techniques to enhance their data quality.

5.6.2 Automation Challenges

The second research question was: What are the technical challenges involved in automating moon captures with a HSI, and how can these be overcome?

Automation of moon captures involves the technical challenges of precise lunar track-

ing, handling large volumes of data, and effectively processing and interpreting this data. Given the dynamic nature of space and the movement of celestial bodies, determining the exact location and trajectory of the moon in relation to the satellite is a significant challenge. Additionally, the volume of data generated by HSI can be substantial, demanding efficient methods for data management and analysis.

To overcome these challenges, we developed a set of scripts that can predict the positions of relevant bodies at given times in the future. This enables us to identify optimal capture times and calculate the necessary pointing quaternions to align the sensor accurately. This approach offers a practical and effective method for automating moon captures with a HSI.

5.6.3 Applications of Findings

The third research question asked: How can the findings from this research be applied to other applications or systems, such as small satellite missions or other planetary observation missions?

The methodologies and tools developed during this research could be of significant benefit to other satellite missions. The scripts for predicting celestial body positions and calculating pointing quaternions can be used by any satellite tasked with observing a specific celestial body. Further, the approach to calibrating hyperspectral imaging sensors can be applied to enhance the accuracy of data collection for a broad range of remote sensing applications.

Furthermore, the methodologies developed during this research, and their resulting findings, can provide a solid foundation for future research and development in related areas. Specifically, the insights gleaned about the timing and process of automating lunar observations can serve as a guide for other applications or systems that might require a similar setup. In addition, the experience and knowledge gained from creating scripts for prediction and automation, while somewhat specific to our use case, certainly opens up a path of exploration for researchers in the field of satellite observation planning and autonomous task execution.

5.7 Limitations

Despite the promising findings and contributions of this study, certain limitations need to be acknowledged, which have the potential to impact the generalizability and applicability of the results.

One of the primary limitations relates to the challenges encountered during the automation process, particularly with respect to precise lunar and satellite tracking. The implemented solution is effective but does not completely eliminate potential inaccuracies caused by orbital deviations, perturbations, or inherent errors in celestial body positioning models. Thus, while we can predict optimal capture times and calculate the necessary pointing quaternions with reasonable accuracy, there remains a certain degree of uncertainty that might impact the quality of moon captures.

Moreover, the challenge of handling the large volumes of data generated by hyperspectral imaging persists. Despite implementing efficient methods for data management and analysis, the sheer size of the data sets can occasionally lead to issues such as downlink-related failures in image capturing and buffering errors. This can potentially limit the volume of successful image acquisitions and ultimately affect the quality and reliability of calibration.

Another limitation of the study pertains to the scope of its application. While the methodologies and tools developed in this research are broadly applicable, they are tailored to the specific case of HYPSON-1 and its mission objectives. Thus, applying these methods to other satellite missions or different hyperspectral imaging applications may require additional customization and validation to ensure optimal results.

Lastly, there is the limitation associated with the use of the Lunar Irradiance Model ESA (LIME) for calibration. Despite our attempts to obtain the complete LIME measurement and model comparison procedure, this was not obtained during the time frame of this thesis. While the Pleiades instrument provides a valuable reference, the absence of a complete model comparison procedure could limit the applicability of our calibration approach.

Acknowledging these limitations is essential for understanding the context and scope of the results and provides a foundation for future work to address these issues. Despite these limitations, the results presented in this thesis contribute valuable insights to the field of hyperspectral imaging for lunar observations, sensor calibration, and satellite automation.

Chapter 6

Conclusion and Future Work

Conclusion

The objective of this thesis was to explore the applications and challenges of hyperspectral imaging for lunar observations, primarily in the context of the HYPSONO-1 mission. The research was directed towards examining how hyperspectral imaging could be used to capture and analyze full moon phases, the technical challenges involved in automating moon captures, and the applicability of these findings to other systems.

