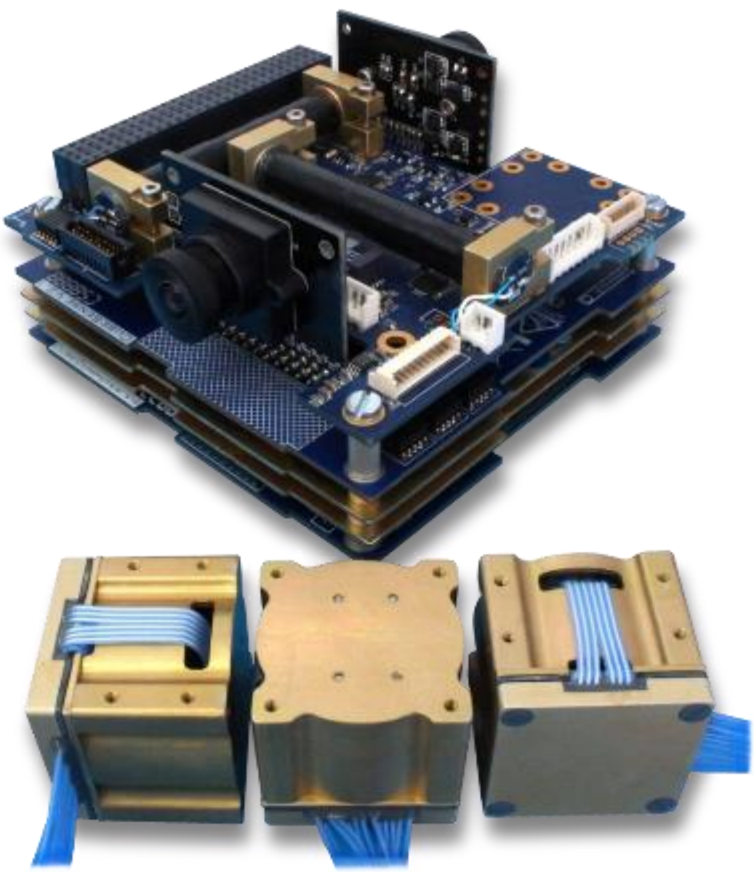
CubeADCS

The complete ADCS solution



User Manual

Contact Us
Phone (0027) 79 945 9957
E-mail info@cubespace.co.za
Web www.cubespace.co.za
Facebook /CubeSpaceADCS
Twitter @CubeSpace_ADCS

Physical Address
CubeSpace
Hammanshand Road
Stellenbosch
7600
South Africa

Document	CubeADCS User Manual
Version	3.09
Domain	Public
Date modified	31 March 2020
Approved by	Name: Christo Groenewald
	Signature:

List of Acronyms/Abbreviations

ACP	ADCS Control Program
ADCS	Attitude Determination and Control System
CSS	Coarse Sun Sensor
ESD	Electrostatic Discharge
I^2C	Inter-Integrated Circuit
MCU	Microcontroller Unit
MEMS	Microelectromechanical System
OBC	Onboard Computer
PCB	Printed Circuit Board
RTC	Real-Time Clock
SBC	Satellite Body Coordinate
SPI	Serial Peripheral Interface
TC	Telecommand
TLM	Telemetry
UART	Universal Asynchronous Receiver/Transmitter

Relevant CubeACP Version

This document serves as reference to the CubeACP with the following version numbers.

	Version
ACP version(s)	3.19, v3.20, v3.21, v6.x
Node type identifier	10
Interface version	6

Relevant reference documents

This document is to be used in combination with the following documents:

Reference	Document name	Document Version
Ref 1	CubeADCS - ICD	V3.16 or higher
Ref 2	CubeADCS – Hardware Configuration Sheet	Completed by user at order of unit
Ref 3	CubeADCS – Health Check	V3.24 or higher
Ref 4	CubeADCS – Firmware Reference Manual	V3.13 or higher
Ref 5	CubeADCS – CubeSupport Manual	V3.0 or higher
Ref 6	CubeADCS – Configuration Sheet	V3.0 or higher
Ref 7	CubeADCS – Commissioning Manual	V3.0 or higher

Table of Contents

1 Introduction

The purpose of this document is to provide the user with:

- instructions of completing the initial setup of the CubeADCS unit for the health check
- instructions for connecting the unit to an OBC
- a guide to mounting sensors inside a satellite and calculating mounting transforms.
- a functional description of the control modes that the CubeADCS unit is capable of executing
- general usage instructions for all major functionality
- an overview of the hardware included in this CubeADCS and their performance

The document is to be used in combination with the **Ref 1** and **Ref 4** to: (a) design the software for interfacing with the ADCS unit; (b) connect the unit electrically and mechanically to the rest of the satellite; (c) configure the unit correctly; and (d) plan the ADCS operations once in space.

A CubeADCS can be configured in various ways with the system having more or less sensors and actuators to accommodate different mission ADCS objectives. This document is applicable to all CubeADCS configurations in general. Certain sections are only applicable to a CubeADCS that contains the relevant sensors or actuators. The user's discretion is required to determine whether or not a specific section is relevant to the CubeADCS under investigation.

2 Handling

Anti-static
The CubeADCS contains a variety of static sensitive devices. The appropriate electrostatic discharge (ESD) protection measures must therefore be implemented. The unit must never be handled without proper grounding.

Cleanliness
It is recommended that the CubeADCS unit be handled in a clean environment. Therefore, an appropriate laminar flow workbench or clean room of ISO class 8 or better, should be used.

Moisture
The unit should be kept free of moisture and liquids, as these could have corrosive effects on the electronics and electronic joints, which may lead to degradation and loss of reliability of the circuits.

Shock
The unit must be handled with care. Dropping or bumping the unit should be avoided completely.

Camera lens cleanliness
The camera lenses should be kept clean and free of dirt that may obstruct the images captured by the camera. Dust should be removed with a microfibre cloth. The lens may be cleaned using ethanol and appropriate lens cleaning equipment, if required. Avoid unnecessary cleaning of the lens should be avoided.

Camera lens structural integrity
The camera sensors are aligned in parallel with the CubeSense PCB. This is important as misalignment of the cameras influence the performance of the system. External forces on the camera modules should therefore be avoided entirely.

Camera lens covers
The Sun and nadir optics are fitted with dust caps which should be removed before flight.

Reaction/momentum wheel(s)
The aluminium housing of the reaction/momentum wheel(s) should NOT be tampered with. Tampering with the housing may damage the wheel. No attempt should be made to loosen or remove the fasteners that secure the housing.

3 Getting Started

3.1 Unpacking and setting up the unit

The CubeADCS will be packaged in anti-static bags inside a Peli-Case. Remove the unit inside a clean environment using gloves. Use a sturdy workbench that is covered with a clean anti-static mat. Any person working on the unit must be grounded with an anti-static wristband.

Magnetometer measurements will be distorted if a workbench containing ferrous parts, such as iron; steel; stainless steel; etc, are used. Working on such a surface will have a dramatic effect on sensor measurements and may cause observed results to differ significantly from reference results.

In addition to the CubeADCS itself, the Peli-Case will contain all or some of the following components:

- Deployable magnetometer
- Redundant magnetometer (optional)
- Coarse Sun sensors
- CubeSense unit(s) (depending on configuration)
- CubeConnect (optional)
- CubeStar (optional)
- CubeWheels
- Harnesses between CubeADCS and its sensors/actuators (depending on configuration)
- UART-to-USB cable
- Flash drive

Remove all parts from their anti-static bags and place them on the bench.

Take care to avoid touching the optics of any of the camera-based sensors.

Next, connect all peripherals to the ADCS stack - refer to **Figure 1**.

1. Connect the GPS to the GPS header on the top of the CubeADCS stack (if GPS is supported)
2. Connect the redundant magnetometer (if redundant magnetometer supplied).
3. Connect the deployable magnetometer using the in-line round connector.
4. Connect the coarse Sun sensor in-line connector to the ADCS stack and connect each photodiode to the in-line connector.

5. Connect CubeSense units to CubeConnect via harness. Make sure to connect CubeSense 1 to the connector for Camera 1, and CubeSense 2 to the connector for Camera 2.
6. Connect wheels to CubeConnect with harness (if not already mounted on CubeConnect)
7. Connect CubeStar to CubeConnect (if CubeStar supplied)

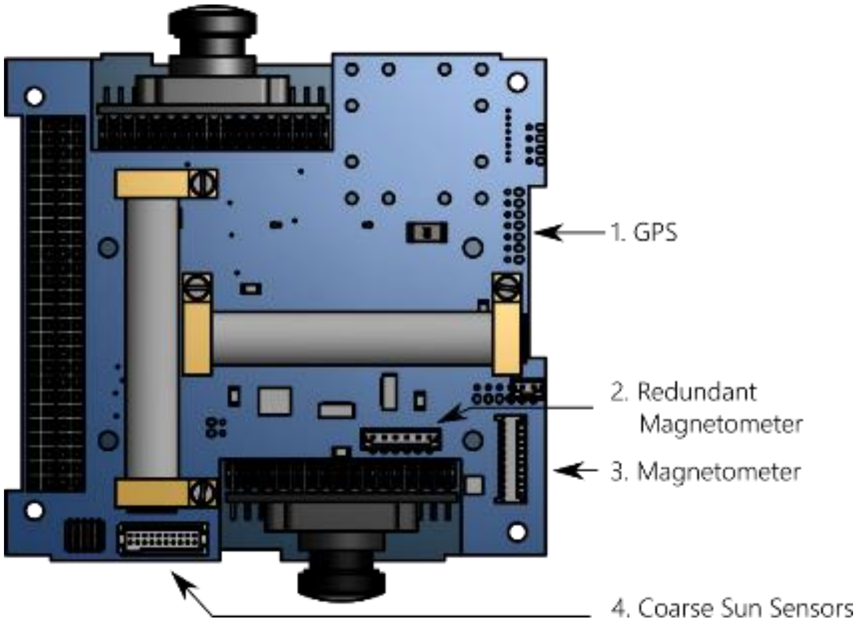


Figure 1 – Peripheral connectors (configuration may differ depending on options).

3.2 Powering the unit

A bench power supply is required to power the ADCS stack. Ensure that this supply is set to the battery voltage that the satellite will operate at and that the current limit is set to at least 1 A. This current limit should be high enough to supply the inrush currents that components such as the GPS (optional) or CubeSense draws upon start-up. Try to keep cables between the power supply and the ADCS short and thick, to try to reduce series resistance.

A current limit set too low or a series resistance set too high in the power connection can cause the system to experience voltage dips when powering up. This may lead to unexpected behaviour and/or resets.

Refer to **Ref 1**and **Ref 2** for the relevant PC104 pin locations when connecting power to the stack. It is preferable to use a voltage regulator to supply the 3.3 V and 5 V sources.

3.3 Performing health check

It's important to perform the CubeADCS Health Check upon delivery to ensure the unit is in full working condition before further testing/integration occurs. Additionally, it's a great intro on how the CubeADCS functions. This testing is done by following process described the CubeADCS Health Check document. This process tests most of the CubeADCS' functions and compares it to tests run on the same unit by CubeSpace, immediately before shipping the unit. The Health Check document can typically be found on the accompanying Flash Drive.

4 Interfacing with the Unit

4.1 ADCS System description

4.1.1 System diagram

To estimate the attitude of the satellite, the CubeADCS unit uses all or a sub-set of the following sensor measurements (depending on hardware selected):

- Magnetometer(s), coarse sun sensors, MEMS rate sensor(s)
- Fine sun sensor, Fine Earth sensor
- Star tracker

Estimation and control algorithms are run on the ADCS on-board computing unit (CubeComputer) that is included in the CubeADCS.

It uses magnetorquers and, if applicable, one momentum wheel or three reaction wheels to control the satellite's attitude.

The CubeADCS unit consists of up to four integrated PC104-standard PCBs and several peripheral components, which are to be mounted separately. A basic diagram of a complete CubeADCS solution with all options is shown in **Figure 2**.

