Attitude Determination and Control System for EyasSAT for Hardware In the Loop Application

by

Christoffel Johannes Groenewald

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Research) in the Faculty of Engineering at Stellenbosch University

Department of Electric and Electronic Engineering

Supervisor: Prof. W.H. Steyn

April 2014

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 2014/02/20

Abstract

Attitude Determination and Control System for EyasSAT for Hardware In the Loop Application

C.J. Groenewald

Department of Electric and Electronic Engineering

Thesis: MScEng (E&E)

April 2014

An Attitude Determination and Control System (ADCS) demonstrator and testing platform was required for satellite engineering students. The ADCS demonstrator and testing platform will allow students to develop insight into the concepts and challenges of ADCS design and implementation. The existing model nano-satellite EyasSAT was used as a design platform for a new ADCS demonstrator. A new ADCS module (ADCS_V2) was developed to replace the existing EyasSAT ADCS module. The new module allows for three-axis ADCS and the demonstration of the ADCS on an air bearing platform. The air bearing allows full freedom of movement for yaw rotations with limited pitch and roll rotations. The actuators and sensors required for the ADCS were developed and integrated into EyasSAT. In addition a new PCB was designed to form the ADCS_V2 module. Attitude determination algorithms and attitude control algorithms were implemented and tested using MATLAB Simulink simulations. These algorithms were then implemented on the ADCS_V2 module. The ADCS was tested using Hardware In the Loop (HIL) techniques and an air bearing. The yaw attitude of EyasSAT could be controlled within 0.4 degrees accuracy with all the sensors active. In order to stabilize the air bearing platform, the pitch and roll angles were rate controlled. The pitch and roll rates were damped to within 6 mrad/s.

Uittreksel

Hoë Werkverrigting Orientasiebepaling en Beheerstelesel vir EyasSAT vir Hardeware-in-Lus toepassing

("High Performance Attitude Determination and Control System for EyasSAT for Hardware In the Loop Application")

C.J. Groenewald

Departement Elektries en Elektroniese Ingenieurswese

Tesis: MScIng (E&E)

April 2014

'n Oriëntasiebepaling en Beheerstelsel (OBBS) demonstrasie en toets platform was benodig vir satellietingenieurswese studente. Die nuwe OBBS sal studente toelaat om insig te ontwikkel met betreking tot die idees en uitdagings wat verband hou met die ontwikkeling en implementering van 'n OBBS. Die huidige nano-sateliet model EyasSAT was gebruik as 'n ontwerpsbasis vir die nuwe OBBS. Die nuwe OBBS was ontwikkel om die huidige module van EyasSAT te vervang. Die nuwe OBBS laat oriëntasiebepaling en -beheer in drie asse toe. Die nuwe OBBS en EyasSAT kan die werking van 'n OBBS demonstreer op 'n luglaerplatform. Die luglaer laat vrye rotasie om die gierhoek toe terwyl die rol- en stygings-as beperk word. Die aktueerders en sensors wat benodig word vir die OBBS is ontwikkel en geïntegreer in EyasSAT saam met 'n nuwe gedrukte stroombaanbord om die nuwe OBBS te vorm. Orientasiebepaling en orientasiebeheer algoritmes is geïmplementeer en getoets met die hulp van MATLAB Simulink simulasies. Die algoritmes was op die OBBS module geïmplementeer en getoets deur gebruik te maak van HIL tegnieke en praktiese toetse op die luglaer. Die rotasie hoek van EyasSAT kan met 'n akkuraatheid van 0.4 grade beheer word indien al die sensors gebruik word. Die rol en stygingshoeksnelheid was gekanselleer om die luglaer stabiel te hou. Die hoeksnelheid van die twee asse kon tot kleiner as 6 mrad/s beheer word.

Acknowledgements

I would like to express my sincere gratitude to the following people:

- Prof W.H. Steyn for his knowledge, guidance, patience as well as the opportunities he provided.
- My colleagues in the Electronic System Laboratory (ESL) for their helpful inputs and advice.
- All the people that were involved in the manufacturing of the new ADCS module. Specific mention, however, needs to go to Mr. Johan Arendse for his efforts.
- My family and friends for their support and understanding during my long hours at work. Especially Nicole de Kwaadsteniet, for her love, encouragement and faith in me.
- God, for giving me the strength, talents and wisdom to do the work I love.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

ADCS	Attitude Determination and Control System
BRF	Body Reference Frame
CAD	Computer Aided Design
COM	Centre of Mass
COP	Centre of Pressure
CSS	Course Sun Sensor
DC	Direct Current
DCM	Direct Cosine Matrix
deg	Degrees
DoF	Degree of Freedom
EKF	Extended Kalman Filter
EPS	Electrical Power Subsystem
ESL	Electronic Systems Laboratory
FPS	Frames Per Second
Gyro	Gyroscope
HIL	Hardware In the Loop
IC	Integrated Circuit
IMU	Inertial Measurement Unit

IR	Infra-red
IRF	Inertial Reference Frame
I/O	Input Output
LED	Light Emitting Diode
LEO	Low Earth Orbit
LSB	Least Significant Bit
m	Mass
MCU	Microcontroller Unit
T_s	Sampling Time
PCB	Printed Circuit Board
PWM	Pulse Width Modulated
rad	Radians
rpm	Revolutions per Minute
RW	Reaction Wheel
s	Seconds
SC	Spacecraft
SIL	Software In the Loop
SPI	Serial Peripheral Interface
USAF	United States Air Force
USAFA	United States Air Force Academy

Greek Letters

ϕ	Euler roll angle
θ	Euler pitch angle
ψ	Euler yaw angle

Φ	The rotation angle about the Euler axis
β	Air bearing friction constant
α	Angle between a light source and the normal of a photocell
ω	Angular rate

Constants

$g =$	$-9.81\,\mathrm{m/s^2}$ Earth gravity constant (towards earth centre or the nadir direction)
$\beta =$	$24.566\mu\frac{Nm}{rad/s}$ Air bearing friction constant

Subscripts

$X_b/Y_b/Z_b$	Body reference frame axes
$X_i/Y_i/Z_i$	Inertial reference frame axes
$X_o/Y_o/Z_o$	Orbit reference frame axes

Syntax and Style

$\mathbf{x}$	The vector **x** (usually lowercase)
$\hat{\mathbf{x}}$	Estimated vector
$\bar{\mathbf{x}}$	Normalized vector value of **x** or unit vector **x**
$\bar{x}$	Normalized value of x
$\mathbf{A}$	The matrix **A** (usually uppercase)

Chapter 1

Introduction

Nano-satellites, such as the popular CubeSat form factor, allow universities to manufacture miniaturized satellites that can be launched for the fraction of the cost of larger satellites. Although these small satellites have limited mass, volume and power, this allows them to be launched as a secondary payload piggybacking on the launch vehicle of larger satellites. Nano-satellites are easily adaptable and are usually made up of "off-the-shelf" components. Therefore small satellites have become popular with universities for research. Students are easily able to learn about satellites and the different subsystems found in satellites, whilst an excellent test bench is provided for hands-on experience in designing a satellite.

One of the subsystems, namely the Attitude Determination and Control System (ADCS), was developed and is discussed in this project. This subsystem is responsible for the orientation control of the satellite in its orbit which allows for the accurate pointing of the satellite payloads towards the earth and the solar panels toward the sun. The ADCS is typically required for earth observation cameras that need to be pointed to targets on the earth's surface.

An ADCS system will be implemented on a nano-satellite to allow the practical demonstration of the ADCS system to satellite engineering students. The objective of this project is to develop a complete ADCS system for a small nano-satellite and Hardware In the Loop (HIL) test procedures are used for evaluation. Thereafter, the system is tested using an air bearing to validate the ADCS. The air bearing also provides an excellent environment to visually demonstrate the working of the ADCS.

The ADCS data must be recorded and displayed by the system on a "ground station PC". This will allow for in-class demonstrations of a functioning ADCS for a nano-satellite, allowing students to get a firm grasp of the ideas involved in ADCS. Although a model satellite, called EyasSAT, was previously developed by the United States Air Force (USAF) for similar goals, it had limited ADCS. Thus this project aims to develop a new

ADCS system for EyasSAT in order to demonstrate more sophisticated ADCS algorithms for small satellites.

1.1 Goals

The goal of this project is to design, build and implement a three-axis ADCS for EyasSAT. Thus a new Printed Circuit Board (PCB) had to be developed to replace the existing ADCS PCB on the EyasSAT stack. Additional sensors and actuators accompanying the ADCS_V2 module are also designed and manufactured. The communications module on EyasSAT is used to implement a wireless data link between EyasSAT and a computer that will act as a "ground station". After the ADCS is developed an air bearing platform will be used for evaluation and testing. Once the module is completed, EyasSAT with improved ADCS capabilities can be used for demonstrations that indicate the functioning of an ADCS for a small satellite.

1.2 EyasSAT

The Department of Astronautics at the United States Air Force Academy (USAFA) developed EyasSAT to teach students more effectively about space systems engineering [5]. EyasSAT literally means “baby FalconSAT” [2]. FalconSat is the name of the flagship program at USAFA in which senior students design, build, launch and operate a real satellite. In preparation for this program, students have to complete a “Spacecraft Systems Engineering” prerequisite course and assemble an EyasSAT unit. Figure 1.1 gives an indication of an EyasSAT unit. Students learn about the different subsystems of a spacecraft and perform acceptance testing, characterization and integration of the EyasSAT subsystems. The course outcome objective is for each team to have a fully-functional space system. Therefore all spacecraft system and subsystem level issues are broadly covered in the Spacecraft Systems Engineering course.

Students experience space systems engineering concepts that are typically found in the Air Force during the EyasSAT course. Unlike the industry, the Air Force focuses on acceptance and verification of purchased resources. The EyasSAT course is in line with this objective and students are given a real indication of their potential speciality in the Air Force after graduation [2]. This approach allows laboratory discovery and hands-on experience at the spacecraft subsystem level.

EyasSAT focuses on six subsystems most satellites consist of. These subsystems consist of structural; power; data handling and communications; attitude determination and control; propulsion; and thermal systems. Additional modules can expand EyasSAT capabilities. Each of EyasSAT’s subsystems is composed of standalone hardware modules

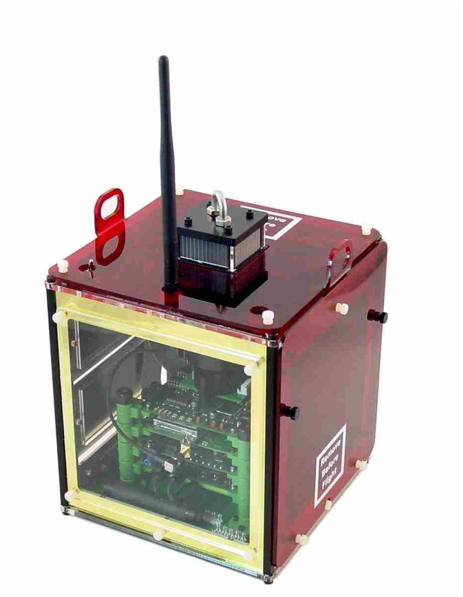


Figure 1.1: *EyasSAT as shown in [1]*

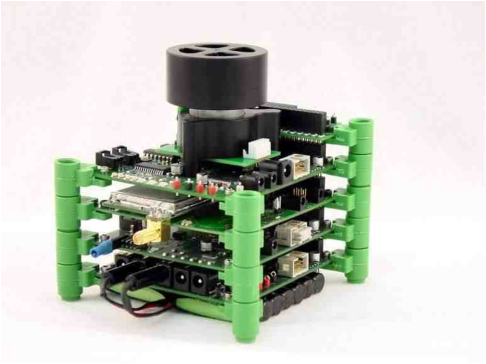


Figure 1.2: *EyasSAT Stack as shown in [1]*
s

built with "off-the-shelf" components that are integrated through a data and power bus called EyaBus. The standard EyasSAT stack is shown in Figure 1.2. The modules from top to bottom are: ADCS module, Communications module, Data handling module and Electrical Power Subsystem (EPS) module with two battery packs.

EyasSAT can be compared to popular CubeSat design methodology. However, the CubeSat design process is an extremely structured approach that has an emphasis on assembly, integration and test procedures. CubeSats are constructed to be flown in space in contrast to EyasSAT which is primarily used for education and to be "flown" in the classroom. A CubeSat is single-mission and single-task oriented and has a strict goal-oriented design. In comparison, EyasSAT is more flexible with more room for adaptation and modification. The existing EyasSAT ADCS module will be replaced with a new module, the ADCS_V2 module, for this project. The original module and sensors are shown above in Figure 1.3.

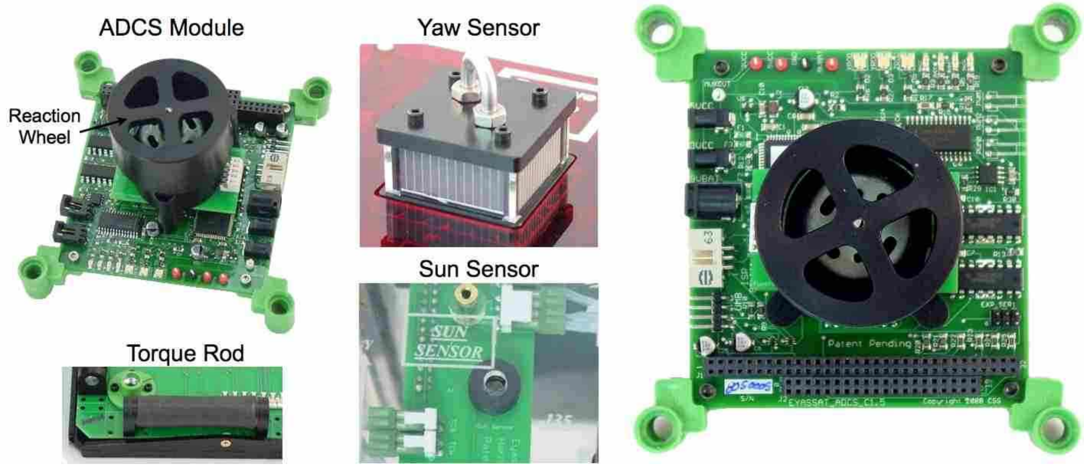


Figure 1.3: *Original ADC Module and Sensors [2]*

1.3 Test Platform

The newly developed ADCS_V2 module will be tested using simulations, HIL and Software-In-the-Loop (SIL) tests. However, ADCS_V2 module needs to be tested in an environment that simulates space as closely as possible for ADCS purposes at low cost. Existing low cost facilities for ADCS testing and validation for nano-satellites have been researched previously.

A number of papers were found describing the facilities developed at different institutions to validate and test the ADCS using low cost environmental simulators. A laboratory facility at the Russian Engineering Technological Centre “Scan Ex” that is used for testing algorithms for ADCS of the satellite “Chibis-M” was described [6]. A rotation air bearing was used by the facility in order to allow for the three-axis freedom of movement of the satellite. The air bearing table also contains a platform balancing system. Helmholtz coils are used to simulate the magnetic field of earth and a sun simulator is also incorporated. Sensors and actuators are mounted onto the platform to test their performance as well as the performance of the ADCS algorithms. In papers [7] and [8] a similar air bearing system is developed to test three-axis control of small satellites by testing the sensors, actuators and algorithms.

A different approach for testing the sensors, actuators and algorithms can be followed by mounting the entire satellite to the air bearing. The development of an ADCS system with "off-the-shelf" components is explained in a research article [9]. Here an entire Cubesat is mounted on an air bearing platform that can be manually balanced. Different control approaches are then implemented, tested and evaluated. According to this paper, future development will include a Helmholtz cage to simulate the earth’s magnetic field in orbit.

A new air bearing test platform is introduced in a conference paper [10] as well as the research projects that could be carried out using this platform. Air bearing simulators have been used for some time and have been proven to be effective tools as demonstrated by the papers previously mentioned. Air bearings are just one way of simulating the space environment, for the validation of ADCS. Another paper [11] describes the commissioning results and testing of the platform. This supports the relevance for applying an air bearing platform to validate the ADCS design. EyasSAT will be placed on an air bearing creating a near frictionless three Degree-of-Freedom (DoF) environment for validation and testing of the ADCS in three-axis.

The paper *Historical Review of Spacecraft Simulators* by Juana L. Schwartz, Mason A. Peck and Christopher D. Hall gives an overview of air bearing spacecraft simulators [12]. The paper discusses a number of examples of planar air bearing systems, as "systems that give freedom to translate and spin", and rotational air bearings, which allow for "three axis satellite attitude dynamics". The rotational air bearing used for this project is of the "Tabletop" configuration as described by [12]. The configuration allows full freedom of movement in the spin or yaw axis whilst the pitch and roll axes are physically limited to about 25 degrees. The air bearing with the nitrogen supply and air bearing mounting platform is shown in Figure 1.4 to Figure 1.6.

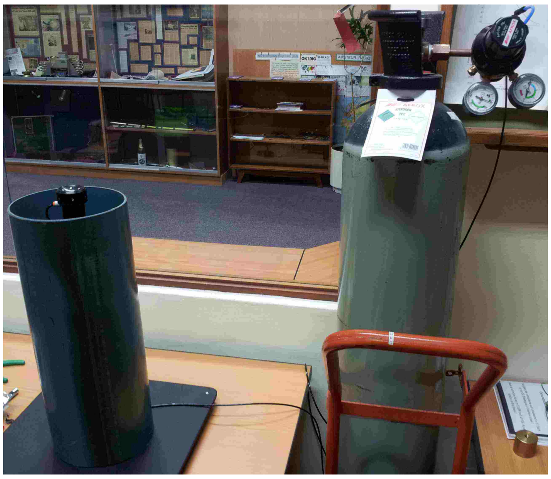


Figure 1.4: *Air Bearing with Nitrogen Supply*

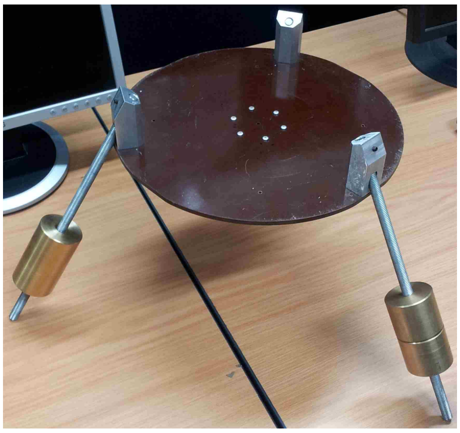


Figure 1.5: *Mounting Surface Top View*

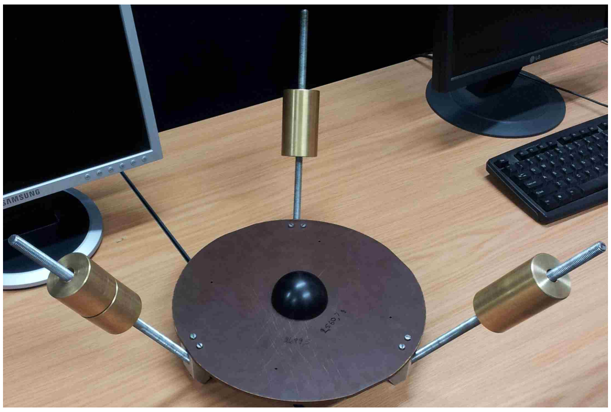


Figure 1.6: *Mounting Surface Bottom View*

The air bearing is supplied with pressurized nitrogen instead of compressed air. Nitrogen contains no contaminates such as water or oil normally found in a compressor air supply. The mounting surface of the air bearing or the "tabletop" has three balancing rods with adjustable counter weights.

The weights are used to offset the weight of EyasSAT and balance out the Centre of Mass (CoM). By appropriately adjusting the weights, the CoM can be moved to the Centre of Pressure (CoP). If the CoM coincides with the CoP, the mounting surface with EyasSAT can be placed in any orientation and it will remain stationary. However, when the weights are adjusted by hand this becomes very difficult to achieve. Instead the weights are adjusted so that the system will have stable pendulum motion by placing the CoM close to the CoP but below the CoP. The CoM will always return to its steady state position directly below the CoP because the gravitational force on the CoM causes a restoring gravity torque. This torque will be modelled as a disturbance torque in the dynamic equations of the system. In addition, this pendulum motion is also used to obtain the distance of the CoM below CoP and determine the moment of inertia of the system. The calculations for determining the position of the CoM and the moment of inertia of the system is shown in Appendix A.

1.4 Thesis Overview

Chapter 2 provides the theoretical background needed for the following chapters. The coordinate frames are defined and an explanation is provided of how the attitude of EyasSAT is described. Dynamic equations are given and disturbance torques involved in the air bearing experiment are discussed.

Chapter 3 describes the sensors that are used for attitude determination. A discussion is provided of the calibration and accuracy of each sensor.

Chapter 4 focuses on the design of the required actuators. The detailed design, assembly and control of the reaction wheels are described and demonstrated.

Chapter 5 discusses the design of the ADCS_V2 module PCB. A description is provided how the PCB is used to read all the sensors and generate command signals for the actuators. The integration of the sensors, actuators and PCB into EyasSAT is also presented in this chapter.

Chapter 6 defines the attitude determination algorithms. This chapter expands on the definitions made in Chapter 2. These algorithms are implemented on the Microcontroller Unit (MCU) found on the ADCS_V2 module PCB.

Chapter 7 highlights the attitude control algorithms implemented on EyasSAT. These algorithms rely on the estimated attitude calculated with the algorithms, shown in Chapter

6 and are used to generate control commands to the actuators.

Chapter 8 demonstrated the overall program flow found on the MCU. An overview is provided of sensor readings, the ADCS algorithms executed and the main program loop programmed on the MCU.

Chapter 9 introduces the MATLAB Simulink simulation environment that is used to evaluate the ADCS algorithms developed in Chapter 6 and 7. The results obtained in this chapter provide a good representation of the responsiveness of the real system. Findings are compared to the results obtained in Chapter 10.

Chapter 10 contains the air bearing test results which are compared to the results obtained in Chapter 9 where applicable. The practical test results indicate how reliable the simulation environment is. The data represented in this chapter proves that the performance of the ADCS can be practically demonstrated.

Chapter 11 contains the final conclusion and recommendations. Future improvements that could be valuable for ADCS performance are discussed.

Chapter 2

Theoretical Background

In order to describe the attitude of a Spacecraft (SC), a set of conventions and definitions have to be established. This chapter introduces the theoretical conventions used. The coordinate frames, in which the attitude description is done, and how the attitude description is expressed are included in these definitions. Furthermore, the equations of attitude motion which describe the orientation of a SC and the effect of disturbance torques on the SC are required to simulate and estimate the attitude of the SC. In this chapter, the dynamic equations are defined and the most prominent disturbance torques discussed.

There are three common reference frames used for attitude description. These are the Inertial Reference Frame (IRF), Orbit Reference Frame (ORF) and Body Reference Frame (BRF). The ORF rotates with the orbital motion with the X_o- and Z_o-axis in the orbit plane and Y_o-axis in the orbit anti-normal direction. The Z_o-axis points towards the centre of earth (nadir direction). The BRF is fixed to the SC body and defined relative to the SC body. The BRF is usually chosen to coincide with the principal moment of inertia axes.

These coordinate frames allow the attitude to be described in terms of the orientation of the BRF with regards to the ORF or IRF. This is done using quaternion representation or Euler angles in a Direction Cosine Matrix (DCM) or transformation matrix. This matrix can transform vectors from the one reference frame to the other, usually from the ORF or IRF to the BRF.

The equations of attitude motion are used in simulations and estimators to predict the motion of the SC. These equations show how the SC reacts to control torques and disturbance torques.

2.1 Coordinate Frames

The attitude or orientation of the SC is described using the coordinate frames. For the purposes of the air bearing the attitude of EyasSAT is described using the BRF relative to an IRF with both origins fixed to the CoP, or pivot point, of the air bearing. The IRF is an approximation as the Earth's rotation is ignored. The IRF axes are fixed in the room while the BRF is fixed to EyasSAT's body. Normally the BRF is used to represent the satellite and its attitude relative to the IRF or ORF. This leads to the problem of finding the attitude of one reference frame (the BRF) relative to anther reference frame (the IRF).

The IRF is defined in terms of the room in which the air bearing tests are done. An image of the room is shown in Figure 2.1. The positive X_i-axis is in the direction of the book shelf, the positive Y_i-axis is toward the window and the positive Z_i-axis is upward or opposite the gravity vector (zenith direction).

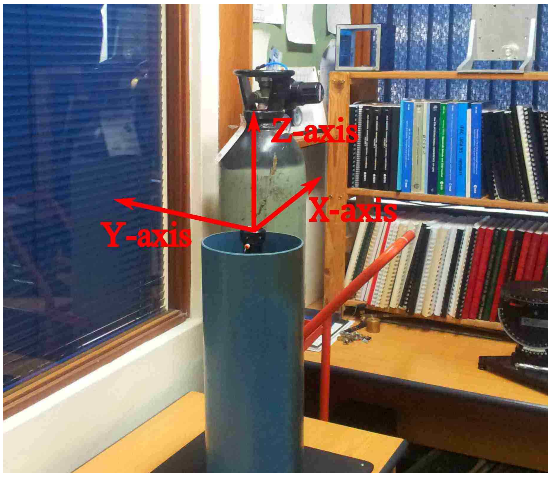


Figure 2.1: *IRF Definition*

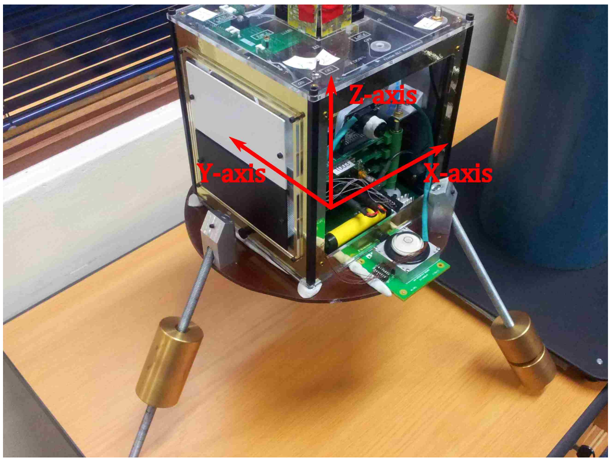


Figure 2.2: *BRF Definition*

The BRF is fixed to EyasSAT's body. The positive X_b-axis points to the panel opposite the thermal testing panel, positive Y_b-axis is in the opposite direction of the camera and Z_b-axis is towards the yaw sensors. The axes are shown in Figure 2.2 on EyasSAT.

2.2 Attitude Description

The attitude of EyasSAT is described in terms of Euler angles (roll ϕ, pitch θ and yaw ψ). The angles represent three consecutive, right-hand positive rotations which will obtain the BRF, starting from the IRF. The order in which these angle rotations take place effects the resulting transformation matrix. The Euler 2-1-3 sequence is used in this project. The first rotation θ is about the Y_i-axis, second rotation ϕ is about the intermediate X'-axis and the last rotation ψ is about the Z_b-axis. An example of an Euler rotation is shown below in Figure 2.3 and obtained from [13].

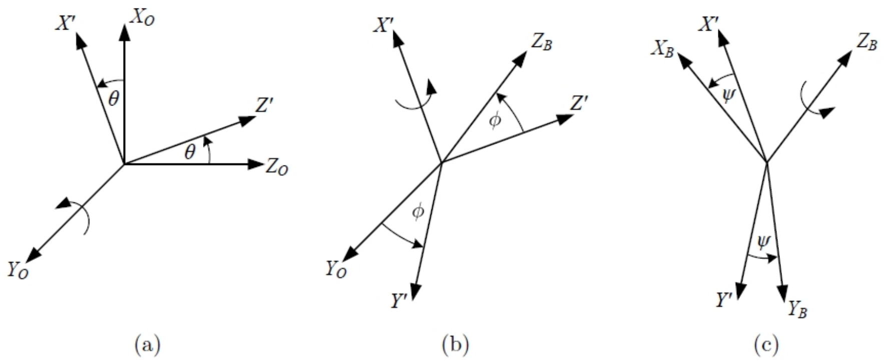


Figure 2.3: *Euler 2-1-3 Rotation*

The DCM or transformation matrix $\mathbf{R}_{bi}$ can be used to transform a vector in the IRF to the BRF. The equation (2.2.1) shows the calculated DCM for the Euler 2-1-3 rotations [13].

$$\mathbf{R}_{bi} = \begin{bmatrix} (\sin\phi\sin\theta\sin\psi + \cos\theta\cos\psi) & \cos\phi\sin\psi & (\sin\phi\cos\theta\sin\psi - \sin\theta\cos\psi) \\ (\sin\phi\sin\theta\cos\psi - \cos\theta\sin\psi) & \cos\phi\cos\psi & (\sin\phi\cos\theta\cos\psi + \sin\theta\sin\psi) \\ \cos\phi\sin\theta & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \tag{2.2.1}$$

The Euler angles can be calculated from the DCM, if it is known, using equations (2.2.2) to (2.2.4).

$$\phi = -\arcsin(R_{3,2}) \tag{2.2.2}$$

$$\theta = \arctan\left(\frac{R_{3,1}}{R_{3,3}}\right) \tag{2.2.3}$$

$$\psi = \arctan\left(\frac{R_{1,2}}{R_{2,2}}\right) \tag{2.2.4}$$

Note that the arctan is implemented as a four quadrant function in these equations.

Euler angles can cause singularities at certain angles when used in numerical computation. The Euler symmetric parameters (quaternion) are a better representation with no singularities and no trigonometric functions in the transformation matrix [14]. The quaternion representation is used during numerical computation. Since the quaternion representation does not have an easily interpreted physical representation, the Euler angles will be used in the present study to represent the attitude during simulations and test outputs.

The quaternion set consists of four parameters, which are defined in terms of the Euler axis. Euler's Eigenaxis Rotation Theorem states that the finite rotation of a rigid body with one point fixed can be expressed by a single rotation around some fixed axis [14]. This fixed axis, the Euler axis, is represented by a unit vector in the IRF or BRF with components $\bar{e} = (e_x, e_y, e_z)$. The four Euler symmetrical parameters, or quaternions, are defined in equation (2.2.5) to (2.2.8), where Φ is the rotation angle about the Euler axis.

$$q_1 = e_x \sin\left(\frac{\Phi}{2}\right) \tag{2.2.5}$$

$$q_2 = e_y \sin\left(\frac{\Phi}{2}\right) \tag{2.2.6}$$

$$q_3 = e_z \sin\left(\frac{\Phi}{2}\right) \tag{2.2.7}$$

$$q_4 = \cos\left(\frac{\Phi}{2}\right) \tag{2.2.8}$$

The quaternion components are also related to one another by the constraint contained in equation (2.2.9).

