Considerations about PeakSat Optical Operations

author

January 30, 2026

Contents

Preface

PeakSat is an 3Ù CubeSat, developed by the SpaceDot team in the Aristotle University of Thessaloniki. It's primary Mission Objective is to perform a Free Space Optical Communication Link from Low Earth Orbit (LEO). What this means in simple words, is for the satellite to send a message to the ground, using a laser, while moving at a speed of approximately $8/, kms^{-1}$. The ground receiver of the laser is called an Optical Ground Station (OGS), and is practically a telescope, with a laser receiver. We will perform a detailed discussion on the mission objectives in section 1.2.

At the time of writing, PeakSat has successfully passed all the testing phase, and is currently waiting to be launched in a closed box.

In this thesis, our main objective is to present a small part of the work that has been done with the satellite, and is directly related to the Optical Operations. In order to understand the methods discussed and their necessity, we need a more spherical knowledge of the satellite. So, we have tried to split the work and discussed in this thesis in 5 relatively independent parts. We tried to keep a linear presentation structure, starting with the satellite introduction, then presenting the Optical Payload and the ADCS, showcasing the alignment measurement performed to bridge the gap between the 2, and finally combining all the above together to construct in-orbit scenarios. However, there are some section, especially in the ADCS chapter, that will require knowledge of the Alignment methods to be fully understood.

To explain in more detail the structure of this work, we start in chapter 1 by introducing the basics of the satellite, it's components the mission objectives, and finally the basic properties of it's orbit, and how can we use them for the optical operations. Then, we continue in chapter 2 by discussing what an optical payload is, some specific details about PeakSat's optical Payload and testing that we have performed to it's Optical Components. Subsequently in chapter 3 we perform a similar introduction to Attitude Determination and Control Subsystem (ADCS), discuss the ADCS of PeakSat and how to configure it. We close chapter 3 by discussing configuration performance simulations that were performed, and explain why are these critical for performing a successful optical pass. In chapter 4, we explain in detail the Misalignment Measurement experiment that we designed in order to be able to characterize the Angular Divergence between our ADCS and Optical Payload, and showcase this measurement's importance. Finally, we close this work in chapter 5, by bridging all the aforementioned topics into creating flight representative Mission Tests, as well as real Operations Optical Pass planning.

Several algorithms and procedures explained here have been programmed into simple `python` notebooks. More details about all the codes used can be found in table 1.3, and if a notebook or script exists for a specific section, it will be explicitly mentioned inside it.

Περίληψη στα Ελληνικά

TODO

Ευχαριστίες

TODO

Chapter 1

Introduction

> Για να μπορέσουν να εκφράσουν την σύγχυσή τους τα άτομα που εργάστηκαν στον δορυφόρο του ΠικΣατ, σχετικά με το πώς λειτουργεί, έφτιαξαν το παρακάτω νοητικό πείραμα. Έστω ΠικΣατ αυτήν την στιγμή βρίσκεται κλεισμένος σε ένα κουτί, και είναι ταυτόχρονα σε μία κατάσταση Ζωντανός-Νεκρός, με πιθανότητα ακριβώς 50-50. Δεν μπορούμε να γνωρίζουμε τι ισχύει, μέχρι να ανοίξουμε το κουτί.
>
> *αστικός μύθος*

1.1 Introduction

As already mentioned, PeakSat is 3U satellite. It consists of: The On-Board Computer (OBC) and Communications board (COMMS) that are developed in-house. The platform includes a deployable UHF antenna, a GNSS antenna, that mainly interface with the Communications Board. As every satellite, it has an Electrical Power System (EPS) and Solar Panels in all the long faces. It also has an Attitude Determination and Control Subsystem, (ADCS, chapter 3) and an Optical Payload (chapter 2) that are discussed with more detail in this work. Illustrations of the above subsystems can be seen in figs. 1.1 and 1.2.

The main objective of PeakSat is to perform an optical data downlink to the Holomon OGS. An optical downlink attempt is performed during an optical pass, or the point of the orbit that the satellite is visible from the OGS. More about the orbit and possible optical passes can be found in section 1.3.

From the platform side, a stable optical link requires a very high precision of position and orientation knowledge of the satekkute, as the downlink laser beam of PeakSat has a divergence of only about 8 arcseconds. This precision is almost impossible to reach using ADCS sensors, so in order for the downlink laser to be reliably directed to the OGS, the OGS will send a beacon laser to PeakSat. When the Optical Payload Beacon Camera (BC), detects laser, it can infer the apparent orientation of the OGS. Using this information, it can steer the downlink laser to the desired direction, using a Fast Steering Mirror (FSM). A very artistic illustration can be seen in

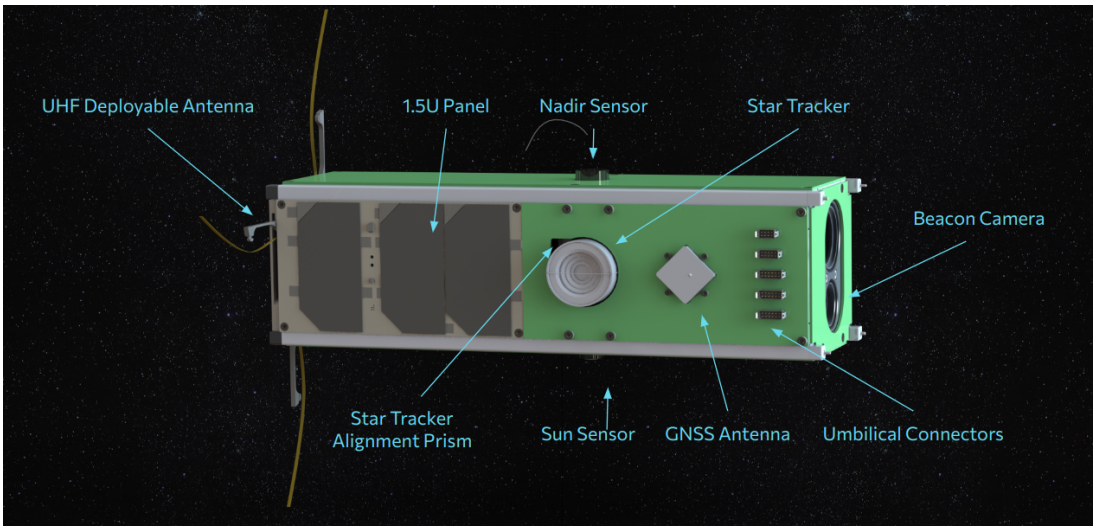

Figure 1.1: Overview of PeakSat. On this picture, the beacon camera, the star tracker, and the alignment prism of the star tracker can be seen. TODO: is it OK to use this pic?

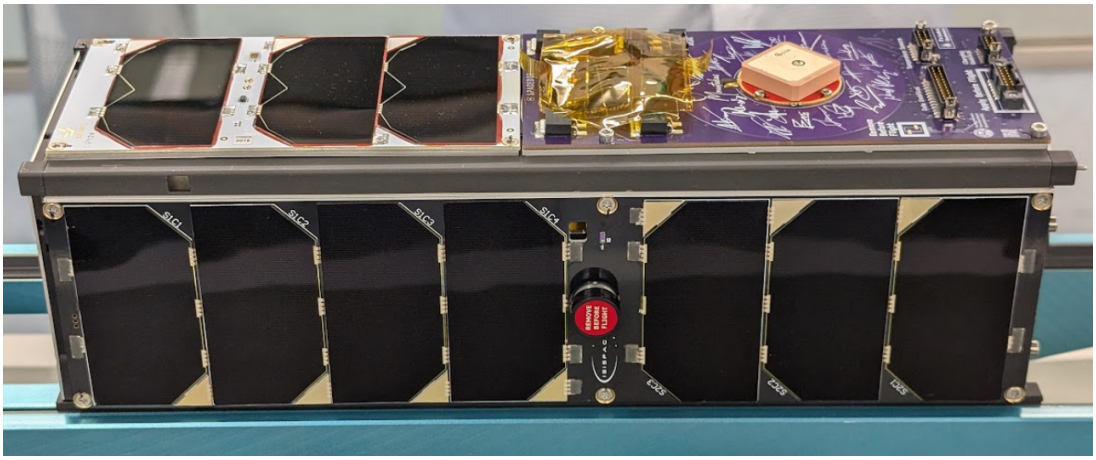

Figure 1.2: Fully assembled PeakSat.

fig. 1.3.

The Beacon Laser - Beacon Camera - FSM steering loop of the down-link laser is a procedure performed by the Optical Payload Provider, Astrolight, (chapter 2) and will not be discussed in detail. It should be obvious though, that we have to ensure that PeakSat can point the Beacon Camera face, towards the ground station with an accuracy of at least the Camera's Field of View (FoV). Moreover, for the link to be stable, the pointing drift also needs to be small.This is understandable if when we consider that there is a control loop with a finite time-step between Beacon Laser Detection and FSM steering. If into this finite time-step, the pointing drifts by an angle larger than the downlink laser divergence, then the downlink data stream will be unstable, or the link will be lost.

As we have already spoken about satellite orientation, this is completely natural to bring the ADCS into the discussion. The ADCS performs the satellite's attitude (the terms attitude and orientation will be used interchangeably throughout this text) estimation using a bunch of sensors, and controls the attitude using actuators (section 3.3). As platform integrators, we have to ensure that the sensors can provide accurate enough estimations, and also that the control algorithms can rotate the satellite adequately. We can check this, if both the absolute pointing

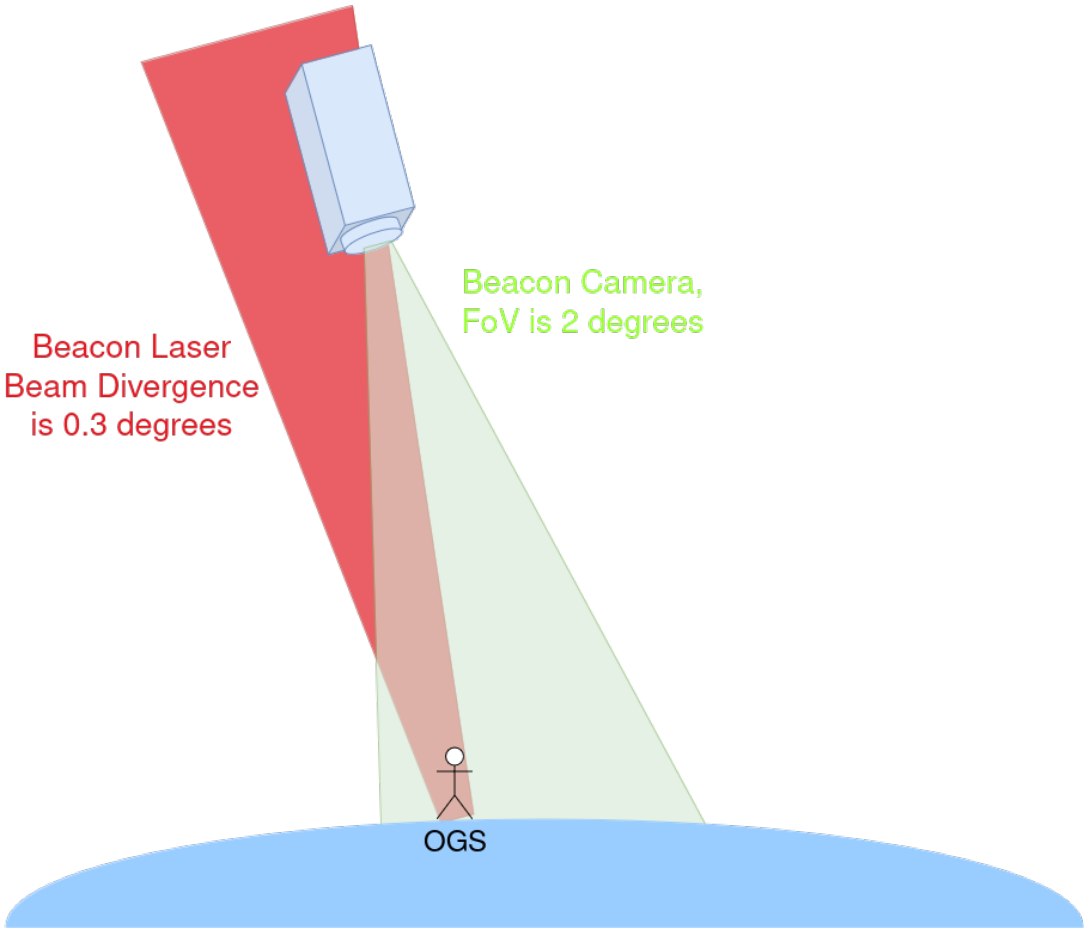

Figure 1.3: Overview of PeakSat. Realistic illustration of PeakSat Point Acquisition and Tracking, where the FoV of the Beacon camera and the OGS beacon laser are visible.

requirement that comes from the BC FoV, and the relative pointing requirement that comes from the BC-FSM steering loop, are satisfied. This pointing performance can depend both on the hardware (sensors, actuators) used, but also in the software ADCS configuration like the control or estimation algorithms to be used, and different parametrizations that can be passed to control and estimation algorithms. At this stage, we can only assess this pointing performance through a series of simulations, so we can be ready for when actual data come, and know what to expect from different ADCS parameter sets. *The ideas presented in the above paragraphs are discussed throughout this thesis in much more detail, primarily in chapters 2 and 3*

Orientations produced by the BC, and by the ADCS will of course be expressed in their own coordinate system. Even though we can have a theoretical transformation between the 2 from the design of the satellite, mechanical tolerances will induce an error to this transformation. Bad knowledge of the ADCS orientation and Payload orientation can result to mis-pointing of the BC. Moreover, during an optical pass evaluation, we will do a post processing of the satellite telemetry, and for this we will need all orientations produced by the satellite to be in the same coordinate system with the best possible accuracy. The above are some of the reasons why we have to perform the Static Misalignment Measurement, that is explained in chapter 4.

Even though the structure of the thesis has already been mentioned in **??**, the after reading the above paragraphs we should understand a bit better on why the above considerations are necessary to perform a successful optical pass. We will finish, by showcasing what we already discussed in a few revision bullet-points:

1. The Payload has to be pointed towards the OGS with an absolute pointing requirement, and a relative pointing requirement.

2. The above requirements can be translated into performance limits for the ADCS. This performance can depend both on the hardware (sensors, actuators) used, but also in the software ADCS configuration (control, estimation algorithms, relative parameters)

3. We have to ensure that the ADCS performance is adequate, and that we can seamlessly refer attitude information produced by the Optical Payload to attitude information produced by the ADCS, and vice versa.

4. This is why examine the possible ADCS configurations we can use during an Optical Pass through meticulous simulations (chapter 3) and why we perform the Static Misalignment Measurement (chapter 4).

1.2 Mission Objectives

TODO: reference the PeakSat document.

Table table 1.1 lists the primary and secondary mission objectives. We practically spoke about the primary mission objectives already in section 1.1, but here we can review them more clearly. In this thesis we will mainly present work that is related to the primary objectives 1,2,4, even though inevitably through the optical link attempts also the evaluation of Holomodas OGS will be performed.

We can take this chance to mention that one of the difficulties of the mission is that both the satellite, and the OGS will be non-validated, so it can be non trivial for the operators to decide on which subsystem needs fine tuning in a possible unsuccessful optical link attempt. It is also showcased that we need to perform the misalignment characterization (chapter 4) so we can narrow down the number of failure points in the Optical pass operations.

Finally, in this work we will not explore the 2nd secondary objective, regarding indirect optical links, neither the first regarding the validation of the in-house OBC and COMMS. We will only mention that there can be several unexpected scenarios, in a satellite that tries to validate its OBC, COMMS, and Optical Payload on the mission.

Table 1.1: Mission Objectives

Number	**Primary Mission Objectives**
1	The mission shall achieve optical communications connection between the spacecraft in LEO and OGS in Greece
2	Evaluate performance of the spacecraft optical terminal
3	Evaluate performance of the Holomondas OGS
4	Evaluate ground and space segment optical link performance under various elevation angles and weather conditions
Number	**Secondary Mission Objectives**
1	Validate in orbit university developed hardware and software
2	Establish indirect optical link between Holomondas and another OGS

1.3 Orbit

PeakSat will be in a Sun-Synchronous Orbit (SSO) at an altitude of 510km. The lift-off date will be no earlier than 29 of March 2026. An SSO orbit, is practically a type of orbit that by fine-tuning the satellite altitude (semi-major axis) and inclination the orbit plane (longitude of the ascending node) changed with approximately $1°/day$, or with the same angular velocity as the Earth's revolution around the Sun. This happens because to the oblateness of the Earth at the equator. At a first approximation can be modeled by assuming the earth is not a perfect sphere, but express its shape using spherical harmonics, and more specifically the J2 harmonic. More about the analytical solution of the effect the J2 perturbation has in the orbital elements can be found in Fitzpatrick (2012), and specifics about SSO orbits in Curtis (2012).

For practical applications, we can note the 2 main advantages of SSO orbits. Firstly, they provide global coverage, and secondly, they provide periodic eclipse periods throughout the mission. This creates a more stable environment, which simplifies the mission budgets, and operations.

Parameter	Value	Alternative	Alt. Value
Date	29 Mar 2026	Epoch	26088.00
Altitude	510 km	Semi-major axis	6881
Eccentricity	0.0001	...	...
Inclination	94.4028°	...	...
LTAN	$14h \pm 1$	RAAN	39.7161°

Table 1.2: Mapping between document sections and corresponding code

As a final comment, we can show an intuitive way to convert between the Local Time of the Ascending Node (LTAN), and Right Ascension of the Ascending Node (Ω). For a sateellite Ω denotes the angle on the equator, between the equinox position γ, and the intersection of the orbit and the equator plane that the satellites moves towards the north hemisphere. The LTAN, denotes the local time down in the earth, when the satellite crosses the equator and moves towards the north hemisphere. For SSO orbits, it is much more convenient to use LTAN, as it is a slowly varying variable, unlike Ω that changes significantly every day.

1.3.1 Passes

Finally, we show some simple orbital analysis that will also be necessary in the Optical Pass planning, and the general mission. We are very interested in knowing how much times we expect to have a pass opportunity. Moreover, we are interested in the maximum altitude of each pass, because it is connected both with atmospheric loss, but also the pass duration. We perform a simple analysis (pass_LTAN_tleprop.ipynb) where we propagate different TLEs, for the different possible launch LTANs of the satellite. Results for our analysis can be found in fig. 1.4, where in the x axis we show the possible LTAN of the satellite, and in the Y axis the maximum expected altitude. The propagation was performed using orekit.