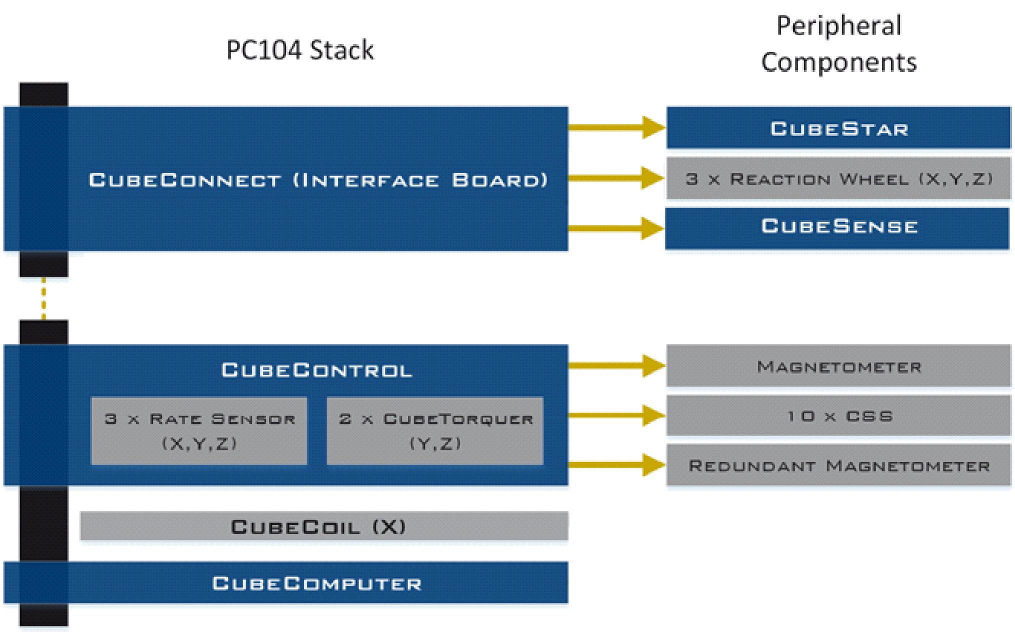
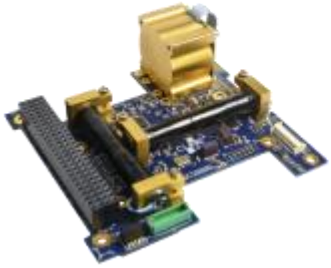
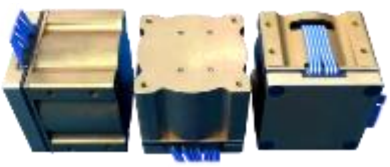


Figure 2 – ADCS stack.

Note that the reaction wheels can be selected as small, medium or large. Small CubeWheels can be mounted on CubeConnect in any axis, whereas medium and large CubeWheels must be mounted separately from the CubeADCS.

A brief description of the core components of the CubeADCS 3-Axis is given in Table 1.

Table 1 – Modules of the CubeADCS solution.

Module	Description
CubeControl 	CubeControl is an actuator and sensor interface module for nanosatellites with advanced attitude control requirements. It can control 3 magnetorquers and a momentum wheel. CubeControl can also interface with 2 magnetometers, 10 coarse Sun sensors, and 3 MEMS rate sensors. **NB: The momentum wheel mounted on CubeControl is not included in the CubeADCS 3-Axis.**
CubeSense 	CubeSense is a compact integrated CMOS sensor that can serve as either a sun sensor or a nadir sensor (in sunlight). The FOV of the fish-eye optics is at least 160°.
CubeComputer 	CubeComputer is a generic CubeSat OBC. It can perform the required ADCS functions, as well as the satellite's main OBC tasks. The module is based on ARM Cortex-M3 architecture and implements error detection and correction (EDAC) techniques.
CubeWheel 	CubeWheel is a compact standalone reaction wheel unit for nanosatellites. It provides the satellite with the ability to achieve 3-axis stability and 3-axis control. Each CubeWheel is magnetically shielded and is mountable in 3 axes. Various sizes are available to suit every need.
CubeStar 	CubeStar is a compact star tracker for nanosatellites. The module is based on ARM Cortex-M3 architecture and has an exceptionally low power consumption.

4.1.2 Coordinate System Definitions

The ADCS coordinate system is related to the satellite body coordinate (**SBC**) system through a transformation matrix. When the ADCS is controlling the attitude to zero roll, pitch, and yaw angles, the SBC system will coincide with the orbit coordinate system (**ORC**, or also referred to as the nominal orientation).

All actuators and sensors each have their own coordinate systems. Each sensor and actuator mounting have to be defined relative to the SBC through a transformation matrix. This means that the transformation matrix for each actuator or sensor should be known.

The mounting information is part of the configurable settings for the ADCS, and has to be specified for correct operation of the unit (see section 5.4 for detail).

The ORC and SBC are further explained in the following subsections.

4.1.3 Orbit reference coordinate (ORC)

The **orbit reference coordinate (ORC)** frame, notated as $\mathbf{X_{ORC}}$, $\mathbf{Y_{ORC}}$, and $\mathbf{Z_{ORC}}$, is defined throughout the CubeADCS literature as:

Axis	Pointing Direction
X_{ORC}	Flight Direction
Y_{ORC}	Orbit anti-normal direction
Z_{ORC}	Nadir direction

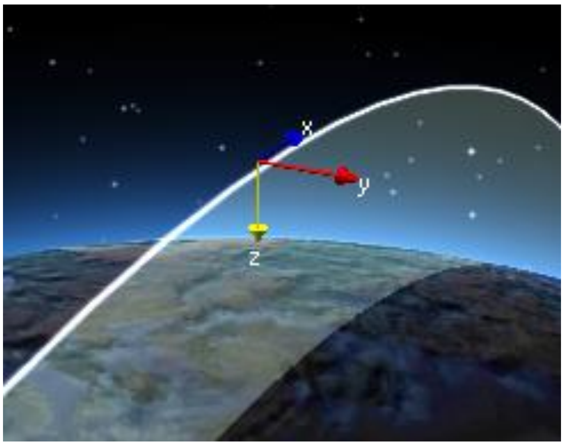


Figure 3 Orbit coordinate system

4.1.4 Spacecraft body coordinates (SBC)

The **spacecraft body coordinates (SBC)** frame is notated as $\mathbf{X_{SBC}}$, $\mathbf{Y_{SBC}}$, and $\mathbf{Z_{SBC}}$.

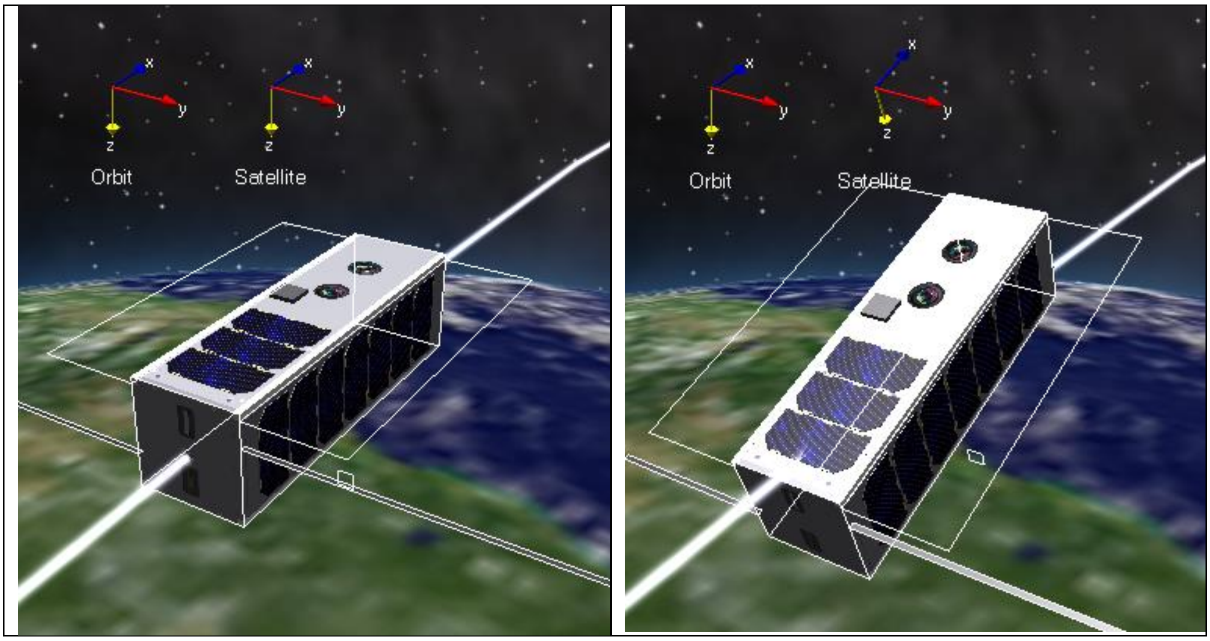


Figure 4 – Satellite (spacecraft) Body Coordinate system, relative to the Orbit Reference Coordinate system for zero roll, pitch and yaw (left image) and a 20° pitch angle (right image)

In nominal flight, the SBC and ORC frames will be aligned for zero roll, pitch, and yaw. The SBC is also not necessarily the coordinate system placed on the mechanical design of the satellite, but rather chosen given the desired flight orientation when the SBC is commanded to follow zero pitch, roll and yaw with the ORC.

The ADCS sensor and actuator mounting configurations are done in the SBC

4.1.5 CubeComputer software options

The computer in the CubeADCS unit (CubeComputer) typically runs only CubeSpace's standalone ADCS control program (ACP). However, the user can optionally purchase the ability to run their own ADCS algorithms (limited to only controllers and estimators) on CubeComputer, while having the standard ACP algorithms as backup. Please contact CubeSpace if this functionality is desired.

4.2 Connecting with OBC

CubeADCS units have the option of UART, I2C and CAN as communication interface. Refer to **Ref 1** and **Ref 2** for the relevant PC104 pin locations of the communication interface to the OBC. The telecommand and telemetry definitions for all interfaces are the same, but the protocols differ slightly. Please refer to **Ref 4** for details on the protocol of each comms interface.

4.3 Mounting in satellite

Placement of the ADCS actuators or sensors in the satellite has a significant influence on the ADCS performance of the system. In some cases, bad placement can even cause the ADCS to completely fail. The following sections will provide guidelines for the layout of the various components of the CubeADCS in the satellite structure.

4.3.1 Magnetometer

The magnetometer is the part of the ADCS that is most sensitive to bad placement. The magnetometer measures the Earth's magnetic field very accurately, and any magnetic disturbances near the magnetometer directly influence the accuracy of the measurements. For this reason, the magnetometer should not be placed in close proximity of any other part of the satellite that:

- **Generates large magnetic fields -** the most common source of such generated magnetic fields is any part that draws large currents.
- **Is ferrous -** ferrous components warp the magnetic field. Even though the ferrous components are not a source of the magnetic field itself, they warp the Earth's field and cause an inaccurate measurement on the magnetometer. Further, these parts can be magnetised by the magnetic torquer rods. In this case the ferrous part keeps a magnetic dipole even after the torquer rod is switched off.

Common parts of the satellite that causes significant disturbances on the magnetometer include:

- **Stainless steel** screws or structural parts – use austenitic types of SS if required.
- **Solar panels** and **solar panel harnesses.**
- Any other **harness carrying large currents.**
- **Electrical motors -** motors have permanent magnets that are of considerable strength.
- **Antennas -** significant magnetic fields are generated specifically at the point where the antennas are fed.

As a rule of thumb, the magnetometer should ideally be kept 80+mm from any of the above-mentioned items.

Another important consideration with regards to the **mounting of the deployable magnetometer is to avoid interference with other sensors**. Fine Sun sensors, nadir sensors, or star trackers typically have a relatively large field of view. If they are placed incorrectly, the magnetometer **can obstruct the view of these sensors after being deployed.**

The magnetometer cable **must not** be routed inside the satellite a manner such that the cable **is tight** where it exits from the magnetometer. **The cable must exit perpendicular to the mounting facet for 1 cm, it should have 2 cm slack and must be loose (movable with slight touch). See Figure 5 for the correct illustration. Failure to conform may influence magnetometer deployment significantly.**

Figure 5 – Magnetometer cable slack.

4.3.2 Coarse Sun sensors

The coarse Sun sensors should be mounted on the external surfaces of the satellite panels. It is necessary to ensure that these sensors are not shadowed by other deployable structures. Ten coarse Sun sensors are supplied. The four extra photodiodes can be placed on any faces, bearing in mind that shadowing is to be minimised. The sensors should be glued down using any space-grade epoxy.