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \tag{2.2.9}$$

The transformation matrix $\mathbf{R}_{bi}$ can be written in terms of quaternions as shown in equation (2.2.10).

$$\mathbf{R}_{bi} = \begin{bmatrix} (q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{2.2.10}$$

In the case where the DCM is known, it can be used to calculate the quaternion parameters with equations (2.2.11) to (2.2.14).

$$q_4 = \frac{1}{2}\sqrt{1 + R_{1,1} + R_{2,2} + R_{3,3}} \tag{2.2.11}$$

$$q_1 = \frac{q_4}{4} \times (R_{2,3} - R_{3,2}) \tag{2.2.12}$$

$$q_2 = \frac{q_4}{4} \times (R_{3,1} - R_{1,3}) \tag{2.2.13}$$

$$q_3 = \frac{q_4}{4} \times (R_{1,2} - R_{2,1}) \tag{2.2.14}$$

2.3 Dynamic Equations of Attitude Motion

A SC's attitude motion can be represented by Newton-Euler dynamic equations. In the present study the dynamic equations are used to simulate the movement of EyasSAT on the air bearing.

2.3.1 Dynamic Equations

The Newton-Euler dynamic equation describes the effects of applied internal and external torques on the angular acceleration of the satellite [13]. The dynamics of a SC in inertial space is governed by the Newton-Euler equation (2.3.1) [13; 15; 16].

$$\mathbf{I}\dot{\boldsymbol{\omega}}_B^I = \mathbf{N}_P + \mathbf{N}_A + \mathbf{N}_D - \omega_{\mathbf{B}}^{\mathbf{I}} \times \left(\mathbf{I}\omega_{\mathbf{B}}^{\mathbf{I}} + \mathbf{h}\right) - \dot{\mathbf{h}} \tag{2.3.1}$$

where

$$\mathbf{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = \text{Moment of Inertia [MoI] tensor} \tag{2.3.2}$$

and,

$$\boldsymbol{\omega}_{\boldsymbol{B}}^{\boldsymbol{I}} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \text{Inertially referenced body angular rate vector} \tag{2.3.3}$$

and,

$$\mathbf{h} = \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = \text{Momentum of the reaction wheels in BRF axes} \tag{2.3.4}$$

and,

$\mathbf{N}_P =$ the pendulum disturbance torque vector, created by air bearing system setup

$\mathbf{N}_A =$ disturbance torque created by air bearing friction vector

$\mathbf{N}_D =$ external disturbance torque vector.

The EyasSAT air bearing and weight rods experiment is considered to be symmetrical. The assumption results in off-diagonal products of inertia elements in the MoI tensor **I** to be zero. There exists a set of body fixed axes for any shape of a rigid body. If the products of inertia are all zero (the moment of inertia matrix is a diagonal matrix), this is known as the principal moment of inertia. In applying the results obtained in Appendix A.5 the MoI is given as:

$$\mathbf{I} = \begin{bmatrix} 252.3518 \times 10^{-3} & 0 & 0 \\ 0 & 226.3869 \times 10^{-3} & 0 \\ 0 & 0 & 162.7543 \times 10^{-3} \end{bmatrix} \text{kg.m}^2 \tag{2.3.5}$$

The Newton-Euler dynamic equation for each axis can now be determined. Furthermore the acceleration of the angular rate can be calculated or estimated given the body angular rate, inertia and state of the RWs.

2.3.1.1 Set of Newton-Euler Dynamic Equations

Equation (2.3.1) is used to find the equation for each axis as shown in equation (2.3.6) to (2.3.8).

$$I_{xx}\dot{\omega}_x = N_{Px} + N_{Ax} + N_{Dx} + \omega_y\omega_z[I_{yy} - I_{zz}] - h_z\omega_y + h_y\omega_z - \dot{h}_x \tag{2.3.6}$$

$$I_{yy}\dot{\omega}_y = N_{Py} + N_{Ay} + N_{Dy} + \omega_x\omega_z[I_{zz} - I_{xx}] - h_x\omega_z + h_z\omega_x - \dot{h}_y \tag{2.3.7}$$

$$I_{zz}\dot{\omega}_z = N_{Pz} + N_{Az} + N_{Dz} + \omega_x\omega_y[I_{xx} - I_{yy}] - h_y\omega_x + h_x\omega_y - \dot{h}_z \tag{2.3.8}$$

The equations can also be written with the angular acceleration as the subject.

$$\dot{\omega}_x = [N_{Px} + N_{Ax} + N_{Dx} + \omega_y\omega_z[I_{yy} - I_{zz}] - h_z\omega_y + h_y\omega_z - \dot{h}_x]/I_{xx} \tag{2.3.9}$$

$$\dot{\omega}_y = [N_{Py} + N_{Ay} + N_{Dy} + \omega_x\omega_z[I_{zz} - I_{xx}] - h_x\omega_z + h_z\omega_x - \dot{h}_y]/I_{yy} \tag{2.3.10}$$

$$\dot{\omega}_z = [N_{Pz} + N_{Az} + N_{Dz} + \omega_x\omega_y[I_{xx} - I_{yy}] - h_y\omega_x + h_x\omega_y - \dot{h}_z]/I_{zz} \tag{2.3.11}$$

2.3.2 Kinematic Equations

The kinematic equation describes the change in the attitude without taking into account the force that caused the change [13]. The quaternion representation of the equation is given by equation (2.3.12).

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{\Omega}\mathbf{q} \tag{2.3.12}$$

with

$$\mathbf{\Omega} = \begin{bmatrix} 0 & w_z & -w_y & w_x \\ -w_z & 0 & w_x & w_y \\ w_y & -w_x & 0 & w_z \\ -w_x & -w_y & -w_z & 0 \end{bmatrix} \tag{2.3.13}$$

and

$$\mathbf{q} = [q_1\ q_2\ q_3\ q_4]^T \tag{2.3.14}$$

This equation can be used to propagate the attitude of the SC given the angular rate.

2.4 Disturbance Torques

2.4.1 Air Bearing Friction

The movement of air in the air bearing can cause very small levels of friction. As a result a torque disturbance is caused on the top part of the bearing. The influence of the air on the bearing is characterized and taken into account by modelling it as a friction force. In order to measure the friction constant, the Z_b-axis Reaction Wheel (RW), was given a known speed and the decay in angular speed over time of the system was recorded. In Figure 2.4, one signal (blue) settles at 2000, which is the speed of the RW in rpm. The second sloping signal (black) is the gyroscope output and shows how the system on the air bearing loses speed as a result of aerodynamic friction. The first data marker shows a gyroscope output of 410 units and the last marker shows an output of 375 units. This indicates a change in speed of 35 units which can be converted to 0.459 deg/s.

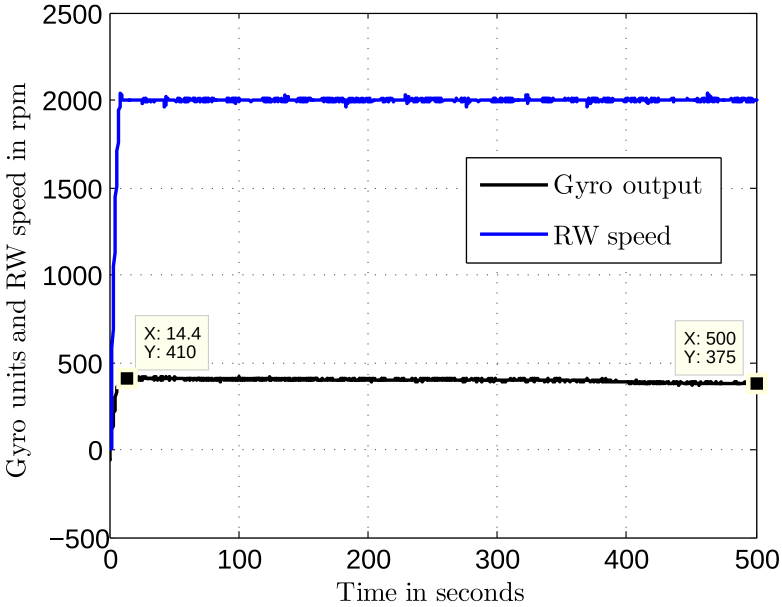


Figure 2.4: *Change in System Angular Rate Given a Constant Speed to the Z_b-Axis RW.*

The friction torque causing this decay is calculated from the result shown and other similar measurements as shown in equation (2.4.1)[11].

$$N_{Az} = I_{zz}\frac{\Delta\omega_z}{\Delta t} = 2.268\mu \text{ to } 2.687\mu \text{ Nm} \tag{2.4.1}$$

From this result obtained in (2.4.1) a friction constant β was derived in equation (2.4.2).

$$\begin{aligned} N_{Az} &= \beta\omega_z \\ \beta &= \frac{N_{Az}}{\omega_z} \approx 25.789\mu\frac{Nm}{rad/s} \end{aligned} \tag{2.4.2}$$

The equation calculating air bearing disturbance torque is defined in equation (2.4.3).

$$\mathbf{N}_A = \beta\boldsymbol{\omega}_{\boldsymbol{B}}^{\boldsymbol{I}} \tag{2.4.3}$$

2.4.2 Pendulum Disturbance Torque

The CoM is near the centre of EyasSAT, when EyasSAT is placed on the air bearing platform. This means the CoM is above the air bearing pivot point, while the air bearing pivot point is the CoP. Therefore the system is unstable, which is visible as the system always tends to fall over. Rods with balancing weights are mounted to the platform to balance and stabilize the system. The weights are moved into position to place the CoM below the CoP, enabling the system to balance and exhibit stable behaviour. Ideally, the CoP needs to be aligned to the CoM, which prevents the system from being impacted by gravitational torques. Once the CoM is slightly below the CoP, the CoM will experience a pendulum motion caused by gravitational torque. Figure 2.5 shows a two dimensional example of the forces experienced by the CoM. At rest there are no resulting forces because the force of gravity (green) and the force of the CoP (black) cancel each other out. If the CoM is not below the CoP, the CoM experiences a resulting force (red) or torque to move back to the position below the CoP.

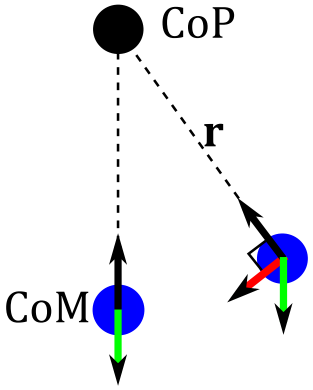


Figure 2.5: *Pendulum Motion*

The torque generated by this pendulum effect can be calculated by equation (2.4.4).

$$\mathbf{N}_P = \mathbf{r} \times \mathrm{m}\mathbf{R}_{bi}\mathbf{g} \tag{2.4.4}$$

where **r** is the position vector of the CoM below the CoP, m is the mass of the system and

$\mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$ is the gravity vector with g the gravity constant -9.81 m/s.

The gravity vector is expressed in the IRF and has to be transformed to the BRF, with the matrix $\mathbf{R}_{bi}$, to calculate the pendulum torque. The pendulum torque can thus be written as showed in equation (2.4.5).

$$\mathbf{N}_P = mg \begin{bmatrix} r_y R_{3,3} - r_z R_{2,3} \\ r_z R_{1,3} - r_x R_{3,3} \\ r_x R_{2,3} - r_y R_{1,3} \end{bmatrix} \tag{2.4.5}$$

Since weights are used to balance the system in an upright position it can be assumed that the CoM is directly below the CoP, the position vector **r** then only has a Z_b-component and equation (2.4.5) is reduced to equation (2.4.6).

$$\mathbf{N}_P = mg \begin{bmatrix} -r_z R_{2,3} \\ r_z R_{1,3} \\ 0 \end{bmatrix} \tag{2.4.6}$$

2.5 Theoretical Background Summary

The theoretical conventions and definitions established in this chapter will be used throughout the rest of the thesis unless stated otherwise. All the following algorithms are based on these coordinate frames and attitude description definitions. The dynamic equations of attitude motion that are implemented in the simulation environment will be discussed in Chapter 9.

In the present study, two reference frames will be used. These frames include firstly, the approximated IRF as defined in the air bearing test room. Secondly, the BRF centred on the air bearing pivot point and defined relative to EyasSAT's body. The attitude is numerically represented by quaternions. However, will be visually represented by roll, pitch and yaw angles in degrees. The dynamic equations are established along with the disturbance torques. Two disturbance torques that are accounted for include the torque caused by the air flow in the air bearing and the effect of gravity on the air bearing setup.

Chapter 3

Sensor Design

Sensors are required to sense or measure the states of the SC and certain references in the observable space, which enable SC attitude determination. The attitude determination is undertaken by finding the orientation between two reference frames, the orientation of the BRF compared to the IRF. The sensors supply information about the BRF whilst mathematical models supply information about the IRF. The sensors normally generate a unit vector that indicates the position of the observable or detectable references relative to the BRF. Mathematical models generate the same vectors in the IRF. Table 3.1 shows the reference sources used for attitude determination as shown in [14].

These references are only useful in space, except for the inertial space references. In the simulation setup, for HIL purposes, references similar to these were modelled. A bright light was used to represent the sun, two small LEDs were used to represent stars and gravity was used to indicate the nadir direction.

There are a number of well-known sensor types that are typically found on satellites, each with different implementation methods. Appropriate sensors are fitted depending on the accuracy required by the SC. Table 3.2 demonstrates some sensors that are commonly used and the range of accuracy that can be expected [17].

Nano-satellites are in the mass range of 1 to 10 Kg and some sensors are not well suited for small satellites such as EyasSAT. These satellites have very restrictive volume, mass and power constraints making the use of large and power intensive sensors impractical in most cases. The most commonly used sensors are forms of sun sensors, earth sensors and magnetometers. These satellites are normally in LEO where the magnetic field of the earth can still be detected strongly and can also be used for magnetic control. Solar panels can double as a power source and an attitude sensor on such small satellites. If the current output can be monitored, then the solar panels can be used as a Coarse Sun Sensor (CSS). The following section will discuss how EyasSAT contains its own form of CSS. A camera is added to EyasSAT to allow for more accurate measurements than can be

Reference	**Advantages using reference**	**Disadvantages using reference**
Sun	Brightest source. Location required for solar cells orientation and equipment protection.	Not always visible (eclipse)
Earth	Always available for IR sensors. Location needed by payload and antenna's.	Horizon needs to be sensed (scanning motion sometimes required). Visual sensors must be protected from sun.
Magnetic Field	Low power usage to sense. Always available for LEO.	Available near earth. Limited by modelling accuracy. SC can influence the sensor.
Stars	High accuracy. Available anywhere in the sky.	Sensor normally larger, complex, time consuming and expensive. Needs to be protected from sun.
Inertial Space(Gyro, Accelerometer)	No external sensors. Orbit independent. Accurate for limited intervals.	Sense change in orbit and attitude only. No absolute measurements are made. Subject to drift.

Table 3.1: Attitude Determination Sensors Reference Sources

Sensor	**Performance**
Gyroscopes	Drift Rate: 0.003 deg/hr to 1 deg/hr
Star sensors	0.0003 deg to 0.01 deg
Earth/Horizon Sensor	0.05 deg to 1 deg (0.1 deg best for LEO)
Sun Sensor	0.0005 deg to 3 deg
Magnetometer	0.5 deg to 3 deg

Table 3.2: ADCS Sensors

supplied by the CSS, and is discussed in Section 3.2. In addition an Inertial Measurement Unit (IMU) is also added to measure body rates and the gravity vector, which will be discussed in Section 3.3.

3.1 Coarse Sun Sensor

EyasSAT has four solar cells arranged on the sides of a cube. These are located on the top facet of EyasSAT, which is called the *Yaw Attitude Sensor* [1]. The sensors are shown in Figure 3.1. Each cell in the sensor is designated with a name and is illustrated in Figure 3.2, along with the body axes. These solar cells convert the energy of light that falls onto the cell into electricity. A CSS is formed by the arrangement of solar cells. The sensor can give information about the position of the sun or a light source relative to the BRF.

Small solar cells like the ones used on EyasSAT can be arranged on a satellite to obtain a rough sun or light vector.

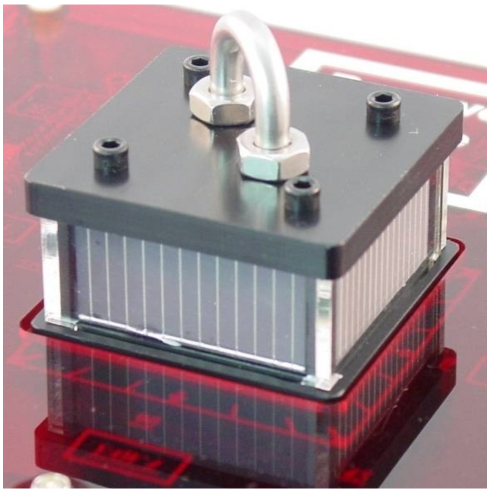


Figure 3.1: *Yaw Attitude Sensor* [1].

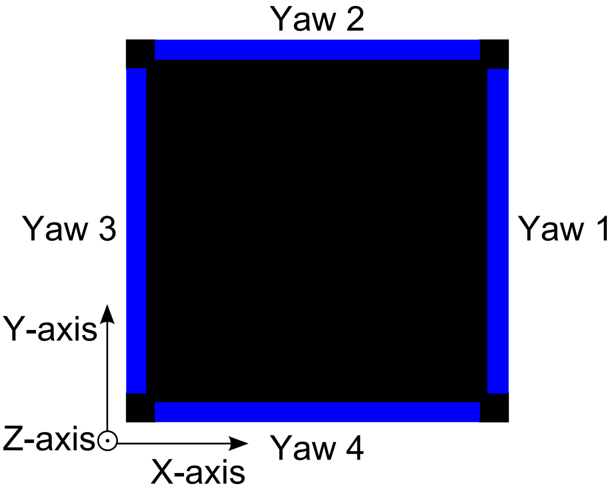


Figure 3.2: *Name Designation*

3.1.1 Working of Coarse Sun Sensor

The CSS generates a unit vector in the BRF that points toward the sun. Analogue sensors, such as the ones on EyasSAT, are based on solar cells, or photocells. Ideally their current output is proportional to the cosine of the angle between the light source and the normal of the photocell. This angle is represented by α for the rest of the text and is illustrated in Figure 3.3.

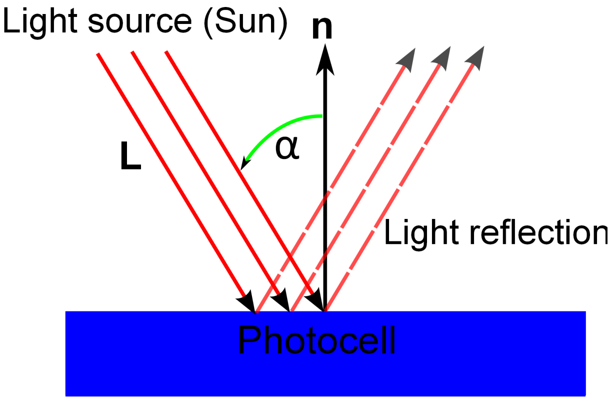


Figure 3.3: *Light Vector on Photocell*

In the ideal case the photocell will output the maximum amount of current when the light falls perpendicular to the cell surface. The output will decrease following a cosine function with the amount the angle α diverges from perpendicular in any direction. The solar cell will also have a 180 degree Field of View (FoV). The response can be described using equation 3.1.1.

$$I(\alpha) = I_0 \cos(\alpha) \tag{3.1.1}$$

In equation (3.1.1) I_0 represents the maximum current output. The output of the ideal photocell as a function of the angle α is shown in Figure 3.4.

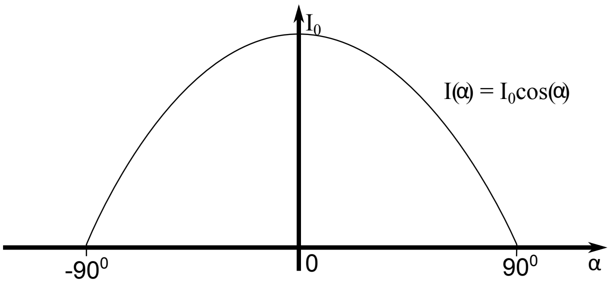


Figure 3.4: *Ideal Response of Photocell*

Normally, the cells are placed in the body axes. This allows the cells to output the components of the unit vector in the X_b-,Y_b- and Z_b-axis. The unit vector will then point in the direction of the light source in the BRF. However in reality, the output will not perfectly follow a cosine function because of Fresnel reflection, the effective photocell area and angle-dependent reflection at the air-cell interface [14]. Once light makes a very small angle to the cell's surface, most of the light will be reflected from the cell. The output will not change but stay at the minimum value because the cell will not respond to such a small amount of light. Therefore it is not ideal to use the photocell to detect light at large angles of α.

Another problem of using one photocell aligned to a body axes is that the measured output indicates that the light source is moving. The output is changing but the cell cannot determine the direction in which the light is moving. The only way to determine the direction of the movement of the light is to rely on cells placed on other axis to also register a change in light intensity. Due to the reduced FoV of the cells, the light source sometimes needs to move significantly before detection by the cell on the next axis.

Two photocells per axis can be used to increase the FoV and detect light at large angles of α [14]. This concept is illustrated in Appendix B.1. The photocell's FoV may be physically reduced with apertures to not detect light at large angles of α. However, the combination of the reduced FoV of both can increase the total FoV.

3.1.2 Evaluation of Yaw Attitude Sensor

Since the sensors are placed on the $\pm X_b$ and $\pm Y_b$ sides of the cube on EyasSAT, the sensors can only supply information of the light source vector in the X_b,Y_b plane of the

BRF. This implies that the vector can only give information about the yaw orientation of EyasSAT.

The short circuit current output is measured by measuring the small voltage drop over a low valued resistor. The measured voltage is amplified with a non-inverting op-amp. The circuit is illustrated in Figure 3.5. The short circuit current is less sensitive to temperature variations than the open circuit voltage of a solar cell [14].

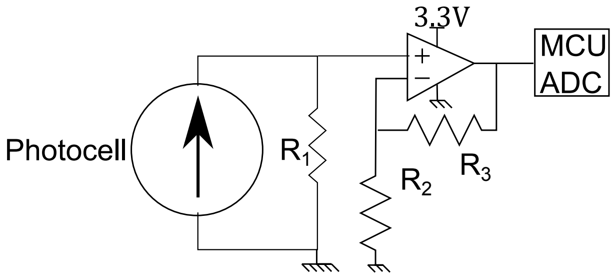


Figure 3.5: *Photocell Circuit*

A unit vector to the source of light is obtained through calibrating the cells by measuring their maximum and minimum output, in order to generate a normalized output. The normalization equation is shown in (3.1.2) [11]. This will allow each cell to output a unit vector component in the X_b,Y_b body plane, which indicates the light intensity that falls on the cell.

$$\bar{PC}\,(\pm X_b \text{ or } \pm Y_b) = \frac{PC_{raw} - PC_{min}}{PC_{max} - PC_{min}} \tag{3.1.2}$$

A unit vector that points toward the light source in the X_b,Y_b plane can be generated using the largest unit vector component of the cells located in the same axis as shown in equation (3.1.3) [11].

$$\mathbf{CSS}_{body} = \begin{bmatrix} \bar{PC}(\pm X_b) \\ \bar{PC}(\pm Y_b) \\ 0 \end{bmatrix} / \sqrt{\bar{PC}^2(\pm X_b) + \bar{PC}^2(\pm Y_b)} \tag{3.1.3}$$

Where the ± symbol implies the selection of the photocell with the largest output in that axis, as there is a cell pointing in the positive and negative direction of the X_b- and Y_b-axis.

The test setup and detailed figures of each solar cell output and the unit vector generated is shown in Appendix B.2. The angle of the light source can be determined in the BRF by calculating the arctan of the X_b and Y_b vector components of the $\mathbf{CSS}_{body}$ unit vector, as shown in Figure 3.6.

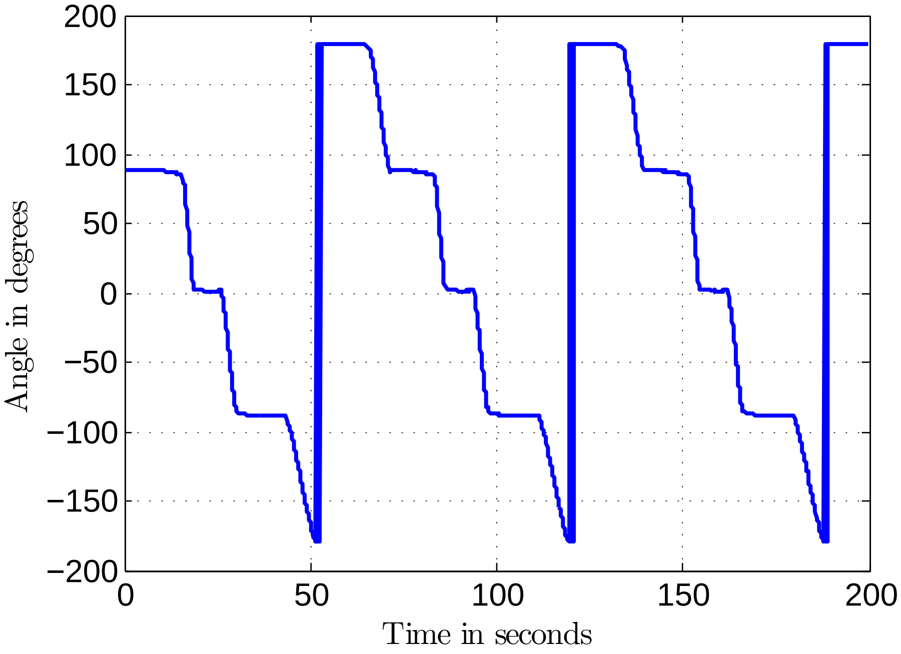


Figure 3.6: *Yaw Angle of Light Source with regards to the BRF*

Figure 3.6 indicates that although the angle was constantly changing, at some points the angle stays constant. This is due to the non-ideal response and reduced FoV of the solar cells. The decreased FoV of the cells can be attributed to the mounting of the cells on EyasSAT, each cell has a small pillar on either side that can cast a shadow on the cell. Perspex pieces are also in front of the cells to protect them, however, this can cause light to be reflected away from the sensor when the light vector makes a large angle with the normal of the solar cell. New photo-diode sensors were installed and used instead of the original photocell sensors to improve the accuracy of the yaw sensor.

3.1.3 Photo-diodes

The solar cells were disconnected and then the photo-diodes were mounted on PCBs. They were attached to the cube on the top of EyasSAT in front of each solar cell. The photo-diodes used were the SLCD-61N8 from Silonex Inc [18]. The datasheet can be viewed in Appendix D.1. Figure 3.7 shows an image of the photo-diode on the mounting PCB. The circuit of the photo-diode is shown in Figure 3.8.

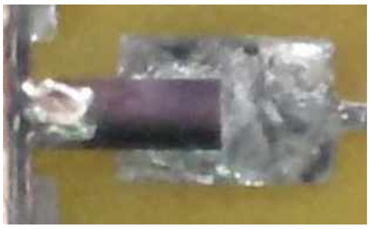


Figure 3.7: *Photo-diode mounted on PCB*

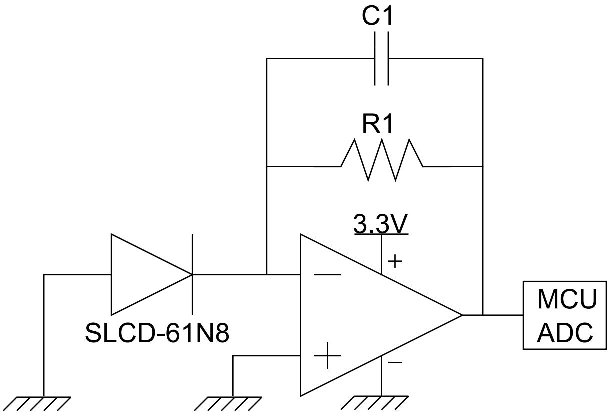


Figure 3.8: *Photo-diode Circuit*

Photo-diode characteristics were measured by rotating each photo-diode through 180 degrees, with regards to a light source, and measuring the output. The detailed test setup and results are shown and discussed in Appendix B.3. One photo-diode's response is shown in Figure 3.9.

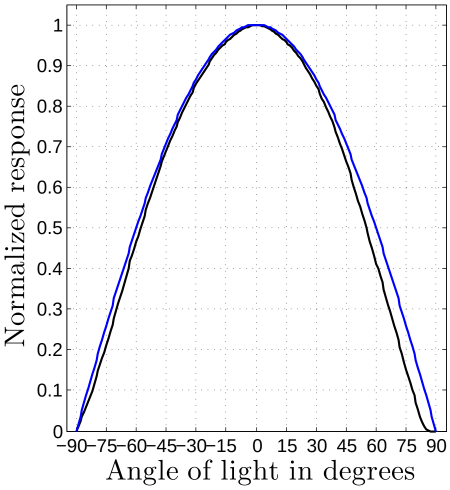


Figure 3.9: *Yaw1 and Ideal Response*

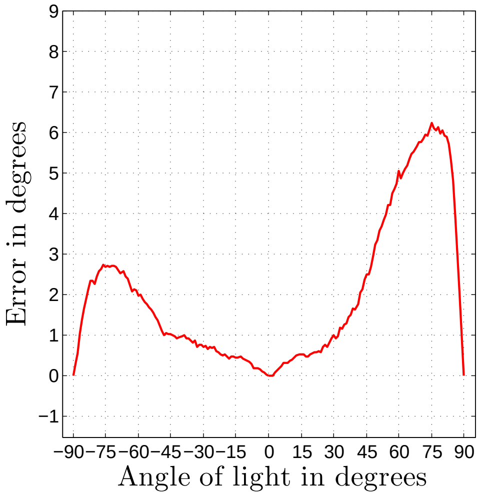


Figure 3.10: *Yaw1 Error*

The blue line indicates the ideal response while the black indicates the true response of the photo-diode. The right side of Figure 3.9 deviates more from the ideal response because the top solder connection cast a shadow on the diode on that side. The solder connection can be seen to the left on the photo-diode in Figure 3.7. The deviation results in an error in angle estimation, which is shown in degrees in Figure 3.10.

Photo-diode Yaw3 made the maximum error, of 8.7 degrees at $\alpha = -75$. At this angle of α the light will fall on sensor Yaw2 at 15 degrees which gives an error of 0.42 degrees. In comparing the angle by calculating the *arctan* of the ideal and true case, the difference is 8.6 degrees. Thus the maximum error made by the unit vector direction to the light source is 8.6 degrees.

The yaw spin test is repeated with the new photo-diodes. Detailed graphs can be seen in Appendix B. In comparing Figure 3.6 to Figure 3.11, the angle of the light source is much more accurately calculated using the same algorithm as used with the solar cells.