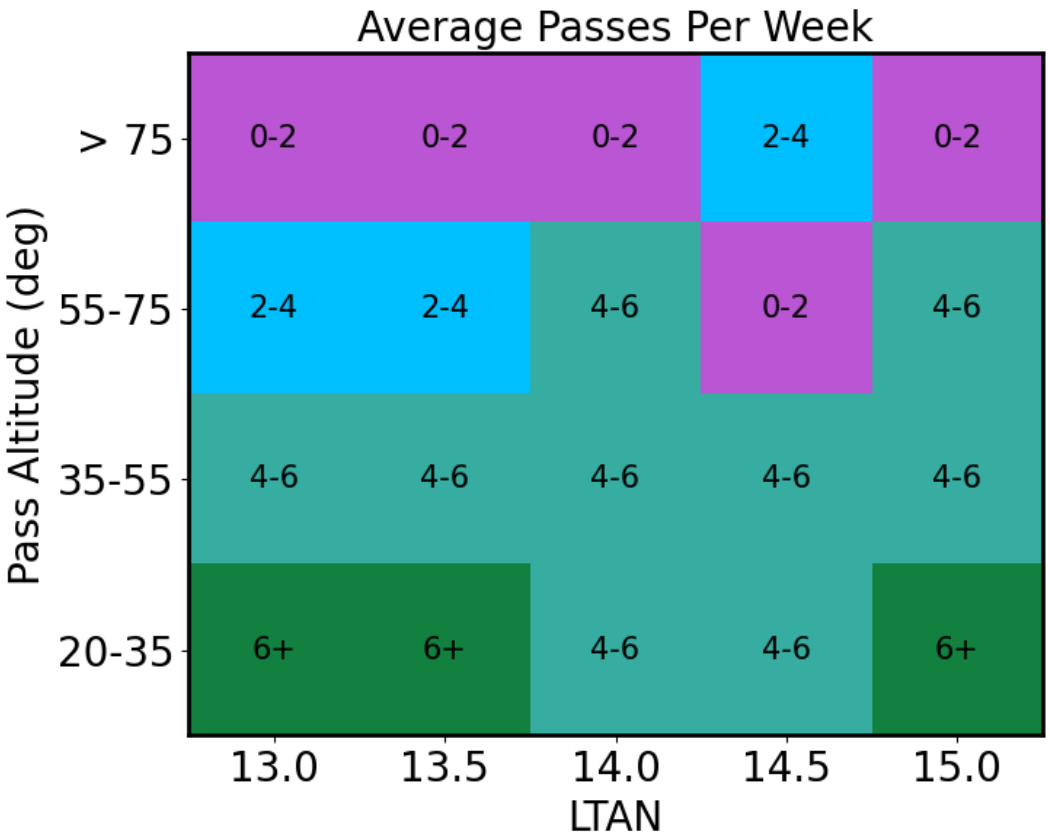

Figure 1.4: Expected number of passes above HolOGS, for the first week after the launch (29 March 2026), as a function of the possible Local Times of the Ascending Node.

1.4 External repositories

In this section, we mention external repositories that have been used in this work.

- `NumPy` Harris et al. (2020)
  https://github.com/numpy/numpy
- `SciPy` Virtanen et al. (2020)
  https://github.com/scipy/scipy
- `AstroPy` Collaboration et al. (2018)
  https://github.com/astropy/astropy
- `Orekit` Maisonobe et al. (2010)
  https://github.com/CS-SI/Orekit
- `pandas` McKinney et al. (2023)
  https://github.com/pandas-dev/pandas
- `Matplotlib` Hunter (2007)
  https://github.com/matplotlib/matplotlib
- `Plotly`
  https://github.com/plotly/plotly.py

1.5 Codes Used

Section	Code Repository	Short Description
Orbit	pass_LTAN_tleprop.ipynb	Simple TLE SGP4 propagation for predicting visibility windows.
Orbit	TLE_elements.ipynb	Mean-Osculating elements transformations, and TLE generation from single point in orbit.
ADCS Coordinate Systems	pass_LOS_error.ipynb	Plotting the Line of sight error drift, from line of sight error.
ESTEC results	calc_misalignment.ipynb	Calculate the final vector of the Payload reflective surfaces, in the ADCS frame, using the theodolite measurements.
BC-BCRS Characterization	00_read_images.ipynb	Fit images captured with the BC and a collimated laser, and fit simple Gaussian Centroids.
Input to the ADCS	str_mountconfig_.ipynb	Calculating the possible Star Tracker mounting configuration to take into account the misalignment measurement.
Rotation Matrices	rotation_matrices.py	Defining 3D rotation transformation matrices and conversions to Euler Angles.

Table 1.3: Mapping between document sections and corresponding code

Chapter 2

Optical Payload

Ένας Λιθουανός και ο Ζέρ μπαίνουν σε ένα μπάρ. Ο μπάρμαν Άτλαντας φτιάχενι ένα κοκτέιλ και το δίνει στον Λιθουανό. Ο Ζέρ το βλέπει και ρωτάει τον Λιθουανό:
- Γιά πες, πως είναι το ποτό, λέει καθόλου;
- Λέει-Ζερ!

ανέκδοτο

2.1 ATLAS terminal

TODO: is anything here confidential?

Figure 2.1: ATLAS terminal. Tx/Rx apperture is on the right, beacon camera is on the left

PeakSat will use ATLAS-1 terminal, that is an optical terminal designed by the Lithuanian start-up company Astrolight, capable of performing Free Space Optical communication from Low Earth Orbit. ATLAS terminal promises to deliver an 100 Mbps download data rate at a wavelength of 1550 ± 3nm, and an 1Mbps uplink data rate at a wavelength of 1535 ± 5nm. The Atlas down-link laser beam will be referred to as the transmit beam, Tx, and the up-link laser beam will be reffered to as recieve beam, Rx. The optical system of the Tx consists of: 1) Laser Diode and a Single Mode Optical Fiber 2)A Signle Mode Fiber Collimator, used to parallerize the light rays 3) Fast Steering Mirror (FSM) used for precise pointing of the beam, and and 4) optical telescope used for further collimating the free space light beam. The telescope has an aperture size of 32.4 nm. (fig. 2.1, the bigger apperture on the right). The Tx is an unpolarized beam, with a power of 0.1W and a beam divergence of 76.5μrad FWHM. ATLAS Rx, consists of the same components as the Tx, however, the laser light ends up in an Avalanche Photodiode (APD). In more detail, the incoming laser beam is 1) Focused through the telescope 2)Pointed with the FSM 3)Coupled to the fiber through the SM Fiber Colimator 4) Passed through a circulator 5)Detected by the APD. The Rx System can receive unpolarized light with a sensitivity of $41\mu W/m^2$.

The smaller aperture on the right of fig. 2.1 is the Beacon Camera (BC), as its main function is to detect the beacon laser that will be transmitted from the OGS location. The purpure of the beacon camera, is to determine the direction of the OGS with respect to the ATLAS terminal, with an accuracy of a few arcseconds. It consists of a) Spectral filters, with transmittance only

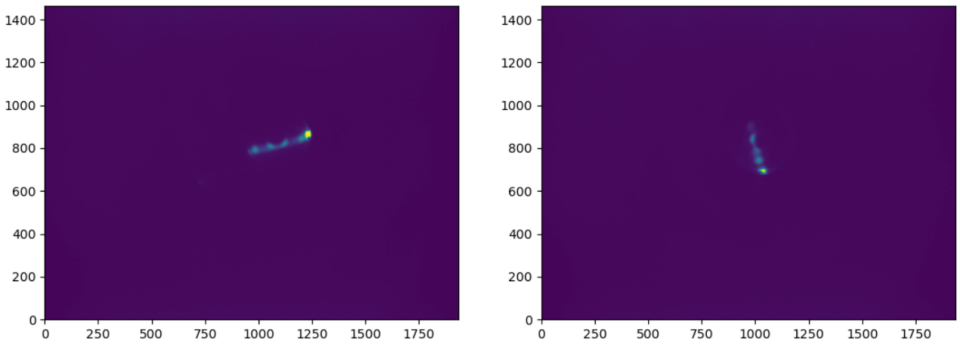

Figure 2.2: ATLAS yaw measurement.

at 808 ± 3nm, b) Imaging optics, that are lenses that will focus the beacon laser light, and c) a CMOS sensor. The CMOS sensor has a pixel resolution of 1464 × 1936 and it's pixel size is 4.5μm. The Beacon Camera has a field of view of 2.6° × 3.4°, resulting in a pixel scale of 5″, and an irradiance of 0.1$\mu W/m^2$ at exposure times of 10ms

For alignment purposes, a coordinate system for the BC needs to be defined.The Z-axis, can be defined as the principal axis of the lens system, and x, y axes as the X-Y position of the pixels in the sensor.

Defining a coordinate system at the Tx Laser is not so straightforward. The Laser beam, is reflected to an FSM before exiting the satellite, in order to control its direction. This FSM position, and the final direction of the laser beam can be used to define a coordinate system for the laser beam. Exact representation is outside the scope of this work, as the characterization of the laser beam and the beacon camera will be done by Astrolight.

2.2 Misalignments

ATLAS consists of a Beacon Camera and a downlink laser, with an FSM that controls its orientation. The FSM zero position is at Beacon Camera pixels x = 918.1 pix / y = 804.1 pix, and all orientations coming out of ATLAS are referenced with respect to the Beacon Camera coordinates (BCC, in pixels) and this center. The conversion from pixel to orientation can be considered linear, and the pixel scale is 31.5 µrad/pixel. However, based on the ATLAS parameters file (pixel size = 4.5 µm, focal length = 143.1 mm), this conversion can be calculated as 31.3 µrad/pixel, which is a non-negligible difference, especially when several pixels are accumulated. Moreover, the BC is tilted by 11.5 degrees with respect to the ATLAS CAD Coordinate System (practically the sides of the cube). We can see this in section 2.2, where we had a laser mounted on a theodolite, that was approximately parallel with the ATLAS Cad coordinate system. The laser was traversing horizontally and vertically across the camera aperture, and with an exposure time of 10 s produced the pictures we see.

It will be essential useful during in-orbit misalignment characterization to know exactly the detected blob positions and corresponding FSM orientations. The beacon-detected positions can be found in the parameters "CameraBeaconCoordinatesX" and "CameraBeaconCoordinatesY". Both can be obtained either from the pat.log file or from the FPGA telemetries, and they will be timestamped. Moreover, we need the orientation of the down-link laser direction. It is recommended to use the same beacon-detected position as the reference, since this is where Tx should point if there are no misalignments. If a spiral scan is performed (to characterize BC and Tx misalignments, check section 2.2), then "Xpix", "Ypix" will denote the offset between the beacon-detected position and the FSM orientation in pixels. If the FSM is spiraling for

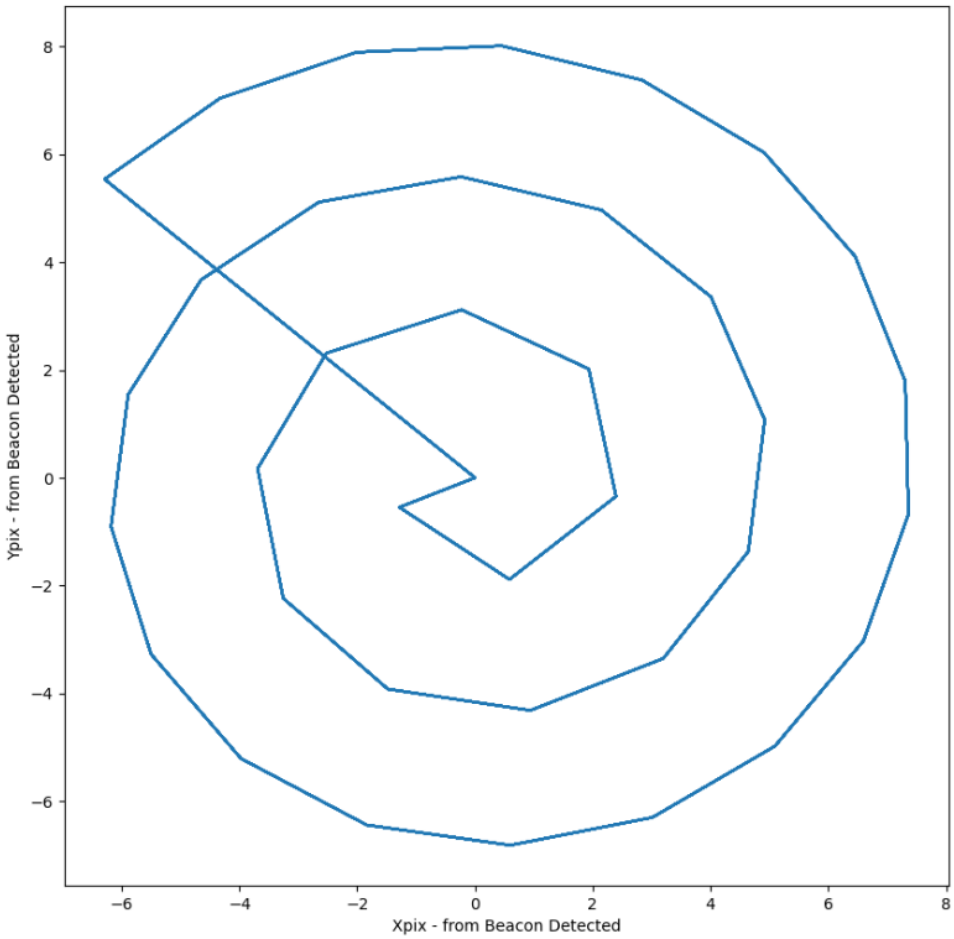

Figure 2.3: Example of Tx FSM Spiraling

characterizing its misalignment with the BC , to get the Tx orientation, we will add: “CameraBeaconCoordinatesX” + “Xpix” etc.

Having said this, we can take the opportunity to explain how we can perform the BC–Tx misalignment characterization.

From telemetry, we have to observe that when the beacon is detected by PeakSat, the OGS cannot detect the downlink laser. We can, to some extent, rule out this being a fault of the OGS (the very famous “chicken and egg” problem).

We decide to perform a spiral search. We configure several spiraling parameters, like the spiraling step, the spiraling period, the spiraling overlap. This will be passed on to the satellite, and during our next optical pass, we observe that we briefly see the downlink laser at time Ts. When downloading the pat.log file from that optical pass, we will see at Ts (time should have been synced using GNSS and PPS) what the xpix, ypix values were in pat.log. These, in theory, should indicate the static BC–Tx misalignment. If we trust these parameters, we can then set the static offset misalignment equal to the corresponding xpix, ypix values.

2.3 Beacon Camera Properties

The Beacon Camera receives the beacon laser. By default, when the BC detects a blob, it will crop the full-frame image to a 64×64 ROI image, with a much faster refresh rate (10 Hz full frame, 100 Hz ROI). The irradiance received by the beacon cannot be directly calculated.

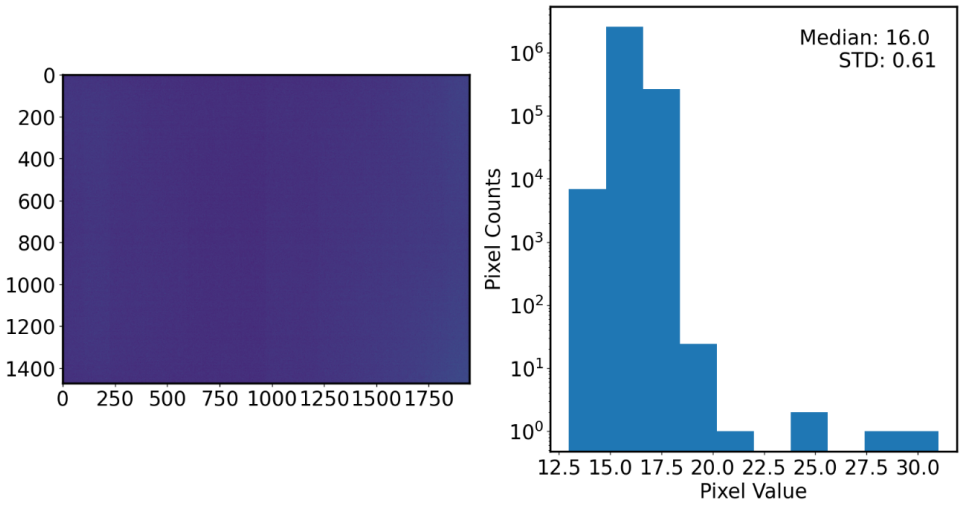

Figure 2.4: BC image, without the protective cup on, and an exposure time of 3 *s*.

However, assumptions can be made based on maximum and mean pixel values of the camera. Moreover, hot-pixel and blob detections are performed by a proprietary algorithms incorporated into the FPGA, which finds the beacon in the image based on several factors (maximum pixel value is not one of them). As one of its key steps, the algorithm uses low-pass filtering, so hot pixels and random shot/radiation noise are not a problem. Of course, if a hot pixel exists, then maximum pixel intensity will not reflect any illuminated power; however, this does not affect the beacon detection.

We have taken several flat and dark images in different exposure times, in order to understand how to operate the BC, and its performance during different conditions. As an example, in section 2.3 we present a flat image taken with the protective cup of the camera on. We observe a mostly homogeneous field, with a few hot pixels (the hot pixels are more visible in higher exposure times). It is finally worth noting that this specific image, needed approximately 6 minutes to download, and is one of our first 3 full frame images we downloaded with the BC.

Chapter 3

ADCS

> Our attitudes control our lives. Attitudes are a secret power working twenty-four hours a day, for good or bad. It is of paramount importance that we know how to harness and control this great force.
>
> Irving Berlin

3.1 Introduction

The Attitude Determination and Control Subsystem (ADCS) is a critical component of a satellite, because as the title says, it controls it's orientation into space. The pointing requirements of different missions can vary significantly. For missions like PeakSat, that aim to perform optical communication experiments from Low Earth Orbit, it is critical to have a reliable and stable ADCS. In this chapter, we will discuss start by briefly presenting PeakSat ADCS (section 3.2) and then mention the different Sensors and Actuators used by our system (section 3.3). We will close section 3.3 by disusing some important considerations about the spin stabilization of the satellite - something that is essential to have before attempting an optical pass. We will continue by presenting the coordinate systems used by the ADCS (section 3.4) in order to be able to understand the ADCS mounting configurations (section 3.5), or the way to tell to the ADCS algorithms the physical orientation of its sensors. Understanding this topic will be very useful for the Optical Pass post-processing analyses (section 3.7). Finally, we will close this chapter, by explaining the different methods on how the misalignment measurement (explained in detail in chapter 4) can be passed in the ADCS.

3.2 PeakSat's ADCS

CubeSpace ADCS Gen2 is the ADCS used for the PeakSat Mission.

The configuration selected for PeakSat, includes the following components:

- CubeComputer
- 2x Gyroscopes
- 2x Gen2 CubeMag Compact Magnetometers

- Gen2 CubeSense sun sensor
- Gen2 CubeSense earth sensor
- TY-Space PTS3S-K4 Star Tracker
- 3x Magnetorqures
- 3x Reaction Wheels

The components will be discussed in more detail in the next Section.

3.3 ADCS Components

In order to understand the optimal ADCS configuration to perform an optical pass, we initially need to understand a few concepts about the ADCS.

3.3.1 Actuators

PeakSat ADCS consists of 2 types of actuators, the magnetorquers, and the reaction wheels. The first can exercise a torque to the satellite by interacting with the Earth's magnetic field, and the latter changes the angular velocity and orientation of the satellite, by exchanging angular momentum with it. The magnetorquers can produce a torque of up to TBD, whereas the reaction wheels TBD.

3.3.2 Sensors

The most accurate attitude estimation sensor of PeakSat is the TY-Space star tracker (PST3S-K4). The star tracker camera can measure the orientation of the satellite with an accuracy of $5''3\sigma$, at angular rates $< 1^\circ/s$, and $8''3\sigma$ at angular rates $< 0.5^\circ/s$

PeakSat ADCS uses 5 types of sensors to find its orientation and rotation state. These are:

- The Gyroscopes, that can find the angular velocity of the satellite with respect to an inertial frame of reference, relative to an initial gyroscopic bias. Gyroscopes can work at any condition, and they do not need any model references to find the satellites angular velocities.
- The Magnetometers, measure the Earth's magnetic field. By having only this measurement the ADCS can not conclude much about the Position and Orientation of the Satellite, however by having an accurate position knowledge (e.g. by a good TLE, or GNSS data) the ADCS can correlate the measured magnetic and find the satellites orientation. Magnetometers can measure earth magnetic field vectors at all times.
- The Earth Sensor, that is practically an IR camera, that can detect the Earth Horizon. This can detect 2 angles TBD. The Earth Sensor can generally take valid measurements as long as the Earth is in its FoV.
- The Fine Sun Sensor (FSS) that can detect the sun within a field of view of approximately 180 degrees full cone. By having accurate Time, and Position knowledge, the ADCS can correlate the modelled sun-vector with the measured sun vector and find the Orientation of the satellite. In general the Sun Sensor can detect the sun, as long as the sun is in its FoV.