Ensure that the photodiodes are kept clear of epoxy.

If solar panels already have photodiodes mounted on them, please contact the CubeSpace team to discuss the possibility of pairing these diodes with the CubeADCS.

The connectors on CubeControl need to be epoxied prior to final integration.

4.3.3 Wheels

CubeWheels are covered in MuMetal to shield the rest of the satellite from the strong magnets inside the electrical motors of the wheel. This shielding significantly warps the magnetic field in close proximity, therefore it should not be mounted within 4 cm of the magnetorquers or the magnetometer.

Even though all CubeWheels are balanced, they still have some residual imbalance that will cause vibrations. Typically, reaction wheel vibrations may cause problems in missions where imaging payloads with narrow FOV are used - the star tracker is a good example. In this case it is important not to mount the wheel directly on the same mounting surface that the camera is mounted. Some structural separation between the wheels and the camera payload would allow for damping of the vibrations through the structure.

4.3.4 Star tracker

As mentioned before, the star tracker should not be mounted on the same surface as the wheels.

The star tracker should be placed in such a way that it will point away from the Sun and Earth in the nominal flying orientation. Deployable structures should also not impede the view of the camera. It is recommended that CubeSpace is always contacted to discuss the star tracker mounting, and the possibility of a baffle to reduce stray light.

4.3.5 CubeSense

CubeSense is less susceptible to noise from the CubeWheels and typically the CubeSense sensors can be mounted in close proximity to the wheels.

The CubeSense sensors must be mounted in an orientation where the nadir camera is pointing nadir during nominal satellite operation. The Sun camera(s) should typically point zenith, or orbit normal towards the Sun's side of the orbit. The exact placement of the Sun sensor(s) would depend on the orbit the satellite is launched in.

CubeSense can mask interfering deployable structures in the nadir camera's field of view, but it is highly recommended to avoid having any obstructions as this will greatly decrease its detection capability. The Sun sensor is more robust against obstructions and does not have the masking capability. However, it is necessary to ensure that there are no surfaces that will reflect direct sunlight to the Sun sensor, as this may cause false measurements.

An important mounting consideration is ensuring that the camera lens protrudes completely through the side panels of the satellite, as demonstrated in Figure 6. The CubeSense cameras have 180-degree fisheye lenses, therefore if the lens does not protrude all the way through the side panel, the sensor will detect reflections from the side panels.

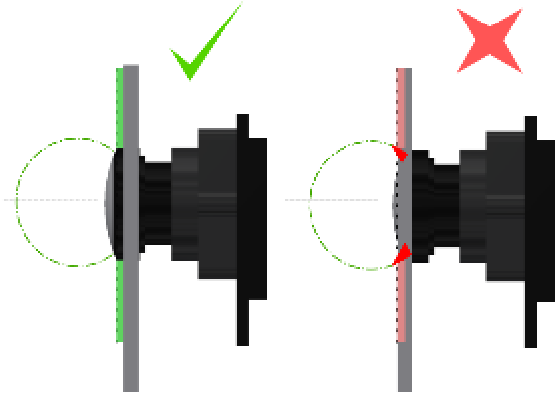


Figure 6 – Correct and incorrect CubeSense camera mounting.

4.3.6 Magnetorquers

Magnetorquers generate a magnetic dipole that interacts with Earth's magnetic field to generate torque. This generated magnetic field can be distorted by any ferrous objects on the satellite. Therefore, it is advised that completely non-ferrous metals are used for all mechanical parts. If any satellite parts contain stainless steel (typically in fasteners), the austenitic types of stainless steel should be used.

Reaction wheels contain relatively large magnets and magnetic shield material which both influence the magnetic field in their proximity. The magnetorquers and reaction wheels should therefore be separated.

Please contact the CubeSpace team when unsure of any mounting or layout of the ADCS components.

4.3.7 Epoxy of connector cables

It is recommended to epoxy all connector cables to peripheral components prior to final integration.

4.4 ADCS operational modes

Whether or not the CubeComputer is running the standalone or unlocked ACP, the ADCS has modes that determine what estimators and control algorithms are run, and at which frequency they are executed. This section gives a description of the different mode-settings of the ACP, as well as the options for each of these modes.

4.4.1 ADCS run modes

The *ADCS Run Mode (ADCS State)* is an enumeration that specifies how the ACP control loop behaves.

- In the *Off* state, the loop does not execute. The ACP is idle but still responds to commands and telemetry requests.
- In the *Enabled* state the control loop will execute with a 1s period.
- In the *Triggered* state, single iterations of the control loop are executed only when a specific command is received. On receiving a *Trigger ADCS Loop* command (see **Ref 4**), one iteration of the control steps execute and the processor will remain idle until another *Trigger* telecommand. This mode is typically used for HIL setups.

This mode only applies to the ADCS control loop. All other functions of the ACP remain active and it will respond to commands and telemetry requests.

4.4.2 ADCS estimation modes

The *Estimation Mode* determines which sensor measurements are used and what information is estimated. The current estimation mode is changed by Set Attitude Estimation Mode telecommand (TC ID 14). The available estimation modes are shown in Table 2, as well as which sensors are used in which mode.

Table 2 – Estimation modes.

Enumeration Value	Estimation Mode	Sensors used	Estimated information
0	No attitude estimation	None	None
1	MEMS rate sensing	X,Y,Z –axis rate sensors	X,Y,Z angular rates
2	Magnetometer rate filter	Magnetometer	X,Y,Z angular rates
3	Magnetometer rate filter with pitch angle estimator	Magnetometer	X,Y,Z angular rates; pitch angle
4	Magnetometer and fine or coarse Sun TRIAD algorithm	Magnetometer; CSS and/or Sun sensor	Roll, Pitch & Yaw angles; X,Y,Z angular rates
5	Full-state EKF	Magnetometer; Sun sensor; Nadir sensor; Star tracker	Roll, Pitch & Yaw angles; X,Y,Z angular rates
6	MEMS Gyro EKF	X,Y,Z–axis rate sensors; Magnetometer; Sun sensor; Nadir sensor; Star tracker	Roll, pitch & Yaw angles; X,Y,Z angular rates; X,Y,Z gyro bias

Estimation Mode 1 does not perform any estimation algorithms, but rather uses the rate sensor measurement directly. It provides angular rate measurements for all the body axes. The rate sensor bias is not estimated, but simply compensated for using a temperature dependant offset. The constant offset can be changed or read back from the ACP using the *Rate Gyro Configuration* messages *(TC ID 23 / TLM ID 138)* and is also included in the *ADCS configuration message (TC ID 20 / TLM ID 206)*.

<u>Estimation Mode 2</u> makes use of successive magnetometer measurements in a robust Kalman filter to estimate the X, Y and Z angular rates. The system noise covariance used in the filter can be changed by the *Estimation Parameters message (TC ID 27 / TLM ID 223)*.

<u>Estimation Modes 3 to 6</u> provide both angular rates and attitude angle estimates, and they make use of modelled vectors for the magnetic field and/or Sun. To calculate these modelled vectors, the orbital position and velocity of the satellite must be known. The latter information is obtained from an SGP4 orbit propagator. ***The mean orbital elements used by the propagator must be updated regularly to ensure that the estimation errors remain low***. Orbital elements are updated by sending a *SGP4 Orbit Parameters* telecommand *(TC ID 45)* to the ACP. The orbit parameters can be stored in flash memory by the *Save Orbit Parameters telecommand (TC ID 64)*.

<u>Estimation Mode 3</u> is the same as Mode 2, but also includes simple pitch estimation from magnetometer measurements only.

<u>Estimation Mode 4</u> combines the magnetometer rate filter of mode 2 with a simple TRIAD algorithm. The latter gives the estimated attitude angles based on matching modelled and measured vectors from the coarse or fine Sun sensor and magnetometer. This mode will not yield attitude angle estimates in eclipse, since the Sun sensor will not give a valid output. The TRIAD estimation mode can be used to initialize the EKF state vector so that a faster EKF convergence time is achieved (this, however, is not a requirement).

<u>Estimation Mode 5</u> uses an Extended Kalman Filter (EKF) to estimate the full attitude state. The system and measurement noise covariance used in the filter can be changed by *Estimation Parameters telecommand (TC ID 27 / TLM ID 223)*. The Kalman filter estimators make use of the satellite moment of inertia tensor, and estimated control torque in order to propagate the attitude state (the moment of inertia and control torque is used in the Euler dynamics equation). ***The satellite moment of inertia tensor should ideally be measured before flight and is part of the configuration block***. The *Moment of Inertia Matrix* telecommand *(TC ID 41 / TLM ID 222)* can be sent to update it. The control torque used in the propagation step is calculated from the actuator commands and model parameters for the actuators. The latter are also configuration parameters modifiable via *Magnetorquer Configuration* telecommand *(TC ID 21 / TLM ID 136)* and *Wheel Configuration* telecommand *(TC ID 22 / TLM ID 137)*.

<u>Estimation Mode 6</u> uses an EKF with the MEMS rate (gyro) measurements to estimate the attitude and rate sensor offsets (gyro bias). As before, the measurement noise covariance can be changed with the *Estimation Parameters telecommand (TC ID 27 / TLM ID 223)*. This EKF does not require the satellite moment of inertia tensor, control or disturbance torques as the angular rates are directly measured and the Euler dynamic model is not required. This EKF is therefore more robust against un-modelled external disturbance torques and actuator modelling errors.

4.4.3 ADCS control stages and control modes

The ADCS has three distinct control stages namely *Detumbling, Y-momentum and, XYZ wheel* control. In each stage, different control modes and estimation modes are used.

The first stage, *Detumbling*, is used to recover from any initial tumble condition and place the satellite in a slow, stable and known tumbling motion - the so-called Y-Thomson or Z-Thomson spin. In this stage the satellite will end-up spinning only about the body Y/Z-axis and the spin axis will align itself with the orbit normal or Sun vector direction.

The second control stage, *Y-momentum* control, can only be activated once the satellite is in a stable Y-Thomson tumbling state. In *Y-momentum* control the satellite will stop spinning and stabilize to the nominal orientation (zero roll, pitch and yaw angles). During *Y-momentum* stabilization the pitch angle may be controlled to a specific pitch reference value using the *Commanded Attitude Angles* telecommand *(TC ID 15 / TLM ID 199)*.

The third control stage, *XYZ wheel* control uses 3-axis reaction wheels with zero momentum bias. This stage has a control mode to track the nadir vector (zero roll, pitch and yaw angles) or constant attitude reference angles which are set using the *Commanded Attitude Angles telecommand (TC ID 15 / TLM ID 199)*. It also has control modes to track the Sun vector with the satellite's solar panels for maximum power or to track a reference target at a specific location on Earth. The target must first be selected with the *Tracking Controller Target Reference* telecommand *(TC ID 55 / TLM ID 200)*.

4.4.3.1 Detumbling

The *Detumbling* stage can make use of 4 control modes.

- Control Mode 1 uses a *Bdot Detumbling* magnetic controller to dump the X- and Z-body rates. Only instantaneous magnetometer measurements are required and only the Y-axis magnetorquer is used.
- Control Mode 2 adds *Y-Thomson* Spin to Mode 1 using the X- and Z-axis magnetorquers. It requires an estimate for the Y-angular rate. Thus, either Estimation Mode 1 or 2 should be active. Control Mode 1 and 2 can only detumble the satellite from angular rates below 30 deg/s. For higher initial rates, Control Mode 3 or 4 can be used.
- Control Mode 9 is also a magnetic controller running at 1 Hz using all magnetorquers and instantaneous magnetometer measurements. Mode 9 can detumble initial body rates in all axes as high as 100 deg/s down to zero.
- Control Mode 8 is similar to Mode 9, but is implemented in the CubeControl subsystem at 10 Hz. Mode 8 can detumble initial body rates in all axes as high as 1000 deg/s down to zero. For body rates above 35 deg/s the Magnetometer Rate Filter (Estimation Mode

2) will not be accurate and the MEMS rate sensors (Estimation Mode 1) should be used. The range of the MEMS rate sensors must be configured with the *Rate Gyro Configuration* telecommand (TC ID 23 / TLM ID 138) depending on the expected maximum body rate. For example, a *RateSensorMult* value equals 1 for a maximum rate of ±75 deg/s, the value equals 12 for a maximum rate of ±900 deg/s, etc.