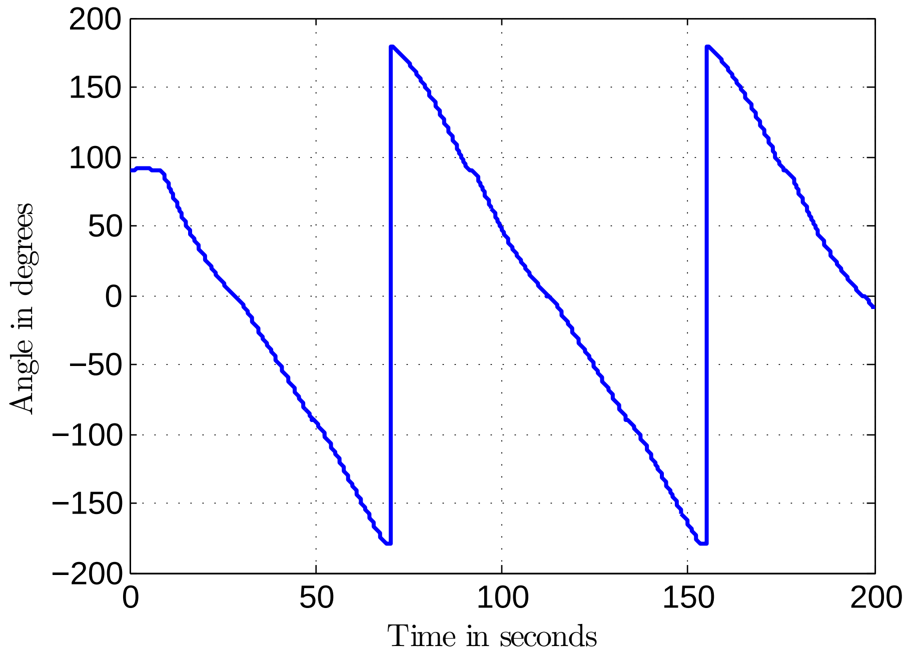


Figure 3.11: *Yaw Angle of Light Source with regards to EyasSAT BRF.*

3.2 Camera

More sophisticated sensors than the CSS are required for very accurate attitude control. A fine sun sensor can have an accuracy of 1 degree or less and a star camera can have an accuracy of less than 0.001 degrees. A camera is added on EyasSAT to have this level of accuracy. The camera is used to detect two LEDs which are located at a fixed location in the room. The LEDs act as a reference point for EyasSAT such as stars for a star camera or the sun for a fine sun sensor. Only one LED is required to generate a unit vector for the attitude determination, movement in two axes can only be detected whilst observing one LED. Since the camera is mounted in the Y_b-axis the yaw and roll motion can be detected and measured. However, the pitch cannot be detected as the LED will stay in the same position with regards to the image sensor. Another LED was added to allow for simple light tracking control to estimate the pitch in observing the orientation of the two LED's with regard to one another.

3.2.1 Camera Module

The AA9653 digital sensor module using OmniVision's CMOS image sensor: OV9121 CMOS (1.3 MPixel) CameraChip was used in this project. The camera module has a

1280 × 1024 image array. Figure 3.12 shows the column and row definition for the camera as well as the image sensor axes definition. The image array is made out of square pixels with dimensions of 5.2 × 5.2 μm, creating an image area of 6.66 × 5.32 mm. The image array has an on-chip 10-bit Analogue to Digital Converter (ADC) and the module can operate at up to 15 Frames Per Second (FPS). To simplify the design only the eight most significant bits were used. The two unused bits were not needed as only light detection is done and 8-bits ADC is sufficient for this purpose. The camera module and part of the image sensor datasheet is shown in Appendix D.2.

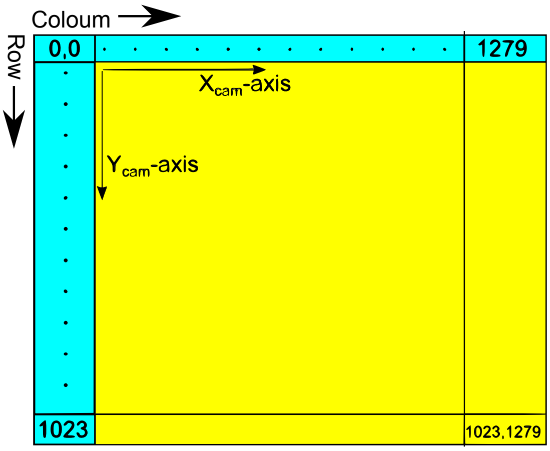


Figure 3.12: *Image Sensor Pixel Array Row, Coloum and Axes Definition.*

3.2.2 Camera Lens

The camera is mounted horizontally on the ADCS_V2 PCB facing the open side panel of EyasSAT. The module was fitted with a 30 degree FoV lens. The camera can measure the position of the LEDs with an accuracy of 0.36 degrees and in turn the attitude of EyasSAT when the LEDs are in the FoV of the camera.

3.2.3 The Camera Circuit Layout

The camera outputs 8-bit pixel data on parallel digital output ports. The settings of the camera can be adjusted through an I2C communications bus, which will be directly connected to the MCU. However, the MCU cannot process the 8-bit data of the camera fast enough and does not have enough memory to store the image. In order to process and store an image, a FPGA and SRAM is used. The FPGA will capture and store the pixel data on the SRAM. The MCU will be able to command the FPGA to capture an image. Then the FPGA will respond when the image has been captured and stored. Additionally, the FPGA will also have the task of finding the LED's pixel position in the image. After the image has been captured and the light detection is done, the MCU will request the data of the position of the two LEDs. The lens disturbance correction will be performed on the MCU. The position of the LEDs can be indicated by the generation of a unit vector in the BRF after distortion correction. The Figure 3.13 shows the layout of the camera circuit with the FPGA, SRAM and MCU. This architecture was obtained from [19] and [20].

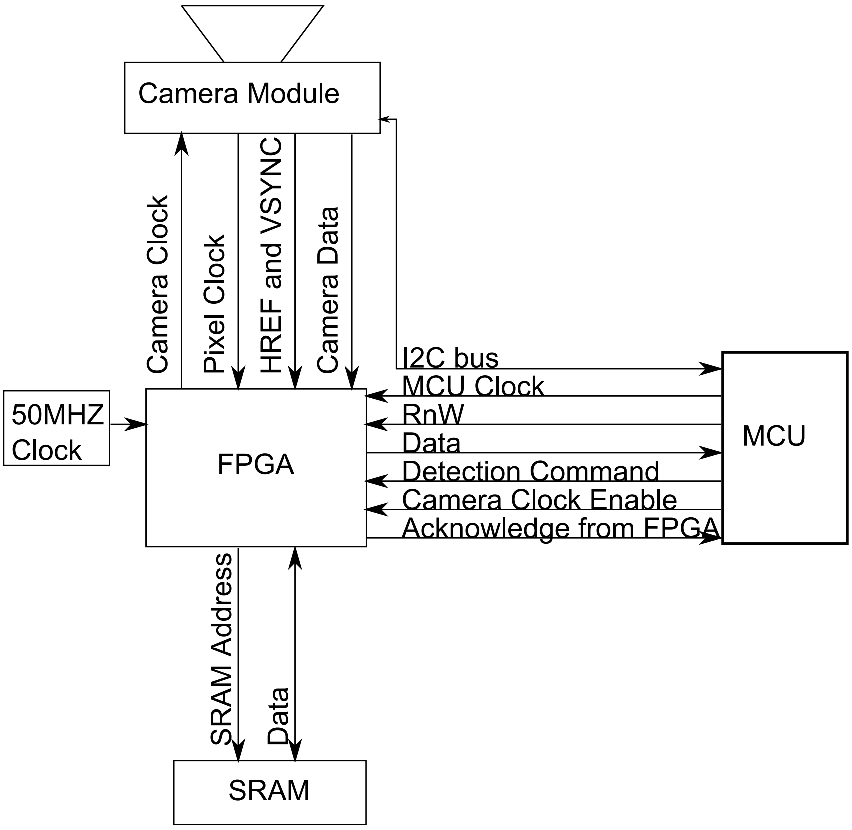


Figure 3.13: *Camera Circuit Layout*

3.2.4 Camera Calibration

A MATLAB Camera calibration toolbox was used to correct for the lens distortions [21]. The intrinsic parameters of the camera are calculated by the calibration toolbox. These parameters describe the camera matrix, as defined in the pinhole camera model, and is used to calculate the distortion model. Brown in 1966 introduced the distortion model which was called the “Plumb Bob” Model [21]. The axes and pixel numbering convention used by the toolbox is the same as for the image sensor as shown in Figure 3.12. The parameters obtained from the camera calibration toolbox is shown in Table 3.3.

Description	Symbol	Value
Focal length:	f = [X Y]	[1897.183 1888.09147] pixels ±[7.89964 7.83003] pixels
Principal point:	cc = [X Y]	[629.13570 481.60794] pixels ±[5.86314 4.27034] pixels
Skew:	α_c	(0.00088) 89.94952 degrees ± 0.01549 degrees
Distortion:	$\mathbf{k}_c = [r_1\ r_2\ t_1\ t_2]$	[-0.40932 0.44964 -0.00042 -0.00389 0] ±[0.01126 0.09598 0.00055 0.00044 0.00000]

Table 3.3: *Camera Calibration Results*

The vector $\mathbf{k}_c$ shown in Table 3.3 contains both radial and tangential distortion's first and second order coefficients. The skew coefficient defines the angle between the X_{cam} and Y_{cam} pixel axes. The angle is assumed to be 90 degrees, because the pixels are square. This implies that the skew coefficient should be zero as shown in Table 3.3. The lens distortion is shown in Figures 3.14 to 3.16.

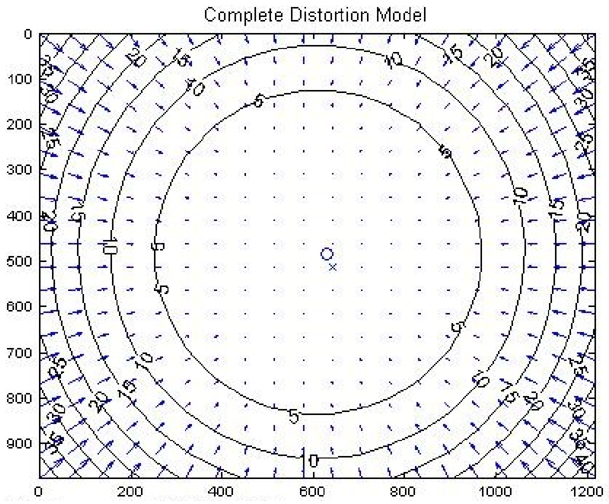
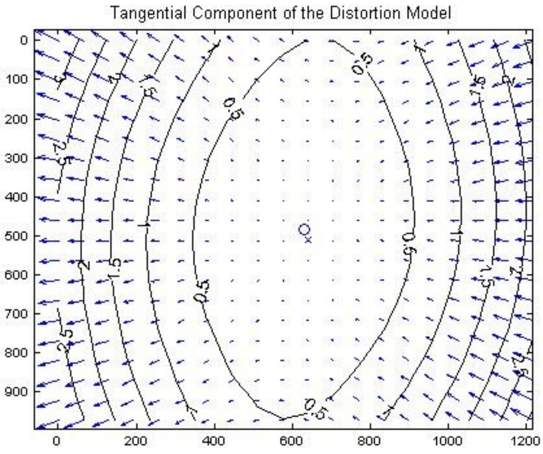


Figure 3.14: *Complete Lens Distortion Model*



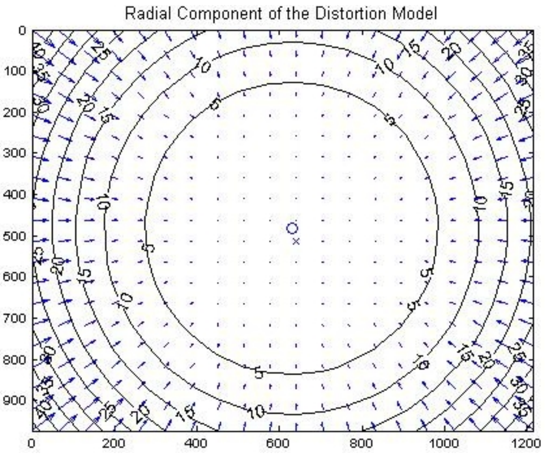


Figure 3.15: *Tangential Component*

Figure 3.16: *Radial Component*

The figures show the effect of the lens distortion on each pixel of the image. Each arrow represents the effective displacement of a pixel induced by the lens distortion. The maximum distortion takes place in the corners of the image and reaches a 40 pixel displacement. The largest distortion component is the radial distortion component which is demonstrated in Figure 3.16.

In order to correct the lens distortion, the following algorithm is applied to each pixel coordinate to correct the pixel displacement [22; 21]:

$$x^{'} = (x_{distorted} - cc_x)/f_x \tag{3.2.1}$$

$$y^{'} = (y_{distorted} - cc_y)/f_y \tag{3.2.2}$$

$$\tag{3.2.3}$$

where $x_{distorted}$ and $y_{distorted}$ are the original pixel position, cc_x and cc_y is the pixel position of the bore sight of lens and f_x and f_y is the focal length.

Let

$$r^2 = x^{'2} + y^{'2} \tag{3.2.4}$$

then

$$x_{undistorted} = x^{'}(1 - k_c[1]r^2 - k_c[2]r^4) - 2k_c[3]x^{'}y^{'} - k_c[4](r^2 + 2x^{'2}) \tag{3.2.5}$$

$$y_{undistorted} = y^{'}(1 - k_c[1]r^2 - k_c[2]r^4) - 2k_c[4]x^{'}y^{'} - k_c[3](r^2 + 2y^{'2}) \tag{3.2.6}$$

where $x_{undistorted}$, $y_{undistorted}$ is the correct undistorted pixel position and $\mathbf{k}_c$ is the vector containing the distortion coefficients, as shown in Table 3.3.

3.2.5 Unit Vector Generation

The camera is facing out of the negative Y_b-axis side panel of EyasSAT. The X_{cam}- and Y_{cam}-axis of the image sensor coincide with the X_b- and Z_b-axis of the BRF. The unit vector is generated by locating the pixel offset position from the boresight of the camera (principal point on the image sensor) in the X_b,Z_b plane of the BRF. The unit vector in the BRF can be calculated with equation (3.2.7), using the average of f_x and f_y as the focal length (f) and the offset of the pixels in the two axes from the principal point.

$$\mathbf{Cam}_{body} = \begin{bmatrix} \frac{P_{x1}}{\sqrt{f^2+P_{x1}^2+P_{z1}^2}} \\ \frac{-f}{\sqrt{f^2+P_{x1}^2+P_{z1}^2}} \\ \frac{P_{z1}}{\sqrt{f^2+P_{x1}^2+P_{z1}^2}} \end{bmatrix} \tag{3.2.7}$$

In equation (3.2.7), P_{x1} and P_{z1} is the LED pixel position from the principal point in the image sensor array in the body reference axes. Examples of the light positions and the definitions of axes are shown in Figure 3.17. The Y-component of the vector is the normalized negative of the average focal length because the camera is directed in the negative Y_b axis direction.

Both LEDs' positions are required to calculate roll, pitch and yaw angles. One LED will be tracked and the other will be used as a reference to track the pitch of EyasSAT. The position of the two LEDs from the principal point are described by $[P_{x1}, P_{z1}]$ and $[P_{x2}, P_{z2}]$. The Figure 3.17 illustrates the two LEDs' light on the image sensor and the description of the axes and principal point.

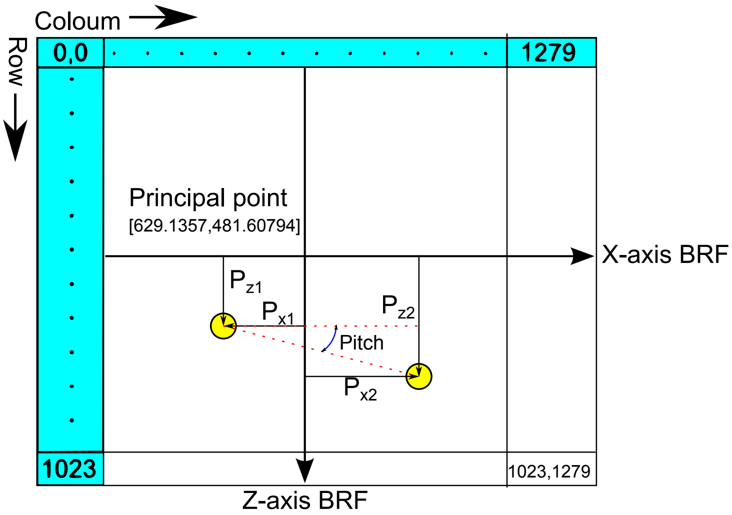


Figure 3.17: *LED Positions on Image Sensor*

The left LED with position $[P_{x1}, P_{z1}]$ is the source to which the unit vector points whilst the second LED is used to calculate the pitch angle of EyasSAT. The roll, pitch and yaw calculations are shown below.

$$Pitch(\theta) = \arctan\left(\frac{P_{z2} - P_{z1}}{P_{x2} - P_{x1}}\right) \tag{3.2.8}$$

$$Yaw(\psi) = -\arctan\left(\frac{P_{x1}}{f}\right) \tag{3.2.9}$$

$$Roll(\phi) = \arctan\left(\frac{P_{z1}}{f}\cos(\psi)\right) \tag{3.2.10}$$

The yaw and roll equations are obtained from trigonometry. Figure 3.18 indicates the focal point with the focal length, light source, pixel position and the unit vector to the light source.

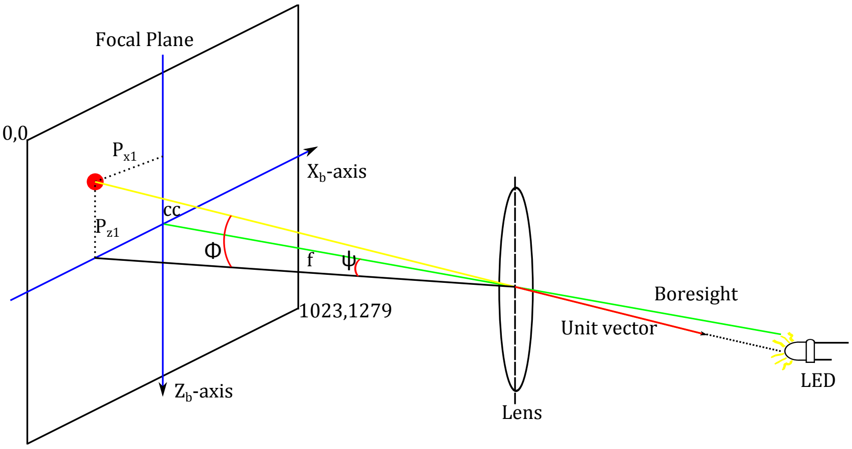


Figure 3.18: *Trigonometry for Calculating Yaw and Roll Angles*

3.2.6 Camera Accuracy Test

EyasSAT with the installed camera module was placed on the two axes rotating platform, as described in Appendix B.3. A LED was lined up and placed in the boresight of the camera. EyasSAT rotated, in 1 degree steps, in two axes while the camera output was monitored and recorded. The comparison of the actual change in degrees and the camera's estimated angle, determines the accuracy.

EyasSAT was first rotated about the Z_b-axis, simulating a yaw movement. Figure 3.19 the X-axis of the figure indicates the amount the platform was rotated in degrees. The Y-axis of the figure shows the camera estimated yaw angle of the platform. The figure shows an almost linear response, indicating that the error made by the camera is very small.

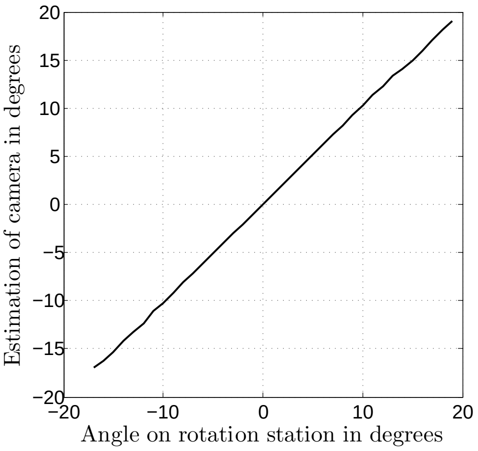


Figure 3.19: *Yaw Rotation*

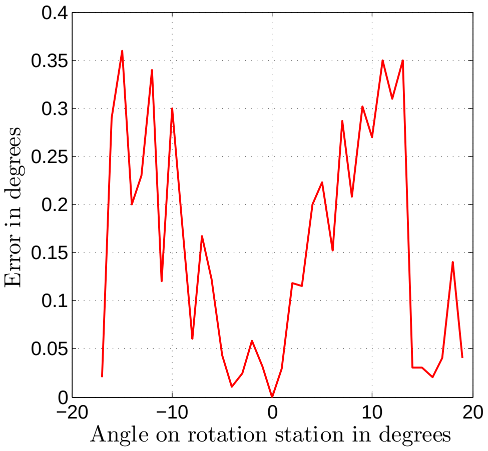


Figure 3.20: *Error in Yaw Estimation*

In Figure 3.20 the absolute error between the actual yaw angle and the camera estimated yaw angle is shown. As the angle increases in each direction the error also increases. The lens distortion increases to the edges of the lens as shown in Figure 3.14 to 3.16. As the

LED moves away from the bore-sight, where lens distortion is at a minimum, the error increases. The maximum yaw angle error is 0.36 degrees.

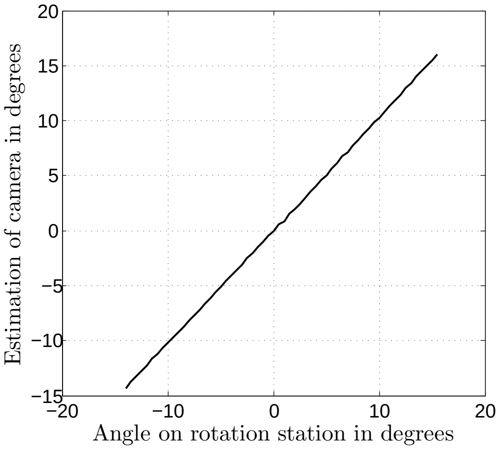


Figure 3.21: *Roll Angle*

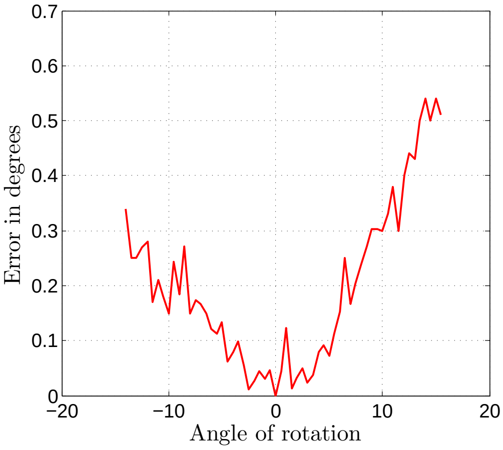


Figure 3.22: *Error in Roll Estimation*

The second rotation is about the X_b-body axis and simulates a roll movement. The true and estimated angles are shown in Figure 3.21. The roll angle error is shown in Figure 3.22. Similarly to the yaw rotation, the error increases as the light source moves to the edge of the lens. However in this case the error for a positive roll angle rotation has a much larger error than for a negative roll angle rotation. The maximum error is 0.54 degrees.

EyasSAT will be mostly manoeuvred around its Z_b-axis on the air bearing, performing yaw movements as disturbance torques will quickly saturate the RWs if a roll or pitch movement is attempted. The roll and pitch motion will rather be controlled to a 0 angle and 0 angular velocity. The camera will not operate regularly in a high roll state, therefore the large error in the roll angle of the camera will not have a great effect on the performance. The camera will be able to estimate the yaw attitude of EyasSAT to an accuracy of 0.36 degrees, when the LEDs' are in the FoV.

3.3 Inertial Measurement Unit

An (IMU) is added to EyasSAT in the form of the ADIS16375 from Analog Devices [3]. The IMU consist of two main sensors which are, the three-axis Microelectromechanical Systems (MEMS) angular rate sensor or gyroscope and the three-axis MEMS acceleration sensor.

As EyasSAT is tested on earth's surface, gravity is used as a reference vector for the ADCS. The accelerometer measures the gravity vector in the room and supply a unit nadir vector to the ADCS. The gyroscope supplies angular body rate information which is used in an Extended Kalman Filter (EKF) to estimate the bias offset of the gyroscope and the orientation of EyasSAT. The EKF is discussed in Section 6.1.3.

The IMU also has the function of numerically integrating the angular rates and supplying the change in angle at every sampling instance. These values can be accumulated on the MCU and provide an estimate of the attitude of EyasSAT. However, these angle measurements drift due to the integration operation on the noisy gyroscope output. As a result the accumulated angle diverges from the true angle over time. The accumulated angle will give a close approximation of the attitude of EyasSAT if the bias of the angular rate signal is estimated, and attempts are made to compensate for it. In comparing the accumulated angle with the estimated attitude, the accumulated angle can indicate whether the attitude determination is working or not. This proved valuable in the testing and debugging stages.

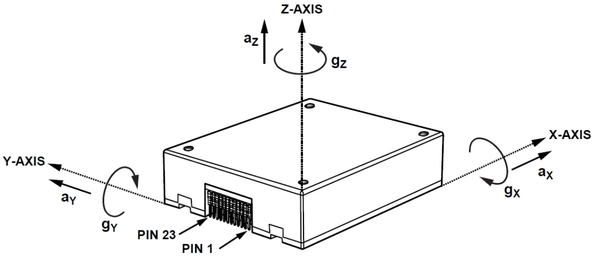


Figure 3.23: *IMU Direction Reference Frame [3]*

The IMU is too big to be placed inside EyasSAT and is placed on the mounting surface along with EyasSAT. A connection cord connects the IMU and ADCS module. The axes of the IMU are defined in the datasheet by the image shown in Figure 3.23. The IMU is placed so that the IMU axes align with the BRF.

3.3.1 Accelerometer

The accelerometer can supply information about the pitch and roll attitude of EyasSAT. The accelerometer output has a 16-bit resolution with 0.8192 mg/LSB. The smallest roll and pitch angle that can be detected by the accelerometer is calculated in equation 3.3.1.

$$\phi_{min}, \theta_{min} = \arcsin(0.8192\text{mg}) = 46.93671 \text{ m deg} \tag{3.3.1}$$

The accelerometer was placed on a level surface and the output monitored during calibration. Assuming that the gravity in the room was one g, the sensor was calibrated to output zero on the X- and Y-axis and one g $\div$ 0.8192 mg $= -1220.7$ units in the Z-axis. The output of the accelerometer is shown in Figure 3.24 and Figure 3.25 with no bias correction.

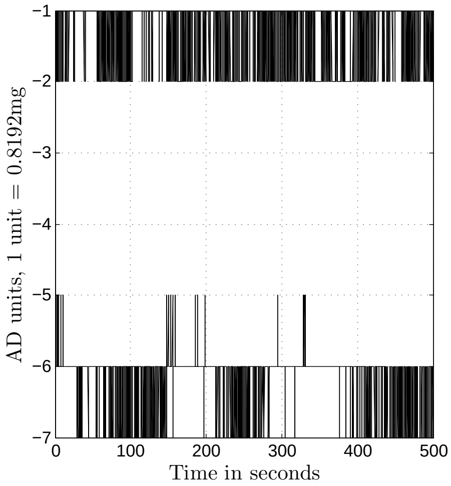


Figure 3.24: *X- and Y-Axis Output*

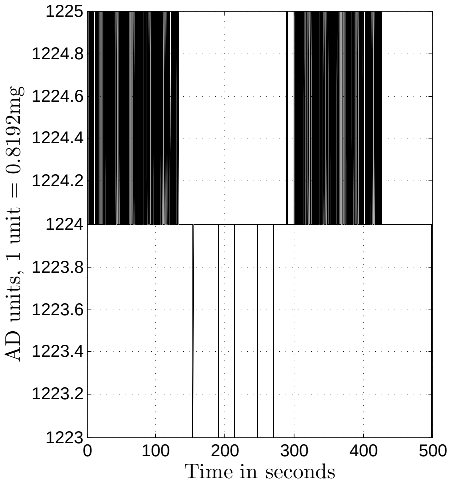


Figure 3.25: *Z-Axis Output*

The noise on the output of the accelerometer has a deviation of 1 unit, which is equivalent to 0.8192 mg. The bias on each axis is determined as +1 unit for the X-axis, +5 units for the Y-axis and -4 units for the Z-axis. The each axis output was also multiplied by -1 to indicate gravity in the downward direction ($-Z_b$-axis in BRF).

The body referenced nadir vector is obtained using bias correction and normalisation of the outputs as shown in equation (3.3.2) [11].

$$\mathbf{ACEL}_{body} = -1 \times \begin{bmatrix} X_{raw}+1 \\ Y_{raw}+5 \\ Z_{raw}-4 \end{bmatrix} / \sqrt{(X_{raw}+1)^2 + (Y_{raw}+5)^2 + (Z_{raw}-4)^2} \tag{3.3.2}$$

The roll and pitch angles can be calculated with equation (3.3.3) and (3.3.4) using the unit vector components.

$$Roll(\phi) = -\arcsin(ACEL_{body}[Y]) \tag{3.3.3}$$

$$Pitch(\theta) = \arcsin(ACEL_{body}[X]) \tag{3.3.4}$$

EyasSAT was placed on the two axis rotation platform and a roll rotation was done to test the accelerometer accuracy. The estimated angle and the real angle are shown in Figure 3.26 and indicates a linear response. The error is very small as predicted by the result of equation (3.3.1). The error made at every measurement is shown in Figure 3.27. The maximum error made is 0.042 degrees. The result for a pitch rotation was found to be the same as for the roll rotation.