The study confirmed that hyperspectral imaging can provide a detailed analysis of lunar surface features and compositions. Importantly, it highlighted the potential for using the full moon phase data as a consistent reference point for calibrating hyperspectral imaging sensors. This finding is particularly significant for improving the quality and accuracy of readings in various satellite missions, beyond the primary objective of observing ocean color along the coast of Norway.

The technical challenges of automation, such as precise lunar tracking and handling large volumes of data, were also addressed. Scripts were developed that could predict celestial body positions and calculate the necessary pointing quaternions to align the sensor accurately. Although some uncertainties remained due to inherent errors in celestial body positioning models and orbital deviations, the implemented solution proved practical and effective for the automation of moon captures.

Moreover, the thesis demonstrated that the methodologies and tools developed could be beneficial for other satellite missions and broader remote sensing applications. These methods, while specifically tailored for the HYPSONO-1 mission, provided a basis for enhancing the accuracy of data collection in a variety of contexts.

However, despite these promising results, the study encountered several limitations that impacted its outcomes. These limitations ranged from uncertainties in lunar and satellite tracking to difficulties in handling large volumes of hyperspectral data, along with the specific tailoring of methodologies to the HYPSONO-1 mission. Additionally, the absence of the complete LIME comparison procedure limited the applicability of our calibration approach.

Nevertheless, the research undertaken in this thesis represents a substantial contribution to the field of hyperspectral imaging for lunar observations. It provides valuable insights into sensor calibration and satellite automation, acknowledging the complexities and challenges involved, and offering practical solutions to overcome them.

Future Work

This study has opened several avenues for future research. Future work should focus on improving the robustness and reliability of the image-capturing and downlinking processes. Enhancements in these areas would reduce data loss, improve image quality, and consequently lead to better calibration results.

Additionally, future investigations are needed to obtain the complete LIME comparison procedure. Access to this information would significantly augment the calibration process, providing a more reliable reference for adjusting the hyperspectral sensors.

The scripts developed in this study for predicting celestial body positions and automating captures could be further refined. Future work should consider improvements in prediction accuracy, factoring in elements such as orbital deviations, perturbations, and potential errors in celestial body positioning models.

Moreover, there is considerable scope for the application of the methodologies developed in this research to other satellite missions and hyperspectral imaging applications. Customizing and validating these methods for different contexts is an interesting path for future research.

Lastly, future work should explore more efficient methods for handling the large volumes of data generated by hyperspectral imaging. This could involve developing improved data management and analysis strategies or designing advanced imaging systems with higher downlink capacities.

By addressing these potential improvements and conducting future research in the outlined areas, we can enhance the accuracy and reliability of hyperspectral imaging for lunar observations, thereby contributing to advancements in this field.