The *Detumbling Control Parameters telecommand (TC ID 38 / TLM ID 208)* can be used to set the controller gains and reference spin rate ($\omega_{y,ref}$) for Control Mode 1 to 3, 8 and 9.

4.4.3.2 Y-Momentum

The *Y-momentum* stage requires pitch angle and body rate estimates, thus either the magnetometer rate filter with pitch estimation (Estimation Mode 3) or the Full-state EKF (Estimation Mode 5) or the MEMS Gyro EKF (Estimation Mode 6) must be active. The *Y-momentum* stage contains two separate control modes. In the initial mode (Control Mode 3), the Y-wheel speed will be ramped up to an initial reference momentum. This should absorb most of the body Y-Thomson spin, but keep a slow pitch rotation about the Y-axis. This will continue until the estimated pitch angle is within 25 degrees of the reference pitch angle. At this point the steady-state Y-momentum control mode (Control Mode 4) will be entered, which uses a PD controller to control the pitch angle to the reference pitch angle. This transition will occur automatically on the ACP and reading the *Current ADCS State telemetry (TLM ID 132)* will confirm if the transition has occurred.

In the steady-state Y-momentum mode (Control Mode 4) the magnetorquers are used to maintain the wheel momentum to a reference value, and to damp nutation in the X- and Z-body axes. The magnetorquer and Y-wheel PD controller gains as well as the reference Y-wheel momentum used by this controller can be changed by *Y-Wheel Control Parameters telecommand (TC ID 39 / TLM ID 209)*.

4.4.3.3 XYZ wheel

The *XYZ wheel* stage implements 3-axis zero bias reaction wheel control and requires full attitude and body rate estimates, thus either the Full-state EKF (Estimation Mode 5) or the MEMS Gyro EKF (Estimation Mode 6) must be active. This stage contains 3 separate 3-axis stabilization control modes. In the first mode (Control Mode 5) the nominal attitude will be nadir pointing (for zero roll, pitch and yaw angles), but any constant roll, pitch and yaw reference attitude can be commanded using the *Commanded Attitude Angles telecommand (TC ID 15 / TLM ID 199)*. The second mode (Control Mode 6) is used to track the Sun vector normal to the satellite's (typically deployed) solar panels for maximum power. The body facet to track the Sun will be preconfigured in the code (currently the +Y, –Y or –Z facet can be preconfigured). The *XYZ reaction Wheel* Q-feedback PD gains can be changed by *Reaction Wheel Control Parameters telecommand (TC ID 40 / TLM ID 217).*

The third mode (Control Mode 7) is used to track a reference ground target at a specific location on Earth. The target must first be selected with the *Tracking Controller Target Reference* telecommand (TC ID 55 / TLM ID 200). The calculation of the Ground Target Reference parameters is discussed in Section 14.1 of **Ref 7**. The *XYZ reaction wheel* Q-feedback PID gains of Control Mode 7 can be changed by *Tracking Controller Gain Parameters telecommand (TC ID 54 / TLM ID 221)).*

The current control mode is changed by *Set Attitude Control Mode* telecommand (TC ID 13).

(The override flag of the latter telecommand is not currently implemented, and is reserved for future use).

A third parameter allows all the control modes involving the magnetorquers to be activated with a time-out. This is meant to be used during the commissioning phase of the control system, e.g. when the magnetorquer polarities are uncertain. Specifying a time-out period other than 0 or 0xFFFF (65535) seconds will let control proceed until the timer has expired. This is useful to verify that all configuration and mounting angles are set correctly, and that the control system is behaving nominally. During commissioning it is advised to always start with a timeout on the magnetic control modes.

The various control modes with the actuators used and required estimated states are listed in Table 3 below.

Table 3 – Control mode actuators and estimation requirements.

Enum value	Control mode	Actuators used	Required estimated state
0	No control	None	None
1	Detumbling control	Y-Magnetorquer	None
2	Y-Thomson spin	XYZ magnetorquers	Y-body rate
3	Y-Momentum wheel (Initial pitch acquisition)	Y-Momentum wheel & XYZ magnetorquers	X-,Y-,Z- body rates & pitch angle
4	Y-Momentum wheel (Steady state)	Y-Momentum wheel & XYZ magnetorquers	X-,Y-,Z- body rates & pitch angle
5	XYZ wheel control	XYZ reaction wheels & XYZ magnetorquers	X-,Y-,Z- body rates & Roll, pitch and yaw attitude
6	R-wheel Sun tracking control	XYZ reaction wheels & XYZ magnetorquers	X-,Y-,Z- body rates & Roll, pitch and yaw attitude

7	R-wheel target tracking control	XYZ reaction wheels & XYZ magnetorquers	X-,Y-,Z- body rates & Roll, pitch and yaw attitude
8	Very Fast-spin Detumbling control	XYZ magnetorquers	None
9	Fast-spin Detumbling control	XYZ magnetorquers	None
10	User Specific Control Mode 1	<reserved>	<reserved>
11	User Specific Control Mode 2	<reserved>	<reserved>
12	Stop XYZ Reaction wheels	XYZ reaction wheels	None
13	User-coded Control Mode	<reserved>	<reserved>
14	Sun-tracking yaw- or roll-only wheel control mode	XYZ reaction wheels & XYZ magnetorquers	X-,Y-,Z- body rates & Roll, pitch and yaw attitude
15	Target-tracking yaw-only wheel control Mode	XYZ reaction wheels & XYZ magnetorquers	X-,Y-,Z- body rates & Roll, pitch and yaw attitude

4.4.4 Valid estimation and control transitions

Table 4 displays the combinations of estimation and control modes which are allowed.

Table 4 – Valid estimation and control combinations.

		Control Mode				
		None	**Detumbling & fast modes**	**Y-Thomson**	**Y-Momentum**	**XYZ wheel (all modes)**
Estimation mode	**None**	OK	OK	X	X	X
	MEMS rate	OK	OK	OK	X	X
	Magnetometer rate	OK	OK	OK	X	X
	Magnetometer rate + pitch	OK	OK	OK	OK	X
	TRIAD	OK	OK	OK	OK	X
	Full-state EKF	OK	OK	OK	OK	OK
	MEMS Gyro EKF	OK	OK	OK	OK	OK

An attempt to switch to an estimation mode or control mode (by sending the relevant commands) will fail if it will result in an invalid combination from the table above. This behavior cannot be overridden.

The green OK entries are the preferred combination, and reflect nominal modes for optimal performance. But orange OK entries are also acceptable – i.e. the ADCS will allow such a combination, but then it should typically be used in consultation with CubeSpace, or during commissioning.

Further, for the control transitions (Table 4) you can go directly from No control to XYZ wheel control, provided the angular rates are low enough. But this is a "Red OK" because a more elegant way to get there is by doing a Y-spin first, followed by a Y-momentum control, and then 3-axis XYZ wheel control.In addition, when switching control mode, the following checks, as found in Table 5, are performed based on the current control mode and state.

Table 5 – Valid control transitions.

Current Control Mode	New Control Mode					
	None	**Detumbling & fast modes**	**Y-Thomson**	**Y-Momentum initial**	**Y-momentum steady-state**	**XYZ wheel (all modes)**
None	N/A	OK	OK	X	OK	OK
Detumbling & fast modes	OK	N/A	OK	X	X	OK
Y-Thomson	OK	OK	N/A	OK	X	X
Y-momentum initial	OK	OK	OK	N/A	OK	X
Y-momentum steady-state	OK	OK	OK	X	N/A	OK
XYZ wheel (all modes)	OK	OK	OK	X	X	N/A

Control mode transitions will not be permitted if a suitable estimator is not active to generate the required attitude and angular rate estimates. Transitions will also not be permitted if the required ADCS subsystems (e.g. CubeControl Signal, CubeControl Motor and CubeWheel) power switches are not enabled.

As with the previous table, orange OK entries in Table 5 reflect control mode transitions that are possible, but not ideal.

5 Usage

5.1 Power management

The ACP controls power to the nodes (including the sensors and actuators they are connected to) using several enable lines, connected directly to the CubeComputer MCU. Refer to **Ref 1** and **Ref 2** for detailed PC104 pin locations for these enable lines. The enabled state of CubeSense, CubeWheel(s), CubeStar and the CubeControl nodes are changed via the ACP - by sending a *ADCS Power Control* telecommand *(TC ID 11 / TLM ID 197)* to the ACP (see **Ref 4**).

The latter telecommand also allows the GPS LNA (low noise amplifier supply to the GPS antenna) to be switched on or off. In this case, the GPS LNA is switched on by the CubeControl Signal MCU only when the ACP sends it a command over I2C.

The *Power Selection* parameter for each component can be either 0 - off; 1 - on; 2 - no change.

5.2 Bootloader

5.2.1 Startup sequence

The bootloader is programmed at internal MCU flash address 0, and it will start to execute when the CubeComputer is powered up. The bootloader will run for 5s before it will attempt to boot the application program.

If, during the 5s interval, before the bootloader attempts to start the application program, an *Identification* telemetry request is made, the bootloader will remain active and will not attempt to start an application program automatically. An application can thereafter be started through an explicit telecommand to the bootloader.

The bootloader will also remain active if the current *Boot Index* is set to 0, or if there have been three or more unsuccessful attempts to start the application program.

The *Identification* telemetry frame can be used to identify whether the bootloader or the CubeACP program is currently running. The *Node Type Identifier* is a numeric value that is unique to the specific application. The CubeADCS firmware and command and telemetry interface is described in detail in **Ref 4**.

5.2.2 Recovery on reset

If an MCU reset or power cycle of the ADCS occurs while controlling the attitude of the satellite, four scenarios, found in Table 6, may occur.

Table 6 – Boot reset scenarios.

Condition at reset/power cycle	Condition at ACP start-up (10 – 20s later)
Satellite is tumbling randomly. Detumbling controller is active (or not)	Satellite is still tumbling randomly
Satellite is in steady-state Y-Thomson spin – Detumbling controller is active	Satellite will still be in Y-Thomson spin
Satellite is in Y-Momentum control mode – stable attitude	Wheel spinning down, satellite spinning about Y-axis (Y-Thomson spin)
Satellite is in 3-Axis pointing mode	Wheels would have had low bias, so satellite will be in slow random spin

The first scenario should only occur during the commissioning of the satellite. In this case, automatic recovery will most likely not be preferred.

In the second scenario, the satellite remains in a Y-Thomson spin after the reset event. The satellite will remain in this state without active control, as long as there are no significant external disturbances. For a typical start-up time of 10s, there will be no change in the spinning motion of the satellite.

If a reset occurs while the satellite is in the Y-momentum stabilized mode, the wheel speed will ramp down because of the friction of the wheel bearings. Angular momentum will be transferred back to the satellite and the satellite will transition into a Y-Thomson spin. The wheel will spin from 2000 rpm to 0 rpm in about 30s, so it is likely that the wheel will still be spinning slowly when the ACP comes online.

If the satellite is in 3-Axis reaction wheel control mode, it will revert to a slow random tumble motion, depending on wheel bias at the reset.

5.3 Firmware management

It is possible to store up to 7 application programs on the external flash memory of the CubeComputer. It is however only possible to have a single "active" application stored in the internal flash memory, from where the application will execute. It is thus necessary to copy the chosen application from external memory to internal flash memory.

The CubeSpace bootloader, application interaction, and firmware maintenance (including firmware reprogramming) is described in detail in **Ref 4**.

5.4 System Configuration

The ADCS makes use of two groups of configuration settings. The first such group is the System Configuration. This includes the following settings:

- Versions of components that are assembled into the bundle
- Definitions of how the ACP should behave (3-Axis reaction wheel, or Y-momentum bundle)
- Support of optional components
- Enable pin configuration
- Magnetometer ground calibration values

The System Configuration is programmed by CubeSpace prior to shipping the bundle. It is possible to reprogram the System Configuration through CubeSupport, but this should be done in consultation with CubeSpace, since setting incorrect settings for the System Configuration will result in malfunction.

5.5 ADCS Configuration

The second group of configuration settings that influences the ADCS operation, is the user-adjustable ADCS Configuration. These settings include

- Mounting angles and orientations for all the sensors and angles
- Calibration values for CubeSense and CubeStar
- MEMS rate sensor offsets
- Magnetometer in-flight calibration delta settings
- Satellite moment of inertia matrix
- Estimator parameters
- Controller gains and parameters

The following sections will explain the various configuration settings in more detail.