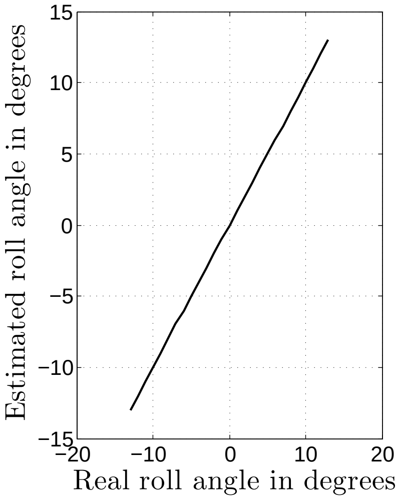


Figure 3.26: *Roll Angle Response*

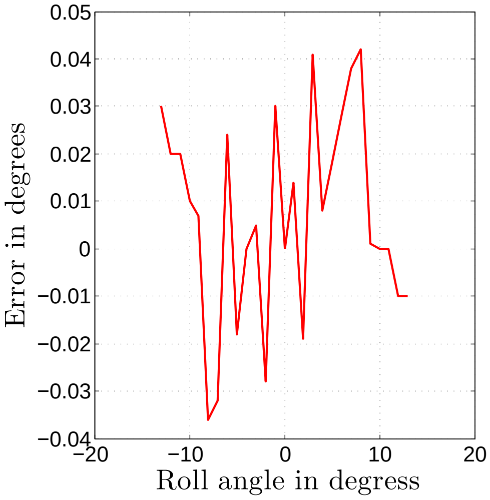


Figure 3.27: *Roll Angle Error*

3.3.2 Gyroscope

The gyroscope also has a 16 bit output and a resolution of 0.013108 deg/s per LSB. The gyroscope output for each axis is shown in Figure 3.28 to 3.30. The measurements were taken while the IMU was stationary and undisturbed on a level surface.

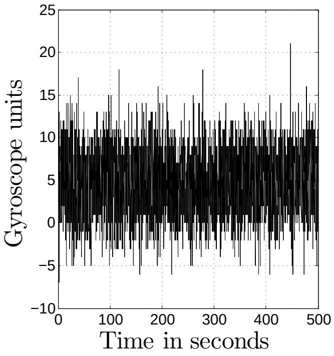


Figure 3.28: *X-Axis*

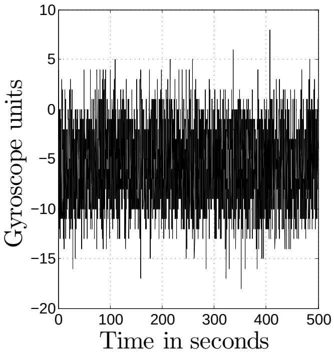


Figure 3.29: *Y-Axis*

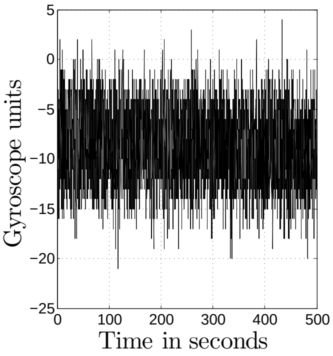


Figure 3.30: *Z-Axis*

The gyroscope output in each axis contains a bias offset. This offset will be estimated by the EKF and be compensated. The noise on each axis causes a peak to peak value of up to 11.8 units. This amount of noise translates to 0.155 deg/s or 0.00269 rad/s. The bias for each axis is estimated as $[4.8484, -5.5237, -8.7143]$. The IMU was placed on a turntable that can be commanded to turn at a certain rate to test the accuracy of the gyroscope. A turn rate of -2 deg/s and 2 deg/s were given and the Z-axis output of the gyroscope monitored. The output of the gyroscope, after bias correction and conversion to deg/s, is shown in Figure 3.31 and 3.32. The red line in the figure indicates the step amplitude.

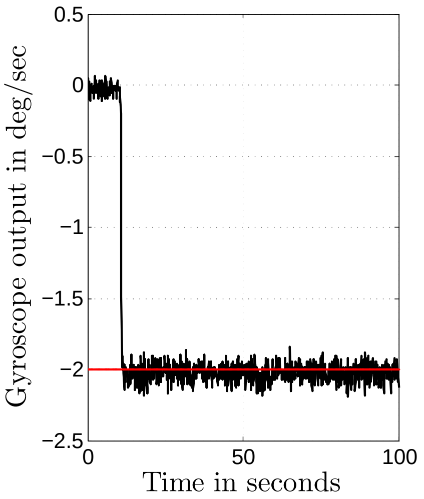


Figure 3.31: *Negative 2 deg/s Rotation*

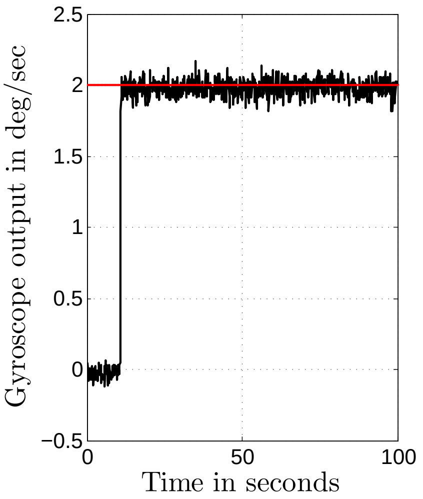


Figure 3.32: *Positive 2 deg/s Rotation*

As can be seen in Figure 3.31, a -2 deg/s step is done at 10s, before 10s the mean of the output is calculated at -0.0284 deg/s. After the step the mean of the output is -2.0013 deg/s. The gyroscope output accurately measures the -2 deg/s step with some noise.

In Figure 3.32 the mean before the 2 deg/s step is -0.0245 deg/s. The mean after the step is 1.9584 which is very close to -2 deg/sec. The gyroscope will supply the EKF with the system body rates in rad/s. The conversion from gyroscope units to rad/s is given in equation (3.3.5). The 0.013108 constant converts the gyroscope units to deg/s and is obtained from the datasheet.

$$Gyro_{rad/s} = Gyro_{raw} \times 0.013108 \times \frac{\pi}{180} \tag{3.3.5}$$

3.3.3 Accumulated angle

The IMU can be setup to generate an interrupt every sampling period (T_s=100ms). The IMU can output the change in body angles using this sampling period, which is the integration of the gyroscope measurements over the sampling period. The formula used for the X_b-axis is shown in equation (3.3.6) [3]:

$$\Delta\theta_x = \frac{\Delta T_s}{2} \times (\omega_{x,n} + \omega_{x,n+1}) \tag{3.3.6}$$

where $\omega_{x,n}$ is the gyroscope X-axis output at sampling instance n and ΔT_s is the sampling period. The output is given in 16 bit format with a sensitivity of 0.005493^0/LSB.

The delta angle output for each axis while the IMU was stationary is shown in Figure 3.33 to Figure 3.35. The red line in the figures shows the mean of each axis. These values will be used to do bias correction. The bias value for each axis are $[1.0478, -1.8808, -2.4921]$ units.

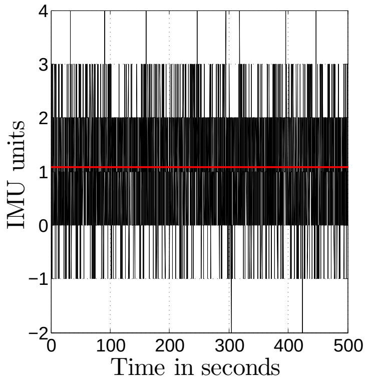


Figure 3.33: *X-Axis*

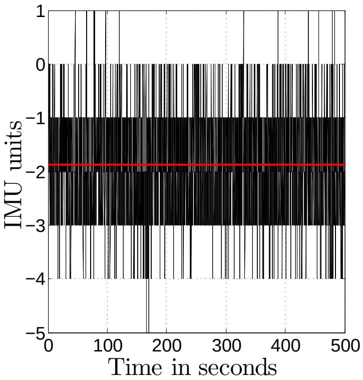


Figure 3.34: *Y-Axis*

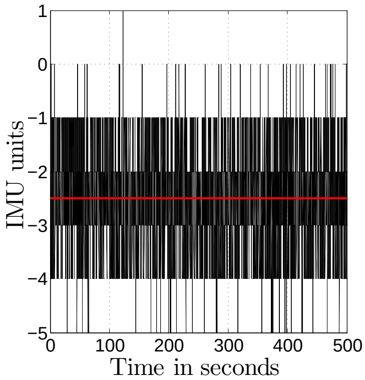


Figure 3.35: *Z-Axis*

The IMU was placed on a level surface in a stationary position and the outputs monitored whilst using the bias offset values for bias correction. The delta angles were accumulated and converted to degrees. The change in attitude from a starting point can be tracked by summing the delta angles in the MCU. The gyroscope output contains noise as shown in the previous section, which causes the accumulated angle to drift. The resulting attitude estimation is shown in Figure 3.36.

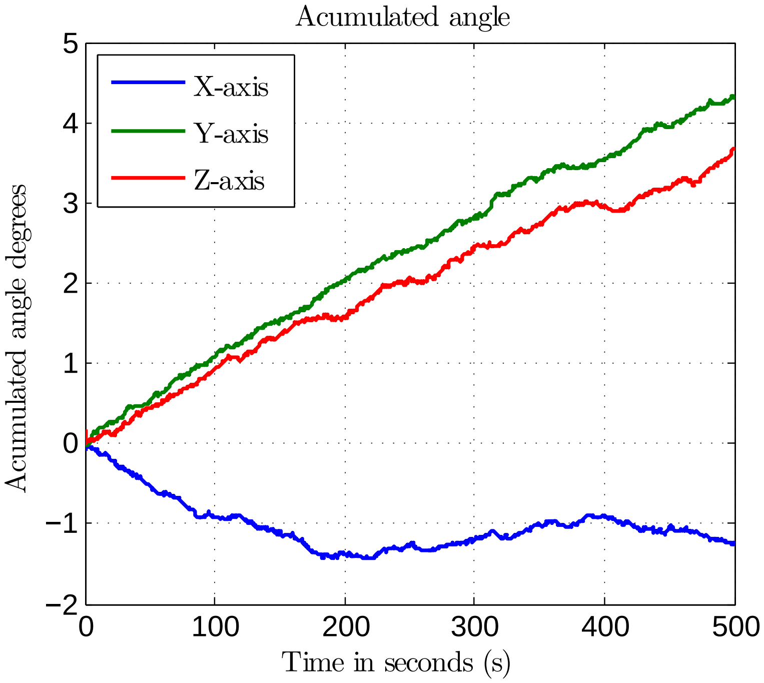


Figure 3.36: *Accumulated Angle in Degrees Around each Axis*

The IMU was stationary during these readings. The accumulated angle over 500 s, however, shows that the IMU is at an orientation of [-1.23, 4.301, 3.664] degrees in, Euler angles, from the starting orientation of [0,0,0]. This is a drift of almost 0.01 degree per second. The drift of the accumulated angle could be limited by keeping the ADCS air bearing test to a restricted time. Another approach is to reset the drift to 0, when the accumulated angle is reset to a known attitude.

3.4 Sensor Design Summary

This chapter concludes the description of the sensors used for attitude determination in this project. The sensors determine how accurately the attitude of EyasSAT can be estimated and in turn how accurately EyasSAT can be controlled to a certain attitude. A CSS, a camera, MEMS accelerometer and MEMS gyroscope were the sensors used.

The CSS is the most in-accurate sensor used and at times is the only sensor that supplies absolute yaw angle measurements. More information on the CSS can be found in Appendix B. The camera is the other sensor able to measure the yaw angle much more accurately but with a limited FoV. The accelerometer can supply accurate roll and pitch measurements. The gyroscope supplies body rates which are used in the EKF to propagate the attitude of EyasSAT.

Now that the attitude can be estimated using the sensors discussed, actuators are required to control the attitude. The actuators are discussed in the next chapter.

Chapter 4

Actuator Design

The mechanism utilized by a control system to manipulate or change the environment or system state is known as an actuator. In the case of satellites, the attitude actuators are used by the attitude control system to control the orientation of the satellite. Small nano-satellites have limited power, weight and available space, which makes the use of thrusters and control moment gyroscopes impractical. The actuators supplied with the original EyasSAT ADCS module is shown in Figure 1.3. However, this chapter will discuss the development of the actuators used in EyasSAT to form the ADCS_V2 module.

EyasSAT contains two magnetorquers, where one is in the X_b-axis and another in the Y_b-axis. Actuators in all three-axis are required for three-axis control, another magnetorquer was added in the Z_b-axis. These magnetorquers can be used for momentum dumping of the RWs or magnetic attitude control. The magnetorquers interact with the earth's magnetic field to generate a control torque.

EyasSAT also comes with a Z_b-axis RW mounted to the original ADCS module. Three new RWs were designed for three-axis control as the entire module was replaced. In comparison to the use of magnetorquers, RWs allow for fast and more accurate control. The RWs are the main actuators for EyasSAT.

4.1 Magnetorquers

The "CubeTorquer" is the new magnetorquer that was incorporated. This magnetorquer was previously developed in the ESL at the University of Stellenbosch. An image of the magnetorquer is shown in Figure 4.1. The datasheet for the magnetorquer is shown in Appendix D.3. The magnetorquers are accessed through the EyaBus by the ADC_V2 PCB and driven by motor drivers. The drivers used are the DRV8800 from Texas Instruments, and are the same drivers used for the RW motors. The drivers are contained on the ADCS_V2 PCB. The intensity and the direction of the generated magnetic field can be adjusted, using the Pulse Width Modulated (PWM) and direction input of the drivers.

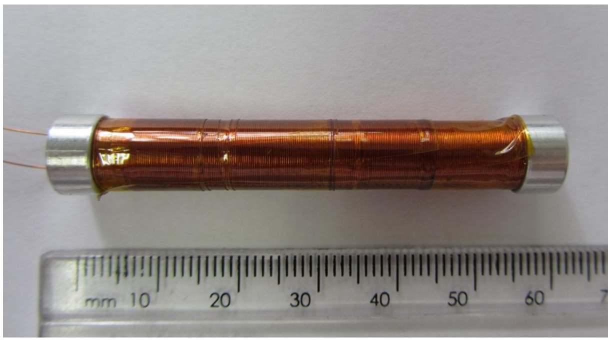


Figure 4.1: *CubeTorquer*

The new magnetorquer is mounted in the Z_b-axis on the same mounting structure used by the RWs and is shown in Figure 4.12. The important connections between the ADCS_V2 PCB and the magnetorquers, as indicated for one magnetorquer, are shown in Figure 4.2.

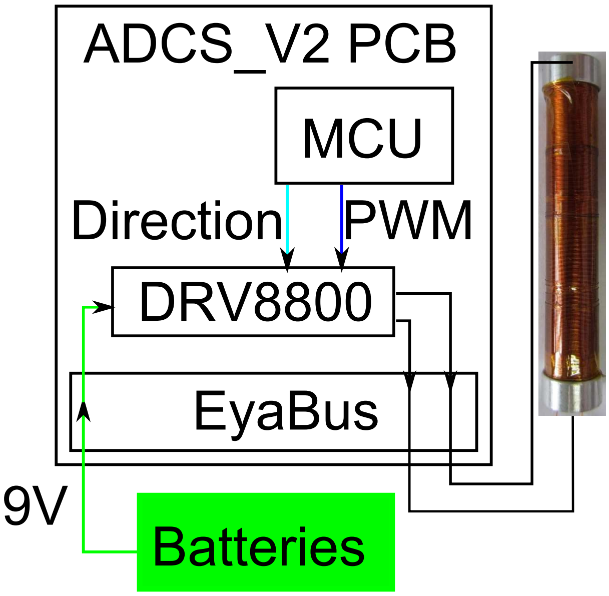


Figure 4.2: *Important Connection for Magnetorquers*

The magnetorquer drivers are powered directly from the batteries. In contrast 3.3 V logic signals, such as the PWM, and direction signals originate from the MCU. The magnetorquers are connected to the drivers through the EyaBus.

4.2 Reaction Wheel Design

The RWs consist of a number of parts which are the motor, flywheel and a PCB. The motor drives the flywheel, and the PCB holds al the electronics required to obtain the RW speed. This allows the speed of the RW to be controlled. Space constraints inside EyasSAT limit the size and capabilities of the RWs. The CAD design seen in Figure 5.8 and Figure 5.9 shows how the space was effectively utilised to obtain the RWs' size constraints. The power constraint placed on the RWs is that of the batteries. The batteries are rated at 9 V and 700 mAh. [1].

4.2.1 Motor

A brushed DC motor was used instead of a brushless DC motor in order to be more cost effective. The motor selected is the “A-max 22”, a 3.5 watt motor from Maxon motors [4] which is shown in Figure 4.3. The motor datasheet with the motor dimensions is shown in Appendix D.4. The motor has a long axis which is accessible from the back of the motor allowing for the placement of a magnet on the rear of the axis.

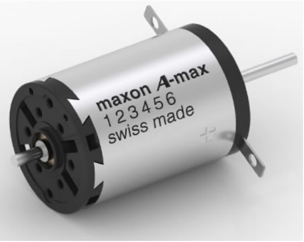


Figure 4.3: *A-max 22 3.5 Watt Brushed DC Motor [4]*

4.2.2 Flywheel

The amount of torque the RW can deliver is constant and determined by the motor used. However, the maximum momentum that can be stored by the RW can be changed. The equation for momentum is shown in equation (4.2.1). In this equation the symbol M is used for momentum, the symbol I for the RW inertia and ω for the RW speed. The equation indicates that the momentum storage can be increased by increasing the inertia and the RW top speed. As the RW speed is limited by voltage, the inertia can be increased. Inertia is added in the form of a flywheel, which is manufactured from brass, and attached to the motor axis. The flywheel is made up of an inner disc which is mounted onto the motor as well as an outer ring, attached to the disc, which covers some of the motor body. The inertia is determined by the mass of the material used and the distance the mass is located from the motor axis. Figure 4.4 shows the dimensions of the flywheel.

$$M_{Rw\_max} = I_{Rw} w_{Rw\_max} \tag{4.2.1}$$

The flywheels were manufactured by the University of Stellenbosch’s mechanical workshop. Due to physical limitations the flywheels could not be made exactly to specifications. The dimensions, mass and inertia of each flywheel is shown in Appendix C

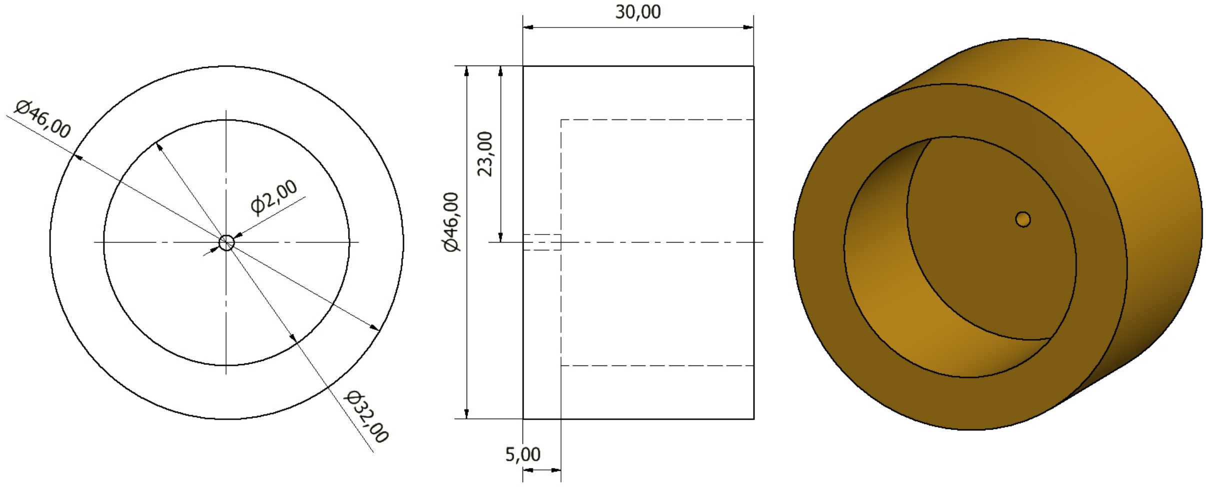


Figure 4.4: *Flywheel Dimensions*

4.2.3 Magnet Holder

A magnet is mounted to the axis of each motor and placed over a magnetic encoder described in Section 4.2.5. The magnet with the magnetic encoder is used for angular position feedback of the motor axis. A magnet holder was designed to hold the magnet in place. The holder's dimensions are shown in Figure 4.5.

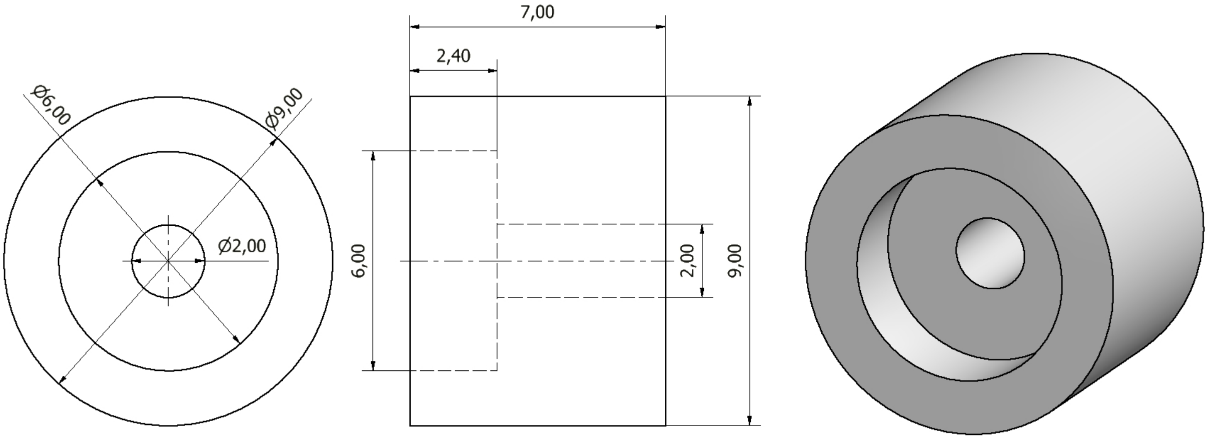


Figure 4.5: *Magnet Holder*

The holder will be mounted to the part of the axis that extends out of the back of the motor. The motor is then positioned over the magnetic encoder. In order for the magnet to be positioned in the desired position over the magnet encoder, a motor mount was designed, and the design will be discussed in Section 4.2.6.

4.2.4 Reaction Wheel Inertia

The detailed calculations for the total inertia of each RW is shown in Appendix C.2. The results are shown in Table 4.1.

Axis	**Inertia** $[kg.m^2]$
X_b	8.2738×10^{-5}
Y_b	7.5354×10^{-5}
Z_b	7.6130×10^{-5}

Table 4.1: *Inertia of Each RW*

4.2.5 Reaction Wheel PCB

The RW's motor driver and magnetic encoder is placed on a PCB and mounted beneath the motor to form a complete RW module. The RW PCB receives power directly from the batteries. Regulated 3.3 V and logic control signals are received from the ADCS_V2 PCB.

The drivers selected to drive the motors are the DRV8800 drivers from Texas Instruments [23]. These drivers accept a PWM input that regulates the voltage output to the motor and a direction logic input that can control the rotation direction of the motor.

The magnetic encoders used are the AS5046 magnetic encoders from Austria Micro Systems, a part of the datasheet is shown in Appendix D.5 [24]. The sensor detects the magnetic field vector generated by a diametrically magnetized magnet rotating through 360 degrees, and encodes the magnetic vector pointing angle in a 10-bit format. The communication and logic signals to the motor driver and magnetic encoder are routed to a header on the RW PCB. This allows the RW PCBs to be attached to the ADCS_V2 PCB through the use of ribbon cables. The RW PCBs were manufactured at the University of Stellenbosch. CAD generated images of the RW PCB design, as seen from the top and bottom, are included in Figure 4.6 and 4.7. In addition the track layout is shown in Figure 4.8.

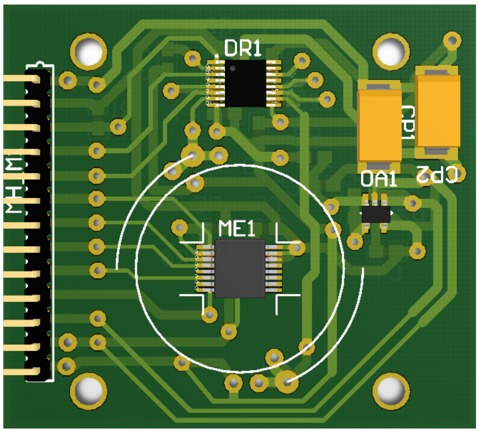


Figure 4.6: *RW PCB Top View*

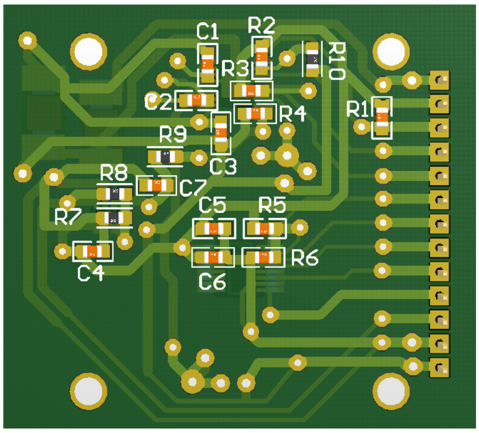


Figure 4.7: *RW PCB Bottom View*

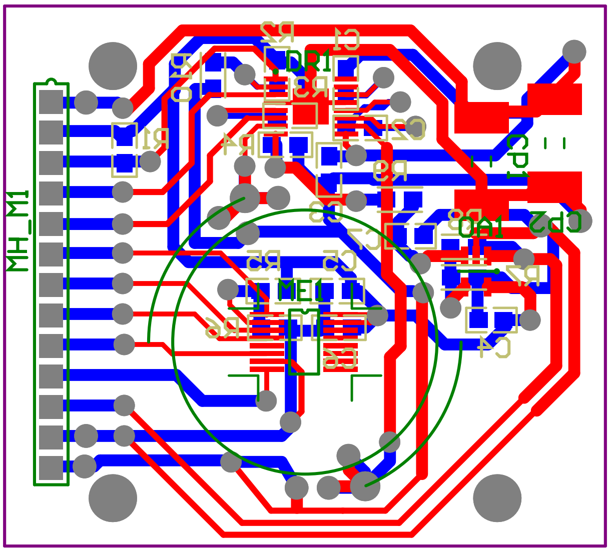


Figure 4.8: *RW PCB Track Layout*

Figure 4.9 illustrates the important logic connection of the RW PCBs. The 9 V is obtained directly from the batteries and the regulated 3.3 V is obtained from the ADCS_V2 PCB. Due to the noise on the ground track induced by the motor, the ground of the motor driver and the encoder is kept separated.

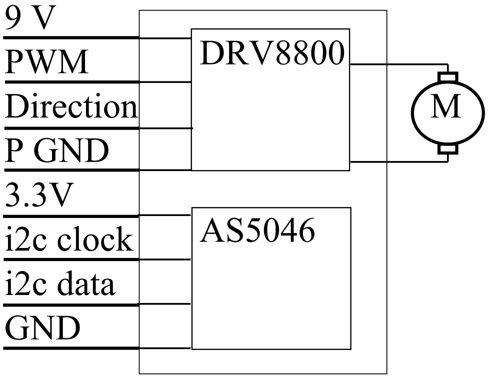


Figure 4.9: *Circuit Diagram for RW PCB*

4.2.6 Reaction Wheel Integration

The motor is mounted to a mounting plate, using PCB spacers, and placed over the RW PCB. This allows for the accurate alignment of the magnet over the magnetic encoder. The CAD design of the mounting plate can be seen in Figure 4.10.

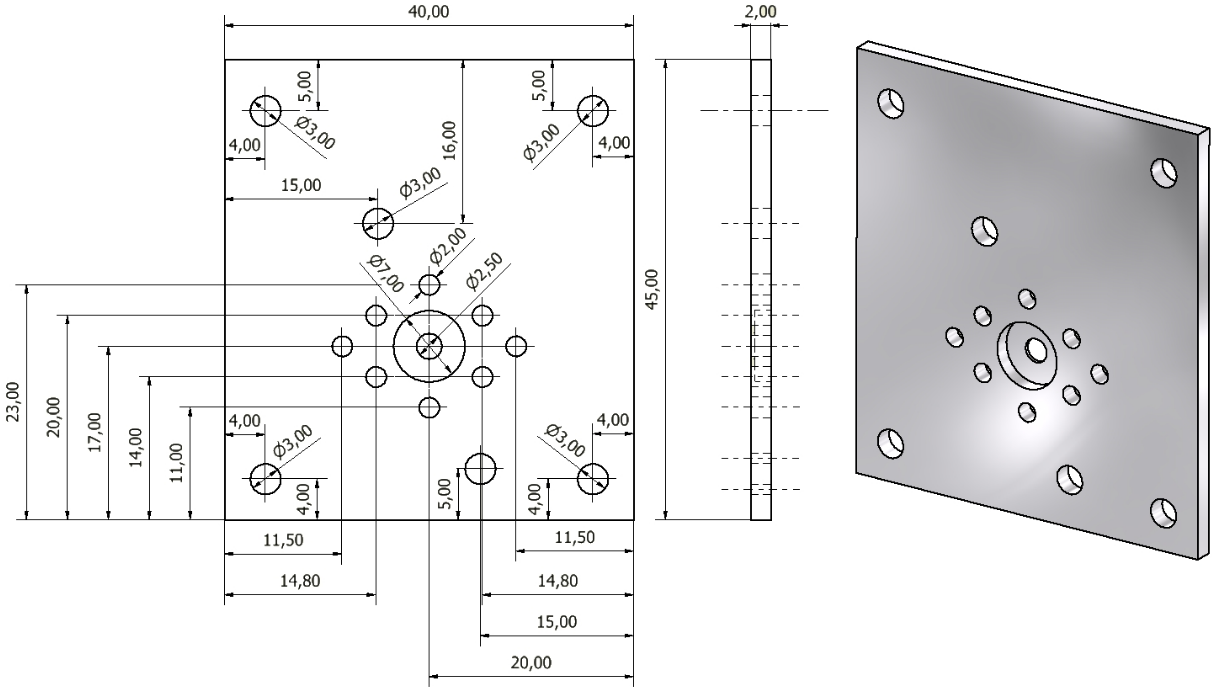


Figure 4.10: *Mounting Plate of Motor*

The complete assembly is shown in Figure 4.11.

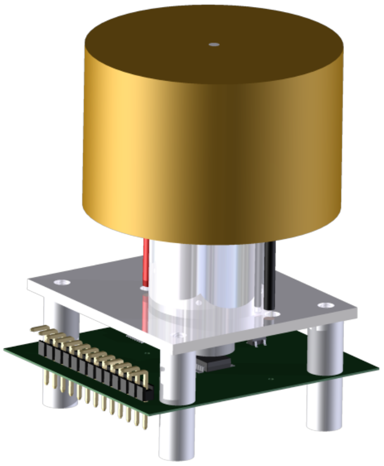


Figure 4.11: *RW Assembly*

Three of these RW modules were manufactured and installed in EyasSAT. To mount the RWs in EyasSAT, a mounting plate had to be designed. The plate is shown in Figure 4.12 with the RWs mounted on it. The mount also houses the Z_b-axis magnetorquer.