3.3.3 Star Tracker

The star tracker is the Sensor that is the most mentioned in this document, so it deserves it's own subsection. It is by far the most accurate sensor of the satellite, that can find the orientations by taking images of star fields, with an arcsecond accuracy. The Star Tracker is the only sensor of the satellite that can provide adequate accuracy in order to be within the limits of the Optical Pass pointing requirements. To understand Star Tracker's estimation accuracy, we need to understand that it mainly originates from the detected positions of the stars in the Star Field. We can define 2 types of accuracy.

- The **pointing accuracy** of the star tracker as the accuracy to find the camera main axis vector (vector that is vertical to the camera plane, and passes from its center). If ps is the pixel scale, an angular change $\Delta\theta$ of this vector will result in a shift in the positions of the stars ΔN to be $\Delta N = ps \times \Delta\theta$. All the stars will change by the same amount of pixels, so it is easier for the Star Tracker to understand such a change.

- The **rolling accuracy**, which is the rotation angle of the orientation of the Star Tracker, around its primary vector. If ps is the pixel scale, and $\Delta\theta$ is a change of the rolling angle, this will result in a shift in the positions of the star ΔN to be $\Delta N = \sin\phi \times ps \times \Delta\theta$, where ϕ is the angular distance from the center of the Star Tracker. A star Tracker will typically have an FoV of $\approx 20°$, which means that it will result in a significantly smaller shift in the detected positions of of the stars in the camera, meaning that it is harder for the Star Tracker to understand such a change, and the rolling angle will have a higher uncertainty.

PeakSat Star Tracker manufacturer claims a pointing uncertainty of $5''$, and a rolling uncertainty of $50''$. We ought to keep this difference in mind, because this will likely to affect the satellites pointing performance, especially during the Optical Passes, and we will possibly have to choose weather to roll, or point the Star Tracker during the pass.

As we will see in chapter 4, we need to align the Star Tracker with the BC. For this, a Cubic Prism is mounted on the Star Tracker body. The sides of the cube can be used to create an orthogonal reference system. The cubic prism and the Star tracker is characterized by TY-Space, and the transformation matrix between the Star Tracker Coordinate system, and the Cubic Prism has been provided.

3.4 ADCS Coordinate Systems

3.4.1 Orbit reference coordinate (ORC)

The orbit reference coordinate is defined as follows:

- X_{ORC} is defined by the flight direction, meaning the velocity vector of the orbit.

- Y_{ORC} is defined by the Orbit Anti-normal direction - that is the vectors that are vertical to the orbit vectors. This vector can be found as the external product of the position and velocity of the vector

- Z_{ORC} is defined by the Nadir direction, and can by found by the opposite of the position vector.

For a circular orbit, this coordinate system can be considered orthogonal.

3.4.2 ORC to ECI transformation

In several cases we will need to transform quaternions provided by the ADCS in ORC, to ECI, in order to refer them with other used vectors. The transformation is defined below, based on the SGP4 propagated positions, that are used by the ADCS. We start with the velocity direction:

$$X_{ORC} = V_{satellite}$$

continue with the nadir direction:

$$Z_{ORC} = -P_{satellite}$$

take their cross product:

$$Y_{ORC} = -X_{ORC} \times Z_{ORC} = V_{satellite} \times P_{satellite} =$$

and calculate it.

$$\begin{pmatrix} v_y p_z - v_z p_y \\ v_z p_x - v_x p_z \\ v_x p_y - p_x v_y \end{pmatrix}$$

It is trivial to show that the final rotation matrix will be:

$$R_{ORC->ECI} = \begin{pmatrix} v_x & v_y p_z - v_z p_y & -p_x \\ v_y & v_z p_x - v_x p_z & -p_y \\ v_z & v_x p_y - p_x v_y & -p_z \end{pmatrix}$$

3.4.3 Spacecraft body coordinates (SBC)

We can express it using $X_{SBC}, Y_{SBC}, Z_{SBC}$. We can practically define it as we want. When the roll, pitch, yaw angles are zero, then the SBC has the same orientation as the ORC. The ADCS defined SBC must be the only coordinate frame that is considered when specifying sensor or actuator mounting configurations, and when interpreting attitude angles. It is very useful, because as we will see later, we use this coordinate system, to refer the sensors mounting configurations. Moreover the relative misalignments discussed in chapter 4 will be finally expressed in this coordinate system. For a more complete understanding of the SBC, section 3.5 should be checked. The graphical representation of the SBC can be seen in fig. 3.1.

3.5 ADCS Mounting Configuration

The position and the orientation of the Actuator and Estimator components of the ADCS is controlled through the ADCS mounting configuration. The mounting configuration of each component is expressed with respect to the Satellite Body Coordinates (SBC) - and the SBC is defined by the mounting configurations of all the components. Moreover, the Ground Tracking vector of the satellite is also expressed in the SBC. Regarding the static misalignment, the 2 main quantities of our interest are the Mounting configuration of the Star Tracker, and the Ground Tracking Vector. The final misalignment measurement will be passed to the ADCS as a function of these 2, meaning that for example a misalignment of 1° can change the Ground Tracking Vector by 1° with respect to the SBC, or it can change the the orientation of the

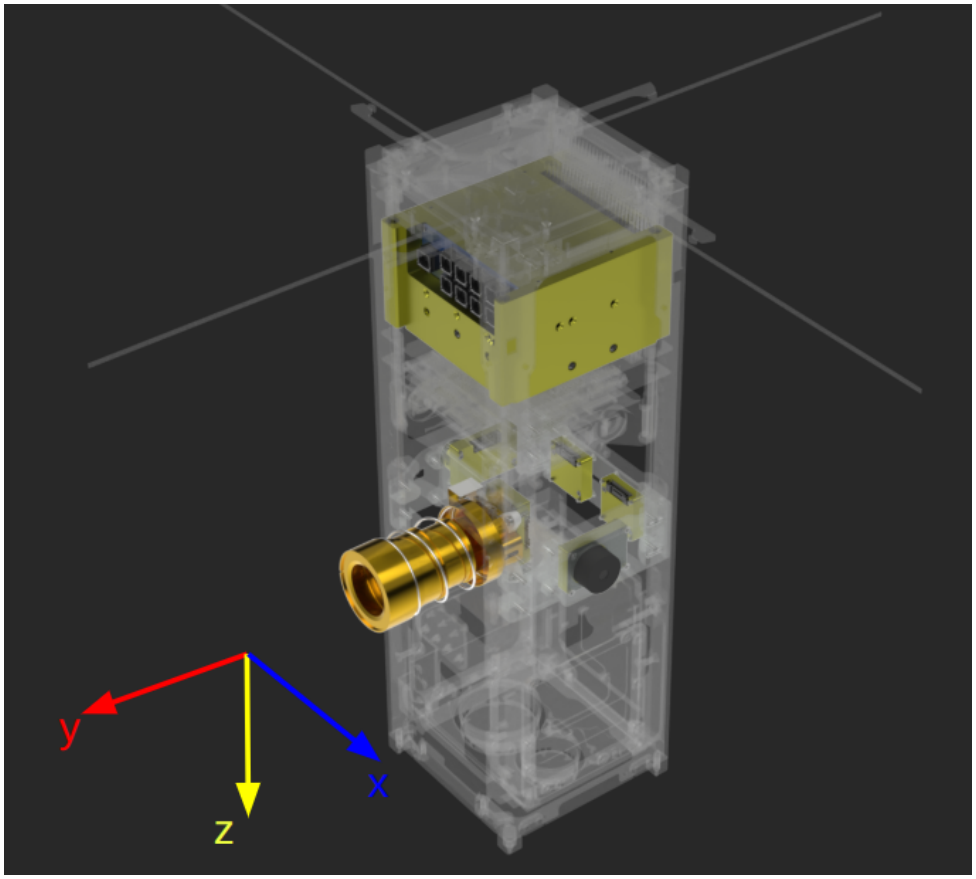

Figure 3.1: ADCS CAD, with highlighted ADCS sensors, and SBC plotted.

STR by the same amount. Note that all these values will be configurable in-orbit, to allow for better calibration and fine-tunning of these parameters to maximize the Point Acquisition and Tracking (PAT) performance.

The mounting of CubeSpace sensors is specified using a Euler 3-2-1 intrinsic angle sequence. The resulting transform will rotate vectors in the local sensor coordinate frame to SBC. The three angles are labeled as: alpha (α), beta (β), and gamma (γ), and they are applied in the same order: α is applied first, around the sensor local Z-axis. β is applied next, around the newly transformed Y-axis, and γ is applied last, around the newly transformed X-axis. Considering the above explanation describes an intrinsic rotation (appendix A.2), in order to mathematically compute the transformation matrix of a sensor to the SBC, we have to use an 1-2-3 extrinsic sequence. The mounting configurations of all the sensors and actuators of the ADCS has been performed using these guidelines, tested through simulations, and validated by the ADCS provider.

3.5.1 Attitude Orientation Format

CubeADCS, when expressing the output quaternion or Euler angles, for some reason follows an Euler 2-1-3 convention for roll, pitch, yaw. What this means is that the attitude of the satellite, is defined by intrinsic rotations, with respect to the ORC (section 3.4) coordinate system.

This means that, in order to describe an intrinsic rotation, we:

1. rotate through y (pitch)
2. rotate through the new x (roll)
3. rotate through the new z (yaw)

We still have to keep in mind, that in order to construct the ORC to SBC rotation matrix (appendix A.2), we have to use a 3-1-2 rotation sequence.

3.5.2 SBC (again)

After we understand what the SBC is, and what the mounting configurations are, we will revisit this subject, in order to gain some deeper insights on it. The SBC is the internal way of the ADCS to know its sensors and actuators orientation. There is no mechanical and physical reference to it, just all the sensors and actuators mounting configurations have to be configured in a way that is consistent with each other.

To better understand this, lets take as an example fig. 3.2, that displays the Star Tracker, and the Sun Sensor (and the ATLAS Payload is on +Z axis, facing downwards). Geometrically speaking (and using for simplicity only the 2 sensors displayed above) these 4 Statements would be equivalent:

- STR mounted on +Y, SS on -X (as in image)
- STR on +X, SS on +Y
- STR on -Y, SS on +X
- STR on -X, SS on -Y

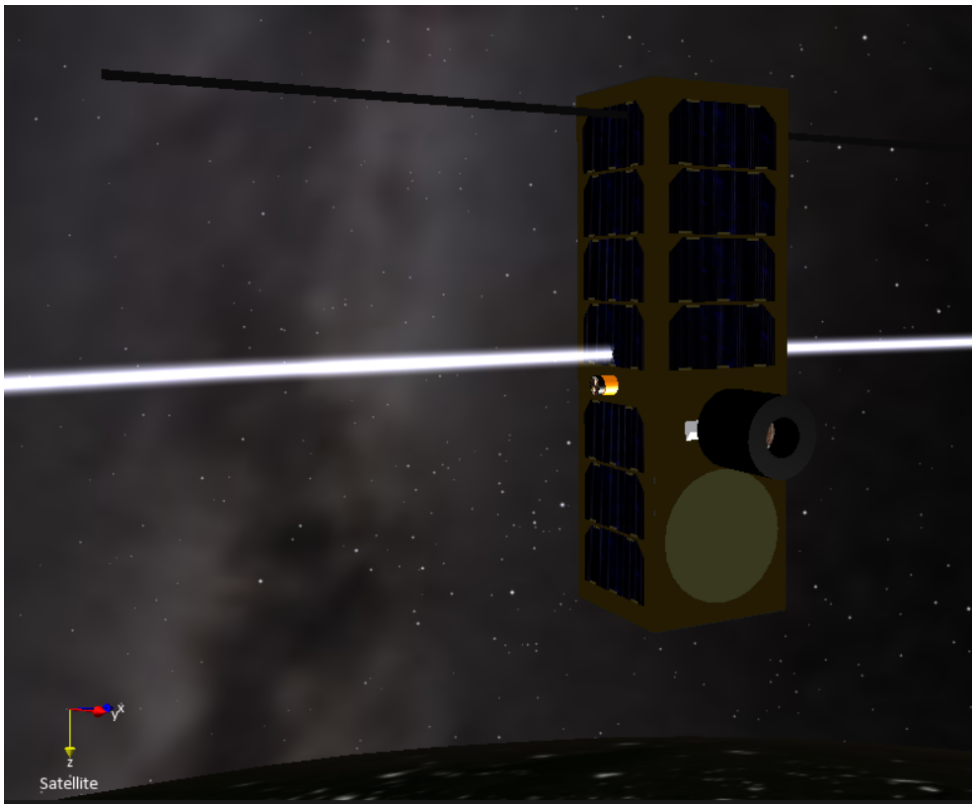

Figure 3.2: ADCS simulation environment, as presented in the D2S2 environment.

All the configurations below are valid and each one defines the SBC differently. However, we do not have the freedom to choose our SBC exactly as we want, as there is some CubeADCS functionality that is implemented only on specific SBC axes. For example there is no XThompson, and the Control Mode ConTgtTrack points the +Z SBC axis to the ground. This means that for PeakSat we can not set the long axis to be the X axis - because the long axis is the only axis a Steady Spin State (ZThompson) can be reached (the other 2 axes have similar moments of inertia, and since there is no dominancy any spin around them will be unstable), and we also have to define the +Z axis to be towards the ATLAS terminal, as there is a control mode that can do exactly this.

A very important constrain exists in the mounting configurations of RWLs, MTQs, GYROs that they are axis quantized. For all other sensors, we define 3 Euler angles, that create a 3D transformation, and so they can be placed in any configuration (physical or theoretical) with each other. However, for the 2 GYROS, the 3 Reaction Wheels and the 3 Magnetorquers, we can only choose if we want them to be in: +X, -X, +Y, -Y, +Z, -Z axes. This is not a problem mechanically, as all the above are included in the CubeADCS stack, but it might be a problem when we want to reach the optimal pointing performance, during an optical pass. Moreover, we can understand that we can choose one out of 24 (6 sides x 4 SBCs on each side) possible SBCs. Our main constrain as we said before is that we want the +Z axis to be facing downwards, at the direction of the ATLAS Terminal. This means that any of the 4 possible SBCs where the +Z axis is “downwards” is acceptable, we still do not have any evidence why to prefer one from the other, so we chose in random.

3.6 Understanding the Estimated Orientation in practice.

It is very likely, if not certain that the mounting configurations of all ADCS sensors and actuators will not have their theoretical values, but will be slightly misaligned with respect to them, mainly because of the mechanical tolerances of the materials. To understand this and its implications conceptually, we will take an example where the Star Tracker is misaligned at about 1 degree w.r.t. its mounting configuration (the Star Tracker orientation in the SBC as the ADCS Computer thinks it is like) Moreover, we will assume that that we are only performing 3Axes Wheel control, at nadir pointing and all the wheels are actually perfectly aligned with the SBC axes.

1. The ADCS will always try to point the most accurate sensor, perfectly. This means that even if this is not the case, the ADCS will try to command the orientation of the satellite, in such way so that the Star Tracker will point perfectly. This, in a way means that the most accurate working sensor will "define" the SBC.

2. In the relative pointing requirements, mounting misalignments can have a non-negligible effect, because. In the nadir pointing example, the satellite thinks it is in Nadir pointing, however, it is not, and we can imagine that its actual orientation is 1 degree shifted w.r.t. the nadir pointing. Now lets say that we command the satellite to perform a yaw rotation, around its long axis. Simplistically, the ADCS will assume that it only needs to power on the Z axis reaction wheel. One would expect that the measured Earth elevation caused by this operation will remain constant, but this will not be the case, as the satellite will rotate about its tilted axis, and not its assumed one. Of course, when an actual ADCS sees this change, it will correct it by also powering on the other Reaction wheels - to correct for the observed change, however this procedure (that will happen during all control loops) will induce some errors - and thus some relative pointing errors.

3. To understand this better, we can also think the example of perfect mounted sensors, and misaligned reaction wheels. Lets say also now we would like to perform a yaw angle rotation, and the ADCS fires only the Z reaction wheel. The sensors will observe that the change in the satellite's orientation is not the expected one, because it will be about the actual reaction wheel axis - and thus the ADCS will have to perform corrections in the control loop.

Implications on the Misalignment

In the section 3.5, we saw that we are defining the SBC with respect to the Actuators and Gyroscopes, and we can only choose one SBC axis for each of them. For the sensors, like the Star Tracker, we can be more flexible, and choose any Euler Angle triad set to define its mounting orientation. Of course, by the CAD and design of the satellite, we can easily find the Mounting Configurations for all the sensors and actuators, but what about misalignments? What if the Sensors are not exactly where we think they are? To answer this, lets first take a quick look at the Estimator Configuration, and the Noise the ADCS Computer is attributing to each sensor. This is a parameter we can change - we do not know exactly how it is used but it seems safe to assume that it is some kind of weight function that is attributed to each sensor measurement, for the final estimation. The noises are:

- Magnetometers: 0.001
- Sun Sensor and Horizon Sensor: 0.0001

- Star Tracker 10^{-8}

It is obvious that since the Star Tracker is by far the most accurate sensor, at any given time when the Star Tracker can produce valid measurements, this will define the SBC, or to call it otherwise an Estimated SBC. This has also been shown in CubeADCS simulations performed by D2S2. The above is helpful for us for 2 reasons:

- We do not really care to characterize neither the Star Tracker, neither the ATLAS terminal with respect to any ADCS component - the knowledge of the misalignment between the 2 is enough, because as we said the Star Tracker "defines" the SBC, or to put it differently the Star Tracker is the main instrument that will create accurate enough Estimations for the SBC, especially during an optical pass.
- We can tackle the misalignment between the 2 by changing the Mounting Configuration of the Star Tracker - and only the Star Tracker. Such a change will change the SBC in such way, so ATLAS pointing direction is exactly on the +Z axis. As we will see below there is a much easier way to tackle the misalignment, however it will become clear why this method will probably be necessary.

3.7 Misalignment Usage

The goal of this Section is to understand how do we plan to use the measured ATLAS terminal misalignment (measured either on Earth, either in orbit) in our ADCS configuration. Of course, this misalignment will affect only the ADCS configuration, and not the ATLAS terminal. Initially we have to remember how do we define the Satellite Body Coordinate System of the ADCS (section 3.5.2), a few details of the ADCS estimation (section 3.6), and then we will explore the possible control modes for ground tracking, and finally understand how the misalignment is going to be handled.

3.7.1 Two Ground Tracking Control Modes

TODO: is it confidential?

We will briefly talk about the 2 main control modes that will be used during our optical passes, and make a performance comparison between them. These control modes can potentially use all sensors - as we want to configure them. The Star Tracker is the only PeakSat sensor that can track the ground with enough accuracy to achieve the optical pass pointing requirements.

ConTgtTrack It is a control mode to track a specified ground target by pointing the +Z SBC axis to it, and uses only the Reaction Wheels as actuators. As we can see in fig. 3.3 that the absolute pointing error for this control mode reaches 0.04°, at a noiseless scenario.