The ADCS Configuration can be adjusted by sending a command (TC ID 20) and can also be read back using a telemetry request (TLM ID 206). It is also possible to make adjustments to smaller subsets of the total configuration, by using appropriate commands or telemetry requests as listed in Table 7.

Table 7 – ADCS Configuration, and subsets of settings.

Settings group	TC ID	TLM ID
Magnetorquer Configuration Axis Configuration (1-3)	21	136
Wheel Configuration Axis Configuration (1-4)	22	137
Rate Gyro Configuration Axis Configuration (1-3) Rate sensor offsets (1-3) Measurement Multiplier	23	138
CSS Configuration Axis Configuration (1-10) Scaling (1-10)	24	139
CubeSense Configuration Camera1 mounting & configuration Camera2 mounting & configuration Earth mask (1-5) Nadir detection parameters	25	203
Magnetometer Configuration Mounting In-flight calibration (Sensitivity matrix and offset vector)	26	204
Redundant Magnetometer Configuration Mounting In-flight calibration (Sensitivity matrix and offset vector)	36	205
Star Tracker Configuration Mounting Calibration model parameters Sensor configurations	37	202
Detumbling Control Parameters Spin and damping controller gains Reference spin rate	38	208
Y-Wheel Control Parameters Momentum management controller gain Wheel control gains Reference wheel momentum	39	209
Reaction Wheel Control Parameters Wheel control gains Y-Wheel bias momentum Facet selection for sun-tracking	40	217
Tracking Controller Gain Parameters Wheel control gains	54	221

Facet selection for target tracking		
Moment of Inertia Matrix Moments of inertia Products of inertia	41	222
Estimation Parameters EKF and Rate-only Kalman Filter system noise Sensor noise covariances Flags indicating which sensors should be used in EKF	27	223
Augmented SGP4 Parameters **<reserved>**	28	227
User-coded Estimator and Control Parameters **<reserved>**	29	226

Table 8 – Orbit Parameters

Settings group	TC ID	TLM ID
Orbit Parameters SGP4 Orbit Parameters	45	207

If any of these configuration settings have been changed, the configuration can be saved to flash permantly using TC ID 63 (Save Config) and 64 (Save Orbit Parameters). If the configuration is not saved, it will be lost after reset of the ADCS OBC.

The CubeSupport software will display tabs with the above configuration settings, displayed as “ADCS Config”, and “Orbit”. The configuration tab also contains a configuration viewer where the components orientation relative to the satellite body is shown, as set by the configuration. This feature can be used to verify mounting settings. Please see **Ref 5** for more information.

5.5.1 Configuration XML files

CubeSupport also has the ability to load configuration settings from an XML file, and then program it onto the CubeACP, also from the *ADCS Config* tab.. CubeSpace can generate an appropriate ADCS Configuration XML file for users when they provide the necessary information about how the ADCS will be used inside their satellite. Clients are encouraged to involve CubeSpace with their configuration settings – to ensure optimal operation.

5.5.2 Mounting transformations

The sensors and actuators in the CubeADCS are either classed as **single** or **multiple axes**. **Single axis sensors or actuators** are sensors or actuators where the alignment of only one axis is relevant, such as a wheel or single-axis rate sensor. In this case the axis about which rotation or sensing takes place is important, but the mounting angle about the rotation or sensing axis is irrelevant.

Multiple axes sensors or actuators are sensors or actuators for which the alignment of all three principle axes are relevant.

The mounting of the sensors and actuators of the ADCS must be defined by:

- **For single axis sensors or actuators:** specifying the principle SBC axis to which the sensor or actuator is aligned.
- **For multiple axes sensors:** specifying mounting transform angles to rotate the sensor or actuator coordinate system to match the SBC.

Single-axis sensors or actuators include the following:

- Coarse Sun sensors (TC 24)
- MEMS rate sensors (TC 23)
- Torquer rods (TC 21)
- Reaction/Momentum wheels (TC 22)

Multi-axis sensors include the following:

- Deployable or redundant magnetometer (TC 26 and TC 36)
- Sun sensor and nadir sensor (CubeSense) (TC 25)
- Star tracker (CubeStar) (TC 37)

A complete guide to identifying these mounting transforms (with examples) is given in ***Ref 6***. Review this document carefully, complete it, and send it back to CubeSpace. These transforms are critical to the correct functioning of the ADCS and require careful review.

5.5.3 Sensor calibration settings

The raw sensor measurements are put through calibration functions before they are used by the ADCS estimation and control algorithms. Each sensor or actuator has its own unique calibration settings, which is explained in the sections below.

5.5.3.1 Magnetometer

The magnetometer pre-calibration (ground calibrated) equation is:

$$\mathbf{B}_{precalib} = \mathbf{S}_{pre}(T)\mathbf{B}_{raw} - \mathbf{d}_{pre}(T)$$

Where:

$\boldsymbol{B}_{precalib}$ is the pre-calibrated measured magnetic field in the body coordinate frame;

$\boldsymbol{B}_{raw} = \left[B_{rawx}\ B_{rawy}\ B_{rawz}\right]^T$ is the raw measurement returned by CubeControl;

$\mathbf{S}_{pre}(T)$ is the temperature dependent sensitivity matrix; and
$\mathbf{d}_{pre}(T)$ is the temperature dependent offset vector.

The magnetometer post-calibration (in-orbit calibrated) equation is:

$$\mathbf{B}_{calib} = \mathbf{S}_{post}\mathbf{M}_{mounting}\,\mathbf{B}_{precalib} - \mathbf{d}_{post}$$

Where:
$\boldsymbol{M}_{mounting}$ is the mounting transformation DCM set as in Section 5.5.2;
$\mathbf{S}_{post}$ is the in-orbit calibrated sensitivity matrix; and
$\mathbf{d}_{post}$ is the in-orbit calibrated offset vector.

The mounting angles will also have to be updated after deploying the magnetometer boom, since the orientation of the magnetometer is now different.

The diagonal components of the **S** matrix are scaling factors for each channel, while the off-diagonal components account for non-orthogonal measurements axes.

The temperature dependent pre-calibration sensitivity matrix $\mathbf{S}_{pre}(T)$ and offsets $\mathbf{d}_{pre}(T)$ are **measured during integration for each magnetometer** and will differ for different sets of hardware. The magnetometer is placed in a Helmholtz cage with a high accuracy flux-gate magnetometer. The sensitivity and offsets are determined, and the magnetometer is also temperature calibrated. Measured values are recorded in the build history for each ADCS module.

Ideally the magnetometer would measure the undistorted ideal magnetic field of the Earth once in space. Once the magnetometer is mounted on the satellite, the magnetic field is however distorted by certain parts of the satellite - like ferrous materials and wires carrying large currents. The magnetic field measured by the magnetometer is therefore not ideal and **recalibration of the magnetometer might be necessary in orbit**. **Please consult Ref 7 for more information on this procedure.**

It is important to note that the ACP performs a check on the calibrated measurement. The post-calibrated magnetic field vector $\mathbf{B}_{calib}$ magnitude must be between 15,000 and 65,000 nT otherwise the measurement is not used by the estimator. In addition, the ACP sets the *Magnetometer Range Error* flag in the *ADCS State* telemetry if the measurement is out of range (see **Ref 4**).

5.5.3.2 Redundant Magnetometer

The redundant magnetometer has the same settings as the main magnetometer. In addition, the following configuration settings are used to determine functionality and modes for the two magnetometers.

The *Magnetometer Mode* setting in the Estimation Parameters group of settings will determine which magnetometer should be used for estimation purposes, and the sampling path to use, since it is possible to sample the main magnetometer either through the CubeControl signal microcontroller (the preferred setting), or through the CubeControl motor microcontroller.

It is possible to also sample the redundant magnetometer while using the main magnetometer for estimation (or vice versa). In this case, the other magnetometer that is only sampled, has measurements returned in the *Secondary Magnetometer Raw Measurements* telemetry frame (TLM ID 215). The *Magnetometer Selection for RAW MTM TLM* setting, also in the Estimation Parameters group, will specify what to return in the secondary RAW Measurements telemetry.

If the *Automatic Magnetometer Recovery* flag in the Estimation Parameters message is set, the ADCS will automatically attempt to switch to the redundant magnetometer in case a magnetometer failure is detected.

5.5.3.3 Coarse Sun sensor

The mounting of the coarse Sun sensors is described in Section 5.5.2.

The CSS photodiodes have bias values even when they are not illuminated. These values are not significant since the CSS only provide a relatively coarse measurement. To reduce the effect of these offsets, and to make the sensors more robust against measurement of reflections off the moon for instance, the CSS measurements are only seen as 'valid' and is only used if the measured value is above a certain threshold, s_{min}. Owing to the varying CSS sensors' sensitivity, the CSS outputs can be scaled. For each of the ten CSSs the configurations settings are then:

- Axis definition
- Scaling
- Minimum threshold

The default scaling and offset configuration can be used for any satellite and the mounting settings must be configured by the user.

5.5.3.4 CubeSense (sun or nadir) sensor

Each CubeSense unit has the following configuration settings:

- Mounting – As described in Section 5.5.2

- Detection threshold
- Auto-adjust exposure flag
- Manual exposure time
- Boresight X & Y

Each of the above are discussed below.

Mounting

The CubeSense unit(s) should be mounted by the user. The camera's axes are defined as illustrated in Figure 7.

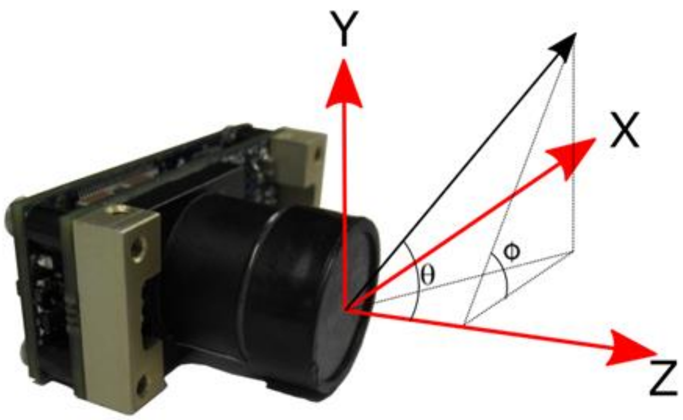


Figure 7 – Camera axes definitions.

The procedure discussed in Section 5.5.2 must be followed to configure the CubeSense mounting angles.

Detection threshold

CubeSense is a visible light image-based sensor. Part of the image processing that is done inside the module is applying a threshold to distinguish between background pixels and pixels that form part of an object – the Sun, Earth and the moon. This threshold must be chosen correctly for a given exposure value. The default threshold values are correct for the default exposure values. If either is changed, the CubeSpace team should first be consulted.

Auto-adjust exposure flag

CubeSense can run an auto-adjust algorithm to automatically adjust the exposure based on the lighting conditions it is experiencing. By default, the sensors run with the auto-adjust off. This should not be changed except if it is for a justified reason that has been approved by CubeSpace.

Manual exposure time

If auto-exposure is switched off the camera exposure is determined by the manual exposure value that is set in the configuration. This value is between 0-1000. Where 0 is the shortest exposure (darkest photo) and 1000 is the longest exposure (lightest photo). These values are correct by default and should only be changed after consulting CubeSpace.

Boresight X & Y

Each CubeSense unit is accurately calibrated during manufacturing. Among other parameters, the calibration provides the boresight of the optics as X and Y coordinates on the image plane.

This boresight value is unique to each camera. Boresights MUST be updated whenever these camera modules are swopped for others.