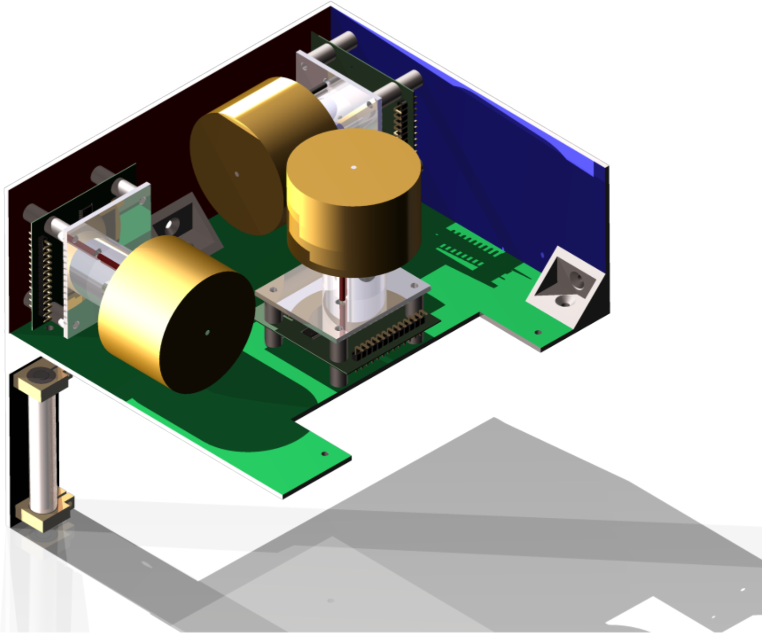


Figure 4.12: *Complete Assembly*

4.2.7 Speed Feedback

The RW PCBs discussed in Section 4.2.5 will be utilised with the ADCS_V2 PCB to determine the speed of the RWs using the motor axis position feedback obtained from the magnetic encoders. The RWs' speed is required for speed control which is discussed in Section 4.2.8. The next section discusses how the speed feedback can be obtained from the position feedback.

The speed of a RW is obtained from the angular position feedback of the motor axis. The speed can be obtained by differentiating the position feedback, as shown in equation (4.2.2).

$$v = \dot{r} = \frac{dr}{dt} = \frac{\Delta r}{\Delta t} \text{ rad/s} \tag{4.2.2}$$

The position is sampled at a fixed sampling period ΔT_s, which allows equation (4.2.2) to be written as indicated in equation (4.2.3). The following section discusses how the required sampling period is determined.

$$v = \dot{r} = \frac{dr}{dt} = \frac{\Delta r}{\Delta T_s} \text{ rad}/T_s \tag{4.2.3}$$

4.2.7.1 Position Sampling

In order to obtain the change in angular position of the motor axis (Δr), the position needs to be sampled fast enough to acquire at least two samples in one rotation. The shortest revolution time was calculated to ensure that two measurements can be taken in one revolution. The shortest revolution takes place when the RW is at top speed. The maximum power delivered to the RWs is 10V with freshly charged batteries. According to the motor datasheet the motor can reach a speed of 4500 rpm given 10 V. One revolution was calculated to be completed every 13.33 ms, given the maximum rotation speed.

The axis position must be sampled at least twice in one revolution time period to obtain an estimate of the angular speed. This implies a sampling time of 6.667 ms or less needed to be chosen, and a sampling time of 5 ms was selected. The motor was given 10 V as input and the magnetic encoder sampled at 5 ms. The position feedback of the encoder is shown in Figure 4.13.

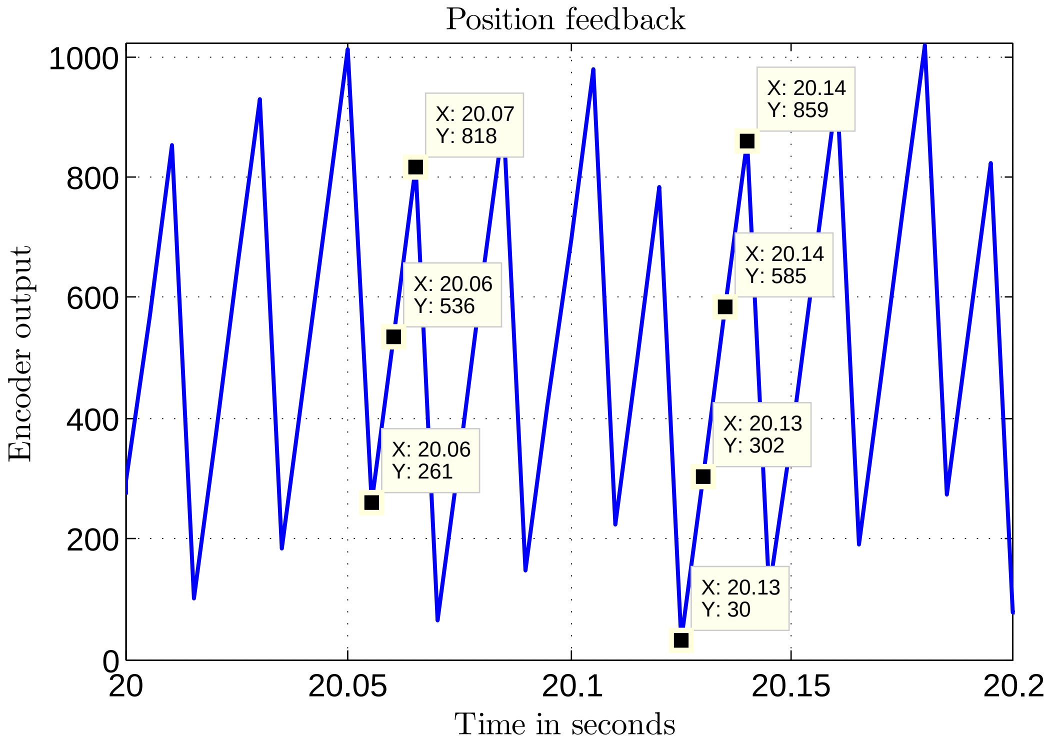


Figure 4.13: *Motor Position Samples at 100% PWM with 10 V Input*

Figure 4.13 shows the revolutions of the RW when reaching a steady state speed. The data points shows where the samples were taken, indicating 3 to 4 samples per revolution. Thus the sample period is short enough to produce accurate speed estimation even at maximum speeds.

4.2.7.2 Speed Estimation

The change in angular position is required as shown in equation (4.2.3) to estimate the speed. The angular distance travelled in one sampling period, was calculated by subtracting the latest position from the previous sampled position. The motors turning direction is indicated by the positive or negative sign of the result. The angular distance travelled should be divided by the sampling time, which remains a given constant, to obtain the speed in encoder units per sampling period. The angular distance travelled, during one sampling period, is directly coupled to the speed and, will be used as a speed measurement to avoid unnecessary computation.

The speed measurement is filtered by summing a number of samples together in order to decrease noise. A speed measurement over 100 ms results from taking 20 angular speed measurements added together. The speed is calculated every 100 ms, or after 20 samples. The speed control loop is then able to run at a sampling time of 100 ms.

The speed can be converted to rpm or rad/s by multiplying with the appropriate scaling constant. Theses constants are calculated in Section C.3.1 and C.3.2. The results are shown in Table 4.2.

Constant	**Resulting units**
$\times 0.586$	rpm
$\times 0.0614$	rad/s

Table 4.2: *RW Speed Conversions*

4.2.7.3 Speed Feedback Noise

The speed control is implemented using the change in angular distance in a 100ms period as a speed measurement. The speed reference value for the speed controller will be in rad/s. However, this will be converted using the conversions constants defined in Table 4.2. One of the RWs was given a constant voltage step of 9V, and the speed response is shown in Figure 4.14. The speed response contains noise with a peak to peak value of about 8 units, which translates to noise of about 4.69 rpm or 0.491 rad/s.

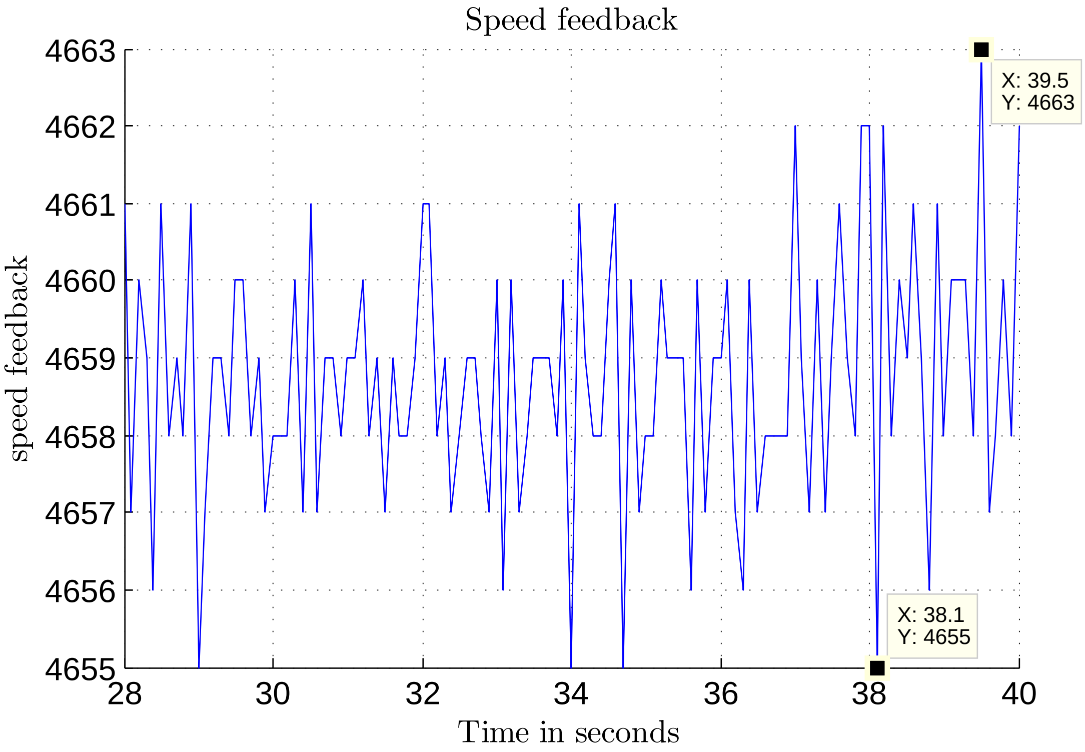


Figure 4.14: *Speed Feedback in Encoder Units per Sampling Period*

4.2.8 Reaction Wheel Speed Control

The speed control design is discussed with examples and values obtained from the RW in the Z_b-axis. Although the other two RWs have slightly different plant transfer functions, the calculations remain the same.

4.2.8.1 Plant Identification

The classical control design approach was taken for the design of the speed controllers for the RWs. Voltage steps were given to the RW and the speed output monitored in order to identify the plant characteristics. The purpose of this approach was to obtain a transfer function that represents the relationship between the voltage input to speed output characteristics. The expected response of the motor speed output, given a voltage step input, is modelled as a first order transfer function shown in equation (4.2.4).

$$G_{ol}(s) = \frac{Speed}{Voltage} = \frac{K}{s\tau + 1} \tag{4.2.4}$$

Where K is the gain of the transfer function and τ is the 63% time constant. This can also be expressed as:

$$G_{ol}(s) = \frac{Speed}{Voltage} = \frac{KT}{s + T} \tag{4.2.5}$$

where

$$T = \frac{1}{\tau}$$

Multiple step responses were obtained by generating voltage steps using a PWM signal and monitoring the speed output. These step responses were then used to derive a transfer function. Ideally the RW will have a linear response resulting in one transfer function fitting all the different step responses. However, this was not the case as the motor contained non-linearities, such as friction, which prevented the linear increase in voltage to result in the expected linear increase in speed.

Two forms of friction, namely static friction and kinetic friction, were contained in the motor. Static friction causes the motor to "stick", when the motor has been stationary for a period of time. A certain amount of voltage is required to overcome the static friction to allow the motor to start revolving. Once the motor is turning kinetic friction is experienced, which is dependent on the speed and direction of the motor rotation. The static friction force is larger than the kinetic friction force. Therefore, the voltage required to overcome the static friction is larger than the voltage required for overcoming the kinetic friction force.

The voltage required by each motor to overcome the static friction was determined and is represented in Table 4.3. This voltage offset was added to the control signal, if the signal was not zero, to more easily overcome the static friction.

RW axis	**Voltage value**
X_b	0.972 V
Y_b	0.810 V
Z_b	0.981 V

Table 4.3: *Voltage Requirement of each RW to overcome Static Friction*

The flywheel attached to the motor axis also experiences friction with air, which increases as the angular speed increases. The effect of air friction on the flywheel causes a decrease in the gain of the plant as the speed of the RW increases. This effect is shown in detail in Appendix C.4.

One transfer function is required to represent the plant in the speed control design. The transfer function obtained with the largest gain was chosen to represent the plant. This decision was made based on the assurance, that the plant will respond as per the design specifications when the plant has a large gain. In the case where less gain is experienced, the plant will respond more sluggish or damped but will still be stable. The chosen transfer function for the plant of the Z_b-axis RW is shown in (4.2.6).

$$G_{OL}(s) = \frac{178.7}{s + 0.2083} \tag{4.2.6}$$

4.2.8.2 Control of Plant

The aim is to control the speed of the RW in order to follow a speed reference with zero steady state error. The direct discrete design method was used as the control algorithm will be implemented using a microcontroller. The s-plane plant open-loop transfer function shown in equation (4.2.6) can be converted to the z-plane discrete ZOH equivalent plant, given a sampling time of 100 ms as shown in equation (4.2.7) [25].

$$G_{OL}(z) = \frac{17.68}{z - 0.9794} \tag{4.2.7}$$

A zero steady state error is required for the RW speed controller and a 2 s settling time was attempted with an over-damped response. The closed loop poles were calculated from these requirements. The detailed calculations are shown in Appendix C.4.1. The resulting desired closed loop poles are given in equation (4.2.8).

$$s_{cl} = -2 \pm j0.9686 \tag{4.2.8}$$

The controller design is done directly in the z-plane. The desired closed loop s-plane poles can be converted to desired closed loop z-plane poles using the transformation in equation (4.2.9):

$$z_{cl} = e^{s_{cl}T_s} \tag{4.2.9}$$

where T_s is the sampling period of 100 ms.

The z_{cl} poles were calculated and shown in equation (4.2.10):

$$z_{cl} = 0.8149 \pm j0.0792 \tag{4.2.10}$$

The PID controller or compensator implemented in the z-plane has the form shown in equation (4.2.11).

$$D(z) = \frac{K(z - a)}{z - 1} \tag{4.2.11}$$

4.2.8.3 Controller Design

The only two unknown constants are the gain K and the zero position a. The value of a can be determined using root locus design with the angle condition to allow the root locus to pass through the desired closed loop poles. The magnitude condition can then be used to obtain the gain K by placing the closed loop poles on the desired positions on the root locus. The detailed calculations to obtain the values for a and K are shown in the Appendix C.4.2. The resulting controller transfer function is shown in equation 4.2.12.

$$D(z) = \frac{0.01977376(z - 0.8840542)}{z - 1} \tag{4.2.12}$$

The control differential equation shown in equation (4.2.13) was implemented on the MCU.

$$u(k) = 0.01977376e(k) - 0.01748108e(k-1) + u(k-1) \tag{4.2.13}$$

4.2.8.4 Control Validation

The RW was given a number of inputs and the response measured in order to validate the speed control design. Figure 4.15 and Figure 4.16 shows the response of the RW to a 100 and 1000 rpm step command and demonstrates the settling time and overshoot response.

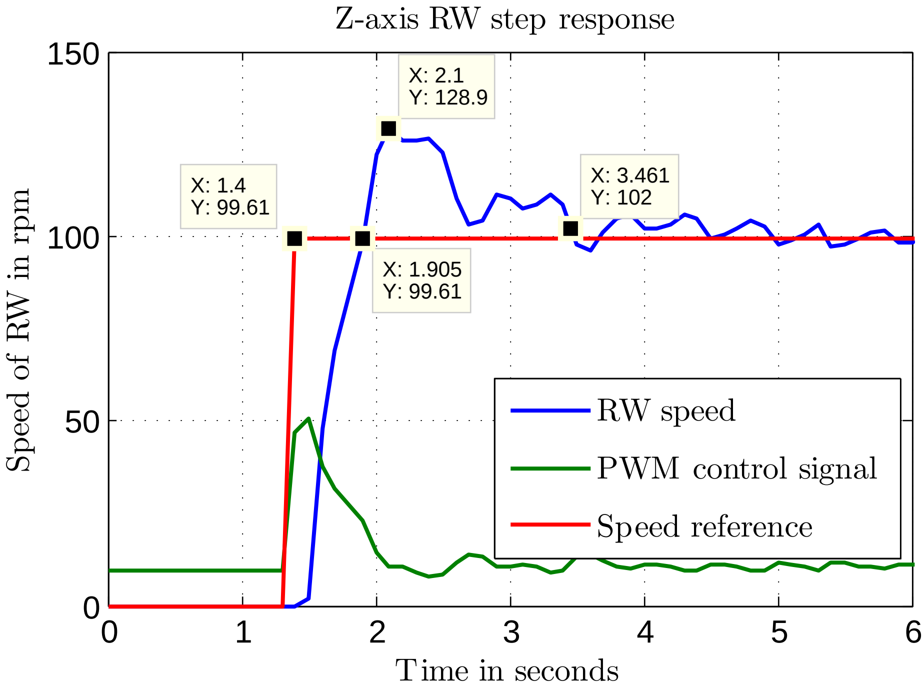


Figure 4.15: *Z_b-Axis RW Step Response to a 100 rpm Speed Reference*

In Figure 4.15, the first data marker indicates that the step command starts at 1.4 seconds. The response shows a 0.506 seconds rise time with 28.9% overshoot and about a 2 s settling time. As demonstrated in Appendix C.4, the plant has more gain at lower speed and a large overshoot results. The response also oscillates around the reference for some time because of the high gain of the plant.

Figure 4.16 shows a larger step that causes the control signal to saturate, resulting in a longer rise time of 2.5 seconds. Due to the speed increase the gain of the plant decreases

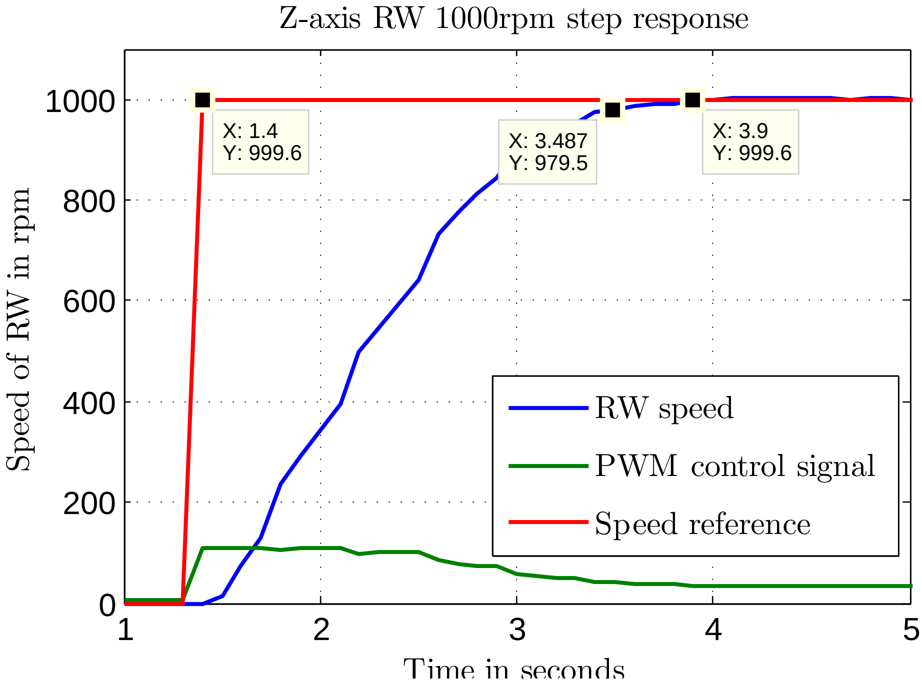


Figure 4.16: Z_b*-Axis RW Step Response to a 1000 rpm Speed Reference*

and no overshoot was obtained. The response has a 2% settling time of 2.1 seconds. The response does settle to the reference speed with some noise and almost no oscillation when compared to the 100rpm step. More input responses are shown in Appendix C.5. The response of the RW speed control was deemed acceptable and the other RWs displayed similar responses.

4.3 Actuator Design Summary

A Z_b-axis magnetorquer was added to EyasSAT allowing for three-axis magnetic control. There were also three RWs designed for fast three-axis control, which are the main actuators used in this project. The speed control of the RWs was designed to have a 2 s settling time with an over-damped response. More information on the RW design can be found in Appendix C.

This chapter concludes the design of the actuators in the project. Thereafter, the sensors and actuators are integrated into EyasSAT. The integration using a new PCB is discussed in Chapter 5.

Chapter 5

Hardware Integration

5.1 ADCS_V2 PCB

The ADCS_V2 PCB contains all the components required to read the sensor data and command the actuators for the ADCS. The PCB also interfaces with EyasSAT through the EyaBus and uses the fast wireless communications module to communicate with a ground station (PC with wireless receiver). The PCB needs to be designed following the EyasSAT PCB design specifications to integrate with EyasSAT. The EyaBus needs to be included and utilized to communicate with the rest of the EyasSAT subsystems. The EyasSAT owner's manual supplies guidelines for designing a PCB that can interface with the other EyasSAT modules [26]. The extract of the guidelines can be seen in Appendix D.6. The data handling module on EyasSAT has the ability to request data from the ADCS_V2 module through the Serial Peripheral Interface (SPI) communication established through the EyaBUS. This data can be added to the telemetry which is transmitted through the communications module. The data handling module will also be able to send commands to the ADCS_V2 module through the EyaBus. Only the fast wireless module is needed for telemetry and is used by the new ADCS_V2 module for the purpose of attitude determination and control. Therefore the SPI communication to the data handling module was not implemented.

The PCB contains a number of components such as voltage translators, opamps and other Integrated Circuits (IC) that allow for communication and logic signals to be transmitted between different components. However, only the main components will be discussed. The most important parts of the PCB are the power supply, microcontroller, camera sensor architecture (as described in Section 3.2.3), the connections to external elements and interface with the EyaBus.

5.1.1 Power Supply Architecture

The EyasSAT EPS module allows access to, switch enabled, regulated power lines through the EyaBus. The regulated switch enabled power was not used as suggested by the owner's manual. The other subsystems do not have the ability to turn off the power to the ADCS_V2 module as it is directly powered from the batteries and regulates its own power. This allows the ADCS_V2 module to function individually with a DC power connection supplying 9 V. This allows for easy assembling, testing and debugging of the ADCS_V2 module without integrating the PCB with EyasSAT. To allow the PCB to function apart from the EyasSAT stack it needed its own power regulators. The different voltage levels and the main components that require them are shown in Table 5.1.

Voltage Level	**Component**
1.5V	FPGA
2.5V	Camera module
3.3V	MCU, FPGA GPIO, Magnetic encoders, Motor driver logic
5V	EyaBus logic level
9V	Motor drivers

Table 5.1: *Voltage Rail Requirements*

The circuit is designed so that the 9 V can be supplied by an external DC power source or by the batteries on EyasSAT through the EyaBus. A jumper is used to select which power source to use. Figure 5.1 shows the block diagram of the power supply circuit.

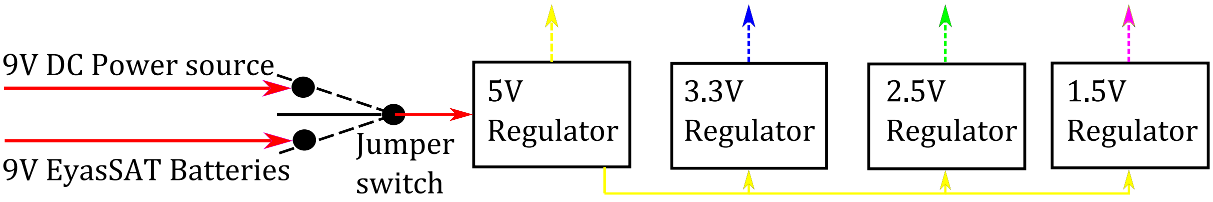


Figure 5.1: *Block Diagram of Power Circuit*

5.1.2 Microcontroller

As the single most important component, the microcontroller pulls all the data collected from the sensors together, runs all the algorithms and outputs commands to the actuators. The microcontroller needs to be fast enough to sample the sensor data and run the algorithms for the ADCS. The microcontroller that was selected is the Giant Gecko 280 from Silicon Labs Energy Micro, which was previously proven effective [27]. This is the same microcontroller that was analysed and used in [28] for an onboard cubesat ADCS computer.

The MCU is based on an ARM Cortex-M3 CPU platform that can run at up to 48 MHz. Multiple timers and communication buses are contained in the MCU. A summary of the MCU capabilities and an extract from the MCU datasheet is presented in Appendix D.8. The Figure 5.2 shows a simplified circuit diagram, indicating the important connections made with the MCU.

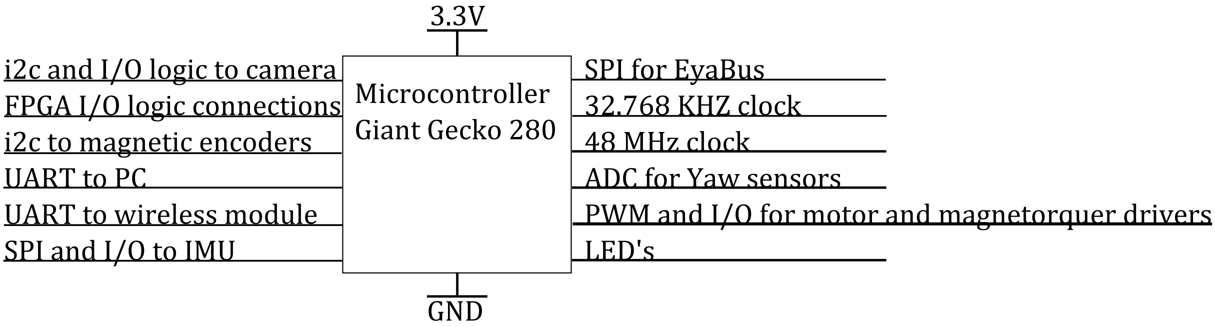


Figure 5.2: *Important Microcontroller Connections*

5.1.3 EyaBus Interface

The user manual defines the EyaBus and Appendix D.7 presents the extract [1]. The connections that are made to the bus are shown in Figure 5.3.

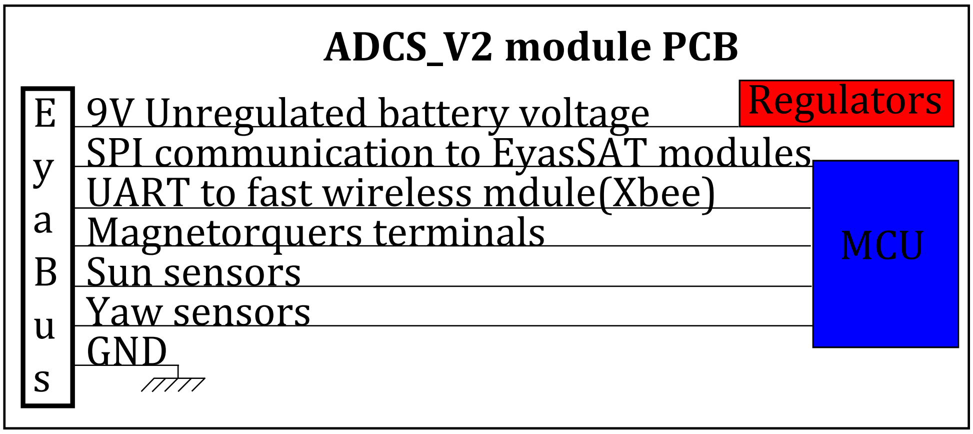

ADCS_V2 module PCB
EyaBus
9V Unregulated battery voltage
Regulators
SPI communication to EyasSAT modules
UART to fast wireless mdule(Xbee)
Magnetorquers terminals
Sun sensors
Yaw sensors
GND
MCU

Figure 5.3: *Important EyaBus Connections*

5.1.4 PCB Layout

The PCB is designed to fit in EyasSAT. According to the owner's manual the PCB must be of a PC104 form factor including the mounting holes. Appendix E.1 indicates the dimensions of the PC104 standard. This form factor significantly limits the area available for placing components. Due to the limited space a four layer PCB was required to fit all the components and buses.

The top and bottom views of the ADCS_V2 PCB are shown in Figure 5.4 and Figure 5.5. The PCB dimensions and tracks are shown in Figure 5.6.