ConGndTrack It is a more flexible control mode that can track a specified ground target by pointing a user configured SBC vector to the target. It uses both Reaction Wheels and magnetorquers as actuators. In fig. 3.4 we can see an example of ConGndTrack pointing error, where the +Z axis is pointed to the ground (should be equivalent with the example above). We can see that the absolute pointing error reaches a maximum of 0.16 degrees, in a noiseless simulation scenario.

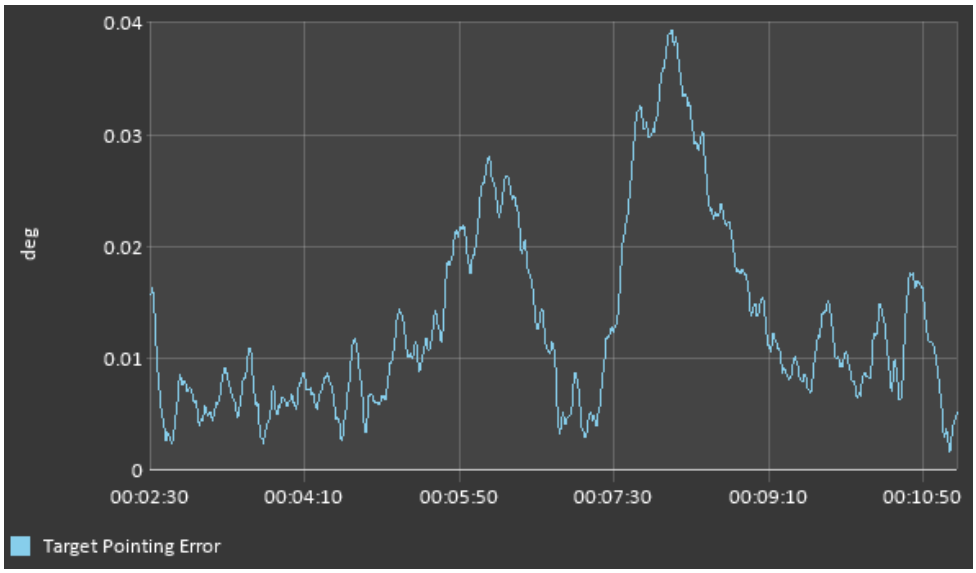

Figure 3.3: Example Target Pointing Error when the Star Tracker, and ConTgtTrack is used.

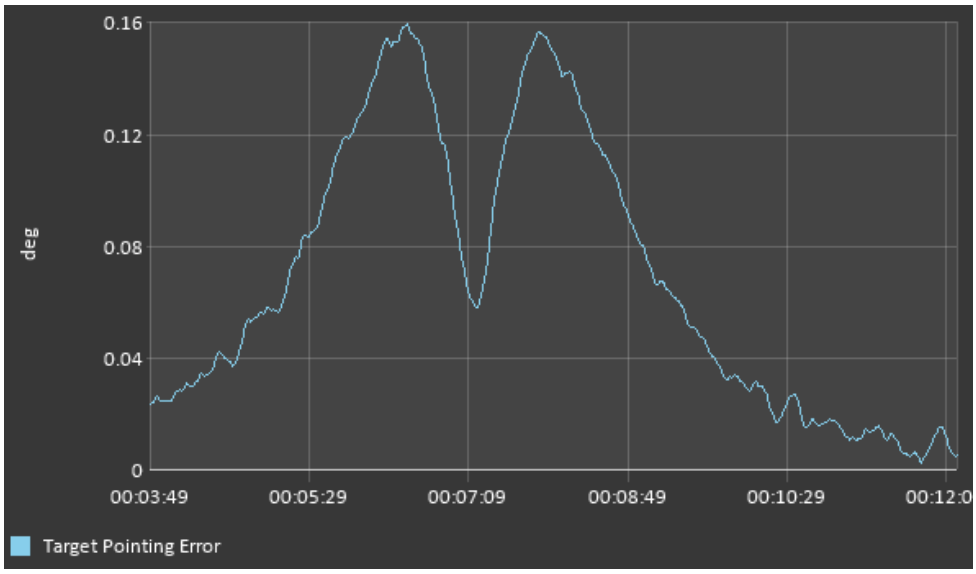

Figure 3.4: Example ADCS Target Pointing error, when ConGndTrack is used.

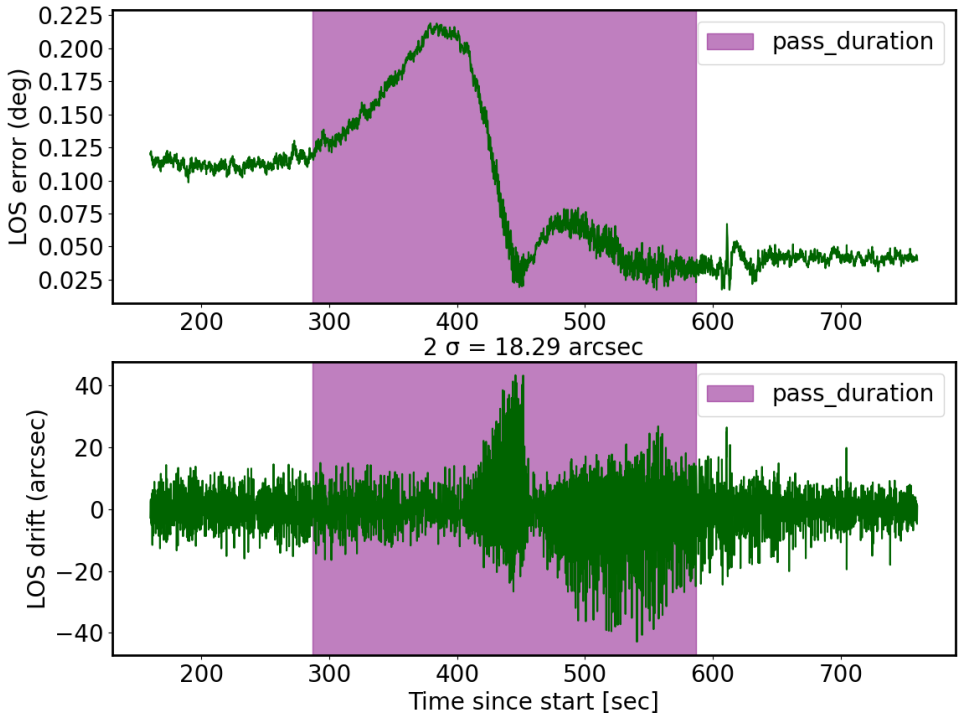

Figure 3.5: Example an ADCS LOS error, for a specific target pointing control mode.

Comparison of Control Modes We saw in figs. 3.3 and 3.4 the difference in pointing performance of 2 possible control modes. We can also see the relative pointing errors of the ADCS in fig. 3.5. In the beginning of the ADCS configuration, we were thinking that any possible measured misalignment can just be put as a SBC target tracking vector into the ADCS and use ConGndTrack. This is indeed simple, however as we can see GndTrack performs quite worse that TgtTrack. This result is amplified in scenarios where misalignments are added to the components mounting configurations. Why this is the case is not exactly clear to us as the exact algorithms each control mode uses are a black box. In any case, simulations show that, using GndTrack, even in a perfect scenario we are marginally reaching our pointing performance requirements so using ConTgtTrack seems like an only way road.

3.7.2 Can we Trust the simulations?

Considering we are about to make several decisions regarding the misalignment usage based on some simulation results, we would like to close this Section (and this Chapter in general) by generally discussing about them. We have a mission that requires very precise pointing performance, and we are using a 3rd party ADCS that runs proprietary algorithms - and a simulation software that is also using proprietary algorithms. Moreover, many of the above results and conclusions have changed in the past year, while we are learning more about how to operating the ADCS, and diving deeper into its details - so this is also expected to happen in the future. However, most of our tiny experience comes from simulations, and it is worth asking, can we trust them?

The short answer is: Maybe.

Maybe it is best to first look at our beard, and, and indeed many mistakes that have been done (and many that we probably have not found yet) are because our inexperience.

However, we can still identify some major limitations in our ADCS simulations. One, is

that they are a black box, we only know a high level of the actual algorithms and propagations that are running. Second, each simulation has to be ran manually, and there is no programmatic interface, which means that we can not perform actual statistical analysis on optical passes. Thirdly, there is not a straightforward way to introduce bias errors and uncertainties in the simulations - so apart from tampering with the mounting configurations we have not thought of other ways to introduce errors, and all this make our results much less reliable. Also the star tracker pointing and rolling error discussed in section 3.3.3 do not seem to be included in the simulations.

ADCS seems to perform well within capabilities during nominal operations (e.g. Sun Pointing, Z-Thompson etc) however PeakSat mission, carrying an optical communications Payload, has very strict pointing requirements, and is going to test the performance of the ADCS to its limits. Details will matter a lot and currently many details are left unanswered, mainly regarding extracting reliable performance statistics. At this point, even though often the results do not make complete sense, what matters is from all the trial and error we have reached a level that when the time comes, we will manage to find the optimal configuration to perform successful optical passes. *Until then, we can just hope it is possible.*

Chapter 4

Misalignment Measurement

σαβουλιάζω: κάνω κάτι κατακόρυφο,
ευθυγραμμίζω / καθετοποιώ

λεσβιακή λέξη

4.1 Introduction

At this stage we should understand the basics terminology of Satellite Optical Terminals, and ADCS systems. We have seen that they present strict pointing requirements, that can be satisfied with correct ADCS configurations. We are now going to see how can we characterize a relative orientation between the Optical Payload and the ADCS, and this will happen with the misalignment measurement. The misalignment measurement, comes to bridge the gap between the ADCS, and the Optical Payload of the satellite. It has 2 final goals:

1. Quantify how the misalignment changes before and after the vibration testing
2. To have an initial guess of the static misalignment in orbit, before the beacon detection.
3. Have a coordinate system transformation matrix, from ATLAS coordinate system orientations, to the ADCS SBC (sections 3.4 and 3.5.2)

During an Optical Pass, a procedure called Point Acquisition and Tracking is performed, which is practically following the satellite from the OGS with the beacon, detecting the beacon laser at the satellite and sending the downlink laser back to the OGS. There are several point of failures at this procedure, like OGS mispointing, satellite ADCS mispointing, thermoelastic misalignments induced to the satellite, uncertain orbital knowledge etc. Before the launch, it is wise to try to reduce as much as possible the uncertainties in cases that can be reduced, and the relative Optical Payload - ADCS misalignment is one of them. In this Section we are going to dive into the specific procedures of this measurement, and finally discuss the results implications on the mission planning.

Figure 4.1 schematically displays the relative PeakSat components that will contribute to a successful point acquisition and tracking, and the procedures that will be used to each of them in order to characterize them. Each component and measurement method will be described in a separate section.

In this section we will start by mentioning the different misalignment characterizations that are being performed by our component providers (section 4.2). After this, we will focus on the

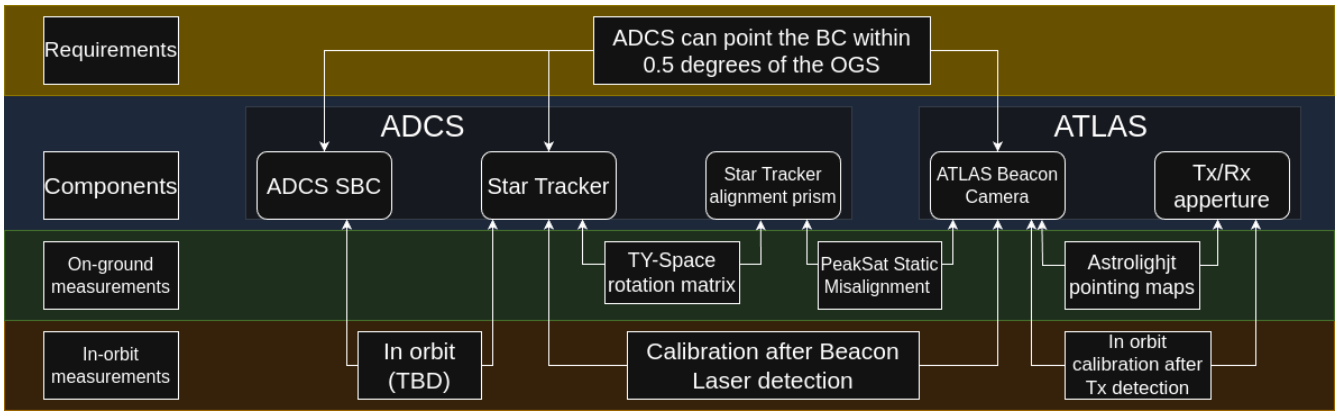

Figure 4.1: Overview of PeakSat components relative to alignment

procedure preparation of the misalignment measurement (section 4.3). One weirdness of this preparation is that we had some gaps on exactly what equipment is provided, and how it is being used. For this reason, our actual measurement did not follow the exactly procedures described in section 4.3, but has some revisions. We decide to keep our original planning here for historical reasons, but we will explain in detail the required revisions in section 4.4. Subsequently, we will explain the post processing of the data we acquired, in order to calculate the final misalignment. One of the revisions we had to do into our procedure, raised the need to perform one more complementary alignment characterization procedure in Thessaloniki, and this is explained in section 4.6. Finally, we briefly mention in section 4.7 the input of our results to the ADCS, however, for a more detailed discussion the user should check section 3.7.

It is also worth mentioning that in order to be able to perform this experiment, we have to have access to at least to sides of the Star Tracker alignment prism. For this reason, a small design change had to take place, and more details about it are in appendix B.1

4.2 On-ground measurements

4.2.1 Star Tracker with alignment prism

The characterization of the Star Tracker and the alignment prism is done once by the Star Tracker provider, TY-Space (section 3.3.3), before integration to PeakSat, and it is provided to us in the form of a rotation matrix. The characterization will take place using a Star Field simulator.

4.2.2 Astrolight pointing maps

The Astrolight pointing maps will provide a correlation between a detected position in the sensor of the Beacon Camera and the output direction of the Tx Beam - that is controlled by the FSM. More information can not be provided here, as the detailed procedures are protected by an NDA.

4.2.3 Star Tracker Prism to Beacon Camera

Accurate measurement of the relative orientation between the star tracker and the beacon camera is necessary to achieve and maintain a stable optical link (fig. 1.1). This misalignment is

called static misalignment (from now on it will be referred as SM), and mainly depends on the mounting of the different optical components on PeakSat. SM is expected to remain constant during the mission, in contrast with dynamic misalignments, like thermal ones, that will change with the passage of time, probably also in short timescales. SM is expected to change only during the Vibration Test and the Launch. This is the main purpose of this Section, and has a goal to measure **the relative misalignment between the STR alignment prism, and the ATLAS beacon camera**, as we will explain in detail in the following Sections

4.3 Preparation of Measurement Procedures

4.3.1 Static misalignment targets

The on-ground SM characterization will take place before and after the EVT, and more specifically before and after the Vibration Testing. As already mentioned, the direct outputs of the SM measurement should be:

- To have an estimation of what variation should we expect to have due to the launch vibration.
- Have a coordinate system transformation matrix, from ATLAS coordinate system orientations, to the ADCS SBC (sections 3.4 and 3.5.2)

Moreover, as said before, the SM is expected to also change during launch. This means that exact prior knowledge of the SM angle in-orbit is not possible. The static misalignment should be measured with an angle with uncertainty less that 0.1°, or that comes from the 5% of the FSM steering range.

4.3.2 Equipment

The minimum necessary equipment:

- **2 autocollimating theodolites**: At least one of them should have a **laser** (preferably at about 625 nm, but this can be discussed) aligned or characterized with the autocollimator main axis. Both theodolites will have to have the freedom of **transverse, and rotational** movement.
- **Mounting Jig, that PeakSat can be safely placed on.**
- Electrical GSE to control PeakSat (umbilical cable, control board) and operate the beacon camera.

4.3.3 Measurement Overview

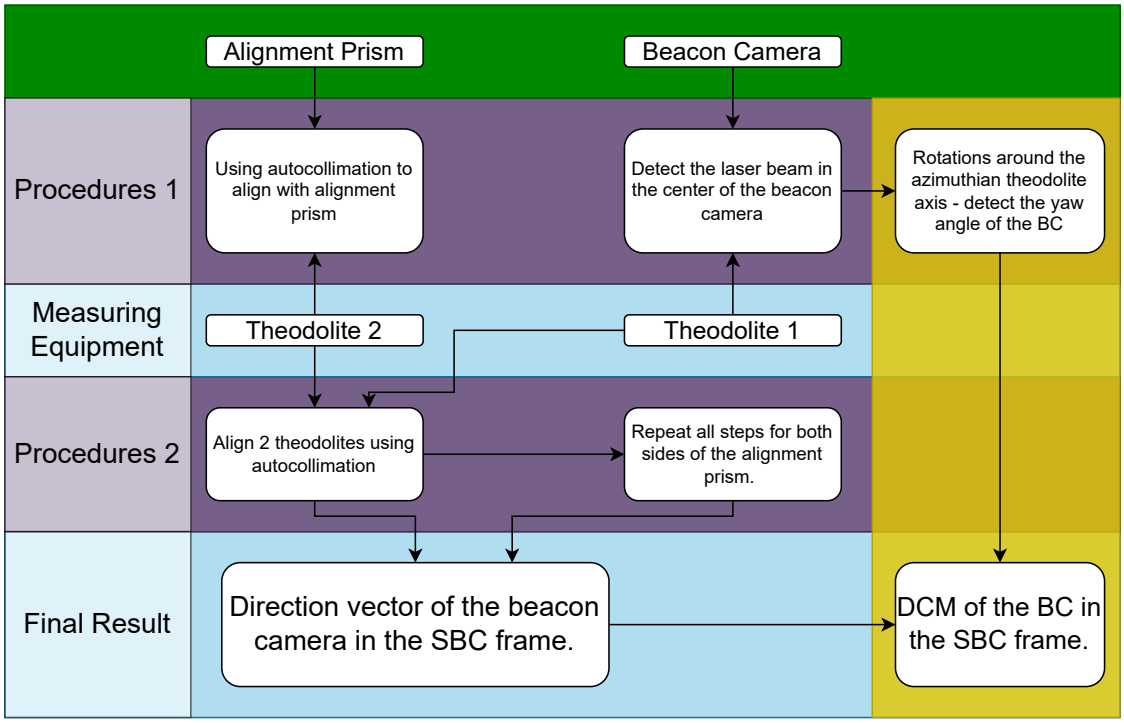

Figure 4.2: Overview of PeakSat aligment procedure

The alignment procedure will be as follows:

1. Refer one theodolite with the beacon camera axis, that is explained in detail in section 4.3.6
2. Refer the other theodolite with the 1 side of the alignment prism, as explained in section 4.3.7
3. Auto-refer the theodolites, in order to calculate the angle θ_t that is the angle between one side of the star tracker alignment prism, and the beacon camera axis. This is explained in section 4.3.8
4. Rotate the satellite, and repeat steps 1,2,3 for one more side of the STR prism. More about having line of sight access to the sides of the alignment prism can be found in appendix B.1
5. Calculate the final misalignment, as explained in section 4.5

4.3.4 Expected Results

The outcome of this experiment will be the the static misalignment of the star tracker with the beacon camera. This result can be expressed in 2 different ways - as a vector in the STR coordinate system, or as a full rotation matrix with respect to the STR coordinate system.

Vector : The first is to find the vector on the principal axis of the beacon camera, in the star tracker coordinate system. This measurement will have 2 degrees of freedom, as we measure a unit vector. The principal axis of the beacon camera, can be found by taking pictures, when a laser is viewed in the center of the sensor (pixel position). More about this characterization can be found on section 4.3.6. As we can see, by this measurement, we will not measure the rotation of the beacon camera around its principal axis. This rotation can mainly be caused by the mounting of the free space optics in the ATLAS frame, and the mounting of the ATLAS frame in the PeakSat frame. The measurement of this angle has been discussed, and it is not considered necessary but it can be useful.