Bibliography

- [1] Hypso, Visited: June 2023. URL <https://www.nanosats.eu/sat/hypso>.
- [2] Sivert Bakken, Marie B. Henriksen, Roger Birkeland, Dennis D. Langer, Adriënne E. Oudijk, Simen Berg, Yeshi Pursley, Joseph L. Garrett, Fredrik Gran-Jansen, Evelyn Honoré-Livermore, Mariusz E. Grøtte, Bjørn A. Kristiansen, Milica Orlandic, Paul Gader, Asgeir J. Sørensen, Fred Sigernes, Geir Johnsen, and Tor A. Johansen. Hypso-1 cubesat: First images and in-orbit characterization. *Remote Sensing*, 15(3), 2023. ISSN 2072-4292. doi: 10.3390/rs15030755. URL <https://www.mdpi.com/2072-4292/15/3/755>.
- [3] Mariusz E. Grøtte, Roger Birkeland, Evelyn Honoré-Livermore, Sivert Bakken, Joseph L. Garrett, Elizabeth F. Prentice, Fred Sigernes, Milica Orlandić, J. Tommy Gravdahl, and Tor A. Johansen. Ocean color hyperspectral remote sensing with high resolution and low latency—the hypso-1 cubesat mission. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–19, 2022. doi: 10.1109/TGRS.2021.3080175.
- [4] Satnews. Ksat to support hypso-1 smallsat with ground station support, Visited: February 2023. URL <https://news.satnews.com/2020/08/11/ksat-to-support-hypso-1-smallsat-with-ground-station-support/>.
- [5] Umamahesh Srinivas, Yi Chen, Vishal Monga, N.M. Nasrabadi, and Trac Tran. Discriminative graphical models for sparsity-based hyperspectral target detection. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1489–1492, July 2012. ISBN 978-1-4673-1160-1. doi: 10.1109/IGARSS.2012.6350822.
- [6] Đorđije Bošković, Milica Orlandić, and Tor Arne Johansen. A reconfigurable multi-mode implementation of hyperspectral target detection algorithms, 2020. URL <https://www.sciencedirect.com/science/article/pii/S014193312030418X>.
- [7] Markus Læg Reid Haldorsen. *HYPISO-1 Data Analysis: QGIS Plugin for Hyperspectral Target Detection*. Project assignment, NTNU 2022, 2022. [Internal document].
- [8] Carmen Quintano, Alfonso Fernández-Manso, Yosio E. Shimabukuro, and Gabriel Pereira. Spectral unmixing. *International Journal of Remote Sensing*, 33(17):5307–5340, 2012. doi: 10.1080/01431161.2012.661095. URL <https://doi.org/10.1080/01431161.2012.661095>.
- [9] Egidio De Benedetto, Wenjing Lv, and Xiaofei Wang. Overview of hyperspectral image classification. *Journal of Sensors*, 2020:4817234, 2020. doi: 10.1155/2020/4817234. URL <https://doi.org/10.1155/2020/4817234>.
- [10] Hongjun Su, Zhaoyue Wu, Huihui Zhang, and Qian Du. Hyperspectral anomaly detection: A survey. *IEEE Geoscience and Remote Sensing Magazine*, 10(1): 64–90, 2022. doi: 10.1109/MGRS.2021.3105440.

- [11] Howard Curtis. *Orbital Mechanics for Engineering Students*. ELSEVIER AEROSPACE ENGINEERING SERIES, 2005.
- [12] Kepler's laws of planetary motion, Visited: February 2023. URL https://en.wikipedia.org/wiki/Kepler%27s_laws_of_planetary_motion.
- [13] Orbital mechanics for engineering students (3rd ed.), 2013. URL [https://books.google.no/books?hl=no&lr=&id=2U9Z8k0T1TYC&oi=fnd&pg=PP1&dq=Curtis,+H.+D.+\(2013\).+Orbital+mechanics+for+engineering+students+\(3rd+ed.\).+Elsevier.&ots=CGXUDJnDyn&sig=wLgPgqccfFswTtaMjV1mG70BUA&redir_esc=y#v=onepage&q&f=false](https://books.google.no/books?hl=no&lr=&id=2U9Z8k0T1TYC&oi=fnd&pg=PP1&dq=Curtis,+H.+D.+(2013).+Orbital+mechanics+for+engineering+students+(3rd+ed.).+Elsevier.&ots=CGXUDJnDyn&sig=wLgPgqccfFswTtaMjV1mG70BUA&redir_esc=y#v=onepage&q&f=false).
- [14] Eberhard Grün. Chapter 34 - solar system dust. In Lucy-Ann McFadden, Paul R. Weissman, and Torrence V. Johnson, editors, *Encyclopedia of the Solar System (Second Edition)*, pages 621–636. Academic Press, San Diego, second edition edition, 2007. ISBN 978-0-12-088589-3. doi: <https://doi.org/10.1016/B978-012088589-3/50038-4>. URL <https://www.sciencedirect.com/science/article/pii/B9780120885893500384>.
- [15] United states space force, Visited: February 2023. URL <https://www.spaceforce.mil/>.
- [16] Brandon Rhodes. Skyfield, Visited: February 2023. URL <https://rhodesmill.org/skyfield/>.
- [17] Pyorbital, Visited: February 2023. URL <https://pyorbital.readthedocs.io/en/latest/>.
- [18] Rotations and euler angles, Visited: March 2023. URL <https://mathworld.wolfram.com/EulerAngles.html>.
- [19] Mario Toso, Ettore Pennestri, and Valerio Rossi. Esa multibody simulator for spacecrafts' ascent and landing in a microgravity environment. *CEAS Space Journal*, 7, 03 2015. doi: 10.1007/s12567-015-0081-5.
- [20] Mark Hughes. Don't get lost in deep space: Understanding quaternions, 2017.
- [21] Python, Visited: February 2023. URL <https://www.python.org/>.
- [22] U.s. naval observatory, Visited: March 2023. URL <https://www.cnmoc.usff.navy.mil/usno/>.
- [23] Astronomical almanac, Visited: March 2023. URL <https://aa.usno.navy.mil/publications/asa>.
- [24] Celestrak, Visited: February 2023. URL <https://celestrak.org/>.
- [25] Ntnu smallsat lab, Visited: February 2023. URL <https://www.ntnu.edu/smallsat>.
- [26] Maciej Neneman, Sébastien Wagner, Ludovic Bourg, Laurent Blanot, Marc Bouvet, Stefan Adriaensen, and Jens Nieke. Use of moon observations for characterization of sentinel-3b ocean and land color instrument. *Remote Sensing*, 12(16), 2020. ISSN 2072-4292. doi: 10.3390/rs12162543. URL <https://www.mdpi.com/2072-4292/12/16/2543>.