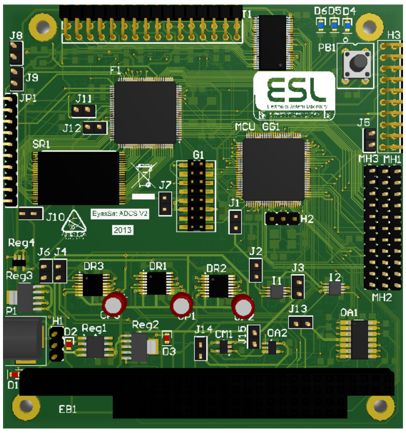


Figure 5.4: *ADCS_V2 PCB Top View*

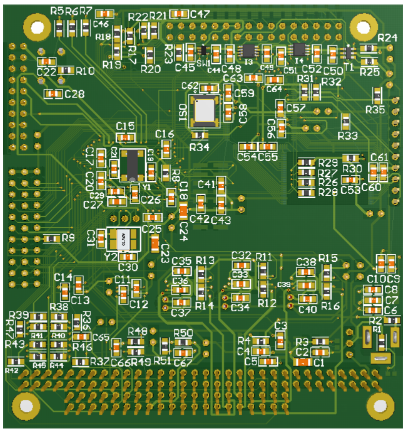


Figure 5.5: *ADCS_V2 PCB Bottom View*

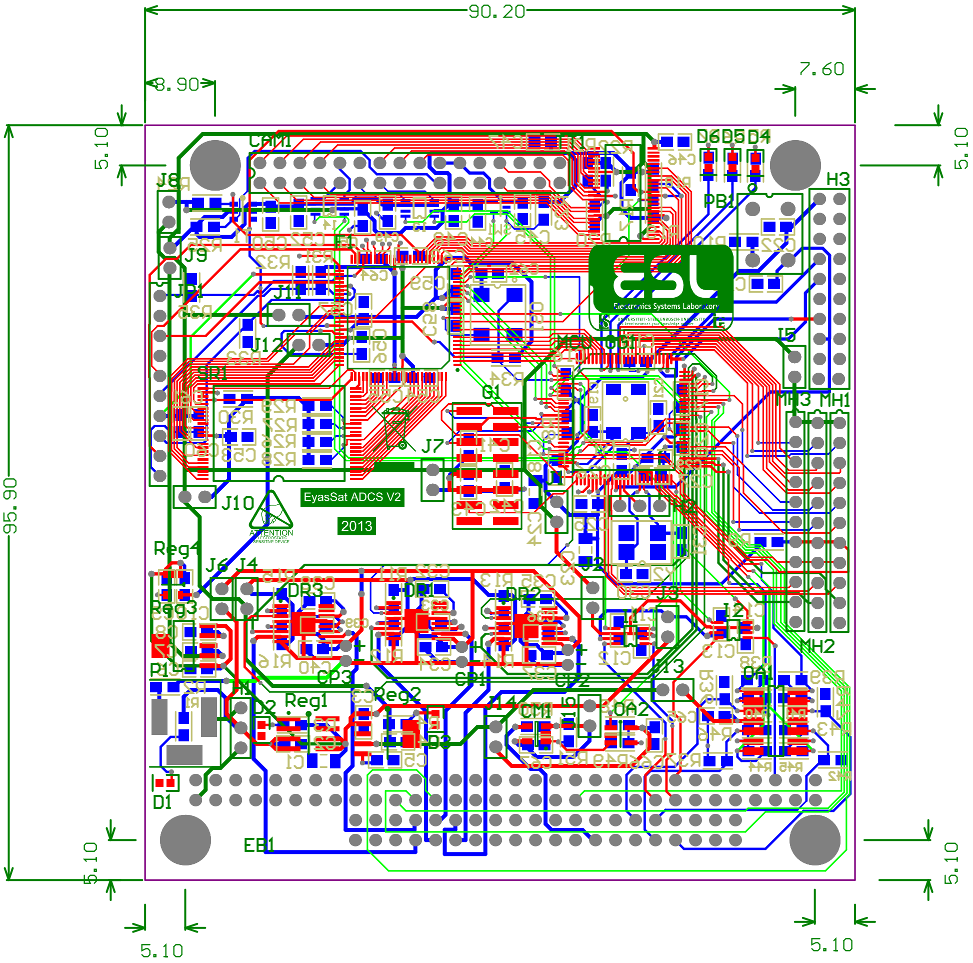


Figure 5.6: *PCB Routing*

5.1.5 External connections

There are a number of external connections that are required on the ADCS_V2 PCB. The headers of the PCB are labelled in Figure 5.7 and a description of each is listed below.

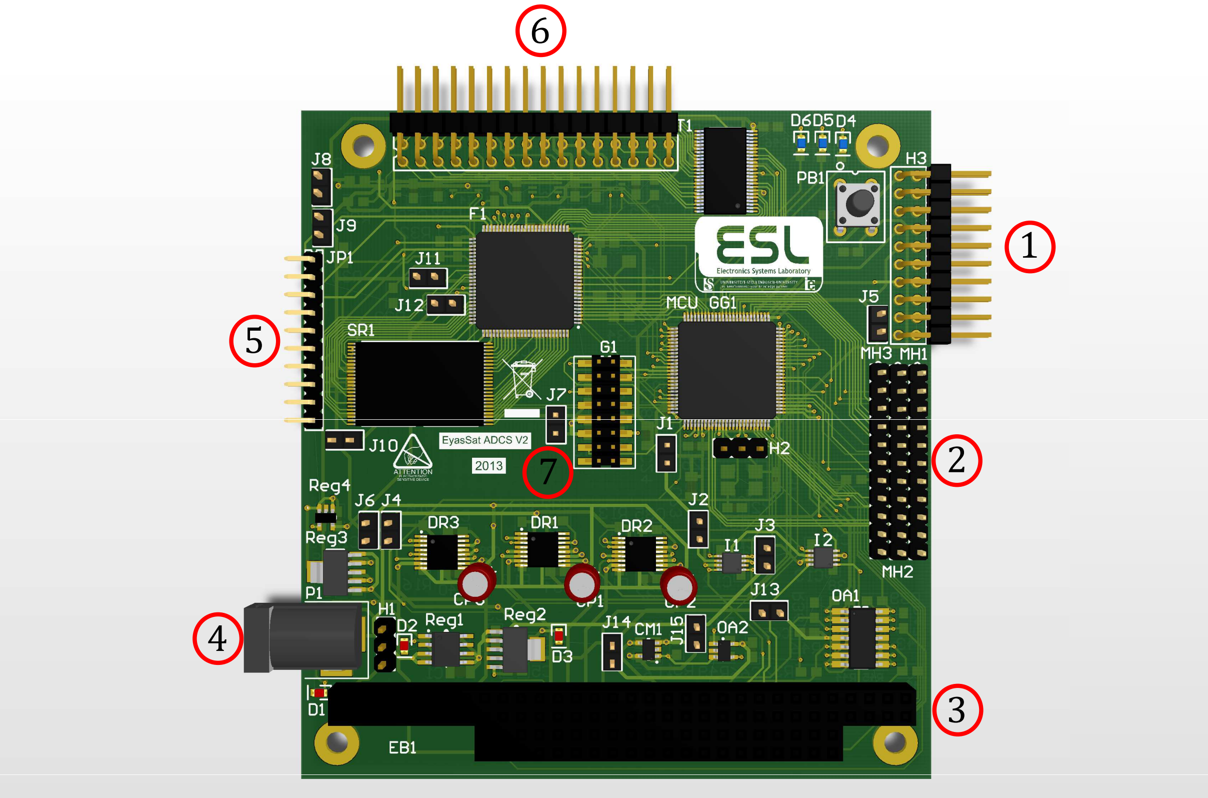


Figure 5.7: *Headers Label*

1. Microcontroller programming header
2. RWs connection header
3. EyaBus
4. DC power connection
5. FPGA programming header
6. Camera Module connection header
7. IMU connection header

5.2 Integration into EyasSAT

Some modifications had to be made to integrate the PCB and RWs into EyasSAT. A CAD model, shown in Figure 5.8 and 5.9, was made to ensure that all the parts will fit into EyasSAT. New screws were made to mount the PCB and RWs' mounting platform and to hold the PCB pillars in place, as the originals were too short. The bases of the pillars were also replaced, which allowed for the use of new screws. These parts were manufactured in the University of Stellenbosch's mechanical workshop.

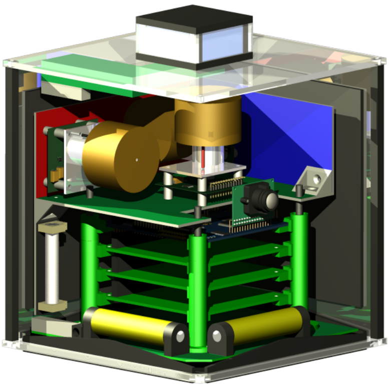


Figure 5.8: *EyasSAT CAD Model*

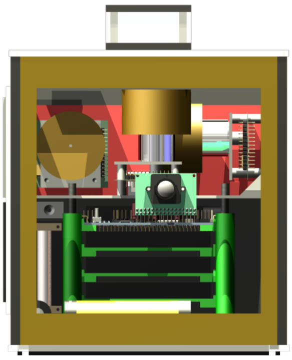


Figure 5.9: *EyasSAT CAD Model Front View*

5.3 Hardware Integration Summary

The integration of the sensors and actuators through the ADCS_V2 PCB to form the ADCS_V2 module was explained in this chapter. The fitting of all the components into EyasSAT was also shown with a CAD design in order to determine the size restraints of the different components. Some of the original parts of EyasSAT had to be replaced to allow the installation of the ADCS_V2 module.

This chapter concludes the hardware section of the project. Therefore only the ADCS algorithms are needed before HIL tests can be done.

Chapter 6

Attitude Determination

Attitude determination algorithms use the available sensor information and knowledge of the SC's orbital position with mathematical models to determine the SC's attitude in a predefined reference frame. This chapter discusses different methods used to determine the attitude of EyasSAT.

Mathematical models are employed to describe the motion and position of celestial objects such as the sun, earth and moon, relative to the SC's given orbital position. These objects serve as references for sensors on the satellite as discussed in Chapter 3. Environmental variables that can be measured, such as the earth's magnetic field direction, are described in lookup tables. These mathematical models generate unit vectors in the IRF or ORF, however, for this project the IRF is used. These unit vectors can be represented by $\mathbf{v}_i$ where the i subscript indicates a vector in the IRF.

Nonetheless, no mathematical models for the HIL experiment are needed as the references are stationary in the room. The light source, LEDs and gravity are the three references. Each one is found in a different axis of the IRF defined in Section 2.1. The light source is placed in the X_i-axis [1,0,0], the LEDs in the $-Y_i$-axis [0 ,-1, 0] and gravity is in the $-Z_i$-axis [0, 0 -1]. However, the references are not in those exact positions. The positions of the references are stored as the modelled vectors, of the IRF, when the attitude estimation is initiated. The orientation of EyasSAT, when the attitude estimation is initiated will be the rest or zero attitude.

The sensors mounted on the SC measure the unit vectors to the reference objects, in relation to the BRF. In the case where the spacecraft body frame aligns with the attitude reference frame (IRF in this case), the sensors will measure the same vectors predicted by the mathematical models. The SC's attitude is then described to be at rest or at the zero position [0,0,0] in Euler angles. The vector generated by the sensors in the BRF can be described by $\mathbf{v}_b$ where the b subscript indicates a vector in the BRF.

At least two vector pairs are required to estimate the attitude. Suppose there is a set of N unit vectors v_k with $k = 1, ..., N$. There is a sensor measurement in the BRF v_{kb} and a mathematical modelled vector in the IRF v_{ki} for each vector, which forms a vector pair. Thus there are two or more vector pairs that are generated. Each vector pair is represented by the same number subscript in the body and inertial reference frame. The vector pair can be related to one another through a rotation matrix. This transforms the modelled vector in the IRF to the vector measured in the BRF as shown in equation (6.0.1). The attitude determination algorithms are used to determine the rotation matrix $\mathbf{R}_{bi}$, as described in Section 2.2, so that:

$$\mathbf{v}_{kb} = \mathbf{R}_{bi}\mathbf{v}_{ki} \tag{6.0.1}$$

Three independent parameters are needed to determine the matrix $\mathbf{R}_{bi}$. A unit vector has two parameters due to the unit vector constraint, however, two different measured unit vector pairs are needed to determine the attitude. Therefore there are three unknowns and four known quantities which results in an over-determined problem [29]. There are several attitude determination algorithms that calculate the $\mathbf{R}_{bi}$ matrix, which are discussed in the following section.

6.1 Attitude Determination Algorithms

The discussion of attitude determination algorithms begins with the simplest algorithm utilized. This algorithm is called the Triad algorithm [29; 13].

6.1.1 Triad Algorithm

The Triad algorithm is based on constructing two triads of orthonormal unit vectors using the sensor and modelled vector information obtained. The two triads are the components of the same reference frame, called the transit reference frame, which are expressed in terms of the BRF and IRF. The one triad is constructed from the measured unit vectors and the other from the modelled unit vectors. The two triads can then be used to represent the rotation matrix from the reference frame of the used vectors to the transit reference frame. The rotation matrix between the BRF and IRF can be obtained from these two transformation matrices. The transit reference frame is obtained by assuming that one of the vector pairs is correct and this direction is used for the first base vector of the reference frame as shown in equation (6.1.1).

$$\mathbf{t}_1 = \mathbf{v}_1 \tag{6.1.1}$$

$$\mathbf{t}_{1b} = \mathbf{v}_{1b} \tag{6.1.2}$$

$$\mathbf{t}_{1i} = \mathbf{v}_{1i} \tag{6.1.3}$$

The second base vector of the reference frame is constructed as a unit vector in the direction perpendicular to the two observations:

$$\mathbf{t}_2 = \mathbf{v}_1 \times \mathbf{v}_2 \tag{6.1.4}$$

$$\mathbf{t}_{2b} = \frac{\mathbf{v}_{1b} \times \mathbf{v}_{2b}}{|\mathbf{v}_{1b} \times \mathbf{v}_{2b}|} \tag{6.1.5}$$

$$\mathbf{t}_{2i} = \frac{\mathbf{v}_{1i} \times \mathbf{v}_{2i}}{|\mathbf{v}_{1i} \times \mathbf{v}_{2i}|} \tag{6.1.6}$$

$$\tag{6.1.7}$$

The assumption made in equation (6.1.4) is that the first vector measurement v_1 is more accurate than the second measurement v_2. The third vector of the reference vector is chosen to complete the triad, as shown in equation (6.1.8).

$$\mathbf{t}_3 = \mathbf{t}_1 \times \mathbf{t}_2 \tag{6.1.8}$$

$$\mathbf{t}_{3b} = \mathbf{t}_{1b} \times \mathbf{t}_{2b} \tag{6.1.9}$$

$$\mathbf{t}_{3i} = \mathbf{t}_{1i} \times \mathbf{t}_{2i} \tag{6.1.10}$$

Two rotation matrices can be constructed by placing the $\mathbf{t}$ vector components into the columns of two 3×3 matrices $[\mathbf{t}_{1b}\ \mathbf{t}_{2b}\ \mathbf{t}_{3b}]$ and $[\mathbf{t}_{1i}\ \mathbf{t}_{2i}\ \mathbf{t}_{3i}]$. The desired transformation matrix $\mathbf{R}_{bi}$ can now be obtained to describe the attitude, as shown in equation (6.1.11).

$$\mathbf{R}_{bi} = \mathbf{R}_{bt}\mathbf{R}_{ti} = [\mathbf{t}_{1b}\ \mathbf{t}_{2b}\ \mathbf{t}_{3b}][\mathbf{t}_{1i}\ \mathbf{t}_{2i}\ \mathbf{t}_{3i}]^T \tag{6.1.11}$$

This completes the Triad algorithm. The Triad algorithm was implemented on EyasSAT. Furthermore the attitude was estimated, however, EyasSAT contains more than two attitude sensors. A statistical method is required to use all the available attitude information. A better estimate of $\mathbf{R}_{bi}$ could be provided by a statistical method that aims to find the rotation matrix $\mathbf{R}_{bi}$ for each of the N vectors. A solution is sought for $\mathbf{R}_{bi}$ that minimizes the overall error for the N vectors. A matrix $\mathbf{R}_{bi}$ that minimizes the cost function, equation (6.1.12), will be the best approximated rotation matrix, describing the attitude, given all the measurements. This problem was first posed by Wahba [14]. In equation (6.1.12), J is the loss function to be minimized and w_k is the weight of the kth vector measurement.

$$J(\mathbf{R}_{bi}) = \frac{1}{2}\sum_{k=1}^{N} w_k|\mathbf{v}_{kb} - \mathbf{R}_{bi}\mathbf{v}_{ki}|^2 \tag{6.1.12}$$

This loss function is a sum of the squared errors for each vector measurement. If the measurements and mathematical models are all perfect, then (6.0.1) will be satisfied for all N vectors and $J = 0$. The smaller J is the better the approximation of $\mathbf{R}_{bi}$. However if there are any errors or noise on the measurements then $J > 0$.

One method for solving this minimization problem is the QUEST (QUaternion ESTimator) algorithm. This is an efficient approximation of the q-method and will be discussed below [29].

6.1.2 Q-Method and Quest Algorithm

The Q-Method is needed as background information in order to discuss the Quest algorithm. A description and the methods used to implement the Q-Method and Quest algorithm was found in [29] and was used to implement the Quest algorithm.

6.1.2.1 Q-Method

The cost function is rewritten as shown in equation (6.1.13). The gain matrix $g(\mathbf{R}_{bi})$ is then defined as shown in equation 6.1.14. Maximizing the gain matrix is the same as minimizing the cost function J. The gain function can be rewritten in terms of quaternions. This leads to the form shown in equation (6.1.15), where the matrix $\mathbf{K}$ is defined in equation (6.1.16).

$$J(\mathbf{R}_{bi}) = \sum_{k=1}^{N} w_k(1 - \mathbf{v}_{kb}^T\mathbf{R}_{bi}\mathbf{v}_{ki}) \tag{6.1.13}$$

$$g(\mathbf{R}_{bi}) = \sum \omega_k \mathbf{v}_{kb}^T\mathbf{R}_{bi}\mathbf{v}_{ki} \tag{6.1.14}$$

$$g(\mathbf{q}) = \mathbf{q}^T\mathbf{K}\mathbf{q} \tag{6.1.15}$$

where

$$\mathbf{K} = \begin{bmatrix} \mathbf{S} - \sigma\mathbf{I} & \mathbf{Z} \\ \mathbf{Z}^T & \sigma \end{bmatrix} \tag{6.1.16}$$

with

$$\begin{aligned}
\mathbf{B} =& \sum w_k(\mathbf{v}_{kb}\mathbf{v}_{ki}^T) && (6.1.17)\\
\mathbf{S} =& \mathbf{B} + \mathbf{B}^t && (6.1.18)\\
\mathbf{Z} =& [B_{23} - B_{32}\ B_{31} - B_{13}\ B_{12} - B_{21}]^T && (6.1.19)\\
\sigma =& tr[\mathbf{B}] && (6.1.20)
\end{aligned}$$

The Q-Method then states that the optimal attitude is an eigenvector of the **K** matrix. The largest eigenvalue of the **K** matrix maximizes the gain function and results in the optimal attitude [29] . QUEST provides a way to estimate the largest eigenvalue λ_{opt} to maximize **K**.

6.1.2.2 Quest Algorithm

The optimal attitude minimizes the cost function in (6.1.21) and maximizes the gain function shown in (6.1.22).

$$J = \frac{1}{2}\sum_{k=1}^{N} w_k|1 - \mathbf{v}_{bk}^T\mathbf{R}_{bi}\mathbf{v}_{ki}|^2 \qquad (6.1.21)$$

$$\begin{aligned}
g &= \sum w_k\mathbf{v}_{bk}^T\mathbf{R}_{bi}\mathbf{v}_{ki} && (6.1.22)\\
g &= \lambda_{opt} && (6.1.23)
\end{aligned}$$

Rearranging these expressions provides a useful result shown in equation (6.1.24)

$$\lambda_{opt} = \sum w_k - J \qquad (6.1.24)$$

The loss function should be small, for the optimal eigenvalue. Thus a good approximation for the optimal eigenvalue is:

$$\lambda_{opt} \approx \sum w_k \qquad (6.1.25)$$

Once the optimal eigenvalue has been estimated, the corresponding eigenvector is calculated which corresponds to the optimal attitude estimate. One way of calculating this eigenvector is to convert the quaternion in the eigenproblem to Rodriquez parameters, defined as:

$$\mathbf{p} = \frac{\mathbf{q}}{q_4} \tag{6.1.26}$$

The problem can be rearranged as:

$$\mathbf{p} = [(\lambda_{opt} + \sigma)1 - \mathbf{S}]^{-1}\mathbf{Z} \tag{6.1.27}$$

Gaussian elimination can then be used to solve equation (6.1.28) rather than calculating the inverse specified in equation (6.1.27).

$$[(\lambda_{opt} + \sigma)1 - \mathbf{S}]\mathbf{p} = \mathbf{Z} \tag{6.1.28}$$

Once the Rodriquez parameters are found, the quaternion is calculated by

$$\mathbf{q} = \frac{1}{\sqrt{1 + \mathbf{p}^T\mathbf{p}}} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \tag{6.1.29}$$

However, the Rodriquez parameters become singular, when the rotation is π radians and this could be problematic. The Triad and Quest algorithm only looks at the unit vectors produced by the different sensors and estimates an attitude given those measurements at that time step. The previous determined attitude is not taken into account. In addition the noise and accuracy of each sensor is also not taken into account. An EKF takes the sensor characteristics, the previously calculated attitude and the gyroscope data into account to estimate the attitude. The Triad or Quest algorithms can be used to calculate an initial attitude that can be used for the initialization of the EKF.

6.1.3 Extended Kalman Filter

The attitude determination algorithm that is mainly used for this project is the Extended Kalman Filter. The EKF can estimate the states of a non-linear system by linearising around the estimated state vector. All available sensor measurements and their corresponding modelled vectors are used in the EKF. The statistical description of the measurement errors and noise are also taken into account to determine the best estimated state vector.

A gyroscope bias and attitude estimation EKF is implemented as derived in [30]. Gyroscope measurements are used to determine EyasSAT's dynamics. Absolute attitude vector measurements are used to determine the attitude and rate bias vectors with the aid of non-linear kinematic state models. The estimated (indicated by the ˆ symbol) full state vector (7 elements) constructed from the perturbation state vector (6 elements) of the EKF is represented in equation 6.1.30, where the estimated quaternion is indicated by $\hat{\mathbf{q}}$ and the estimated gyroscope bias by $\hat{\mathbf{b}}$.

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{q}} \\ \hat{\mathbf{b}} \end{bmatrix} \tag{6.1.30}$$

where[1]

$$\hat{\mathbf{q}} = \delta\mathbf{q} \otimes \hat{\mathbf{q}} \tag{6.1.31}$$

and

$$\delta\mathbf{q} = \begin{bmatrix} \delta\mathbf{q_{vec}} \\ \sqrt{1 - ||\delta\mathbf{q}||} \end{bmatrix} \tag{6.1.32}$$

The (6 elements) state perturbation vector is defined in equation (6.1.33).

$$\delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{q_{vec}} \\ \boldsymbol{\Delta}\mathbf{b} \end{bmatrix} \tag{6.1.33}$$

The system model equations are the kinematic equation (2.3.12) described in Section 2.3.2 and gyroscope model shown in equation (6.1.34).

$$\boldsymbol{\omega}_B^I = \mathbf{u} - \mathbf{b} - \boldsymbol{\eta}_1 \tag{6.1.34}$$

Where the gyroscope measurement is $\mathbf{u}$, the bias vector is $\mathbf{b}$ and the zero mean Gaussian distributed gyroscope measurement noise vector is $\boldsymbol{\eta}_1$. Gyro bias is described as a random walk noise process and is shown in equation (6.1.35).

$$\dot{\mathbf{b}} = \boldsymbol{\eta}_2 = \Delta\mathbf{b} \tag{6.1.35}$$

Where the zero mean Gaussian gyroscope drift noise vector for the rate random walk model is $\boldsymbol{\eta}_2$. Including the above equations, the perturbation model for the EKF becomes,

$$\delta\dot{\mathbf{q}} = -\hat{\boldsymbol{\omega}}_B^I \times \delta\mathbf{q} - \frac{1}{2}(\Delta\mathbf{b} + \boldsymbol{\eta}_1) \tag{6.1.36}$$

with, $\delta\dot{\mathbf{q}}$ the vector part of the error quaternion.

The perturbation state model becomes:

$$\Delta\dot{\mathbf{x}} = \mathbf{F}\Delta\mathbf{x} + \mathbf{G}\mathbf{s} \tag{6.1.37}$$

[1] The symbol $\otimes$ represents quaternion multiplication.

with,

$$\mathbf{F} = \begin{bmatrix} \hat{\boldsymbol{\omega}}^{\mathbf{I}}\times & -0.5\mathbf{I}_{3\times3} \\ [0]_{3\times3} & [0]_{3\times3} \end{bmatrix} \tag{6.1.38}$$

$$\mathbf{G} = \mathbf{I}_{6\times6} \tag{6.1.39}$$

$$\mathbf{s} = \begin{bmatrix} -0.5\eta_1 & \eta_2 \end{bmatrix} \tag{6.1.40}$$

where

$$\hat{\boldsymbol{\omega}}^{I}\times = \begin{bmatrix} 0 & \omega_y & -w_z \\ -\omega_x & 0 & \omega_z \\ \omega_x & -\omega_y & 0 \end{bmatrix} \text{ the inertial referenced angular rate cross product matrix,} \tag{6.1.41}$$

$\mathbf{s}$ = zero mean with covariance $\mathbf{Q}$, system noise vector.

6.1.4 Measurement equation

The EKF measurement model produces the innovation by taking the cross-product between the measured attitude sensor vector in body coordinates and the modelled sensor vector in a known reference coordinate frame that is transformed to body coordinates. The innovation is given in equation (6.1.43).

$$\mathbf{e} = \mathbf{v}_{meas} \times \hat{\mathbf{v}}_{body} \tag{6.1.42}$$

$$= \mathbf{v}_{meas} \times \mathbf{A}\{\hat{q}\}\mathbf{v}_{model} \tag{6.1.43}$$

with

$\mathbf{v}_{meas}$ = The measurement unit vector in the BRF.

$\mathbf{v}_{body}$ = The modelled unit vector in the BRF.

$\mathbf{v}_{model}$ = The modelled unit vector in the IRF.

$\mathbf{A}\{q\} = \mathbf{A}\{\delta q\}\mathbf{A}\{\hat{q}\}$ the true transformation matrix from the reference of the body coordinates

and

$$\mathbf{v}_{meas} - \mathbf{m}_1 = \mathbf{A}\{q\}(\mathbf{v}_{model} - \mathbf{m}_2) \tag{6.1.44}$$

$$\mathbf{v}_{meas} = \mathbf{A}\{\delta\mathbf{q}\}\hat{\mathbf{v}}_{body} + \mathbf{m}_3 \tag{6.1.45}$$

with

$$\begin{aligned}\mathbf{m}_1 &= \text{The measurement noise vector.}\\ \mathbf{m}_2 &= \text{The modelling noise vector.}\\ \mathbf{m}_3 = \mathbf{m}_1 = \mathbf{A}\{\mathbf{q}\}\mathbf{m}_2 &= \text{The sum of uncorrelated measurement and modelling noise vectors}\end{aligned}$$

thus,

$$\mathbf{e} = \mathbf{A}\{\delta\mathbf{q}\}\hat{\mathbf{v}}_{body} + \mathbf{m}_3 \times \hat{\mathbf{v}}_{body} \tag{6.1.46}$$

$$= \mathbf{H}\{\hat{\mathbf{v}}_{body}\}\delta\mathbf{q}_{\mathbf{vec}} + \mathbf{m} \tag{6.1.47}$$

with

$$\mathbf{H}\{\hat{\mathbf{v}}_{body}\} = 2\begin{bmatrix}\hat{v}_{by}^2 + \hat{v}_{bz}^2 & -\hat{v}_{bx}\hat{v}_{by} & -\hat{v}_{bx}\hat{v}_{bz}\\ -\hat{v}_{by}\hat{v}_{bx} & \hat{v}_{bx}^2 + \hat{v}_{bz}^2 & -\hat{v}_{by}\hat{v}_{bz}\\ -\hat{v}_{bz}\hat{v}_{bx} & -\hat{v}_{bz}\hat{v}_{by} & \hat{v}_{bx}^2 + \hat{v}_{by}^2\end{bmatrix} \tag{6.1.48}$$

$\mathbf{m} = \mathbf{m}_3 \times \hat{\mathbf{v}}_{body} =$ zero mean, covariance $\mathbf{R}$, measurement plus modelling noise vector.

6.1.5 EKF Discrete Implementation

The EKF state variable and covariance propagation can be done in a discrete implementation that follows the numbered steps shown below in time and measurement updates.

6.1.5.1 Time Update

1) The gyroscope measurements are obtained. Equation (6.1.34) is used to obtain the estimated angular rate vector:

$$\hat{\boldsymbol{\omega}}_B^I(k) = \mathbf{u}(k) - \hat{\mathbf{b}}(k-1) \tag{6.1.49}$$

2) The attitude quaternion is propagated from a discrete kinematic equation, see equation 6.1.50. [14]

$$\mathbf{q}^-(k) = \left[\cos(0.5\bar{\omega}\Delta T)\mathbf{I}_{4\times4} + \frac{1}{\bar{\omega}}\sin(0.5\bar{\omega}\Delta T)\boldsymbol{\Omega}\{\hat{\boldsymbol{\omega}}_B^I(k)\}\right]\hat{\mathbf{q}}^+(k-1) \tag{6.1.50}$$

with

$$\bar{w} = ||\hat{\omega}_B^I(k)|| \tag{6.1.51}$$

$$\Delta T = \text{Gyroscope sampling period (0.2s)}$$

3) The state covariance matrix is propagated

$$\mathbf{P}^-(k) = \boldsymbol{\Phi}(k)\mathbf{P}^+(k-1)\boldsymbol{\Phi}^T(k) + \mathbf{Q}(k) \tag{6.1.52}$$

with

$$\mathbf{\Phi}(k) = e^{\mathbf{F}(k)\Delta T} = \text{ The discrete state transition matrix.} \tag{6.1.53}$$

$$\approx \mathbf{I}_{6\times6} + \mathbf{F}(k)\Delta T + \frac{1}{2}\mathbf{F}^2(k)\Delta T^2 \tag{6.1.54}$$

$$\mathbf{Q}(k) = \text{The discrete system noise covariance matrix.} \tag{6.1.55}$$

6.1.5.2 Measurement Update

The EKF state variable vector and state covariance matrix can be updated when the attitude sensors have valid measurement vectors available at the sampling instances (0.2s). This means the EKF will propagate the gyroscope measurements at 0.2 second intervals and perform an update depending on the availability of the attitude sensor measurements.

4) If new sensor measurements are unavailable:

$$\hat{\mathbf{q}}^+(k) = \hat{\mathbf{q}}^-(k) \tag{6.1.56}$$

$$\hat{\mathbf{p}}^+(k) = \hat{\mathbf{p}}^-(k) \tag{6.1.57}$$

$$\tag{6.1.58}$$

and repeat steps 1-3.

If new measurements are available go to step 5.

5) The measured and modelled sensor vectors are used to calculate the innovation vector from equation (6.1.43):

$$\mathbf{e}(k) = \mathbf{v}_{meas}(k) \times \mathbf{A}\{\hat{\mathbf{q}}^-(k)\}\mathbf{v}_{model}(k). \tag{6.1.59}$$

6) Calculate EKF update gain

$$\mathbf{K}(k) = \mathbf{P}^-(k)\mathbf{H}^T(k)[\mathbf{H}(k)\mathbf{P}^-(k)\mathbf{H}^t(k) + \mathbf{R}_i]^{-1} \tag{6.1.60}$$

$$= [\mathbf{K_q}(\mathbf{k})\ \mathbf{K_b}(k)]^T \tag{6.1.61}$$

with

$$\mathbf{R}_i = \text{The measurement noise covariance matrix for the } i\text{th sensor.} \tag{6.1.62}$$

7) Update the estimated quaternion and gyroscope bias vectors:

$$\delta\mathbf{q}_{vec}(k) = \mathbf{K}_q(k)\mathbf{e}(k) \text{ and } \Delta\mathbf{b}(k) = \mathbf{K}_b(k)\mathbf{e}(k) \tag{6.1.63}$$

$$\hat{\mathbf{q}}^+(k) = \delta\hat{\mathbf{q}}(k) \otimes \hat{\mathbf{q}}^-(k) \tag{6.1.64}$$

$$\hat{\mathbf{b}}(k) = \hat{\mathbf{b}}(k-1) + \Delta\mathbf{b}(k) \tag{6.1.65}$$

8) Update the state covariance matrix:

$$\mathbf{P}^{+}(k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)]\mathbf{P}^{-}(k)[\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)]^{T} + \mathbf{K}(k)\mathbf{R}_{i}\mathbf{K}^{T}(k) \tag{6.1.66}$$

The EKF algorithm was implemented and tested. Chapter 9 contains the simulation results and the HIL results can be seen in Chapter 10.