3D rotation matrix : The second way is to find the full rotation transformation between the beacon camera and the star tracker. This measurement will have 3 degrees of freedom, and it will be similar to the first way, with the addition that we will rotationally move the beacon camera theodolite, along its azimuthian axis, to measure the relative rotation of the beacon camera and the theodolite. We can choose to also make this measurement, we have secured that we can reliably perform the first one, and there is time to perform it.

4.3.5 Main procedures

Theodolites are instruments that, if leveled properly, can reach a measuring angle precision of $2''$. The theodolites will be used in autocollimation mode, that would have a typical uncertainty of $5 - 10''$. The main SM measurement procedure will use 2 theodolites and will take place inside a clean-room. We can start by placing PeakSat in on an optical bench, or on a table on an assembly jig in a sturdy way. Then, the theodolite with the collimated laser laser beam (theodolite 1), we will be aligned with the beacon camera.(section 4.3.6). Then, theodolite 2 will be aligned with using autocollimation with one side of the alignment prism of the star tracker, as can be demonstrated in fig. 4.3. (section 4.3.7) After this procedure has been completed, one of the 3 axis of the star tracker alignment prism coordinate system will have been transfered to theodolite 2, and the main axis of the coordinate system of the beacon camera to theodolite one. Afterward, by using one of the methods explained in **??** the relative misalignment between theodolite 1 and theodolite 2 will be measured. This procedure will be repeated, characterize 2 sides of the star tracker alignment prism, because 2 of them are required to get the full static misalignment characterization.

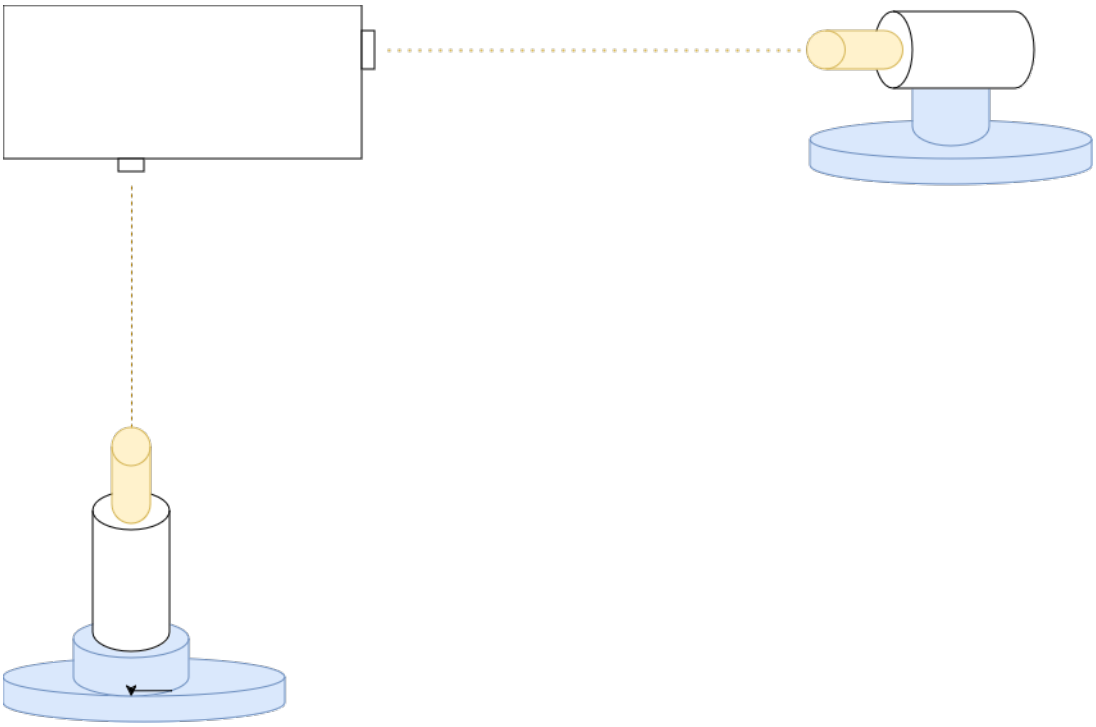

Figure 4.3: The initial position of the theodolites during alignment. Theodolite 1 on the right is pointing at the Beacon Camera, and Theodolite 2 on the lower left part of the figure is pointing at the STR alignment prism.

4.3.6 Aligning Theodolite with the Beacon Camera

Relevant information

- Pixel scale of the beacon camera is approximately 6.5″
- Theodolite angle measurement precision is about 2″

Implementation

- The theodolite laser, has to be either aligned, either characterized with the main axis of the autocollimator. If this is not the case a prori, it will be aligned by us. This alignment is assumed to have a precision of 10″
- For this step, a 625nm standard theodolite laser will be used. However, the option of having a back-up 808nm laser is also considered.
- The beacon camera has an FoV of 2 degrees, so the theodolite will have to be positioned very precisely in order for the beam to be inside the FoV of the Beacon Camera.
- Since beacon camera has a field of view of 2 degrees if the theodolite is at a distance of 2 meters from the beacon camera, the it has to be placed within 6cm to the right position in order to be detected.
- After the theodolite is positioned, it will be detected by the beacon camera, by downloading pictures from it. Determining the position of the laser in the pixels of the camera, when we have an image of the camera, is fairly straightforward:

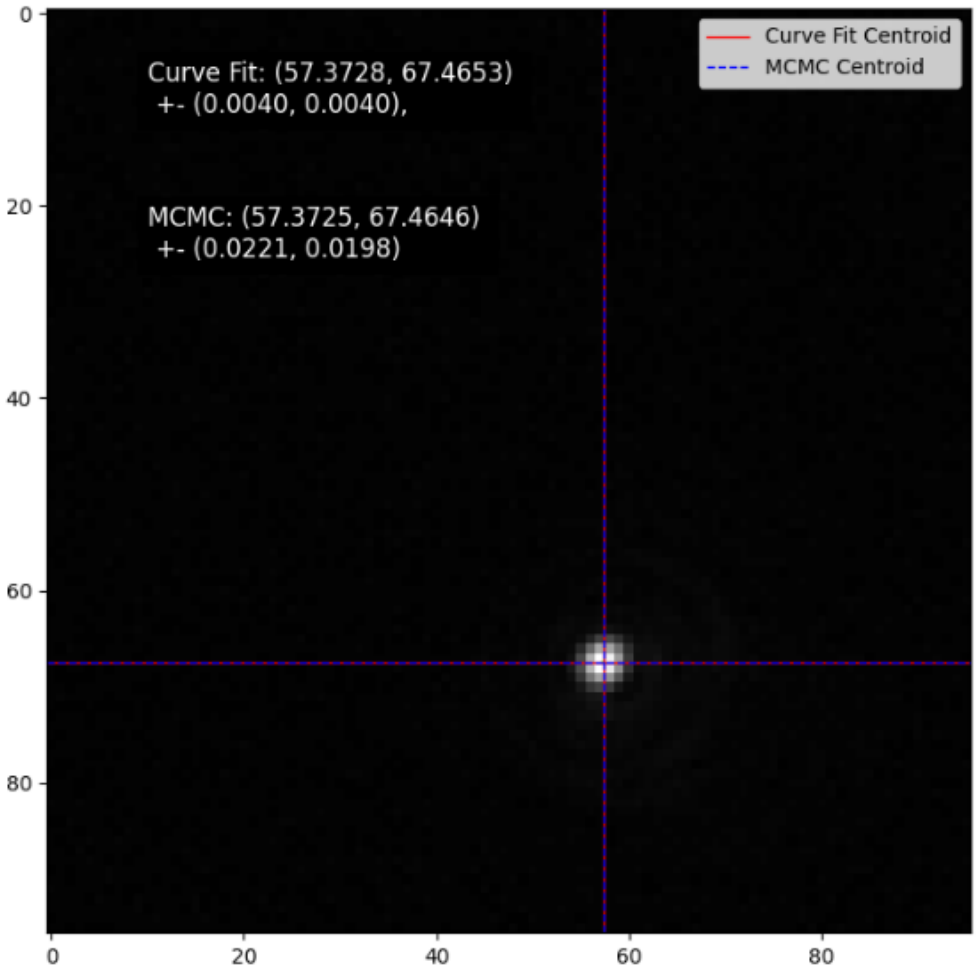

Figure 4.4: Picture from the beacon camera, using a collimated laser source at 625 nm. The entore FoV is not shown, and we can see only a Region of Interest with dimentions 128x128 pixels.

  - Either we can fit a PSF (that will just be a 2D Gaussian). Fitting a Gaussian this method can give better accuracy to the detected position of the laser, but it will not work if the peak is saturated. Example of fitting a 2D Gaussian to find the center of the Beacon Laser can be seen in fig. 4.4
  - Define the geometrical center of the circle, by seeing the most bright pixels of the image. This should work in any case, but it is not so robust and reliable.
  - Astrolight said they can provide us with the detected positions of the laser in the beacon camera using their own algorithm. The risk in this method is that the laser beam might not be collimated, and it might be in a different wavelength that the BC native wavelength, so it is not guaranteed that Astrolight's algorithm will work.

- If the laser beam is detected in the center of the beacon camera then the axis of the camera, is close to parallel with the axis of the laser.The main effect that can add uncertainties is the chromatic aberration of the convex lens. Theoretically the lens is designed to operate on 808nm, so pointing an 625nm laser to it will yield in some angle error. This error can be mitigated in the following way: By transversely moving the theodolite, with trial and error, we can try to ensure that we are pointing the laser in the geometric center of

the beacon camera. Moreover, by rotationally moving the theodolite, we can check, if symmetrical rotations of the theodolite, with respect to the center, also have symmetrical results in the traces on the beacon camera.

- Finally, this part will be done, as part of the 3D rotation matrix approach of section 4.3.4. The procedure will measure the rotation of the beacon camera with respect to its primary axis. It starts by making small rotational movements of the theodolite, to see the trace of the laser detected position changes in the beacon camera sensor. (fig. 4.5). This way the angle ϕ is calculated, that shows how much tilted will the beacon camera is with respect to the theodolite.

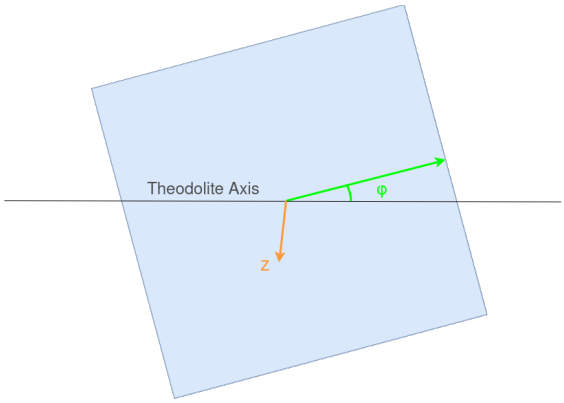

Figure 4.5: Representation of the beacon camera detector. Green line, that coincides with the x-pixel direction of the beacon camera ϕ with the theodolite tilting axis. Z axis (orange line) represents the pointing axis of the beacon camera

Sources of uncertainty

- **Lens errors, like spherical abberation** - that is why we will have to also make some transverse movements when pointing inside the beacon camera with the theodolite, so we are actually pointing close to the center of the lens. Just to comment this will not add up the uncertainty, it will add bias. We could arbitrarily increase the uncertainty to include this bias.

- **Not collimated laser beam**, with a beam divergence that can reach 0.3 degrees divergence

- **Saturating/damaging** the beacon camera - Saturation will affect how well we detect the position of the beacon in the CCD. I think we can live with that - Considering this is a very sensitive camera, that can detect a laser from LEO, firing a laser inside it from a distance of a few meters, could have unexpected results. We will have to make sure what is the maximum laser power that we can use, and if necessary, use the necessary filters. *This will most probably will not be a problem if we use a 625 nm laser, as the filters allow a very small percentage of the light rays to pass.*

- **Beacon camera pixel scale**: The beacon camera has a pixel scale of 5″. We can use a rule of thumb, and say that we can detect the center of the centroid with an uncertainty of half the pixel scale, so 2.5″.

Final estimated uncertainty: Since quantification of most of the uncertainty sources is not straightforward, we can assume that the total uncertainty will come from the detection of the centroid in the beacon camera, spurious errors like spherical aberration, with $\sigma_{cenroid} = 10''$ (double the pixel scale) and $\sigma_{abberation} = 20''$ respectively. This means that $\sigma_{bc} = 30''$ and it will denote the verticalization uncertainty. The uncertainty in measuring the angle ϕ of the beacon camera, can also be considered to be governed by the pixel scale, and the trail the beacon laser leaves in the beacon camera, and we assume it will have an uncertainty of $\sigma_{\phi} = 10''$.

4.3.7 Aligning theodolite with star tracker prism

Implementation

- The main complication in this setup, is that out FOV to the alignment prism is relatively small. So, the theodolite has to be treated with extra delicacy, if we want to have a reliable autocollimation. It seems that adequate access to both sides of the alignment prism is guaranteed. For CAD images and more detailed explanation check appendix B.1

- The exact position for the theodolite also depends on the aperture of the autocollimator. Typical autocollimators, have an aperture of about 4cm. So the theodolite has to be positioned within these 4cm of the beacon camera. We use the autocollimation mode of the theodolite, to align it with the STR prism with the mirror. The exact autocollimation accuracy will of course depend on autocollimator specifications. For now we will assume $5 - 10''$, as a typical value.

Sources of uncertainty

- **Autocollimation uncertainty**
- **Flatness of alignment prism**
- **Characterization of the alignment prism with the star tracker**: This is completely out of our control, but we have to consider that the relative angles between the alignment prism and the star tracker will not be perfectly characterized, and they also may change before and after testing. We do not seem to have a way of checking this for now. (The discussion of simulating a stellar field has been raised, but it finished very quickly.)
- **Peculiar reflections / diffraction we might have because of the small visible mirror surface, and other parts of the satellite**

Final estimated uncertainty: Assuming that a typical autocollimation measurement uncertainty is about $10''$, and the fact that other sources of uncertainty can not be quantified, then the final uncertainty will be considered to be $\sigma_{autocol} = 10''$.

4.3.8 Auto-referring theodolites

Implementation

- By rotating the theodolites so they will autocollimate with each other's light (fig. 4.6)

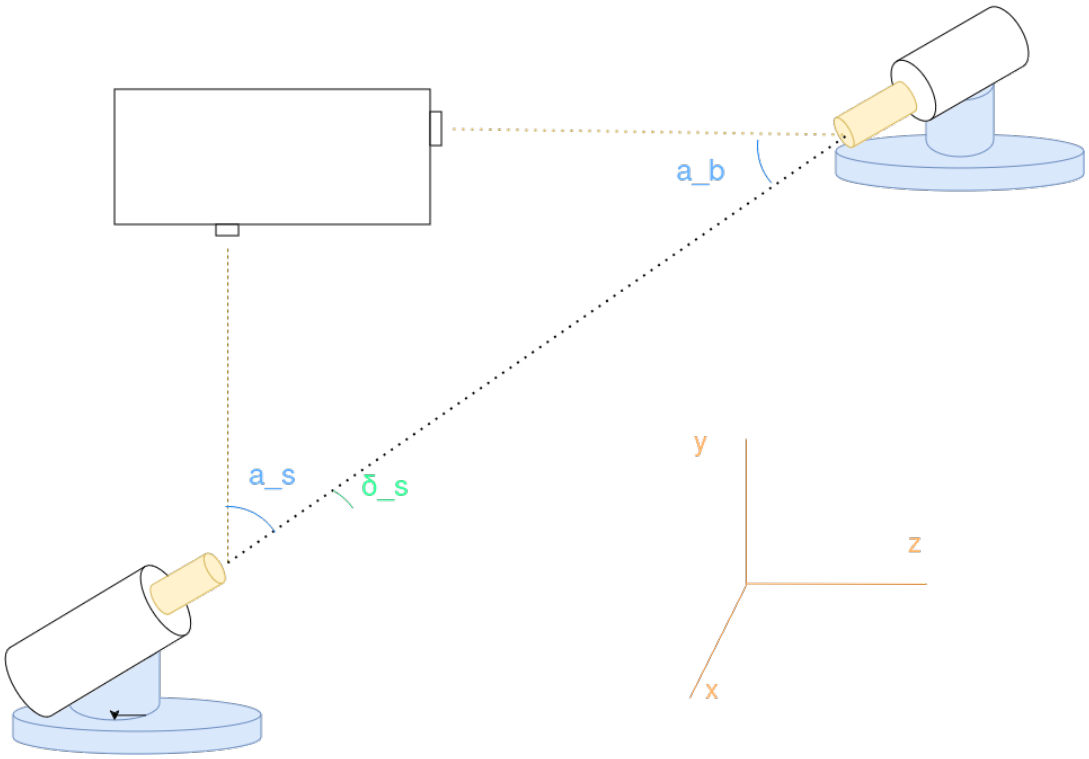

Figure 4.6: Theodolites after they have rotated, and now they are in line of sight with each other

Mathematical angles representation As an be seen in fig. 4.6 the theodolite azimuthian angle will be given by the relation

$$a = 180 - |a_s| - |a_b|$$

where $a_s = a_{sf} - a_{si}$ is the change in angle from the initial position (fig. 4.3) where the theodolite is aligned with the star tracker, to the final one where the 2 theodolites have the same line of sight, and similarly $a_b = a_{bf} - a_{bi}$ for the beacon camera. The angle δ would be $\delta = \delta_{si} - \delta_{bi}$. Of course since the vertical scale of the theodolite is calibrated using gravity, if the theodolites are set up correctly, there is no need for auto-referencing on these axis. Auto-referencing also on the declination axis can potentially be performed, deoending on the geometry of the problem, for completeness and confirmation of the results. If we can perform autocollimation by moving the declination axis of only 1 theodolite, then:

$$\delta = \delta_{si} - \delta_{bi} = \delta_{si} - \delta_{sf} = \delta bf - \delta bi$$

where of course the subscript b, means the theodolite that is characterized with the beacon camera, and s, the theodolite that is characterized with the star tracker alignment prism. The subscripts i and f mean initial and final.

Then the angle θ_t (t stands for theodolites) that can be seen in fig. 4.7 are easily found by the cosine law:

$$\cos\theta_t = \cos a \cdot \cos\delta$$

. This angle represents a 3D angle of one axis of the alignment prism, to the main axis of the beacon camera. This angle can give the direction cosine of the beacon camera vector, and will be measured 2 times per experiment. More about how will we handle this measurement can be found on **??**

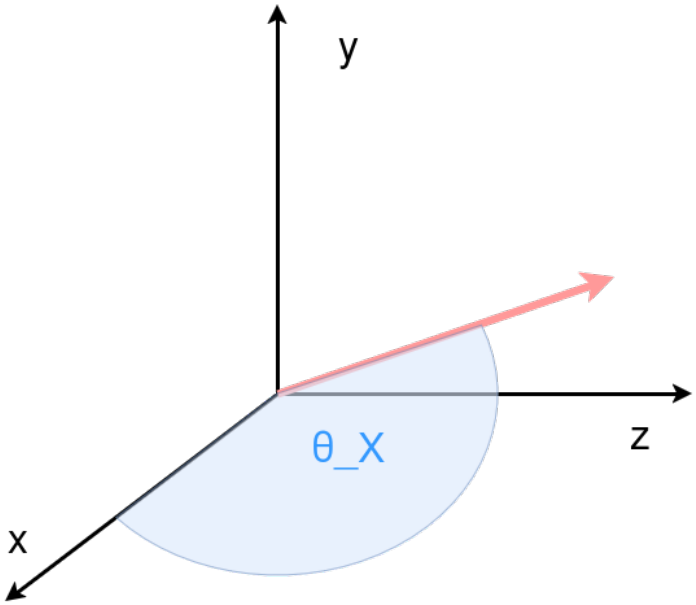

Figure 4.7: XYZ shows the star tracker alignment cubes reference frame, the red vector shows the principal of the camera and angle θ_X (blue color) shows the 3d angle of the beacon camera vector with respect to the x-axis of the star tracker reference frame. θ_X is directly measured. In order to fully characterise this vector, also the θ_y angle is necessary, that is measured in a similar way.