Nadir detection parameters

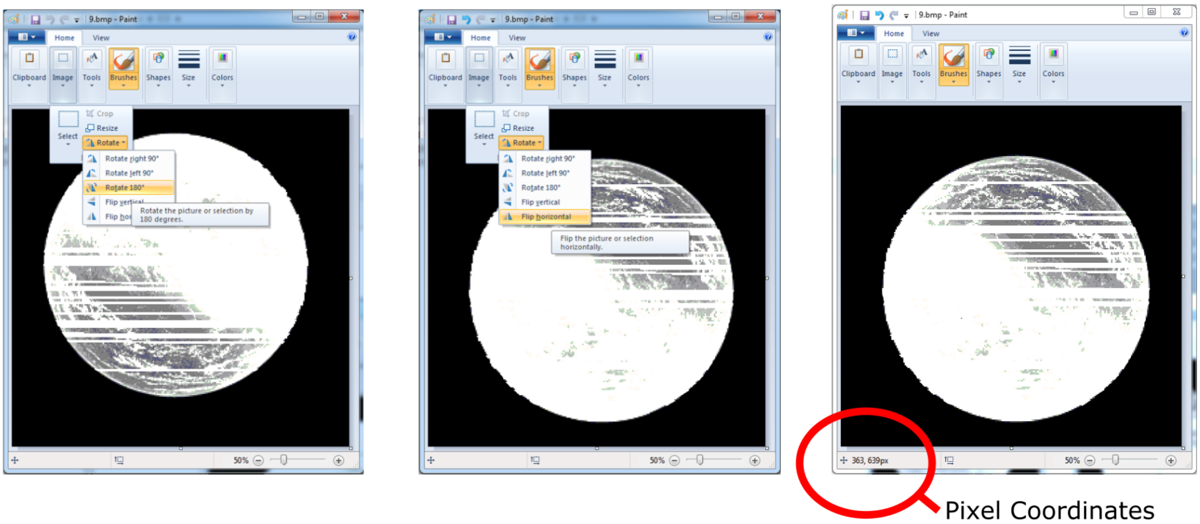
There are four parameters in the *CubeSense Configuration* message, that influence the nadir detection algorithm – the *Max Deviation Percentage, Max Bad Edges, Max Radius* and *Min Radius* parameters.

The Nadir sensor algorithm attempts to fit a circle to the detected edge points. The *Min Radius* and *Max Radius* parameters can be used to specify the valid range of the circle radius. This will depend on the the height of the satellite in its orbit. The *Max Deviation Percentage* is the tolerance that are allowed for all detected points to lie on or close to the fitted circle. If there are more than *Max Bad Edges* such points, a detection error is reported.

Sensor masking

The CubeSense sensor may in some configurations have antennas or other deployable structures in its field of view. For these cases, a mask on these images is required to avoid false detections. To mask an object, the coordinates of a square can be set in the *CubeSense Configuration* message. Every pixel that falls within this square will be ignored by the detection algorithms. To mask more complex objects in the FOV, multiple squares (up to 5) can be defined. To find the coordinates that should be entered, a full resolution bitmap image should be downloaded. The procedure below can be done in any capable image editing software, but Microsoft Paint will be used as example.

Open the image in *Microsoft Paint*, rotate the image 180°, flip the image horizontally and then hover the mouse at the top-left of the object. This is the location of the minimum X and Y pixel coordinate. Hover the mouse at the bottom-right corner to find the maximum X and Y pixel coordinate. These steps are shown in the figure below.



Sampling and errors

The ACP will reject a Sun or nadir measurement if the sensor reports that it is still busy performing the detection or if the detection result is anything other than successful. If the sensor is busy at the time the Sun or nadir telemetry is requested, the ACP will set the *Sun Sensor Busy Error* or *Nadir Sensor Busy Error* flags in the ADCS State telemetry frame. **This should not occur under normal operating conditions**.

If the sensor reports a detection failure, the *Sun Detection Error* or *Nadir Detection Error* flags in the *ADCS State* telemetry frame will be set. These flags will also be set if the measured Sun azimuth and elevation angles are larger than 80 deg (or smaller than -80 deg), and the measurement will not be used for estimation. For the nadir sensor, the valid azimuth and elevation range is limited to -70 to +70 deg. These flags will regularly be set during nominal operations since the Sun will often not be in range, or the Earth will only be partially visible. **The user does not have to manage these flags. The ACP will automatically use the measurements only when they are valid. When a valid measurement is received, these flags are automatically cleared by the ACP.**

In some situations, such as commissioning, it will be required to sample raw Sun or nadir measurements without using the calibrated measurements in the EKF estimator.

The *Use Sun Sensor* and *Use Nadir Sensor* flags in the Estimation Parameters will determine if the calibrated measurement vectors will be used in the EKF. The sampling of the CubeSense measurements will also be skipped if the ADCS computes that it is currently in eclipse, based on current TLEs and unix time.

The *Nadir sensor terminator test* flag in the Estimation Parameters message will further determine if the Nadir sensor should be sampled if a terminator (line between shadow and sunlit part of the Earth) is present in the field-of-view.

5.5.3.5 Rate sensor

Axis definition of the rate sensors is done as described in Section 5.5.2. The *RateSensorMult* parameter is a multiplier that scales the raw rate sensor measurements. This multiplier can either be *x1* (±75 deg/s), *x2* (±150 deg/s), *x4* (±300 deg/s) or *x12* (±900 deg/s). The reason for this multiplication is to make the rate sensor range adjustable. The rate sensor TLM measurements are formatted as a 16-bit integer. If a satellite is spinning at very high rates, the *x12* multiplier is implemented giving the rate sensor a ±327. 68 deg/s (±32k centi-deg/s) measurement range and a centi-deg/s TLM resolution, given the 16-bit variable size. When the satellite is spinning at low rates, it is desirable to have milli-deg/s resolution for more accurate measurement. In this case the *x1* default multiplier is implemented which decreases the range to ±32.768 deg/s (±32k milli-deg/s) that can fit in the 16-bits, but providing a milli-deg/s TLM resolution.

The multiplier is set to *xN* (*N* = 1, 2, 4, 12), when the *RateSensorMult* parameter in *Rate Gyro Configuration* TC ID 23, is set to *N*. By default it sets to *x1* (*N* = 1). In extreme cases where a satellite is tumbling at rates higher than ±32.768 deg/s, the multiplier can be set to *N* = 2, 4, or 12.

Further, the rate sensors are calibrated by subtracting a factor to compensate for temperature variation, and subtracting a permanent offset value. The temperature calibration is hardcoded into the ADCS and the only configuration parameter that should be set for the rate sensors are the permanent offsets set by the *Rate Gyro Configuration* TC ID 23.

5.5.3.6 CubeStar

The CubeStar configuration parameters are listed in Table 9. All CubeStar units are pre-calibrated and the default configuration parameters are set to optimal values. Most of these parameters can be altered using TCMD 20 or TCMD 37. However, under nominal circumstances, altering should not be necessary. Please contact CubeSpace before altering any of these values. The configuration parameters can be read using TLM 202.

Table 9 – CubeStar configuration parameters.

Parameter	Default Value
Exposure Time	2704
Detection Threshold	10
Star Threshold	5
Maximum Stars Matched	15
TimeoutTime	500
Maximum Star Pixels	160
Minimum Star Pixels	4
Error Margin	85
Centroid X	640 ± 40

Centroid Y	510 ± 30
Focal Length	6.2 ± 0.2
K1 distortion coefficient	-
K2 distortion coefficient	-
P1 distortion coefficient	-
P2 distortion coefficient	-
Sync Delay	500
Tracking Window Width	23
Tracking Margin	10
Validation Margin	50
Tracking Module Enable	0
Tracking Prediction Enable	0
Tracking Search Width	210

5.5.3.6.1 Exposure

The exposure parameter is a 16-bit value. This value translates to an exposure time which can be calculated using the formula:

$$Exposure_{ms} = Exposure_{16\ bit\ paramter} \times 0.1795 + 14.565ms$$

The default exposure value of 2704 translates to an exposure value of 499.93 milliseconds. Increasing the exposure will cause the overall processing time and noise of an image to increase. An increase in exposure time may also prevent CubeStar from providing new measurements every second. This value should not be changed without consulting CubeSpace.

5.5.3.6.2 Detection and star threshold

The star detection algorithm searches though the image by evaluating every second pixel against the *Detection Threshold* value. If the pixel value is greater than the threshold value it indicates that this pixel may be part of a star. Once such a pixel is found, the algorithm searches the neighbouring pixels to find all the pixels with values more than or equal to the *Star Threshold*.

Increasing the *Detection Threshold* will cause dimmer stars not to be detected.

Decreasing the *Detection Threshold* value will cause more noise to be found and will increase the time the detection algorithm takes to execute. It may also cause CubeStar to find dimmer stars not detected previously. There are limits on the number of stars and how much noise can be found in an image, therefore decreasing the detection threshold will cause the algorithm to reach the noise and stars detection limits earlier in its search, which will affect performance.

Increasing the *Star Threshold* will result in fewer pixels being detected as part of a star. This will decrease the execution time of the detection algorithm, but may cause dimmer stars to be seen as noise.

Decreasing the *Star Threshold* will result in more pixels being detected as part of a star. This will increase the execution time of the detection algorithm. It may also cause more than the maximum number of pixels to be detected for a star. This will cause larger stars to be seen as invalid stars.

Once these threshold values have been set, it takes immediate effect. The *star pixel threshold* must always be less than the *detection threshold*.

5.5.3.6.3 Max and min pixels per star

Once all the pixels above or equal to the *Star Threshold* have been found, it is necessary to determine whether these pixels form a star. This is done by evaluating how many pixels were found, against the minimum and maximum pixels a star can have. If the number of pixels is greater or equal to the minimum number of pixels and less or equal to the maximum number of pixels they form a valid star. CubeStar counts the number of valid stars that is found as it searches through the image.

If the number of pixels found is less than the minimum number, it is considered as noise detected. CubeStar counts how many of these invalid detections are found as it searches through the image. The pixels or small grouping of pixels that fall below the minimum number of pixels for a valid star, is referred to as noise in CubeStar.

If the number of pixels per star is found to be more than the maximum number of pixels allowed, the pixels are considered to be part of a bright object in the field of view, e.g. The Moon. CubeStar counts how many of these invalid objects are detected as it searches though the image. These clusters of pixels above the threshold values that exceed the maximum number of pixels for a star are referred to as invalid stars.

The largest value for the maximum number of pixels that can be set is 99. If an attempt is made to set the value higher than 99, an error flag will be set.

Decreasing the minimum pixels to three or less pixels will cause noise to be considered as stars. Increasing the minimum pixels will cause small stars to be rejected as noise.

Decreasing the maximum pixels might cause larger or brighter stars to be seen as invalid stars.

The minimum and maximum pixel values also depend on the detection threshold value. A high detection threshold will cause less pixels to be detected, while a lower threshold value will cause more pixels to be detected.

5.5.3.6.4 Max stars

As stated earlier CubeStar keeps track of detection execution time, as well as the number of invalid and valid stars that have been detected, while it searches through the image. The detection algorithm needs to ensure that the execution time is not too long, and that the image being processed is valid. The default limits are shown in Table 10:

Table 10 – Detection limits.

Parameter	Default limit
TImout Time	500
Invalid stars	5
Valid stars	15

If any of these limits are reached during the star search, the star search is ended and an error flag is set. This does not necessarily mean that the image could not be processed. For example, in the case where the maximum number of stars is detected, the search is stopped and the max stars detected error flag is set.

Once the detection is complete, the identification algorithm will execute, as long as three or more stars were detected.

5.5.3.6.5 Error margin

CubeStar uses an error margin when calculating the distance between stars. This error margin is a percentage of the distance. The default value is 0.85 % or as a gain value: 0.0085. This telecommand allows this margin to be changed and accepts an 8-bit parameter.

This 8-bit parameter is the gain value multiplied by 10 000.

$$parameter = gain \times 10\,000$$

For the default gain of 0.0085 the parameter will have the value of 85.

When CubeStar receives the parameter it is divided by 10 000 to obtain the correct gain to be used in the calculations.

Owing to the conversion and the byte size of the parameter, the gain value can only be between 0% and 2.55% (gain value: 0 to 0.0255).

5.5.3.6.6 Centroid

The *Centroid* is the position on the image plane in line with the point on the lens with no distortion. This is close to the center of the lens and is determined during calibration.

5.5.3.6.7 Distortion coefficient

The *Distortion coefficients* (K1, K2, P1, P2) represent the distortion model of the lens. These values are precalibrated.

5.5.3.6.8 Focal length

The *Focal Length* is a value in millimetres. This value is determined during calibration and should be close to 6.2 mm. An error flag will be set if the focal length is not set between 5 and 7 mm.