-
- [27] Lunar irradiance model esa: Lime, Visited: March 2023. URL <https://calvalportal.ceos.org/lime>.
- [28] Lunar irradiance model documentation, Visited: March 2023. URL <https://calvalportal.ceos.org/lime-documents>.
- [29] U.S. Geological Survey. ROLO Lunar Model and Database. <https://www.usgs.gov/media/files/rolo-lunar-model-and-database>, Visited: June 2023.
- [30] Barycenter, Visited: February 2023. URL <https://en.wikipedia.org/wiki/Barycenter>.
- [31] Brandon Rhodes. Load, Visited: February 2023. URL <https://rhodesmill.org/skyfield/api-iokit.html#the-loader-class>.
- [32] Brandon Rhodes. Earth satellites, Visited: February 2023. URL <https://rhodesmill.org/skyfield/earth-satellites.html>.
- [33] Hanxiang Hao, Sriram Baireddy, Kevin J. LaTourette, Latisha R. Konz, Moses W. Chan, Mary L. Comer, and Edward J. Delp. Improving building segmentation for off-nadir satellite imagery. *ArXiv*, abs/2109.03961, 2021.
- [34] Adriaensen Stefan. *COMPARISON OF THE LUNAR SPECTRAL IRRADIANCE MODEL TO SEVERAL DATASETS*. VITO, 2021. [Internal document].
- [35] Attitude control strategies for an imaging cubesat - scientific figure on researchgate. Available from: https://www.researchgate.net/figure/Reference-frames-ECI-body-LVLH-Target_fig1_334308050. [Accessed 26 May, 2023].
- [36] Ntnu smallsat lab cal-char-corr. URL <https://github.com/NTNU-SmallSat-Lab/cal-char-corr>. Accessed on June 7, 2023.
- [37] Astropy, Visited: June 2023. URL <https://www.astropy.org/>.

Attachments

The entire program structure, including the code for the SatNav repository, is available on GitHub. You can access the repository through the following link: <https://github.com/NTNU-SmallSat-Lab/SatNav>

Additionally, you can download a compressed file of the repository by clicking on one of the paperclip icons below. There is a paperclip icon for the file type '.zip' and another paperclip icon for the file type '.7z'. This is because different PDF readers support different file formats. To determine which one works with your PDF reader, you can double-click on the paperclip icon and see if it opens or not. If none of them work, it is recommended to download **Adobe Acrobat Reader DC**, as it is the most common PDF reader and it has been confirmed to allow attachments and downloads of files with the '.7z' file type.

Click on the paperclip icon to download the repository as a .zip file: 

Click on the paperclip icon to download the repository as a .7z file: 



 **NTNU**

Norwegian University of
Science and Technology