If we decide to make the 3D rotation measurement, we can also find the angle ϕ that denotes the rotation of the rotation of the beacon camera (fig. 4.5) with respect to its primary direction vector. Assuming small misalignment approximation, from the geometry of the problem, it can be shown that this rotation angle ϕ will be

$$\phi = \phi_\alpha - \delta_{\alpha i} = \phi_\beta - 90 - \delta_{\beta i}$$

Where α, β denote the measurements for the 2 sides of the star tracker. The fact that we can measure the same angle twice, can work as a validation of our result.

Sources of uncertainty

- **Theodolites Angular measurement precision** $2''$ each
- **Theodolite autocollimation uncertainty**, we do not know yet, but we will assume an angle of $10''$
- **Non-perfectly flat theodolites with respect to gravity**, that we assume it adds an uncertaitny of $10''$

From the initial position of the theodolites, to the final one, the uncertainty in the angle measurement will contain all the above uncertainties. For now we do not know how to validate the flatness of the theodolites, and what the uncertainty would be, so we can arbitrarily assume the final uncertainty will be double the sum of the 2 first uncertainties, meaning:

$$\sigma_{\theta_t} \approx 30'' = 0.5'$$

4.3.9 Propagation of measurement uncertainties

All the error propagation is done usng the RMS approach as explained in appendix C.1. As discussed before, the final result should be a rotation matrix from the coordinate system of the star tracker, to the coordinate system of the beacon camera, or a vector of the beacon camera vector in the coordinate system of the star tracker. For the next session a simpler approach will be followed. First the beacon direction of the Beacon Camera vector will be calculated, and then the rotation of the beacon camera with this vector (considering that we can measure this rotation). This approach is simpler to understand, helps in calculating a relative an absolute misalignment measurement uncertainty. Also it can help in estimating any asymmetrical errors (that might come from for example not having adequate line of sight to one side of the star tracker). Finally this vector will be given as a pointing offset to the ADCS. Using the previous measurement methods, the measured results will be:

- θ_X, θ_Y that refer to the angles the beacon camera vector makes with respect to the x, y axes of the star tracker alignment prism.(fig. 4.7) These angles can be measured using section 4.3.8,
- The rotation angle ϕ of the beacon camera with respect to a vector vertical to it.(fig. 4.5) This rotation angle can be obtained only through section 4.3.6.

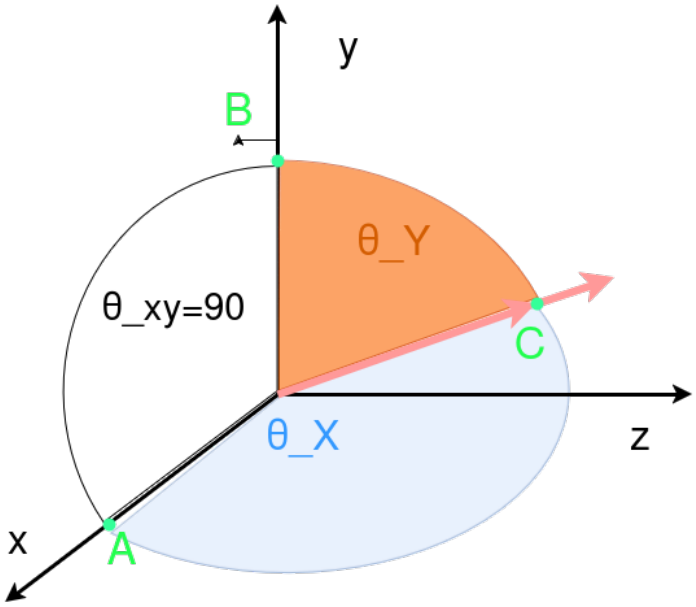

Figure 4.8: xyz is the Cartesian coordinate system of the star tracker alignment prism. ABC triangle is the spherical triangle that will be used in order to transform from the measured angles, to the coordinates to the star tracker coordinate system.

The, the beacon camera vector in the star tracker frame of reference will be:

$$\vec{R_b} = \cos\theta_X\hat{x} + \cos\theta_Y\hat{y} + \sqrt{1 - \cos^2\theta_Y - cos^2\theta_Y}\hat{z}$$

with:

$$\sigma_x = sin\theta_X\sigma_{\theta_X}, \sigma_y = sin\theta_Y\sigma_{\theta_Y}$$

$$\sigma_z = \sqrt{\left(\frac{\sin(2\theta_X)}{2z}\right)^2 \sigma_{\theta_X}^2 + \left(\frac{2\sin(2\theta_Y)}{2z}\right)^2 \sigma_{\theta_Y}^2}$$

$\vec{R}_b$ is also the vector that will be given to the ADCS as an offset. Using this vector, and calculating the dot product with the z axis, we find that the absolute misalignment angle will be

$$\omega = \arccos(\sqrt{1 - \cos^2\theta_X - cos^2\theta_Y})$$

and the uncertainty in this measurement will be

$$\sigma_\omega = \frac{1}{z\sqrt{2}\sqrt{1-z^2}} \sqrt{\sin^2(2\theta_Y)\sigma_{\theta_X}^2 + sin^2(2\theta_Y)\sigma_{\theta_Y}^2} \tag{4.1}$$

(the above equations are prone to have numerical errors, and will be rechecked several times before their application) Of course, due to the discontinuity of the arccos function for $z = 1$, the above uncertainty equation does not hold, if $\theta_x = 0$ or $\theta_y = 0$. The above relationship is not very intuitive, mainly because as θ_x, θ_Y approach to 90°, then the denominator approaches to zero.

For that reason, when trying to model and estimate the uncertainty of our measurement, we will use the much simpler RMS approach:

$$\sigma_\omega \approx \sqrt{\sigma_{\theta_X}^2 + \sigma_{\theta_Y}^2} \tag{4.2}$$

4.3.10 Expected Measurement Uncertainty

eq. (4.2) gives an estimation for the uncertainty of the SM between the Star tracker BC alignment prism, and the beacon camera. This equation is much simpler to use for making assumptions. Using eq. (4.2), we see it only depends on the $\sigma_{\theta_X}, \sigma_{\theta_Y}$ angles and not on ϕ as expected. Considering that the same method is going to be used in order to measure θ_X, θ_Y it is safe to assume that they will present the same uncertainties.

Also, the total uncertainty of θ_X, θ_Y of the proposed experimental setup will consist of:

- Autocollimation uncertainty of Theodolite 2 with star tracker alignment prism, that has a value of 10″ (section 4.3.7)
- Uncertainty in aligning 625nm laser with beacon camera that has a value of 10″. (section 4.3.6)
- The uncertainty of parallerization of the 625 nm laser and the beacon camera.
- The uncertainty in autocollimating the 2 theodolites, that is assumed to have a value of 20″
- The Uncertainty in the flatness of the 2 theodolites, that has a value of 10″

The above uncertainties are summarized in the following table 4.1

Procedure	Unc. Source	$''$, 1σ	$''$, 2σ
Theodolite 2 - Aligmnet Prism	Autocollimation unc.	10	20
Theodolite 1 - Beacon Camera	Detection of Laser Centroid	30	60
Autocollimator and Laser	Mechanical	10	20
Autocollimating Theodolites	Autocollimation unc.	20	40
Flatness of Theodolites	Calibration	10	20
1 angle RMS	-	29	58
Total (RMS)	-	41	82
Yaw angle of the BC	Detection in the Laser Centroid	10	20

Table 4.1: Misalignment error propagation. Note that the yaw angle of the Beacon Camera is not taken into account in the final uncertainty, as it is independent of the measurement.

Now, if 50% measurement margin is included, and 50% total margin is added, then the total uncertainty will be

$$\sigma_{predicted} \approx 4'(2\sigma) \tag{4.3}$$

Comparing the required and the final measured misalignment, we can deduce that theoretically, the current measurement setup if capable of measuring the relative misalignment with the necessary accuracy.

4.4 Measurement at ESTEC

The Misalignment characterization measurement was performed before the vibration testing, and after the vibration testing, in the Metrology Lab at ESTEC, in the Netherlands. We will explain both measurements separately.

Before this, we will mention one big change that we did not include in our main procedures. **A theodolite autocollimation takes place in Phase 1, and Phase 2**. In phase one, the theodolite is autocollimated with the reflective surface as explained in section 4.3.7. The results are saved at this point, and then, a button is pressed at the theodolite, that rotates both axes by 180°. This points the theodolite approximately at the same orientation as before, with a small deviation from the initial autocollimation. If this deviation is fixed, then the results are taken again. The difference between the 2 calculated vectors, is the measurement uncertainty.

4.4.1 Pre-Vibe Measurement

Into the first alignment measurement, that was also the first time we were using the necessary equipment, came will problems, but was after all partially successful Initially, in our procedures preparations, it was assumed that both theodolites will be movable, so we can autocollimate with the Star Tracker (STR) prism, and the BC. However, when we arrived at the facility, it was clear that one theodolite was fixed at a point, and we had to move/rotate the satellite.[1] Moreover,

[1]It is worth mentioning that this was not actually True, and the Theodolite that was considered fixed, could actually move transversely, however there was not trained stuff in the laboratory during out attempt to let us know about it.

after some hard trial and error, we found out that the theodolite laser which we were expecting to detect with the BC, was not detectable (Because, as we mentioned the BC has a 808 ± 5 bandpass, and the laser was at $670\,nm$). So the procedure was modified as follows:

1. Collimate with one side of STR MC with the "Fixed theodolite", as explained in section 4.3.7, while moving the satellite, and keeping the theodolite fixed.
2. Collimate with the reflective surface of the BC, as explained again in section 4.3.7, keeping the satellite still, and moving the "Movable theodolite".
3. Auto-refer the theodolites, as explained in section 4.3.8.
4. Change the Satellite Orientation, so the other side of the MC is visible through the "Fixed theodolite", and repeat the above steps.

In the addition to the draw-backs mentioned above, a height adjustment jig that was supposed to lift the satellite failed, and while attempting to perform step 1, we moved the satellite so high, so the movable theodolite could not go so high. The actual measurement, took place in our last 3 hours at the facility, so we only measured the Beacon Camera Reflective Surface (BCRS) in the SBC coordinate system.

4.4.2 Post-Vibe Measurements

During our second visit, things went much more smoothly regarding the measurement. The first thing we noticed, was that the previously "Fixed" theodolite, was now movable [2]! Moreover, the mounting jig of the satellite was significantly more stable, and when we also encountered the problem that the movable theodolite could not go as high as necessary, it was lifted up by the laboratory stuff. We performed exactly the same steps explained in the previous Section. However, this time we also measured the reflective surface of the transmit laser (TxRS) in the SBC coordinate system, and repeated the measurement once. In a course of $1.5\,d$ we measured the BCRS vector in the SBC twice, and the TxRS vector also twice.

4.5 ESTEC results

In section 4.3, we presented our preparation for performing the misalignment measurement, and in section 4.4 we outlinled the changes in the procedures. It should be obvious by now that the ATLAS - ADCS rotation matrix can not be measured, but only the BC vector in the SBC frame. In this Section we will show the analysis we did on the theodolite measurements. This calculation is performed in calc_misalignment.ipynb.

Definitions

- The 2 theodolites will be reffered as fixed, and movable. The fixed theodolite is always auto collimating with the Mirror Cube of the Star Tracker, and the movable one with the reflective surface of the Beacon Camera.
- The superscript will denote with respect to which coordinate system this vector is being calculated to. For example R^m is a vector measured in the moving theodolites coordinate system.

[2] Even though in our notes we kept referring to it as fixed.

- CS -> Coordinate System
- MC -> Mirror Cube

Coordinate Systems The transformation from unitary spherical coordinate system, to a Cartesian is necessary, in order to transform the theodolite measurements to vectors, and auto-refer their coordinate systems. The theodolite is measuring the Horizontal (θ) angle counter clockwise, so to transform to a right hand Cartesian system we will use the usual spherical to orthogonal equations:

- $x = \sin\phi\cos\theta$
- $y = -\sin\phi\sin\theta$
- $z = \cos\phi$

Calculating Vectors

- A vector will be calculated by using a combination of Phase 1 and phase 2 measurements. Phase 1 measurements explain the relative angle of the Beacon Camera, with respect to the Side of the Star Tracker prism that is oriented towards the star tracker, and Phase 2 refers to measurements taken with respect to the sun sensor side of the Star Tracker prism.
- Initially, 2 vectors will be calculated using the above equations, one for phase 1, and one for phase 2. Then, the average vector will be the measured vector.
- The uncertainty in each XYZ coordinate will be the half-difference in each coordinate.
- Even though the uncertainty can more easily be calculated in angles (as this is how the theodolite measures), we prefer to denote the uncertainty in vector components, because denoting angular uncertainty will require to perform more complicated equations that include trigonometric, inverse trigonometric functions and their derivatives in uncertainty propagation.
- Uncertainty propagation in vectors is fairly simple and less prone to errors. We will do it by the root of the square sum.

Transforming Vectors

- We will have to transform vectors from one theodolite coordinate system, to the other, and ore specifically from the moving to the fixed one. This means we will have the Camera Vector in the Star Tracker Mirror Cube Vector.
- When transforming, we have to keep in mind, that if $\hat{z}$ is the vertical unit vector of each theodolites coordinate system, then $\hat{z}^m = \hat{z}^f$, because the theodolites have been leveled with gravity.
- Any vector at the moving theodolite CS, can be transformed to the fixed theodolite CS by the equation $\vec{R}^f = R_z(\theta)\vec{R}^m$ where A is a 3x3 rotation matrix, that will have the form:

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- When the theodolites are referring with each other, we will measure $\vec{v}^m_m$ that is the moving theodolite auto-referring vector in the moving theodolite CS, and $\vec{v}^f_f$ that is the fixed theodolite auto-referring vector in the fixed theodolite CS.
- From the geometry of the problem, we can easily infer that when the theodolites are looking at each other, the moving theodolite auto-referring vector in the fixed theodolite CS, will be the opposite of the fixed theodolite auto-referring vector in the fixed theodolite CS, or
$$\vec{v}^f_m = -\vec{v}^f_f(1)$$
- $\vec{v}^m_m$ will be transformed to the fixed theodolite CS with the equation:
$$\vec{v}^f_m = R_z(\theta)\vec{v}^m_m(2)$$
- By using equations (1), (2) we get:
$$-\vec{v}^f_f = R_z(\theta)\vec{v}^m_m$$
- If we denote with $c \equiv \cos\theta$ and $s \equiv \sin\theta$, we get the 3 following equations:
  - $-v^f_{fx} = c \cdot v^m_{mx} - s \cdot v^m_{my}$
  - $-v^f_{fy} = s \cdot v^m_{mx} + c \cdot v^m_{my}$
  - $-v^f_{fz} = v^m_{mz}$
- The first 2 equations make a linear system with 2 unknowns, that can be solved. By solving it we can calculate the angle θ and the final rotation matrix to transform between one theodolite CS and the other
- The 3rd equation, is a measure to how accurate our leveling was. The difference in the 2 vectors $(v^f_{fz} + v^m_{mz})$ will be added as an extra uncertainty to our calculations.

Calculating BC vector and misalignment Now that we have the rotation matrix to transform from one theodolite CS to the other, we can easily Transform the BC vector from the movable theodolite CS, to the fixed one. At this point we have the MC vector and BC vector in the same CS. Using a simle dot product we can find the relative angle between them. This practically leaves us with 2 cones (see fig. 4.9), and we can find their intersection with positive Z, to find the BCRS (or TxRS) vector in the MC coordinate system (simple linear system with 2 independent equations and 2 unknowns). This can be transformed to the SBC by applying the MC to STR transformation, and then STR to SBC.

As an example, previous to Vibration, only the BCRS vector with respect to the SBC was measured, and can be seen in fig. 4.9. The cones are:

- Cone1 (STR Side) = 90.1324 ± 0.007°
- Cone2 (SS Side) = 87.7922 ± 0.008°

And the final vector in the SBC can be plotted in fig. 4.10.

Before and After Vibration, we calculated the vectors using exactly the same procedure. The full results of the final measured vectors of both the BCRS, and the TxRS can be found in table 4.2, and the projection of their X, Y coordinates in fig. 4.11. As a conclusion, we see that the change between the BCRS vectors before and after vibe is less than 0.011 degrees - which is much less than the 2 degrees field of View of the Beacon Camera.

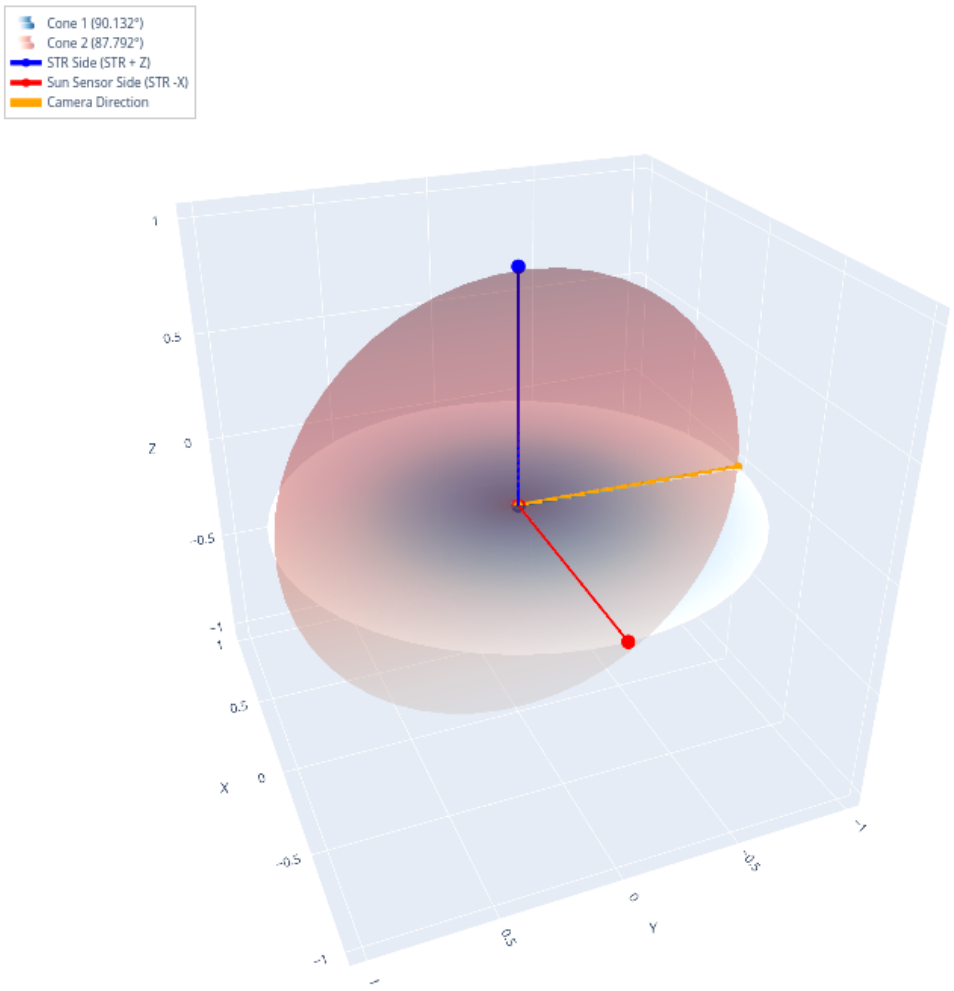

Figure 4.9: The intersection of the cones is the final vector we are looking for.