6.2 Attitude Determination Summary

The EKF algorithm was the main attitude estimation algorithm used. However, a simulation was done to test the performance and operation of the algorithm. The simulation results are discussed in Chapter 9. The algorithm was also implemented on the ADCS_V2 module and tested on the air bearing. The results of the practical test can be found in Chapter 10.

The Triad and Quest algorithms were also implemented on the MCU. These algorithms can be used for demonstration purposes. An initialization attitude can be given to the EKF to begin with when the EKF is started by using these algorithms. This will allow EyasSAT to be placed in any orientation on the air bearing and the attitude will be estimated, given the appropriate modelled reference vectors.

Chapter 7

Attitude Control

The attitude controller will be able to change the attitude of the SC using the on-board actuators given that the SC attitude is determined. Quaternion feedback is used on EyasSAT to generate torque commands for the three RWs.

As described in Section 4.2, the RWs are placed in the body axes. Any angular momentum imparted to the wheels are balanced by an equal momentum in the opposite direction in the satellite body. This is the conservation of angular momentum and is described by equation (7.0.1). If a wheel in one axis turn's with the angular rate of ω_{RW} the spacecraft will rotate around that axis with the angular rate of $\frac{I_{RW}}{I_{nn}}\omega_{RW}$.

$$I_{nn}w_{nn} = I_{RW}w_{RW} \tag{7.0.1}$$

In equation (7.0.1) the *nn* subscript denotes any of the three body axes. The total angular momentum **L** of the satellite can be written as shown in equation (7.0.2) [15].

$$\mathbf{L} = \mathbf{I}\boldsymbol{\omega}_B^I + \mathbf{h} \tag{7.0.2}$$

where

$$\mathbf{h} = \mathbf{I}_{RW}\boldsymbol{\omega}_{RW} = \text{The RWs' angular momentum vector.} \tag{7.0.3}$$

with,

$$\mathbf{I}_{RW} = \text{The diagonal RWs MOI matrix.} \tag{7.0.4}$$

$$\boldsymbol{\omega}_{RW} = \text{The RW angular rate vector referenced to satellite's body} \tag{7.0.5}$$

The equation (7.0.6) results from the time derivative of **L** being taken [15]. If the external torques on the satellite are neglected, then the total angular momentum is conserved and the left hand side of the equation adds up to zero.

$$\mathbf{I}\dot{\mathbf{w}}_B^I + \dot{\mathbf{h}} + \boldsymbol{\omega}_B^I \times (\mathbf{I}\boldsymbol{\omega}_B^I + \mathbf{h}) = 0 \tag{7.0.6}$$

The vector $\dot{\mathbf{h}} = \mathbf{I}_{RW}\dot{\boldsymbol{\omega}}_{RW} = \mathbf{N}_{RW}$ is the net torque applied to the wheels by the spacecraft body. According to Newton's third law of motion $-\dot{\mathbf{h}}$ is the torque applied to the spacecraft body. Equation (7.0.6) is equivalent to equation (2.3.1), without the external torques.

7.1 Quaternion Feedback

The attitude of the satellite is described by quaternions. A desired orientation can be related to quaternions. An error quaternion can be obtained as described by equation (7.1.1) [31; 32].

$$\begin{bmatrix} q_{1e} \\ q_{2e} \\ q_{3e} \\ q_{4e} \end{bmatrix} = \begin{bmatrix} q_{4c} & q_{3c} & -q_{2c} & -q_{1c} \\ -q_{3c} & q_{4c} & q_{1c} & -q_{2c} \\ q_{2c} & -q_{1c} & q_{4c} & -q_{3c} \\ q_{1c} & q_{2c} & q_{3c} & q_{4c} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{7.1.1}$$

where the e subscript denotes the error quaternion and the c subscript the command quaternion.

The quaternion feedback law used in this project is given in equation (7.1.2) [15; 13; 33]. The RW control torque is calculated in the control law by using a proportional gain with the quaternion error and a derivative gain with the body rate.

$$\mathbf{N}_{RW} = \mathbf{K}_p\mathbf{I}\mathbf{q}_{e\_vec} + \mathbf{K}_d\mathbf{I}\omega_B^I \tag{7.1.2}$$

In equation (7.1.2), the MoI matrix defined in equation (2.3.5) is $\mathbf{I}$, the proportional gain matrix is $\mathbf{K}_p$, and the derivative gain matrix is $\mathbf{K}_d$. The gain matrices are shown in equation (7.1.3) to (7.1.5). The $\mathbf{q_{e\_vec}}$ symbol represents the vector component of the quaternion error matrix as defined in equation (7.1.5).

$$\mathbf{K}_p = \text{diag}[k_{px}\ k_{py}\ k_{pz}] \tag{7.1.3}$$

$$\mathbf{K}_d = \text{diag}[k_{dx}\ k_{dy}\ k_{dz}] \tag{7.1.4}$$

$$\mathbf{q_{e\_vec}} = [q_{1e}\ q_{2e}\ q_{3e}]^T \tag{7.1.5}$$

The control gains can be designed for a certain amount of damping and natural frequency. The equations used to calculate the gains are shown in equation (7.1.6) and (7.1.7).

$$k_p = 2w_n^2 \tag{7.1.6}$$

$$k_d = 2\zeta\omega_n \tag{7.1.7}$$

7.1.1 EyasSAT implementation

In order to calculate the gains of the quaternion feedback law, the desired response of the system needs to be defined. An over-damped response, with a settling time of 5 s in the Z_b-axis and 10 s settling time in the X_b- and Y_b-axis, was determined to be the desired control specifications. These settling times were chosen based on experimentation and the damping response was chosen to obtain a smooth stop motion for EyasSAT on the air bearing. The X_b- and Y_b-axis RWs were used to damp the body rates on the air bearing [11]. This will result in the X_b- and Y_b-axis RWs control laws only containing differential gain components. The damping and natural frequency shown in Table 7.1 are a result of these specifications.

Over-damped:	$\zeta = 1$
10s settling time:	$w_n = 0.4$ rad/s
5s settling time:	$w_n = 0.8$ rad/s

Table 7.1: *Control Specifications*

These specifications allow for the calculation of the gains which are shown in equation (7.1.8) and (7.1.9).

$$\mathbf{K}_p = \text{diag}[0\ 0\ 1.28] \tag{7.1.8}$$

$$\mathbf{K}_d = \text{diag}[0.8\ 0.8\ 1.6] \tag{7.1.9}$$

The resulting control torque equations used to calculate the command torque for each RW is given in equation (7.1.10) to (7.1.12).

$$N_{RWx} = 0.2019\omega_x \tag{7.1.10}$$

$$N_{RWy} = 0.1811\omega_y \tag{7.1.11}$$

$$N_{RWz} = 0.20833q_{3e} + 0.26041\omega_z \tag{7.1.12}$$

7.2 Torque Command to Speed Command Conversion

The quaternion feedback control generates a torque command for each RW. The RWs, however, only have speed controllers. The torque commands need to be transformed to speed commands for the speed controllers. This is achieved by integrating the torque command. Torque is the derivative of momentum as shown in equation (7.2.1).

$$N_{RW} = \dot{M_{RW}} = I_{RW}\dot{\omega_{RW}} \tag{7.2.1}$$

The desired momentum can be obtained by integrating the torque over time. The momentum, as shown in equation (7.2.1), is the speed of the RW multiplied by the inertia of the RW. The inertia is known and constant, thus the speed can be calculated given a desired momentum.

The torque command stays the same for the entire control sample period of 200ms. Therefore, the torque can be numerically integrated over this time period and the required momentum calculated. The equation for calculating the desired momentum is shown in equation (7.2.2).

$$M_{RW}(k) = M_{RW}(k-1) + 0.5T_s(N_{RW}(k) + N_{RW}(k-1)) \tag{7.2.2}$$

The speed can then be calculated for each RW by dividing the desired momentum by the inertia of the specific RW, as shown in equation (7.2.3).

$$\omega_{RW} = \frac{M_{RW}}{I_{RW}} \tag{7.2.3}$$

Thereafter, the speed command is sent as a reference speed to the RW speed controller discussed in Section 4.2.8.

7.3 Proportional Feedback for Light Tracking

A light tracking controller, that allows EyasSAT to track a light source was also implemented. This controller does not require attitude determination because the LED is only attempted to be aligned with the boresight of the camera. The pixel position of the tracking LED and the pitch angle between the two LEDs were used to generate speed commands for the RWs, using the equations described in Section 3.2.5. The control gains that gave good responses were obtained through trial and error. The gains used are shown in equation (7.3.1) to (7.3.3). In the equations P_{x1} and P_{z1} are the light pixel position described in Section 3.2.5. The *Pitch* variable is the calculated pitch angle of EyasSAT in radians, as described in equation (3.2.8).

$$\omega_{RWx} = 10P_{z1} \tag{7.3.1}$$

$$\omega_{RWy} = 20\left(\frac{180}{\pi}\right) Pitch \tag{7.3.2}$$

$$\omega_{RWz} = 10P_{x1} \tag{7.3.3}$$

This controller allows for the demonstration and testing of the camera sensor and the RWs speed control.

7.4 Attitude Control Summary

Quaternion feedback was implemented to generate torque commands for the RWs. The torque commands are converted to speed commands and the references sent to the RWs speed controller. The quaternion feedback was designed to obtain a 5 s settling time in the Z_b-axis and a 10 s settling time in the X_b- and Y_b-axis. An over-damped response for all the axes was also specified. The performance of the quaternion feedback controller is simulated in Chapter 9 and the practical test results are discussed in Chapter 10.

Chapter 8

Software Development

This chapter describes the implementation of the software and algorithms on the MCU as an expansion of the description provided in Section 5.1.2. The PCB connects the MCU with all the sensors and actuators. This allows the program on the MCU to read the sensor data and output command signals to the actuators. An overview of the program flow will be provided in this chapter to show how and when the sensor data is read and when the ADCS algorithms are executed in the MCU.

8.1 Program Flow

The program flow is described by a flow chart shown in Figure 8.1. Each block in the chart will be briefly described in the following sections.

8.1.1 Initialization

The initialization of global variables and variables required in the main function is done after the MCU powered up. The chip and clock initialization are executed. In addition, other required peripheral initialization functions are performed. After this setup procedure the main loop is ready to start. The timer controlling the timing interrupts is started before the main loop is entered. A single timer is used to generate accurate sampling time interrupts. When a timing interrupt occurs, the appropriate flag, or global variable, is set to 1 to indicate that the interrupt occurred and all the appropriate functions for that time period needs to be handled in the main loop.

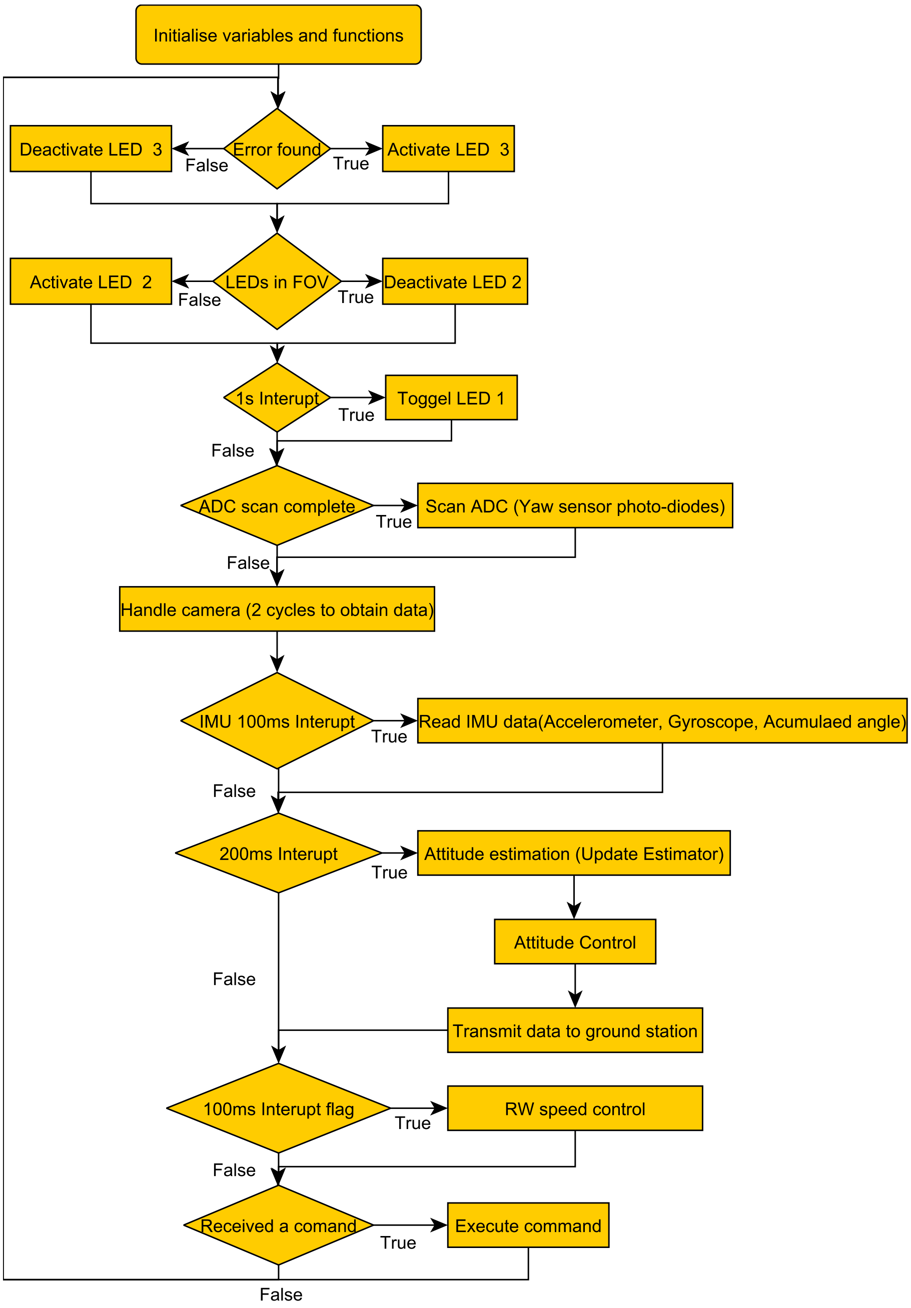


Figure 8.1: *Program Flow Chart*

8.1.2 Error Condition Monitoring

A number of ICs used on the ADCS_V2 PCB and on the RW PCBs, possesses fault pins that can indicate if a fault condition arises in the IC. Fault conditions include but are not limited to under voltage and overheating conditions of the ICs. If any fault condition is detected the third status LED on the ADCS_V2 module PCB is activated.

The most likely error condition that will be detected is the under voltage condition of the motor drivers. These drivers require 8 V or more to operate and will assert a fault condition when the voltage of the batteries drops below 8 V.

8.1.3 LED in FoV

The camera sensor, discussed in Section 3.2, has a reduced FoV. The second status LED, on the ADCS_V2 PCB, is deactivated with the detection of the reference LEDs in the camera's FoV. Thus the status LED will be on if the camera does not detect the tracking LED in the camera's FoV. This status LED is useful to indicate that the camera is working, when the air bearing test and demonstration is done.

8.1.4 Interrupt: 1s

The 1 second interrupt flag is used to toggle the first status LED on the ADCS_V2 PCB. This flashing status LED acts as an indicator that the main program loop is running.

8.1.5 ADC Scan

The analogue to digital converter scans the pins connected to the CSS's outputs and filters the results through oversampling. This is done as fast as possible to ensure that the latest results are obtained from the CSS. The digital to analogue conversion output is handled by Direct Memory Access (DMA). Therefore, the program flow is not halted while the analogue to digital conversion is undertaken.

8.1.6 Handle Camera

The amount of time required by the FPGA to output the data from the camera, makes this a time consuming process. As a result the process of obtaining data from the camera has been broken up into two steps. The MCU first requests that a picture be taken. Then an acknowledgement from the FPGA needs to be received that signals that the picture has been stored and the LEDs detected. Thereafter, the MCU can request the LEDs' pixel positions. Lens distortion correction is then performed on the LEDs' positions and the unit vector generated. The process is shown in Figure 8.2.

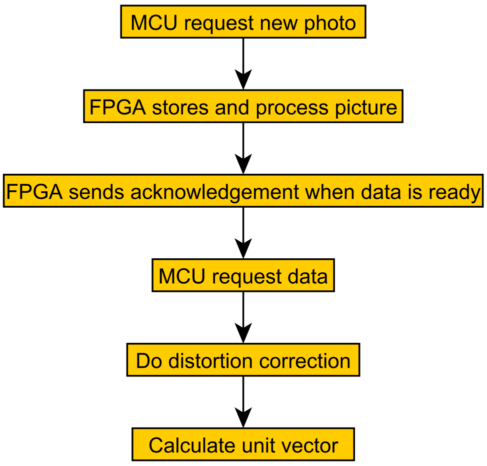


Figure 8.2: *Camera Flow*

Instead of waiting for the picture to be stored and obtaining an acknowledgement from the FPGA, the MCU will request the picture to be taken, when the camera handling function is reached, and then continue with the main program loop. The MCU will verify if the FPGA responded with an acknowledgement, when the camera handling function is reached for the second time. If not, then the program flow continues, whilst waiting for the FPGA to respond. If the acknowledgement is received, then the data is requested and the distortion correction and vector generation is done. The program flow is shown in Figure 8.3.

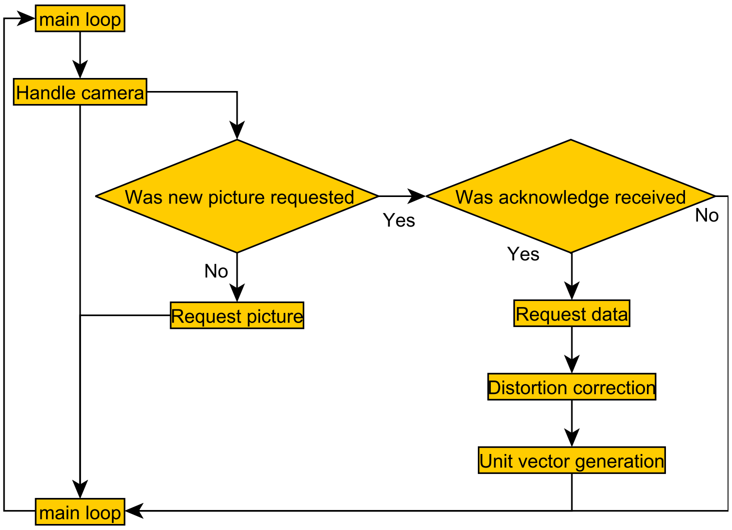


Figure 8.3: *Camera Interface Loop*

8.1.7 IMU Interrupt: 100 ms

The IMU generates an interrupt every 100 ms indicating that new data is ready. Over the 100ms, the accelerometer, gyroscope and accumulated angle is read from the IMU through an SPI interface.

8.1.8 Interrupt: 200 ms

The 200 ms interrupt signals the execution of the EKF attitude determination algorithm as defined in Section 6.1.3. This involves the time and measurement update if the sensor data is available. After the current attitude has been estimated, the attitude control is implemented for the current time step. Thereby the reference signals to command EyasSAT to the correct attitude can be generated. The attitude controller implemented was discussed in Chapter 7.

8.1.9 Interrupt: 100 ms

The 100 ms interrupt signals the RWs speed control. The speed controllers use the reference signals, generated by the attitude controller in the 200ms interrupt, to obtain a speed reference signal. The RW in each axis is then controlled to the desired speed to change the attitude of EyasSAT. The speed controller of the RWs was discussed in Section 4.2.8.

A 5 ms interrupt is used to sample the RW axis position, which is used for speed estimation. This interrupt does not set a flag but immediately executes the sampling function. This interrupt can occur at any time during the main program loop.

8.1.10 Command Receiver

The ADCS_V2 can receive commands through wireless communication. These commands are defined in the code and implemented by receiving the appropriate identification (ID) character. This part of the code monitors the received buffer for a command ID. If a command ID has been received, then the command is looked up and executed before returning to the main program loop.

8.2 Software Development Conclusion

The MATLAB Simulink environment, the ground station PC, sent the commands received by the program, when a certain HIL test or demonstration is done. These commands can also be sent manually using a terminal interface. The program flow was monitored for timing problems by measuring the time taken to execute each part of the program. However, the main loop was found to be running fast enough not to cause any timing problems.

Chapter 9

Simulation Test and Results

A MATLAB simulation was created in Simulink to develop and evaluate the attitude determination and control algorithms. The dynamics and kinematics of the system were simulated using the equations of attitude motion, as discussed in Section 2.3. The sensor data and outputs were also simulated to resemble the output of the sensors as closely as possible. The attitude determination algorithms were implemented to estimate the state of the system, given the sensor outputs. The simulation ensured and confirmed that the attitude determination algorithms were correctly implemented and would perform as expected. After the attitude determination was tested and confirmed to perform as expected, the attitude control algorithms were implemented to control the system. The attitude determination and control simulation represented how the real system would respond and how fast the attitude of the system could be controlled. This chapter provides an overview of the MATLAB Simulink simulation environment that was used.

9.1 Simulation Components

A simplified block diagram of the simulation is shown in Figure 9.1.

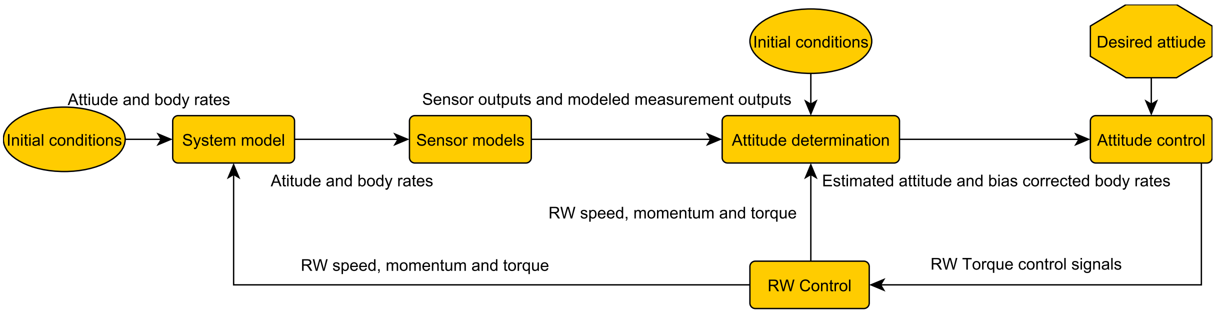

Initial conditions
Desired attiude
Attiude and body rates
Sensor outputs and modeled measurement outputs
Initial conditions
System model
Sensor models
Attitude determination
Attitude control
Atitude and body rates
Estimated attitude and bias corrected body rates
RW speed, momentum and torque
RW speed, momentum and torque
RW Torque control signals
RW Control

Figure 9.1: *Simulation Flow Block Diagram*

9.1.1 M-file

In order to start the simulation a MATLAB m-file was created to store the system constants. The equations of motion, as shown in Section 2.3, require these constants. Below the constants are listed.

- Inertia of the system, defined in Section A.5.
- Total mass of system, calculated in Section A.3.

The disturbance torque calculation requires the following constants:

- the friction constant of the air bearing, calculated in Section 2.4.1,
- the distance of the CoM below the CoP as calculated in Section A.2,
- and earth's gravity constant.

The simulation also requires starting conditions in the form of:

- initial attitude in Euler angles as defined in Section 2.2,
- and initial body rates.

These constants are loaded into the MATLAB workspace when the m-file is executed. The Simulink simulation is then ready to be executed.

9.1.2 The System Model Block

The system model block contains the dynamic equations shown in Section 2.3.1 in block diagram format. The system model block and the inner workings of the block are shown in Figure 9.2 and Figure 9.3.

The system model block also inputs the torque and momentum of the RWs as required by the dynamic equation. The attitude and angular rate of the system can be simulated given these variables and the constants of the system. The system model outputs the disturbance torques, the DCM matrix, the body rates and the attitude in a quaternion representation. The sensor outputs can be simulated using the body rate and DCM outputs. The attitude determination will attempt to estimate the output of the system model given the simulated sensor input.

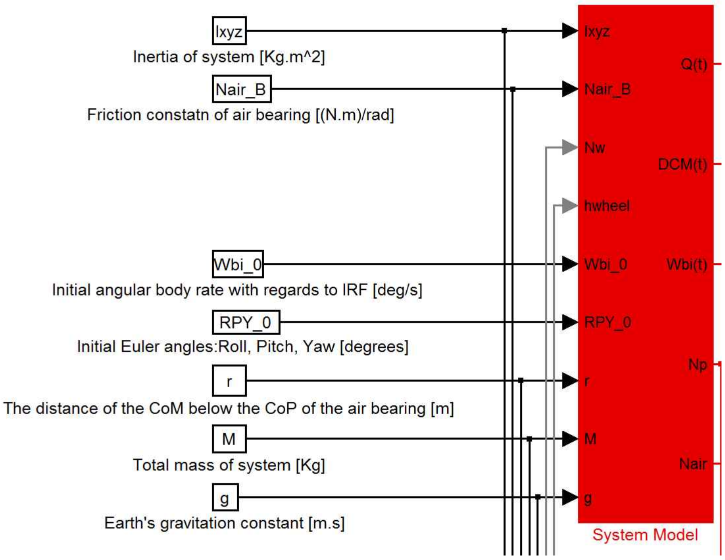

Ixyz
Inertia of system [Kg.m^2]
Nair_B
Friction constatn of air bearing [(N.m)/rad]
Wbi_0
Initial angular body rate with regards to IRF [deg/s]
RPY_0
Initial Euler angles:Roll, Pitch, Yaw [degrees]
r
The distance of the CoM below the CoP of the air bearing [m]
M
Total mass of system [Kg]
g
Earth's gravitation constant [m.s]
Ixyz
Nair_B
Nw
hwheel
Wbi_0
RPY_0
r
M
g
Q(t)
DCM(t)
Wbi(t)
Np
Nair
System Model

Figure 9.2: *System Model Block*

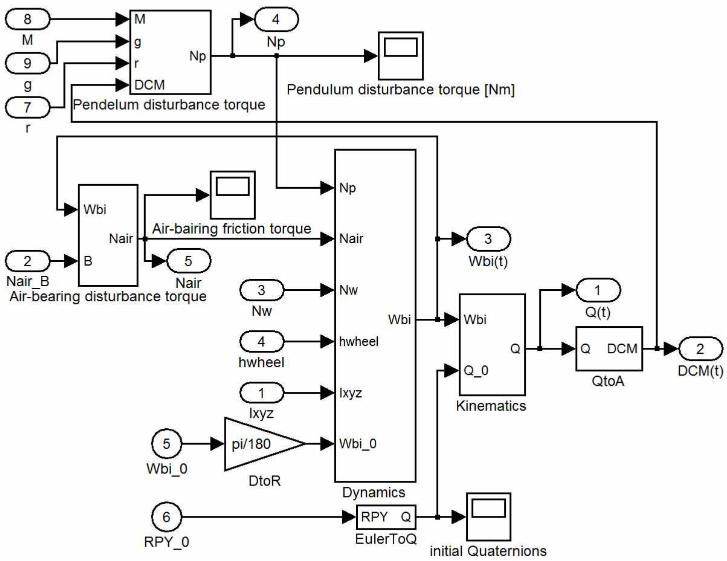


Figure 9.3: *System Model Block Inner Block Diagram*

9.1.3 Sensor Models Block

The sensor models have inputs from the DCM matrix and the body rates. The measured unit vector in the BRF of each sensor can be calculated given the DCM matrix and the modelled unit vector in the IRF. The modelled or simulated unit vector in the IRF for each sensor can be determined due to the fixed references in the room for the air bearing test. This is equivalent to the modelled unit vector to the sun, stars or earth in the IRF for a SC in orbit. However, these unit vectors stay constant to the references as they are fixed in the room. The modelled measurements are also given with the sensor measurements as outputs.

The output of the sensors can be simulated by taking the IRF modelled unit vector and transforming it to the BRF with the transformation matrix or DCM ($\mathbf{R}_{bi}$). This will lead to an ideal unit vector as the actual DCM is used. The sensor measurements can be simulated accurately if the appropriate transformations are performed and noise is added to resemble the raw output of the sensors as (shown in Chapter 3).

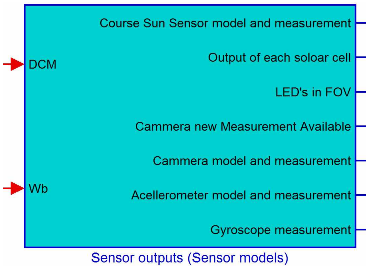

DCM
Wb
Course Sun Sensor model and measurement
Output of each soloar cell
LED's in FOV
Cammera new Measurement Available
Cammera model and measurement
Acellerometer model and measurement
Gyroscope measurement
Sensor outputs (Sensor models)

Figure 9.4: *Sensor Model Block*

For the CSS, the output of each photo diode is also given. This is to monitor and see if the outputs resemble the outputs shown in Appendix B.3. The yaw angle of the light source was also calculated and monitored to compare the results to the results in Figure B.8. For the camera sensor there is also an indicator, indicating if the lights are in the FoV and if a new measurement is available. The sensor data is sent to the attitude determination block.

9.1.4 Attitude Determination Block

The attitude determination block take the dynamic equation constants and the sensor model outputs as input. The attitude determination block outputs are the estimated DCM, body rates and disturbance torques, the innovation error of the EKF, the state covariance matrix and the estimated gyroscope bias.

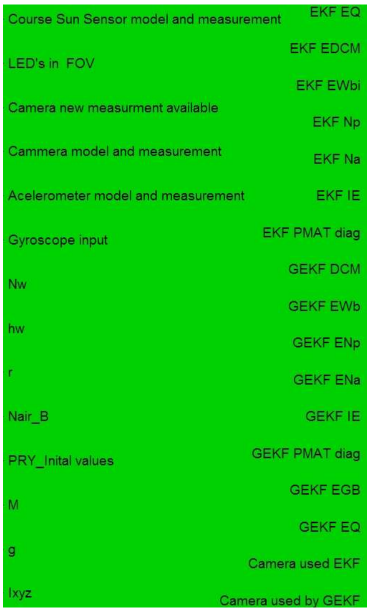


Figure 9.5: *Attitude Determination Block*

The algorithm implemented in the attitude determination block was discussed in Section 6.1.3.