5.5.3.6.9 Delay Time

The Delay Time setting is used internally to control synchronisation of the ACP with the CubeStar capture and detection loop. The one second update rate is achieved by overlapping the capture of a new image with the processing of the previous image. After the ACP issues a trigger command to the CubeStar, it will start a delay timer. When the timer reaches the timeout value it triggers a new image capture.

The Delay Time in milliseconds must be a value between 0 to 0.99 s. This implies that the parameter value will range between 0 and 1000.

5.5.3.6.10 Tracking Parameters

CubeStar comes preprogrammed with an experimental tracking mode that can be enabled through the *Tracking Module Enable* and *Location Prediction Enable* configuration parameters.

Setting the *Module Enable* bit to a '1' enables the advanced tracking mode and, by default, the rate estimation module. When the tracking module is enabled, a region-limited search scheme is used to allow for increased algorithm performance and faster star matching. Setting this bit to a '0' disables both the rate estimation module and the advanced tracking module.

Setting the *Location Prediction* bit to a '1' enables feedback from the rate estimation module to the advanced tracking module, allowing for more accurate star location prediction during higher slew rates.

The *Window Width* parameter represents the size of the search window used when attempting to detect stars during tracking mode. By increasing this value, the region of interest around each estimated star position is increased.

The *Tracking Margin* specifies how strict the matching algorithm used during initial matching is. This parameter signifies the maximum error between an identified vector pair, and a detected star pair.

The *Validation Margin* value specifies the percentage of the total stars that have to validate the identities of any star for a match to be made. A register value of '70' would therefore lead to validation being much stricter than when using a value of '50'.

Both the *Tracking Margin* and *Validation Margin* use a scaling factor of 100. To set the gain value to a 0.1 degree margin, a parameter value of 10 is required. The required parameter value can therefore be determined with the following formula:

$$parameter = gain \times 100$$

The *Search Width* parameter specifies the field of interest (FOI) used during the experimental tracking mode. The FOI influences the furthest star that would be searched for during the new star identification stage. Setting this parameter to a value of 255 sets the field of interest to 51 degrees. The star furthest away from the boresight that can still be identified would therefore be half the FOI, or, in this case, 25.5 degrees away.

$$paramter = gain\ \times 5$$

5.5.4 Actuator configuration settings

5.5.4.1 Magnetorquers

The only configuration for the magnetorquers is the mounting, as described in Section 5.5.2. The *Axis Selection* enumeration (see **Ref 4**), for each magnetorquer rod, is used to map the control command to torquer output.

5.5.4.2 Reaction wheels

The axes of each of the reaction wheels are set using the Wheel Configuration command (see **Ref 4**). Figure 8 displays the polarity of the reaction wheel.

Figure 8 – Reaction wheel polarity.

5.5.5 ADCS controller configuration

The ADCS controller configuration consists of settings for detumbling, Y-wheel, and reaction wheel controller.

5.5.5.1 Detumbling controller settings

The detumbling controller setting contains three gain values, *Ks* ,*Kd* and *Kdf*. These values are specified by CubeSpace at delivery and should not be changed unless consulting CubeSpace.

Further, the reference spin rate for the detumbling controller can be specified. The default value for the spin rate is -2 deg/s. This can be adjusted in specific cases, however CubeSpace should be consulted before changing this value.

5.5.5.2 Y-Wheel controller settings

The Y-Wheel controller settings consist of four gain values *Kh*, *Kn*, *Kp1*, and *Kd1*. These gains are set at delivery of each ADCS unit and should only be changed after consultation with CubeSpace.

Further, the reference wheel momentum for the Y-Wheel controller can be specified. The default value for the wheel momentum depends on the satellite's Y-axis MOI and will be specified by CubeSpace at delivery. This can be adjusted for specific cases, however CubeSpace should be consulted before changing this value.

5.5.5.3 Reaction wheel controller settings

The 3-axis reaction wheel controller setting consists of two gains, *Kp2* and *Kd2*, having values which are specified by CubeSpace at delivery. The values should not be changed, unless CubeSpace is consulted. . It is also possible to specify which facet of the satellite should be angled towards the sun when using the *Sun-tracking yaw- or roll-only wheel control Mode*. Valid options are PosY, NegY or NegZ.

It is possible to set a bias wheel momentum to use for the Y-axis reaction wheel in three-axis wheel control mode, by specifying the *Y-Wheel Bias Momentum* parameter. Use this setting in consultation with CubeSpace.

Setting the *Automatic Control Transition due to Wheel Errors* flag will cause the ADCS to automatically transition to the Y-Spin control mode in case a failure of one of the reaction wheels is detected. A reaction wheel failure is registered when the measured wheel speed differs too far from the commanded wheel speed.

5.5.5.4 Tracking mode controller settings

The wheel control tracking mode uses three gain parameters - *Kp3, Kd3* and *Ki3*. The values should not be changed, unless CubeSpace is consulted. It is also possible to specify which facet of the satellite should be angled towards the target when using the *Target-trackig yaw-only wheel control Mode*. Valid options include PosY or NegY. For the normal *target tracking control mode*, the PosZ axis will be pointed toward the target.

5.5.6 ADCS estimator configuration

The estimator setting consists of the EKF noise parameters, sensor masking, and a few other settings, some that were discussed under earlier sections (Terminator check for nadir sensor, main and redundant magnetometer behaviour).

5.5.6.1 Measurement noise

The estimator algorithms make use of filters that combine sensor measured vectors and modelled vectors. The measurement noise parameters in the ADCS configurations settings determine how heavy or light, in weight, the measured vectors are, relative to the modelled vectors. These values are correctly set up at delivery, and CubeSpace should be consulted if they are to be changed.

5.5.6.2 Sensor masking

The estimator algorithm will use whatever measurements are available on the satellite. All sensors that are switched on and enabled will be sampled by the ACP and will be available for use in the estimators. **In some cases, especially during commissioning, it is desirable to sample sensors but not to use it in the filter. In this case, the sensor can be masked. The sensor is included in the filter, if the corresponding parameter (*Use Star Tracker, Use CSS*, etc.) is set to 1 and excluded, if it is set to 0.**

5.5.6.3 Automatic Estimation Transition due to Rate Sensor Errors

The *Automatic Estimation Transition due to Rate Sensor Errors* flag will cause the ADCS to automatically switch to the magnetometer-only RKF in case a failure of one of the MEMS rate sensors is detected.

5.5.7 General satellite configuration

5.5.7.1 MOI and POI

The moments of inertia and the products of inertia for a specific satellite must be correctly set for the ADCS to function nominally**. If any deployable structure on the satellite is activated, the MOI and POI of the satellite will typically change. It is critically important that the MOI and POI in the ADCS configuration is updated when the deployables are activated**.

5.5.7.2 Orbit parameter configuration

The orbit SGP4 parameters can be set using a single command TC ID 45, and read back using TLM ID 207. The SGP4 parameters must be regularly updated to account for a changing orbit. This is especially important at LEO altitudes.

It is recommended that the TLEs be updated daily during commissioning, and at least once a week, thereafter. See Figure 9 below.

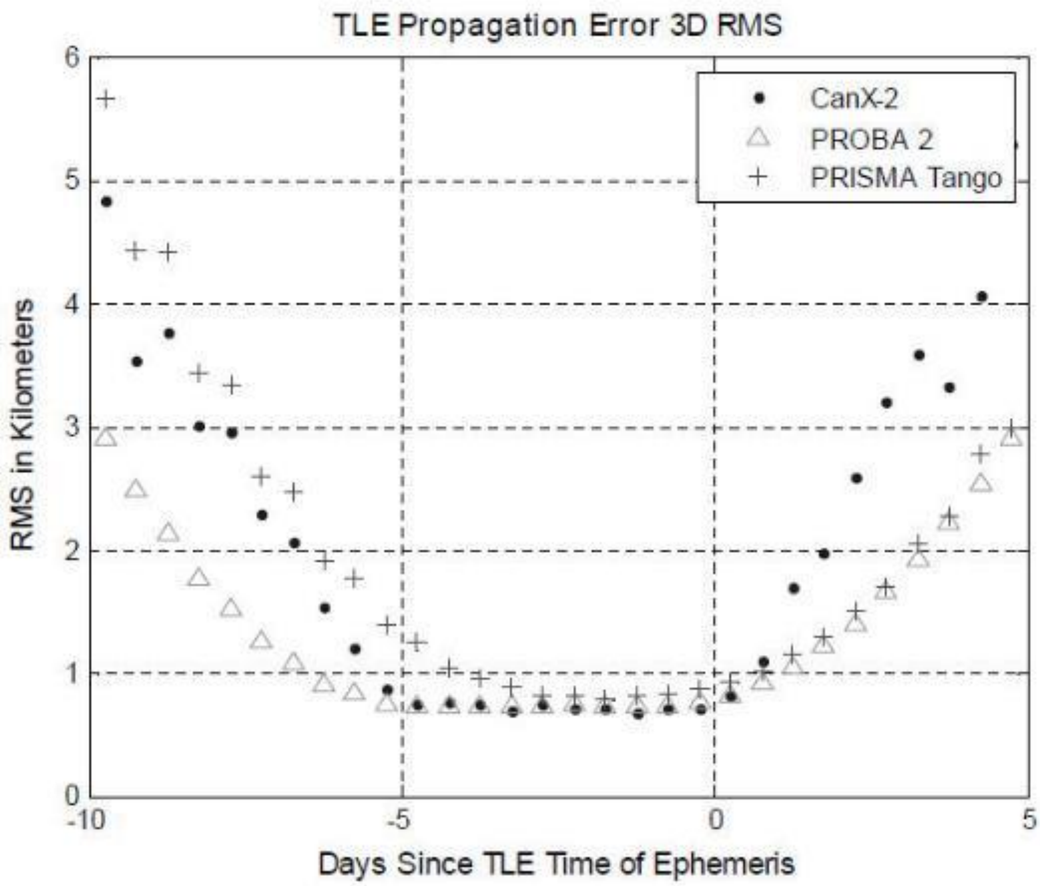


Figure 9 – Example of TLE Time since Epoch error for different satellites.

5.5.8 General considerations with configuration

Both the configuration block and SGP4 orbit parameters are stored in flash memory on the CubeComputer and persists between power cycles.

On start-up, the configuration in flash memory will be read and transferred to RAM. The configuration block in RAM is what will be used by the ACP for controller and estimation parameters, and sensor calibration. If the configuration block in flash memory is corrupt or has not been programmed yet (a CRC check is performed when doing the initial read) a flag in the *ADCS State* telemetry frame (see **Ref 4**) is set.

The current RAM configuration can be changed by sending the *Set Configuration* telecommand (see **Ref 4**) or smaller individual targeted telecommands as described in the previous sections. The current RAM configuration can be read via the *Configuration* telemetry request (see **Ref 4**).

The current RAM configuration is written to flash memory upon receiving a *Save Configuration* command (see **Ref 4**).

The process for the current SGP4 orbit parameters is the same. For command and telemetry formats, refer to the *Set SGP4 Orbit Parameters* telecommand and the *SGP4 Orbit Parameters* telemetry request.

5.6 Status and error flags

The *ADCS State* telemetry (see **Ref 4**) contains the status and error flags. Error flags are latched whenever the particular error occurs, and must be cleared by telecommand.

5.7 Logging

It is possible to configure logging through the CubeACP application (the CubeComputer bootloader does not support logging. Logging can be enabled for a variable selection of telemetry frames, with selectable period, up to 1s resolution.

It is possible to log telemetry to the on-board SD card or as "unsolicited" telemetry on the UART communications link. For logging to the single SD card, two independent logging selections can be configured. This allows for logging of different telemetry frames at different periods. This can be used, for instance, when most telemetry frames are only required at a slow interval (house-keeping telemetry) but certain telemetry such as magnetometer measurements, are needed at 1s interval.

The logging selections are configuration items – they can be set via command and read back using telemetry request. See **Ref 4** for SD Log1 Configuration, SD Log2 Configuration and UART Log Configuration message format.

If the on-board telemetry logging functionality is not being used, and telemetry logging is to be performed by an external master, it is important that the logging ensures synchronization of telemetry data.

Since the ACP loop executes at 1s periods, telemetry frames containing estimated attitude state and sensor measurements will be updated with a 1s period. For logging purposes, these frames should be requested from the ACP with integer second periods and preferably synchronized to the ADCS loop. The ideal sample time would be to request the frames after one control loop has executed, before the next iteration.