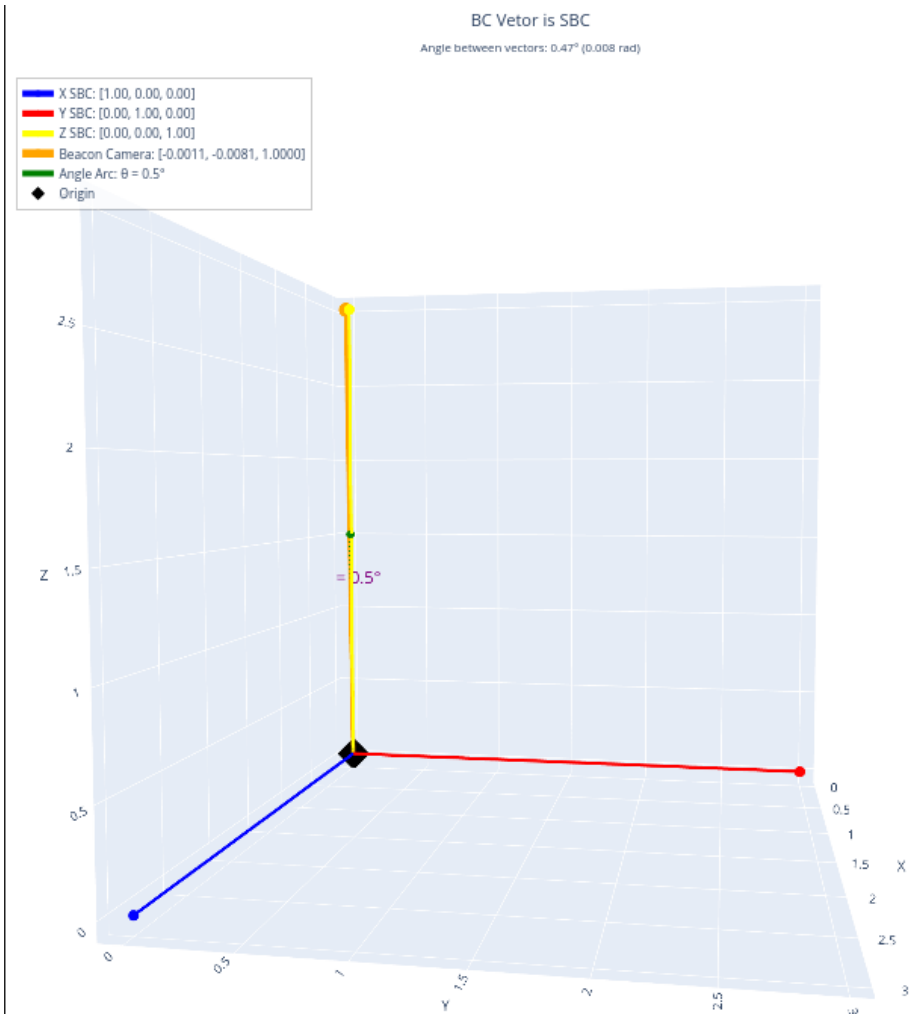

Figure 4.10: Example of Calculated Vector from the alignment measurement.

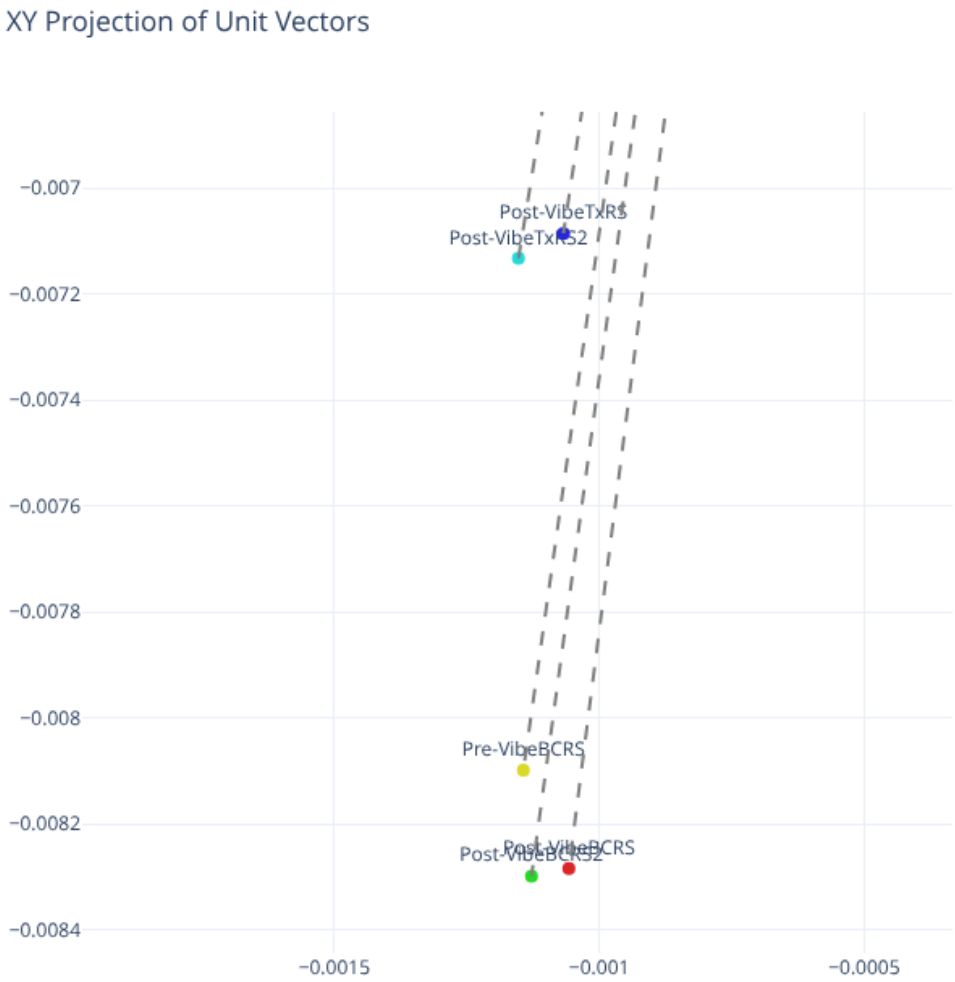

Figure 4.11: All measured vectors in the ADCS coordinate system, with only their XY coordinates plotted.

Configuration	x	y	z
Pre-VibeBCRS	-0.00114	-0.00810	0.99997
Post-VibeBCRS	-0.00106	-0.00828	0.99997
Post-VibeBCRS2	-0.00113	-0.00830	0.99997
Post-VibeTxRS	-0.00107	-0.00709	0.99998
Post-VibeTxRS2	-0.00115	-0.00713	0.99998

Table 4.2: Unit Vectors of BCRS, and TxRS.

4.6 BC-BCRS Characterization

After measuring only the vector of the Beacon Camera Reflective Surface, and not the sensor, it became obvious that we should have a way to characterize both to some extent. We are going to explain the procedure to do this very briefly, as it had relatively high uncertainty. For this we used a laser collimator, primarily used to align telescopes. The goal is to make the collimator vertical with the BCRS, and then capture images, detect the laser at a X, Y position in the BC, and estimate the relative misalignment between the 2. It is finally worth noting that this was done before, and after vibration also.

The collimator used can be placed vertically with a reflective surface, by seeing the reflected laser go through the transmission hole. The uncertainty in this verticalization (collimation) depends on the distance of the collimtator to the reflective surface, and the whole diameter. In our case:

- Distance: 1.4m
- Whole diameter: 3mm
- Thus, the collimation uncertainty is 3.5′, which is of course much higher than the measurement we performed at ESTEC.

The setup can be seen in the figs. 4.12 and 4.13. In the right we can see the collimator mounted on the theodolite. The theodolite is used in order to make small angular measurements of the collimator. We have opened the ATLAS BC cover, in order to capture images. The images are downloaded and analyzed in the laptop via the SPI connector.

Beacon Camera

- The total Resolution is (1464, 1936) px.
- The sensor middle is at: (732, 968) px.
- The camera pixel scale is 4.5″
- The FSM zero is at: (807, 916) px.

In total we captured 12 images, during the pre-vibe measurement, and 22 in the post vibe measurement. For all of them, we changed the lateral position of the reflected ray. We do not have any way to measure where exactly on the BCRS the ray was incident to, but by eye we can

Figure 4.12: Image acquisition setup.

make an estimation on where we see it. This is very important, as it will answer the question, weather the detected position depends on the incident ray beam position.

To detect the centroid of each laser blob, we individually fitted each image with a Gaussian.In the **Pre-Vibe** measurement, the maximum absolute distance of the furthest centroids is 33px, which translates into 2.5′, and the centroid 1σ is approx 15px, or 1′. In the **Post-Vibe** the maximum absolute distance of the furthest centroids is 35px, which translates into 2.5'. Results of the fitted optical centers, can be viewed in fig. 4.14.

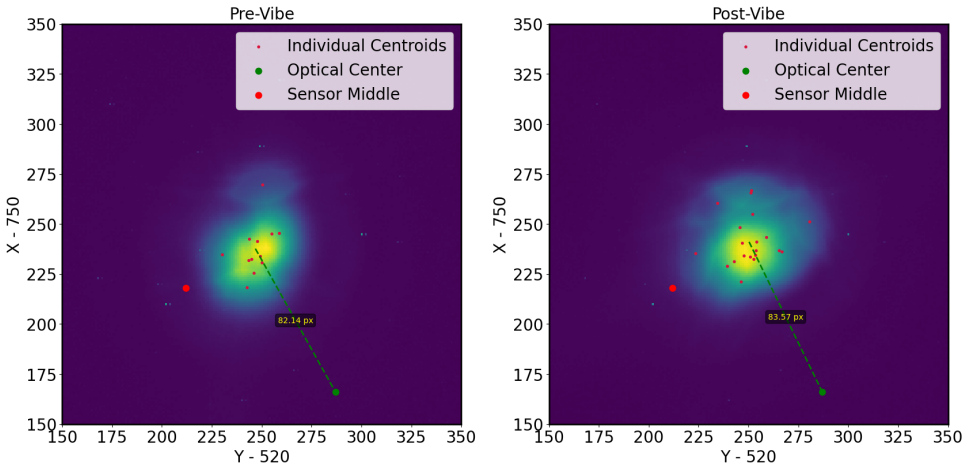

Figure 4.14: Comparing the measurement results before and after the vibration testing.

4.6.1 Conclusions

We find that the **pre-vibe** center of the position (767, 985) ± (14, 17). The **post-vibe** detected center of the position is (771,989) ± (13, 15) and they are in very well agreement to each other.

We were able to characterize the misalignment between the BC, and BCRS, with an uncertainty of 1′ , and we found a misalignment of 6′, which is within the expected misalignment tolerance.

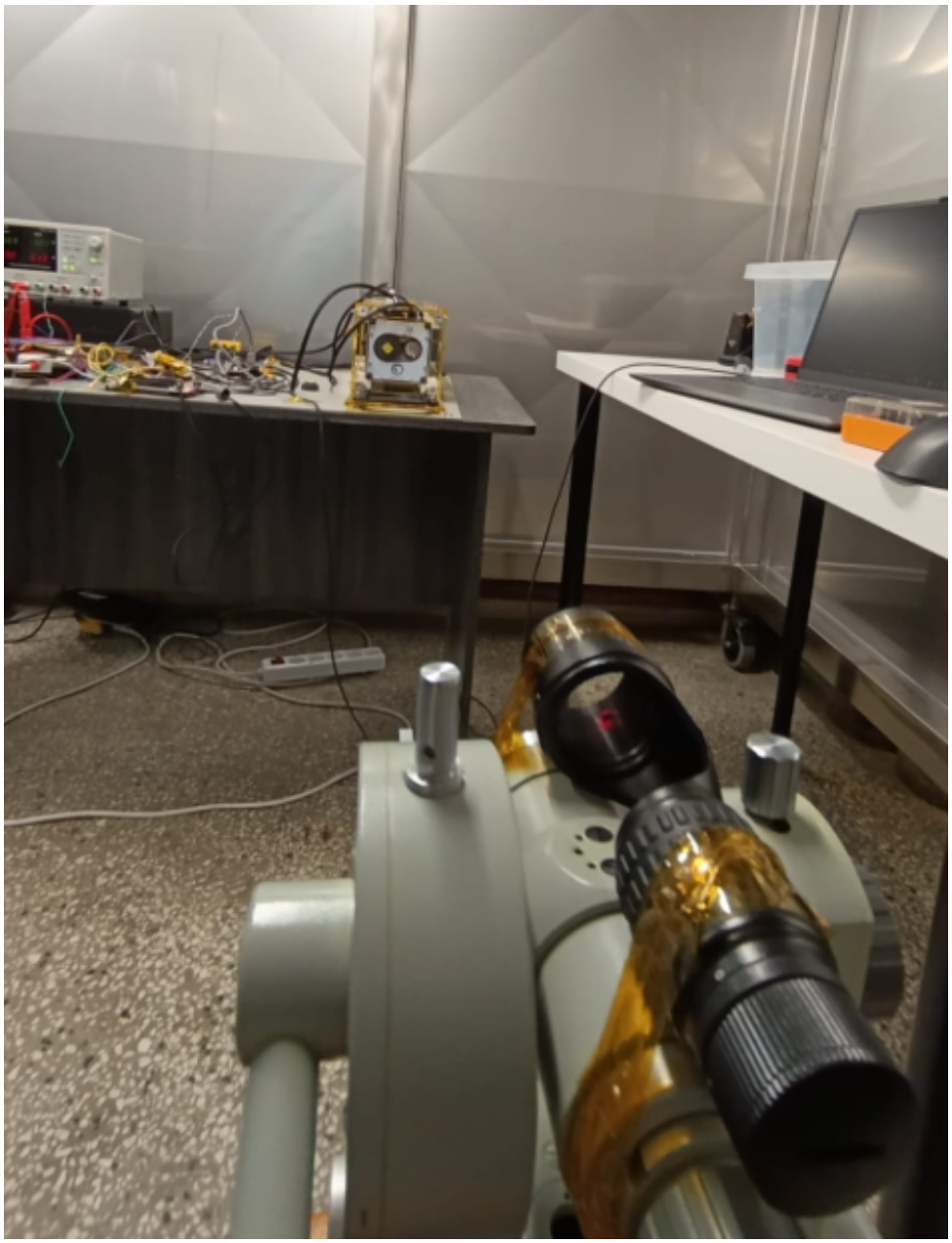

Figure 4.13: Image acquisition setup.

4.7 Input to the ADCS

The information, we have gathered so far, are going to be used for finding the following 2 quantities:

1. The Beacon Camera Center (FSM zero position) in the SBC frame, and this will be the final vector we have to point to the ground
2. An initial guess of the full transformation from SBC to BCC.

We will also use 2 pieces of information, that are provided by the Optical Terminal Provider:

- The FSM zero position in the BCC is: x=918.1 pix / y=804.1 px
- The TxRS vector in the BCC: x=971.1 pix / y=694.7 px
- Beacon Camera Orientation is 11.5 degrees with respect to the ATLAS CAD (approximate, not measured). [3]

Having the above in mind, by definition a vector in the BCC is calculated by :

$$v_{BCC} = \begin{bmatrix} -(p_x - p_{cx}) \cdot p_s \\ -(p_y - p_{cy}) \cdot p_s \\ f \end{bmatrix}$$

where: p_x, p_y are the pixel detected positions, and p_{cx}, p_{cy} are the reference center pixels. Then, the Z axis transformation (appendix A.1) from a BCC vector to the newly transformed BCC1, that has matching X, Y axes as the SBC, is

$$R_{BCC \to BCC1} = R_z(11.5^\circ) \begin{bmatrix} 0.9799247 & 0.19936793 & 0 \\ -0.19936793 & 0.9799247 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The above coordinate system would be exactly the SBC, if we ignored all the misalignment. We will make the approximation, that the 11.5° is happening around the FSM zero position. That is why we can also ignore the misalignment in this calculation, because the larger change in $v^{BCC}_{BCCenter} - v^{BCC}_{TxRS}$ is happening by the rotation. This means that the camera center in the SBC frame, will be given by:

$$v^{SBC}_{BCCenter} = v^{SBC}_{TxRS} + R_{BCC \to SBC} \cdot (v^{BCC}_{BCCenter} - v^{BCC}_{TxRS}) \tag{4.4}$$

where we take the 2 vectors difference in BCC, rotate it accordingly, and add it to the TxRS in SBC.

In order to find the initial guess of the full rotation matrix say from BCC to SBC, we assume that there are 2 sequential rotations, one that is due to the misalignment. Of course, the one that is due to the misalignment, can not be explicitly found, as we have only one vector as reference.

[3] In theory, we have 2 vectors in the BCC: TxRS that is provided by Astrolight, and BCRS measured in section 4.6. We also have the same 2 vectors in SBC, both measured as explained in section 4.5. Assuming that they are linearly independent, we can find the 3D transformation between the 2 coordinate systems. However, they are both very close to each other, and BCRS in BCC, is measured with a relatively high uncertainty. That is why we prefer to use the CAD 11.5° rotation, instead of our measurements.

	BC Center in SBC
x	−0.00097
y	−0.00746
z	0.99997

Table 4.3: BC Center in SBC vector components (rounded to 5 decimals). This is the misalignment to pass to the ADCS.

We can assume though, that the rotation is happening on the plane that is defined by the SBC Z vector, and the BCC Center in SBC. Rodriguez formula explained in appendix A.4 is very useful in such a case, as it can calculate the rotation matrix given an angle and a rotation axis. The rotation axis is given by the cross product of SBC Z vector, and BCC Center in SBC. We can calculate the transformation due to misalignments R_{msl}:

$$R_{msl} = \begin{bmatrix} 0.9999995 & -0.0000036 & -0.0009729 \\ -0.0000036 & 0.9999722 & -0.0074560 \\ 0.0009729 & 0.0074560 & 0.9999717 \end{bmatrix} \tag{4.5}$$

that as defined, can transform a vector from BCC to SBC. Then, the final BCC to SBC transformation will simply by:

$$R_{BCC \to SBC} = R_{msl} \cdot R_{BCC \to BCC1} \tag{4.6}$$

and trivially

$$R_{SBC \to BCC} = R_{BCC \to SBC}.T \tag{4.7}$$

The vector of table 4.3, is in practice the misalignment that can be put into the ADCS body pointing vector. As has been explained, putting it as a vector is not preferred, so this misalignment is going to be encoded into the Star Tracker mounting Configuration. Of course, there is an infinite amount of mounting configurations (as it requires 3 constrains, but we have 2 from the unit vector) that can produce put the BC Center in the +Z SBC axis. We need one that is closed to the actual Star Tracker mount configuration, and this is calculated, by finding the rotation matrix using the Rodriguez formula (appendix A.4), that will rotate the Star Tracker Mounting Configuration, around the cross product axis defined by the theoretical SBC Z, and the beacon camera measured SBC Z.

As a reminder, the ideal Star Tracker mounting configurations are:
Alpha (yaw): -2.5
Beta (pitch): 0.0
Gamma (roll): 90.0

The best Star Tracker mounting configuration, in order to point the center of the BC (FSM Zero Position) to the ground, are:
Alpha (yaw): -2.500169
Beta (pitch): -0.037053
Gamma (roll): 90.429225

We close this section, by showing the 2 possible ADCS configurations, that will take into account the misalignment we have measured, and they can be compared in table 4.4.

Table 4.4: Comparison between 2 possible ADCS configuration for Tracking the OGS.

Method	**Change Vector**	**Change STR MountConfig**
Star Tracker Mount Configuration	yaw = −2.5° pitch = 0° roll = 90° (the perfect mounting)	yaw = −2.500169° pitch = −0.037053° roll = 90.429225°
Control Mode to Use	ConGndTrack	ConTgtTrack
Target Tracking Vector	$\vec{BC}$ = (0.0067, −0.0253, 0.9997)	–

Chapter 5

Conclusions

> Sometimes you have to bend the rules to make them work.
>
> *Cpt. Kirk, Star Trek TOS*

Before Jumping into the conclusions, we are going to mention 3 operational examples, that combine most of the topics we explored before, in order to get a better understanding of the usefulness of our analysis.

5.1 Operational Scenarios

5.1.1 Optical Pass Planning

From the PeakSat platform perspective, the OBC has to mainly command the ATLAS terminal and the ADCS during optical pass attempts. PeakSat will perform 2 types of Payload Operation Modes (POM). No change takes place in the ADCS configuration during the pass PeakSats Commands the ADCS to change the Ground Target Tracking Vector of the ADCS during the Optical Pass, based on detected Position of the Beacon Laser by the Beacon Camera. This correction may become necessary, if during the in-orbit optical alignment phase we find that dynamic (e.g thermal, we can keep in mind that Optical passes can occur a few minutes after the satellite enters the Earth's shadow) misalignments between the platform and the ADCS are larger that a threshold. The Operator will be able to configure a series of parameters in each optical pass, in order to ensure the satellite will reach its maximum performance. Below, a high level explanation of how simulations show that these parameters can be configured will be made. These explanations are not catholic, as new results - either experimental or practical - may arise. Any change of parameters should be made with due care.

5.1.2 Optical Link Post Processing

We are going to get, GNSS, ADCS orientation and rates, and atlas detected positions We have tested that we know what position ADCS propagates, so we can assess the control error.

Since measurements from these 3 subsystems will not be on the same timestamps, we are going to bring them to one. We are going to interpolate the GNSS data and Beacon Camera detected positions to the ADCS timestamps. For the GNSS data we are going to fit an orbit to them, and this way we both cancel out random GNSS noise, and also we can propagate to the

ADCS timestamps as we want. Also, because the ATLAS BC detected position are going to be the most high frequency data that we will get, can easily choose the 2 closest points around an orientation point of the ADCS, and interpolate the X, Y position using linear interpolation to the ADCS timestamps.

5.1.3 In orbit Optical Alignment (BCC-SBC)

For this misalignment (which is practically the ATLAS beacon camera, as an SBC defined by the Star Tracker estimation), we already have an initial guess from the on-ground optical alignments. Since the measurements were performed both pre- and post-vibe, we do not expect it to change significantly (at least the static misalignment).

If the beacon is not detected reliably by the Beacon Camera—which can be understood from the pat.log file—then a trial-and-error phase will be performed. Initially, the ADCS pointing performance will be assessed by downloading Star Tracker telemetry, actual satellite positions (GNSS), and SGP4-propagated positions (used by the ADCS for pointing), in order to assess the ADCS control error. If this error is high, we will consult CubeSpace and perform simulations in order to identify which parameters must be changed.

If ADCS ground-tracking performance is verified, then we will change the ADCS ground-tracking vector (tgtTrackBodyVecX, Y, Z) in 1.5° steps. In this way, we will scan a grid (with a different step in each optical pass) to try to find the beacon laser. Even though this is not expected, such a misalignment can exist if, for example, we have an incorrect STR–STR MC characterization or a Star Tracker estimation bias.

After the static misalignments have been fitted, dynamic misalignments (such as thermoelastic ones) should be addressed. First, we should keep in mind that, since the satellite will be in an SSO orbit, all night passes will occur under similar thermal conditions; thus, large thermoelastic changes are not expected. However, this cannot be quantified further. Nevertheless, once pointing is stable, we will assess whether the SBC–BCC transformation appears to have any correlation with temperature.

The final product of this phase is the SBC–BCC transformation, which will allow us to seamlessly use vectors produced by the ADCS and ATLAS. If ADCS ground-tracking performance is verified, then we will change the ADCS ground-tracking vector (tgtTrackBodyVecX, Y, Z) in 1.5° steps. In this way, we will scan a grid (with a different step in each optical pass) to try to find the beacon laser. Even though this is not expected, such a misalignment can exist if, for example, we have an incorrect STR–STR MC characterization or a Star Tracker estimation bias.

After the static misalignments have been fitted, dynamic misalignments (such as thermoelastic ones) should be addressed. First, we should keep in mind that, since the satellite will be in an SSO orbit, all night passes will occur under similar thermal conditions; thus, large thermoelastic changes are not expected. However, this cannot be quantified further. Nevertheless, once pointing is stable, we will assess whether the SBC–BCC transformation appears to have any correlation with temperature.

The final product of this phase is the SBC–BCC transformation, which will allow us to seamlessly use vectors produced by the ADCS and ATLAS.

5.2 Conclusions

In this work we started from basic satellite concepts, and discussed with more detailed important considerations concerning an Optical Communications Mission, PeakSat. We started by

explaining the orbit of the satellite, and its visibility windows. Then we made an introduction to ATLAS, the optical payload used. The Payload has 2 apertures, one Beacon Camera, for detecting the OGS orientation, and one for the downlink laser. Then, we started presenting topics regarding the ADCS of a satellite, focusing on the ADCS of PeakSat. We saw that because for an Optical Communications Mission strict pointing requirements apply, the correct configuration of the ADCS is a necessary step towards the mission success. We put our main focus on the Star Tracker, as the most accurate estimation sensor.

Combining the 2 topics above, we made clear why a misalignment characterization between the Optical Payload and the ADCS is necessary, and we saw that this was done using the characterized alignment prism of the Star Tracker, and the reflective surfaces of the optical Payload apertures. We saw the technical details of the measurement procedure. From the measurement Beacon Camera reflective surface, that is the only measurement we have both before, and after the vibration testing, we saw that its orientation with respect to the star tracker did not change more than 0.0011°. This means that we do not expect major launch induced misalignments. Moreover, using the transmit aperture reflective surface measurement, we calculated an initial guess of the Beacon Camera Coordinates to ADCS coordinates transformation. We saw how we can explore with simulations what is the most accurate way to pass the misalignment into the ADCS, and that we decided to add it as the mounting configuration of the Star Tracker. This way we minimize both the absolute, and the relative pointing errors.

Concluding, we showed some preparation scenarios planned for when the satellite is in orbit, and combine all the topics discussed in this theses. *If they will actually take place in orbit, or not, only time will tell.*

Appendix A

3D Rotations

A.1 Rotation Matrices

Rotations (and orientations) in 3 dimensions can be denoted by several representations. One of them is the rotation matrices. A rotation matrix around a specific axis defines a rotation, and it is an one-parametric function of the rotation angle. Rotation matrices around the x, y, z axes of a Cartesian coordinate system can be found below. Rotation matrices can be understood more easy, if we think that they represent an extrinsic rotation sequence.

$$\mathbf{R}_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \tag{A.1}$$

$$\mathbf{R}_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{A.2}$$

$$\mathbf{R}_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{A.3}$$

Combining the above equations, we end up with the full rotation matrix in 3 dimensions. For example, we can denote a if we rotate first about the Z axis with an angle γ, then around the Y axis with an angle β and finally around the X axis with an angle α, we can multiply the matrices and write:

$$R(\alpha,\beta,\gamma) = R_z(\gamma)R_y(\beta)R_x(\alpha)$$

$$R(\alpha,\beta,\gamma) = \begin{bmatrix} \cos\beta\cos\gamma & \cos\gamma\sin\alpha\sin\beta - \cos\alpha\sin\gamma & \cos\alpha\cos\gamma\sin\beta + \sin\alpha\sin\gamma \\ \cos\beta\sin\gamma & \cos\alpha\cos\gamma + \sin\alpha\sin\beta\sin\gamma & -\cos\gamma\sin\alpha + \cos\alpha\sin\beta\sin\gamma \\ -\sin\beta & \cos\beta\sin\alpha & \cos\alpha\cos\beta \end{bmatrix} \tag{A.4}$$

In such an rotation, the angles α, β, γ represent a rotation sequence (check appendix A.1), and they are called Euler Angles (Landis (2019)). The angles α, β, γ can also be refereed to as the roll, pitch and yaw respectively.

Finally, a rotation matrix can be interpreted in 2 ways: The first is to transform a vector from Cartesian coordinate system, say A, to an other coordinate system B. The second, is to consider it as an attitude quaternion, that shows the relative orientations of the vectors of A, in the B coordinate system. The latter can be also interpreted as taking the unit vectors os B and rotate them until they match the unit vectors of A, but with the "opposite" angle directions with respect to the vector rotation approach.

A.2 Rotation sequences

Intrinsic Rotations: Axes of the new coordinate system are used. In a 3-2-1 rotation, first the angle gamma is applied, then beta, then alpha. Following the scipy convention, these will be denoted with capital letters "XYZ". *Extrinsic Rotation* Axes of a global coordinate system are being used. Following the 'scipy' conversion, small letters 'xyz' will be used to show the specific rotation sequence.

Theorem: Any extrinsic rotation is equivalent to an intrinsic rotation by the same angles but with inverted order of elemental rotations, and vice-versa. For instance, the intrinsic rotations $x - y' - z''$ by angles α, β, γ are equivalent to the extrinsic rotations z-y-x by angles γ, β, α. A very useful link to understand the above can be found here.

A.3 Quaternions

Quaternions are a convenient way of representing rotations and orientations in 3D space. A quaternion, is practically an extension of the imaginary numbers. It can be represented as:

$$q = q_w + iq_x + jq_y + kq_z \tag{A.5}$$

Several interesting properties of the quaternions can be found here. Using these, we show that there is a relation of rotation matrix orientations and quaternion orientations. The quaternions present the following advantages: 1) They do not present Gimbal Lock, so they can be used more convenient in control algorithms 2) Quaternion multiplications represents a sequence of rotations, as matrix multiplication represents sequence of rotations. Quaternion multiplication uses less calculations. It should be noted that matrix rotating a vector, uses less equations than quaternion rotating a vector. Finally, it should be noted that quaternions, as rotation matrices, can either represent a transformation, or an attitude representation.

Transforming from a matrix to a quaternion, needs to perform a few checks on the matrix numerical properties (Landis 2019). The quaternion multiplication of quaternions q, w is given by.

$$\begin{aligned} q_{new\_x} &= w_1x_2 + x_1w_2 + y_1z_2 - z_1y_2 \\ q_{new\_y} &= w_1y_2 - x_1z_2 + y_1w_2 + z_1x_2 \\ q_{new\_z} &= w_1z_2 + x_1y_2 - y_1x_2 + z_1w_2 \\ q_{new\_w} &= w_1w_2 - x_1x_2 - y_1y_2 - z_1z_2 \end{aligned} \tag{A.6}$$

Using this, we can find transform a rotation matrix to a quaternion: For example, for matrix M, if the Matrix Trace is larger than zero, the components of the quaternion will be given by:

$$
\begin{aligned}
q_w &= \frac{1}{2}\sqrt{1 + m_{00} + m_{11} + m_{22}} \\
q_x &= \frac{m_{21} - m_{12}}{4q_w} \\
q_y &= \frac{m_{02} - m_{20}}{4q_w} \\
q_z &= \frac{m_{10} - m_{01}}{4q_w}
\end{aligned}
\tag{A.7}
$$

The opposite relation from a quaternion to a matrix is given by:

$$
R = \begin{bmatrix}
1 - 2(q_y^2 + q_z^2) & 2(q_x q_y - q_z q_w) & 2(q_x q_z + q_y q_w) \\
2(q_x q_y + q_z q_w) & 1 - 2(q_x^2 + q_z^2) & 2(q_y q_z - q_x q_w) \\
2(q_x q_z - q_y q_w) & 2(q_y q_z + q_x q_w) & 1 - 2(q_x^2 + q_y^2)
\end{bmatrix}
\tag{A.8}
$$

Rotation sequences in quaternions, can be denoted using quaternion multiplications using eq. (A.6).

A.4 Rodriguez Formula

Rodriguez formula is very useful, as it can compute the rotation matrix, if we provide to it an axis that the rotation will happen around it, and an angle theta. Following Xie (2022), if $c = \cos\theta$. $s = \sin\theta$, $t = 1 - \cos\theta$ and u_x, u_y, u_z are the unitary axis components, then the rotation matrix is given by:

$$
R = \begin{bmatrix}
c + u_x^2 t & u_x u_y t - u_z s & u_x u_z t + u_y s \\
u_y u_x t + u_z s & c + u_y^2 t & u_y u_z t - u_x s \\
u_z u_x t - u_y s & u_z u_y t + u_x s & c + u_z^2 t
\end{bmatrix}
\tag{A.9}
$$

Appendix B

Optimizing Access to Aligmnent Prism

B.1 Optimizing Access to Alignment prism

The star tracker alignment prism is an extruded orthogonal trianlge shape, with the triangle sides to be 1cm x 1cm, and the extrusion sides to be 1cm. To perform a successful optical alignment we need to have an access to at least 2 sides of the alignment prism. As can be seen in **??** from the $+x$ side of the star tracker we have line of sight access to the whole surface of the alignment prism. From the $+y$ side (**??**) we have to make a cut-out to the PCB of the panels. If the star tracker is placed on the frame, with zero yaw, pitch and roll, then from the y side we will have line of sight to a surface of about $0.3cm^2$. This limit, does depend on the positioning of the star tracker in the frame, and the position of the alignment prism coincides with a part of the frame. If we also imagine that the star tracker will not be positioned perfectly, with its axes aligned with the satellite frame aces, and also that the alignment prism, will not be perfectly mounted to the star tracker (of course it will be characterized) then this $0.3cm^2$ surface can potentially reduce significantly.

The solution that we propose is the following: to rotate the star tracker with an angle, that we call ω. Angle ω can be visualized in fig. B.1 and **??**. We chose for it to have an value of 2.5° because that way we can optimize the vertical line of sight that we can have to the alignment prism of the star tracker, without interfering with other components of the satellite. That way, the surface of the alignment prism that we can see vertically, will be $4.6cm^2$ that is about the same as the surface from the $+x$ side.

Design of this rotation performed by Mr. Theodoris Balsis, (TODO), and more details can be found in his thesis.

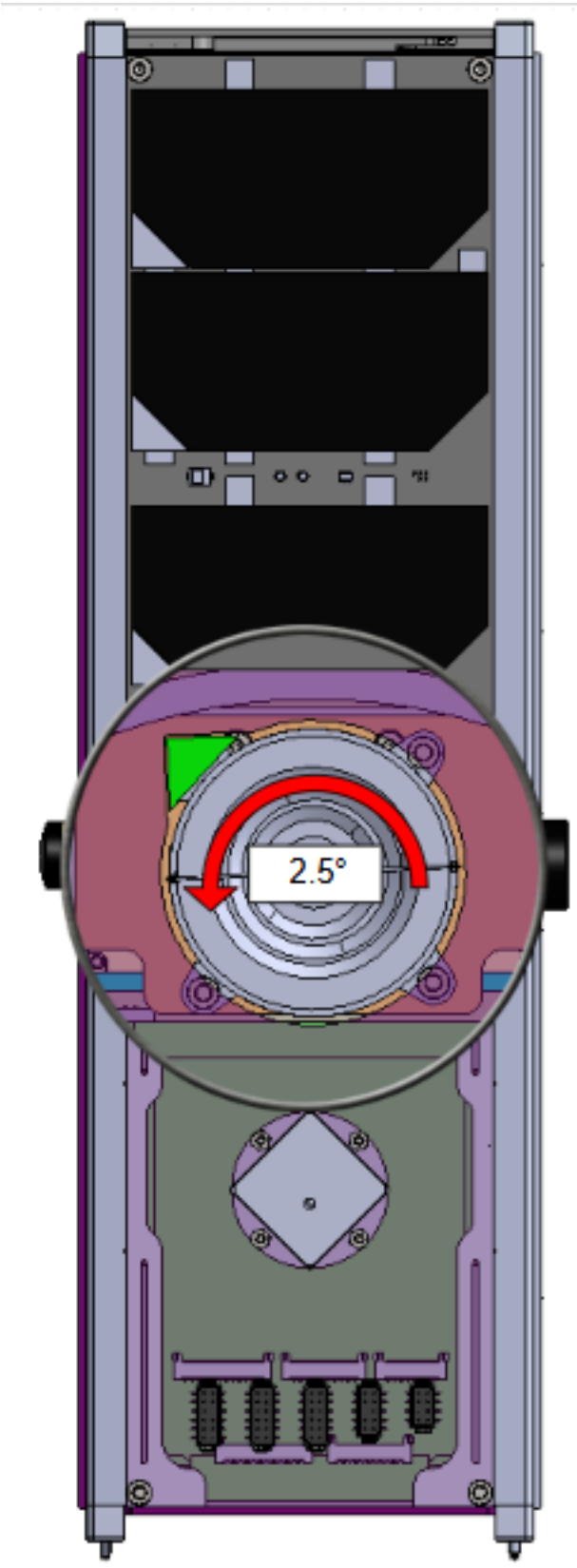

Figure B.1: Overview of the rotation of the star tracker, as seen from the +x side of the satellite.

Appendix C

Gaussian Error Propagation

C.1 Error propagation

All measurements are assumed to follow a Gaussian distribution, with the measurement value to be μ and the uncertainty to be σ. In the event of multiple measurements of the same value, the final probability distribution will be given by:

$$\overline{X} = \mathcal{N}\left(\overline{X}, \frac{S^2}{N} + \frac{\sigma^2}{n}\right)$$

where $\overline{X}$ is the average of the individual values:

$$\overline{X} = \left(\sum_i^n \frac{\mu_i}{n}\right)$$

σ^2 is the measurement uncertainty, and S^2 is the sample variance:

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

As it is obvious, S^2 is added to the standard deviation of $\overline{X}$, in order to take into account also the spread of the specific experiments. If we also assume that different measured quantities are independent with each other, and a value that we want to deduce is given by $y = f(x_1, x_2..., x_n)$ where x_i are measurements, then the error of y will be propagated by the equation:

$$\sigma_y^2 = \sum_{i=1}^{n}\left(\frac{\partial f}{\partial x_i}\right)^2 \cdot \sigma_{x_i}^2$$

Bibliography

Collaboration, A., Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, The Astronomical Journal, 156, 123

Curtis, H. 2012, Orbital Mechanics for Engineering Students, iSBN: 9780080977478

Fitzpatrick, R. 2012, An Introduction to Celestial Mechanics (Cambridge: Cambridge University Press)

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, Nature, 585, 357

Hunter, J. D. 2007, Computing in Science & Engineering, 9, 90

Landis, L. 2019, Fundamentals of Spacecraft Attitude Determination and Control

Maisonobe, L. et al. 2010, Proceedings of the 4th International Conference on Astrodynamics Tools and Techniques

McKinney, W. et al. 2023, Zenodo

Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, Nature Methods, 17, 261

Xie, Y. 2022, Spacecraft Dynamics and Control