This can be achieved by using the *ACP Execution State* telemetry request (TLM ID 220). This telemetry frame can be polled until the *Current Execution Point* changes to *Idle*. At this point it will be safe to sample the required telemetry. Care must also be taken to ensure the required TLM sampling will complete before the next iteration starts.

5.8 Magnetometer boom deployment

Deployment of the magnetometer is required to limit the magnetic disturbances caused by the satellite bus, e.g. currents flowing in solar panels, wire loops and reaction wheel magnets. The Comissioning Manual (**Ref 7**) describes the suggested boom deployment procedure. Both CubeControl MCUs should be on when sending the *Deploy Magnetometer Boom* command. The *Deploy Magnetometer Boom* command format is given in the Reference Manual (**Ref 4**). In addition, the *ADCS Run Mode* should be set to *Enabled*. A deployment timeout of 2s is suggested. This timeout can be increased in case of non-deployment.

5.9 Image downloads

The CubeACP can download images from the CubeSense and CubeStar cameras onto the CubeComputer's SD card. For this functionality to be available, an SD card must be present in the CubeComputer SD card slot. Image downloads are initiated by the *Save Image* telecommand (see **Ref 4**). A parameter in the command selects which camera to download from (CubeSense1, CubeSense2, or CubeStar) and the size of the image to download to the SD card.

The time that the download process takes to complete is dependent on the size of the image that is selected. The status of the download can be polled by reading the *Status of Image Capture and Save Operation* telemetry frame (see **Ref 4**). While the download is taking place, the relevant sensor will not perform any detection. If the ADCS loop is running while an image is being downloaded, the relevant sensor will automatically be masked (i.e. not used in the ADCS algorithms).

Once the download is complete, the file can be downloaded from the ACP using the protocol described in **Ref 4.**

5.10 GPS Measurements

The CubeControl board has the capability to interface with a Novatel OEM719 GPS receiver module. When enabled, it is possible to read the GPS provided position, velocity and time measurements via TLM request. The GPS receiver measurements are not used or required by the ADCS. Note that the GPS measurements will have a latency of up to 2 seconds, since it is collected by an intermediate MCU on the CubeControl board before being made available in the CubeComputer telemetry frames.

The use of the GPS receiver requires a harness and specific components on the CubeControl board of the ADCS module to be populated. The harness and components are not populated on standard ADCS units. The mechanical standoffs for the GPS receiver are also not attached by default. Users of the ADCS should explicitly request that these components are populated.

Even when the GPS interfacing components are fitted to the ADCS, the GPS receiver itself is never supplied with the ADCS module due to export restrictions. Please contact Novatel or a local distributor directly for more information.

6 Special Considerations

6.1 FOV of sensors

Obstacles in the FOV of the optical sensors like CubeSense and CubeStar, may degrade the quality of the sensor significantly; may prevent the sensor from outputting measurements at all; or in the worst case, may output incorrect measurements that lead to control issues.

6.2 Magnetic loops and solar panels

Current flowing in solar panels can form a loop and cause a magnetic moment vector with Sun dependent direction - see Figure 10. This can cause a disturbance torque that leads to spin-up of the satellite body.

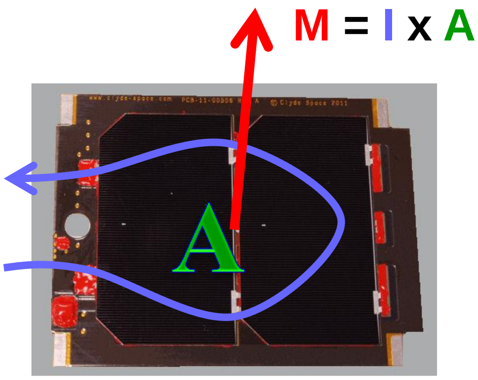


Figure 10 – Solar panel magnetic moment disturbance.

It is therefore advised to make use of back-wiring to cancel the disturbance magnetic moment, see the figure below.

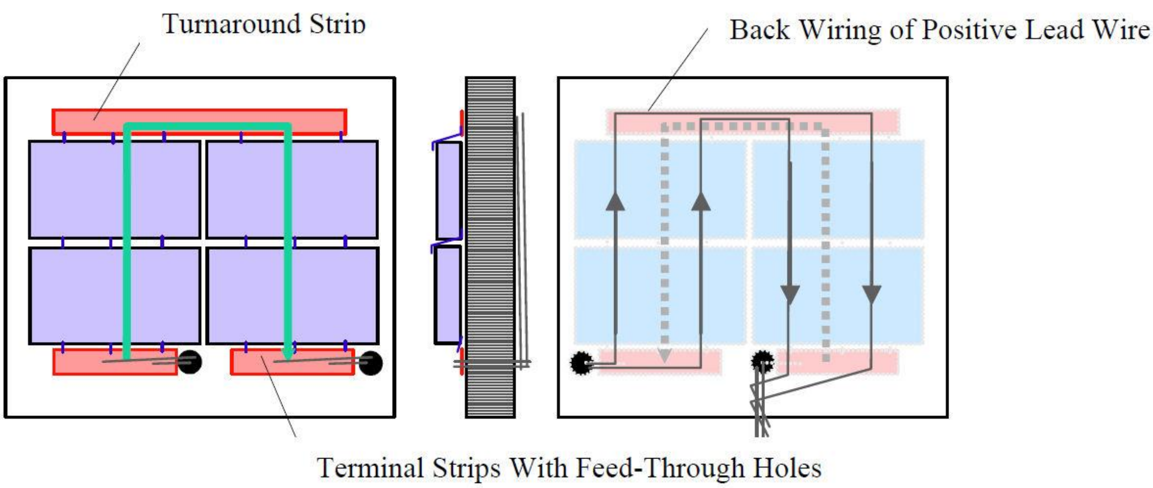


Figure 11 – Back-wiring of solar panels to cancel magnetic moments.

6.3 Configuration and mounting

If controllers are activated with mounting configurations (for sensors or actuators) that do not represent the actual physical mounting of the sensors or actuators, relative to the satellite coordinate frame, the ADCS can have the incorrect effect on the orientation of the satellite. This may, in extreme cases, cause the satellite to spin up to a very high rate, which may be fatal to the satellite mission.

For this reason, it is important to verify that sensors and actuators are configured correctly on the ground before launch. Sensors should be stimulated, and the calibrated vector outputs should be compared with the test setup to confirm that the vectors are correct.

The axis definitions are described in Section 4.1 and **Ref 1.**

The mounting transformations for the Sun and nadir sensor(s) are set in the CubeSupport *ADCS Config* tab. The mounting transformation must be set to transform the camera coordinate frame to the satellite coordinate frame as used by the ADCS.

Similarly, actuators' axes must be defined correctly and should be checked. In the CubeSupport *ADCS Config* tab, the ADCS bundle defined axes (1, 2, 3) must be set to the corresponding satellite axes (PosX, PosY, and PosZ). Figure 12 is an example of a possible setting.

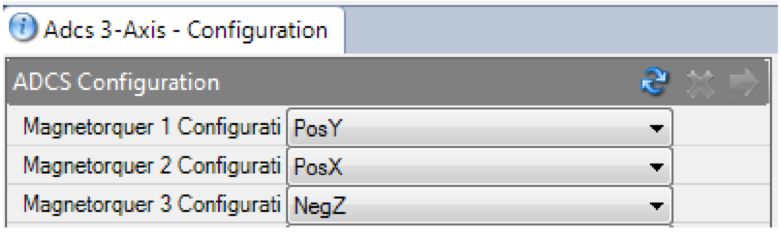


Figure 12 – Magnetorquer mounting transform.

Using the above setting, if a positive X Magnetorquer command is given, the ADCS unit's axis 2 magnetorquer will be activated and can be measured with a compass or external magnetometer as shown in Figure 13 (be aware that the magnetometer pulses on and off).

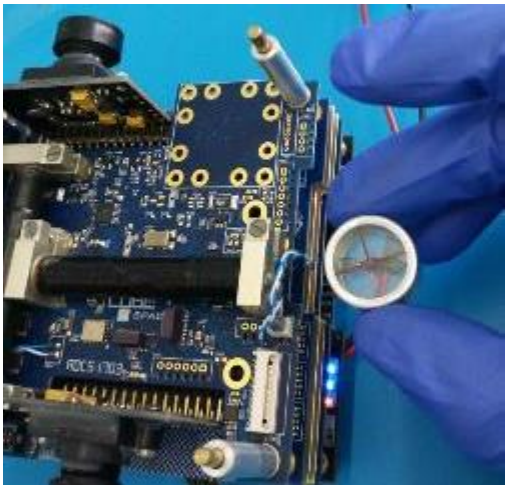


Figure 13 – Magnetorquer axes transformation verification.

6.4 Deployable magnetometer

Placement of the magnetometer is EXTREMELY critical for effective ADCS performance. Placement of the magnetometer next to magnetic parts of the satellite will severely affect the magnetometer, and may compromise the ADCS performance, even to the extent where the ADCS is unusable. Items to avoid having in close proximity to the magnetometer include:

- Electrical motors (reaction wheels, deployment wheels, etc.)
- Batteries
- Steel (or any ferrous) screws
- Torquer rods

6.5 Series resistance of power supply & inrush currents

The CubeADCS unit requires a good power supply to operate reliably. Most satellite EPS systems are designed to handle the power requirements of typical CubeSat subsystems and connects through the PC104 connection.

When the CubeADCS unit is tested on ground though, a normal bench power supply is often used. These power supplies are typically rated for supplying large currents, and do have the capability to power the CubeADCS unit. The supply is, however, usually connected to the ADCS unit through jumper wires, which very often have large resistances or unreliable connections. When the ADCS draws large currents, especially in-rush currents, the resistance of the wire or connection causes a voltage drop. This voltage drop, in many cases, is large enough to cause failures in the CubeADCS electronics. It is therefore recommended to use thick wires to connect bench power supplies to the CubeADCS, and to avoid using leads that have crimp connections.

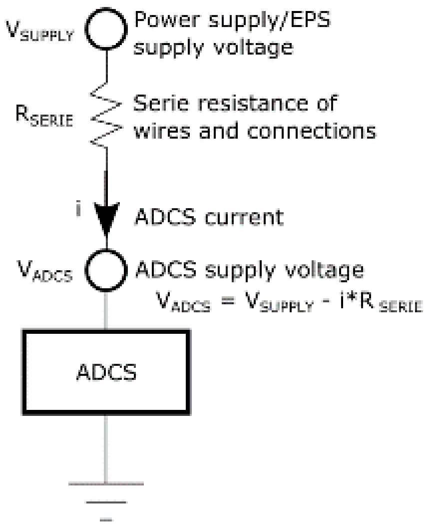


Figure 14 – Power supply to ADCS.

6.6 Y-axis MOI not the largest for Y-Thomson control

In some satellite configurations the Y-axis MOI is not the largest or at best only similar in size to another axis. In these cases, the magnetic Y-Thomson controller (Control Mode 3) will have difficulty to sustain a Y-spin. By commanding a constant Y-wheel momentum (*Set Wheel Speed* TC ID 17), the magnetic Y-Thomson controller will be able to work as expected. A negative Y-wheel speed of 5% to 20% of the maximum speed will be sufficient. The speed required depends on the Y-MOI/max-MOI ratio, e.g. 1 = 5%, 0.8 = 20%.

7 Document History

Version	Responsible person(s)	Pages	Date	Description of change
3.0	MK	ALL	20/03/2017	V3 First draft
3.01	J	13, 14	23/03/2017	Adding epoxy note of connectors
3.02	MK	ALL	22/06/2017	Merge Y mom and 3-axis
3.03	CG	37-41	04/07/2017	CubeStar Configuration
3.04	WHS	ALL	12/07/2017	Update Control Modes, Magnetometer calibration, Rate sensor range
3.05	CJG	ALL	24/07/2017	Formatting and version number correction.
3.06	MK	ALL	27/07/2017	Accept/Reject changes
3.07	CCH	All	25/05/2018	General language editing
3.08	HW, JG, LV	All	27/08/2019	Expanded Getting Started Section General Updates
3.09	GJVV	Various	31/03/2020	ACP and ref doc version updated, several updates for CubeSense V3, removed deprecated sections