9.1.5 Attitude Control Block

The attitude control block implements the algorithm that was discussed in Section 7.1. The inputs to the block are:

- the estimated quaternion,
- the command quaternion,
- the bias corrected body rates,
- the inertia of the system,
- the inertia of the RWs,
- and the RWs' speeds.

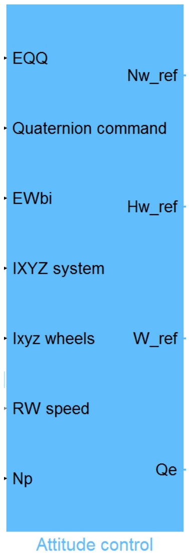


Figure 9.6: Attitude Control Block

The torque, momentum and speed reference for each RW are calculated using the quaternion feedback algorithm and the appropriate conversions. The speed reference is sent to the RW speed controllers

9.1.6 Reaction Wheel Controller

The RW controller implements the speed controller as was discussed in Section 4.2.8. This block also contains the model (plant transfer function) of each RW. As input the RW controller accepts the speed reference of each RW. Outputs are the speed, momentum and torque of each RW. These values are then used as input to the system model block and attitude determination block.

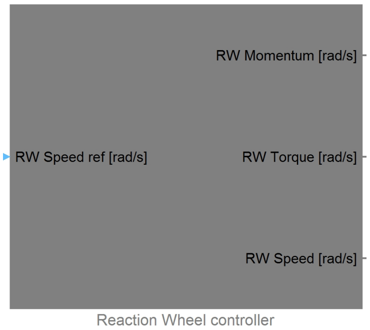


Figure 9.7: *Reaction Wheels Control Block*

A realistic starting condition is provided in the simulation as the system experiences on the air bearing. Movement is then simulated and the attitude determination algorithms implemented to estimate the attitude.

9.2 Attitude Determination Simulation

9.2.1 Simulation With All Sensors On

The system model and attitude determination block were given the initial conditions, as defined in Table 9.1, in order to validate the attitude determination algorithms. The system model block outputs the real attitude of the system whilst the attitude determination block uses the modelled sensor data to estimate the attitude.

Block	**[Roll,Pitch,Yaw]**	**Body rates**$[\omega_x, \omega_y, \omega_z]$
System block	[0,0,0]	[1,1,0.1] deg/s
Sensor: Gyroscope bias		$[1.1092, -1.26369, -1.99362]$ mrad/s
Attitude determination: EKF	[0,0,0]	[0,0,0] deg/s

Table 9.1: *Starting Conditions*

The reference LEDs will always remain in the FoV of the camera, with the starting conditions given in Table 9.1. Thus all the sensors are used to estimate the attitude. The output in each axis is shown in Figure 9.8 to 9.10.

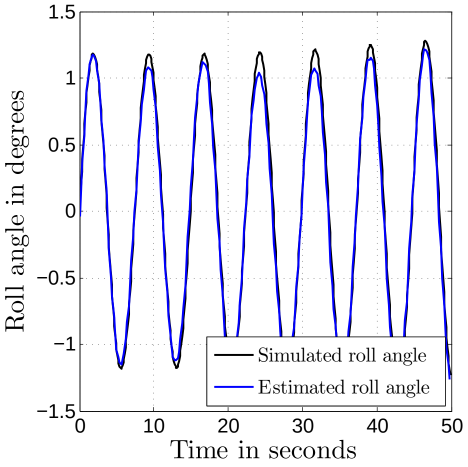


Figure 9.8: *Roll Angle*

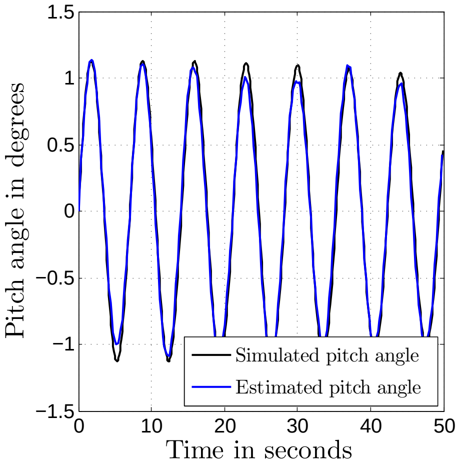


Figure 9.9: *Pitch Angle*

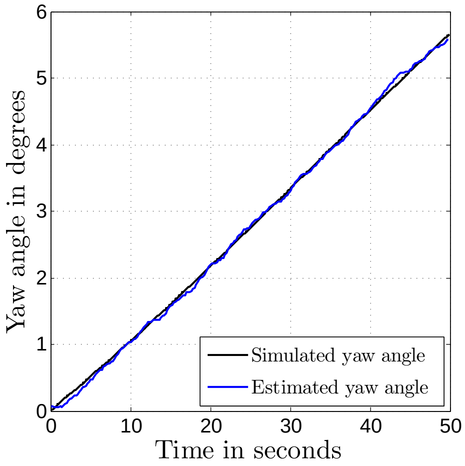


Figure 9.10: *Yaw Angle*

There is a small error between the estimated and simulated attitude. The error in degrees for each axis is shown in Figure 9.11 to Figure 9.13.

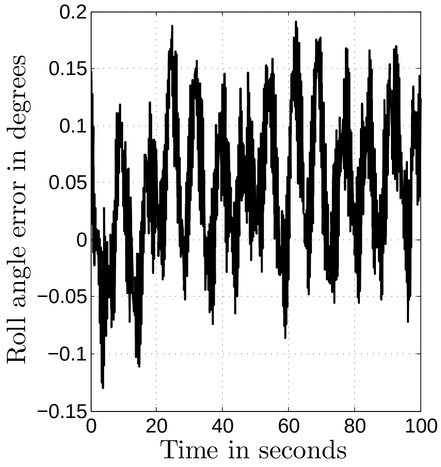


Figure 9.11: *Roll Angle Error*

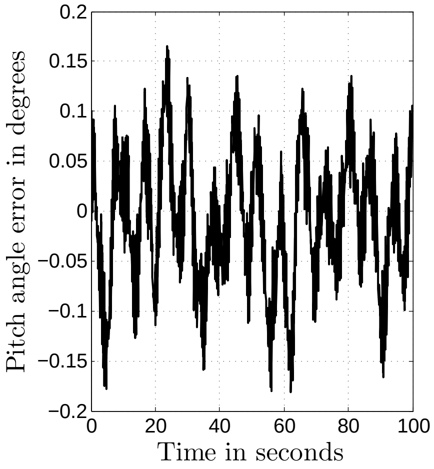


Figure 9.12: *Pitch Angle Error*

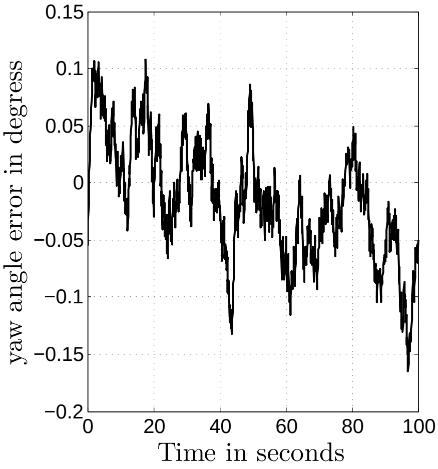


Figure 9.13: *Yaw Angle Error*

The error for each axis in the above test remains below 0.2 degrees. However, this will change if the camera turns off when the LEDs moves out of the camera FoV. The bias of the gyroscope is also estimated in each axis. The gyroscope modelled measurements were given a bias for each axis as defined in Table 9.1. Figure 9.14 shows the bias estimation values settling close to the specified gyroscope bias values which indicates that the bias was correctly estimated. The bias estimation takes some time to settle close to the correct values. The initializing of the EKF bias close to the correct settling values will result in the bias estimation to settle faster.

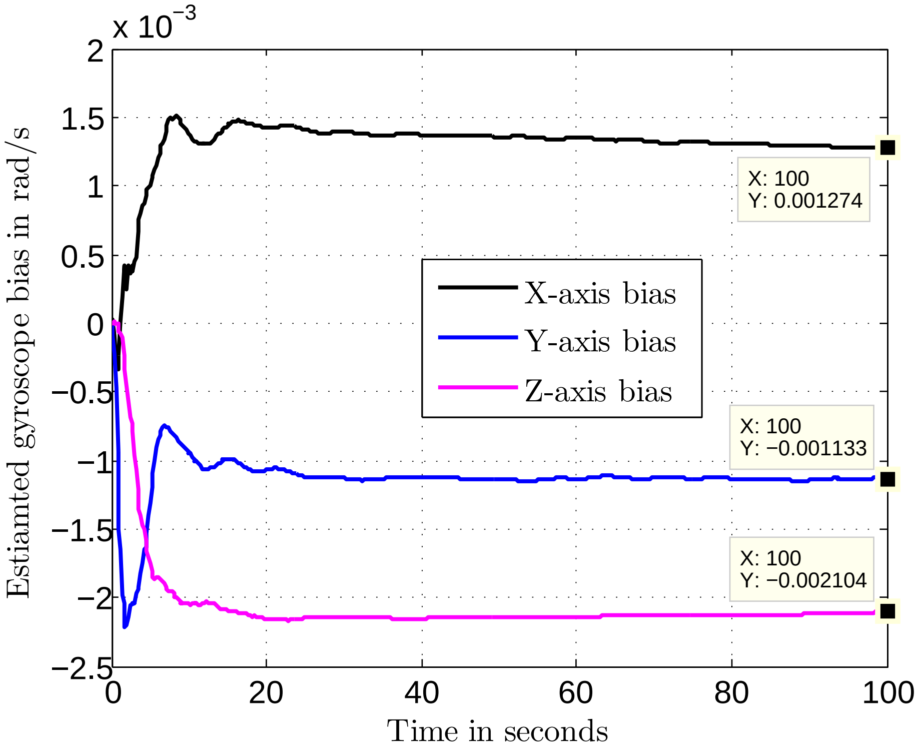


Figure 9.14: *Estimated Gyroscope Bias*

9.2.2 Simulation With Camera Turning On And Off

This section demonstrates what happens when the reference LEDs moves out of the camera's FoV and the camera turns off. Therefore the next simulation will be initiated with a larger yaw body rate. The EKF will have to rely on the gyroscope and CSS measurements to estimate the attitude. The starting conditions are shown in Table 9.2.

	[Roll,Pitch,Yaw]	$[\omega_x, \omega_y, \omega_z]$
Simulation	[0,0,0]	[1,1,-4] deg/s
Sensor: Gyroscope bias		[1.1092, −1.26369, −1.99362] mrad/s
EKF EKF Gyroscope bias	[0,0,0]	[0,0,0] deg/s [1.1, −1.2, −1.9] mrad/s

Table 9.2: Starting Conditions

Figure 9.15 to Figure 9.17 indicate the simulated and estimated roll pitch and yaw angles.

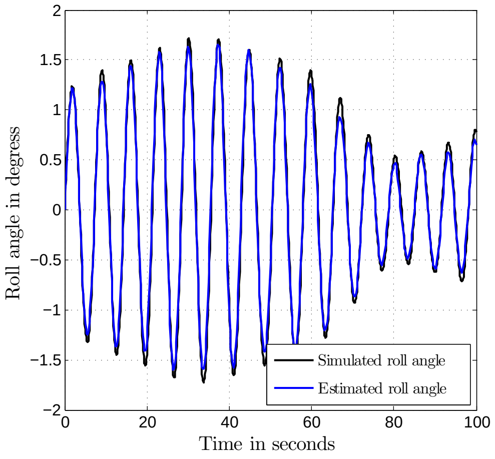


Figure 9.15: *Roll Angle*

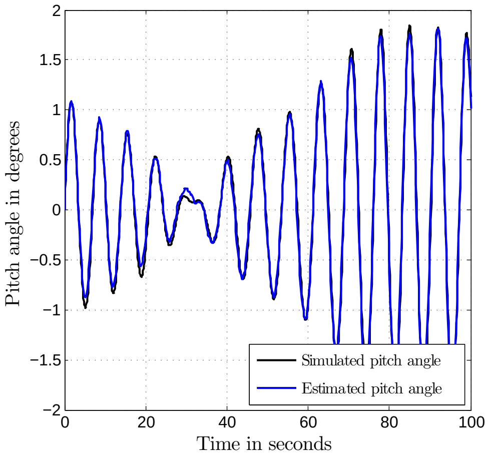


Figure 9.16: *Pitch Angle*

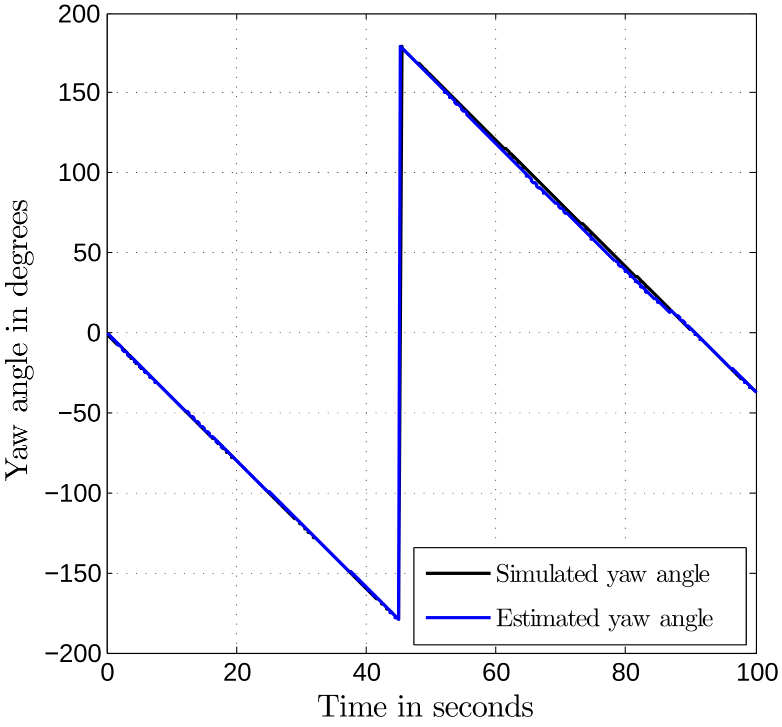


Figure 9.17: *Yaw Angle*

Figure 9.17 shows the yaw angle estimation and true yaw angle. The LEDs move out of the FoV and the camera turns off at about 4 seconds . The yaw angle is propagated with the CSS and gyroscope readings. The estimated angle starts to deviate from the simulate angle due to the CSS not being as accurate. The errors between the estimated and simulated angles are shown in Figure 9.18 to 9.20.

The estimated gyroscope bias also starts to deviate as shown in Figure 9.21. The camera turns on again at 87 seconds. Then the large errors in attitude and bias estimation are corrected.

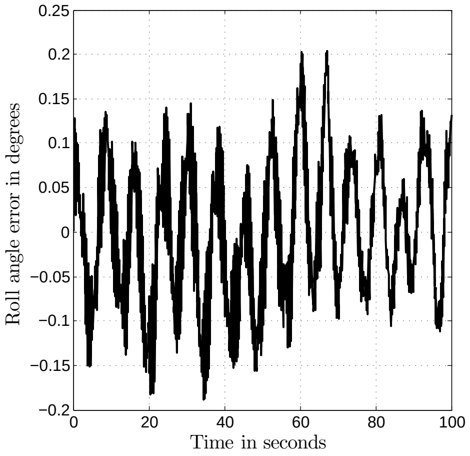


Figure 9.18: *Roll Angle Error*

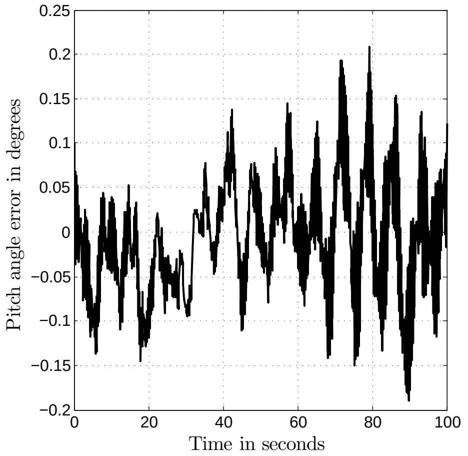


Figure 9.19: *Pitch Angle Error*

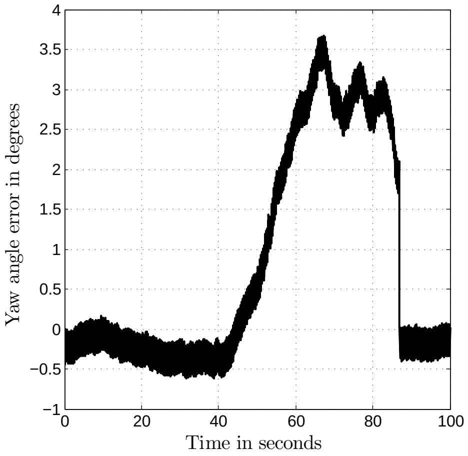


Figure 9.20: *Yaw Angle Error*

If the more accurate camera sensor is not used, the estimated attitude will begin to deviate from the simulated attitude. Then as the camera turns on the error between the estimated and simulated attitude is reduced.

9.3 Attitude Control Simulation

The quaternion feedback attitude controller was implemented in C-code and contained in the attitude controller simulation block (as discussed in Chapter 7). The modelled torque and momentum generated by the RWs' are fed back to the attitude determination and system model blocks. Then the system and estimator can react to the changing states of the RWs.

The performance of the attitude control can be demonstrated by giving the attitude control block different command attitudes and monitoring the system's responses. Although, there is an error between the actual attitude of the satellite and the estimated attitude,

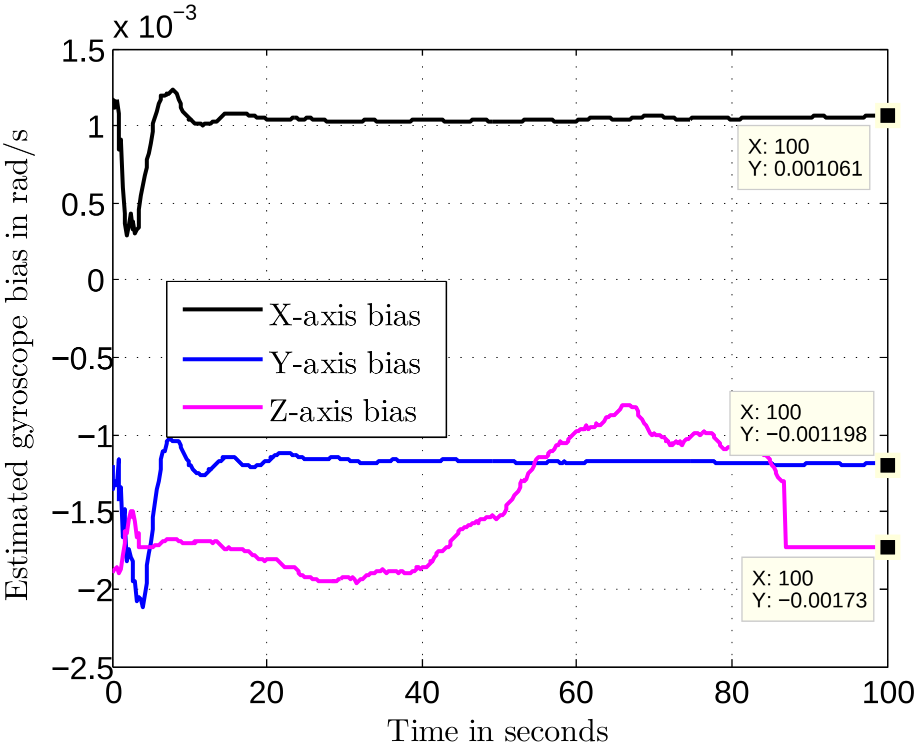


Figure 9.21: *Estimated Gyroscope Bias*

this error cannot be measured in reality. The error can be obtained in simulation because the estimated and simulated attitudes are known. The attitude controller assumes that the estimated attitude is the true attitude and the control commands are generated based on the estimated attitude. Only the response of the estimated attitude will be considered in the evaluation of the controller. As shown in Section 9.2.2, the error between the estimated attitude and true attitude can become large, but the use of the more accurate camera sensors can correct this error.

9.3.1 Quaternion Feedback Simulation

9.3.1.1 10 Degree Yaw Step

The simulation begins with all the sensors being active and the tracking LED present in the FoV of the camera. The system is initialized with zero body rates and a zero or rest attitude. At ten seconds a 10 degree yaw step command is given. This step command allows the LEDs to remain within the FoV of the camera. Thus the step will be completed with a small error from the actual attitude. The yaw angle step is shown in Figure 9.22, with a close up view of the settling angle in Figure 9.23. The yaw angle settles within 0.1 degrees of the commanded angle. The error between the simulated and estimated yaw angle is also small as shown in Figure 9.24. The roll and pitch angles are kept within 0.1 degree of the zero attitude. Figure 9.25 shows the roll and pitch angles. The step is completed with a settling time of 8.65 seconds and not the specified 5s as designed in

Section 7.1.1. This is due to the limited torque available from the motors.

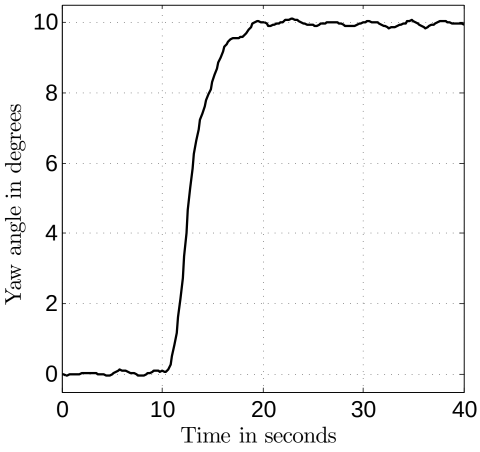


Figure 9.22: *Yaw Angle*

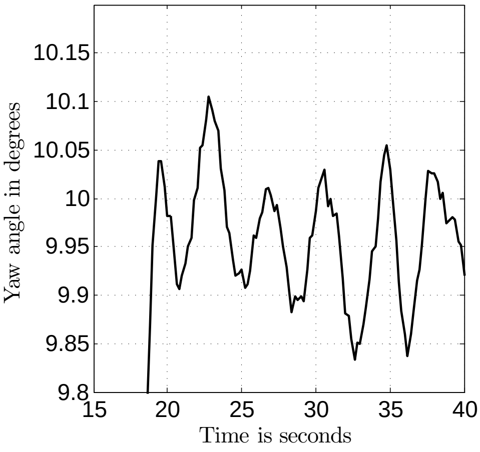


Figure 9.23: *Close-Up on Settling Yaw Angle*

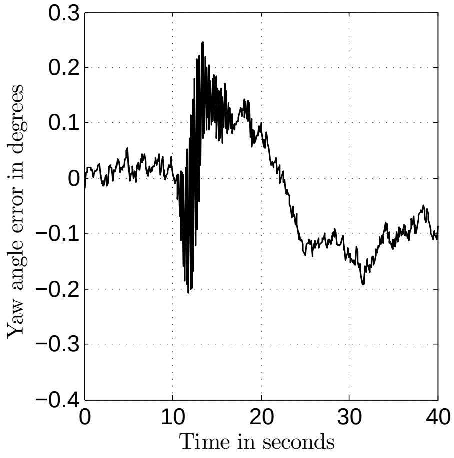


Figure 9.24: *Yaw Angle Error*

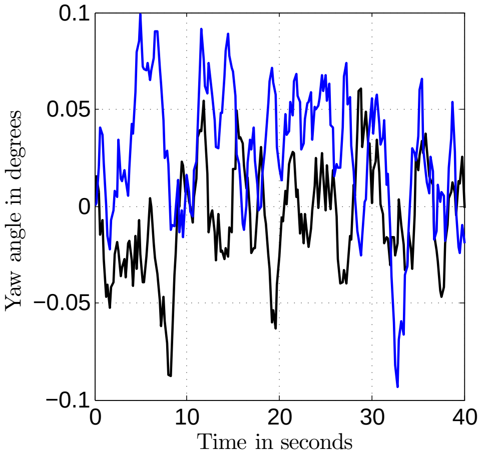


Figure 9.25: *Roll and Pitch Angles*

The torque command to each RW is shown in Figure 9.26 to Figure 9.28. The torque commands looks noisy because of the noisy body rates obtained from the gyroscope. The torque command saturates at 5mNm because this is the maximum continuous torque the motors can supply. The Z_b-axis RW immediately receives a maximum torque command at the 10 s step to accelerate the motor. Thereafter, a maximum torque command in the opposite direction is given to slow the motor down again.

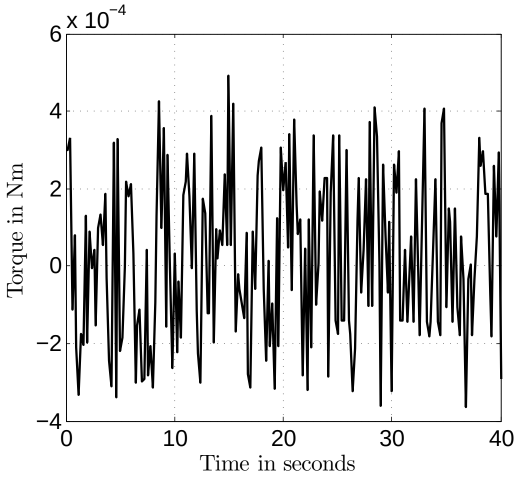


Figure 9.26: *X_b-Axis RW Torque*

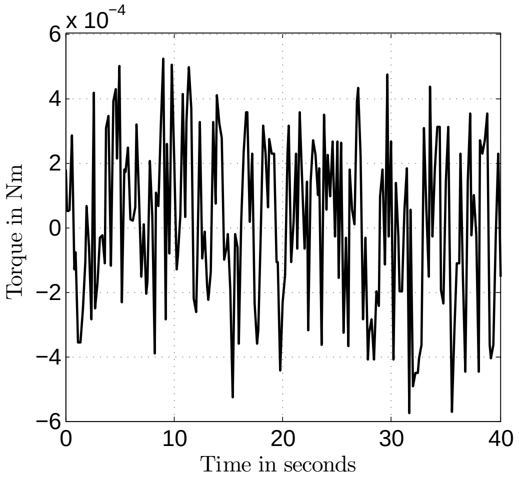


Figure 9.27: *Y_b-Axis RW Torque*

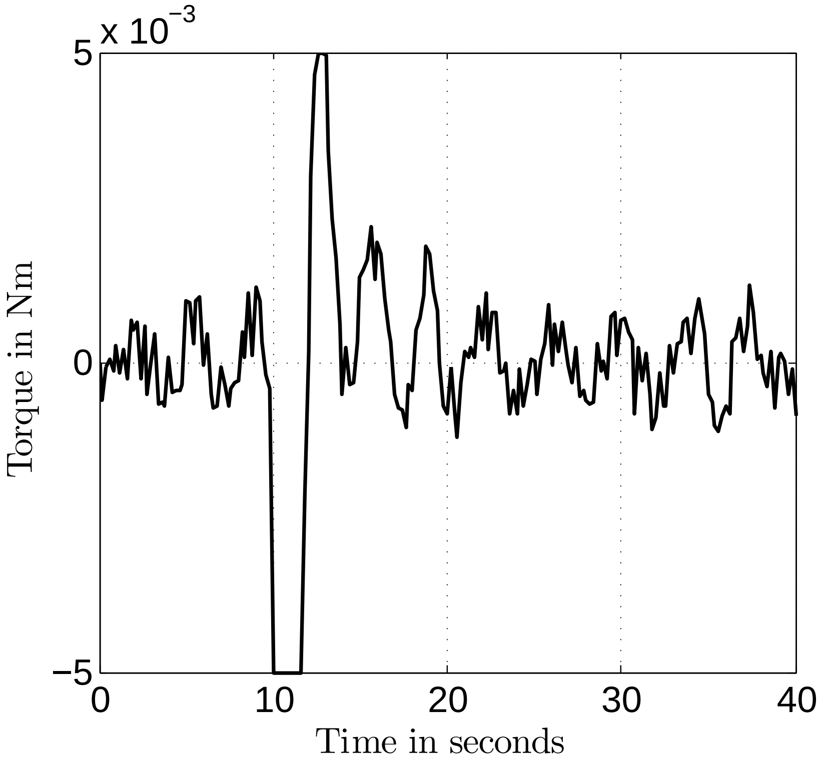


Figure 9.28: *Z_b-Axis RW Torque*

The speed of each RW is shown in Figure 9.29 to 9.31. The Z_b-axis RW is accelerated as fast as possible until a deceleration occurs to stop the rotation before the 10 degree yaw mark is overshot. The X_b- and Y_b-axis RWs' speed varies slowly to compensate for the roll and pitch body rates caused by the disturbance torques.

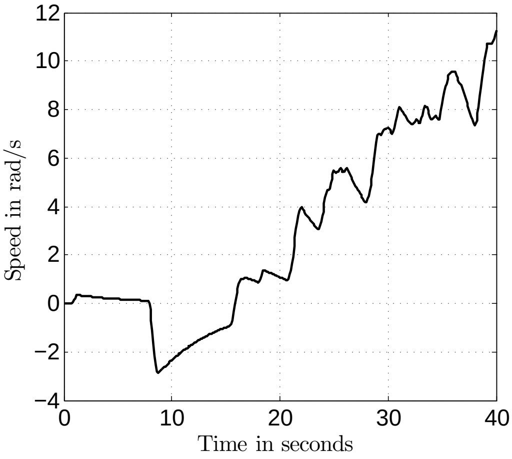


Figure 9.29: X_b-*Axis RW Speed*

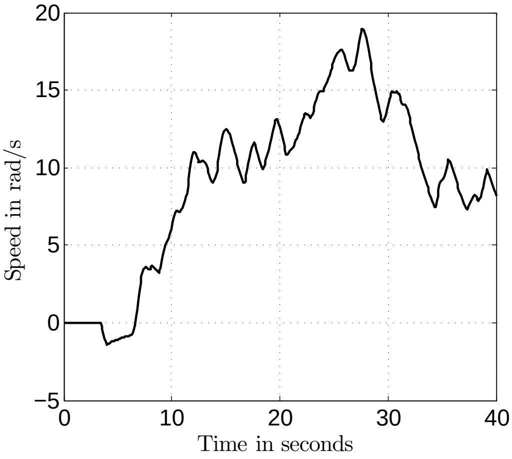


Figure 9.30: Y_b-*Axis RW Speed*

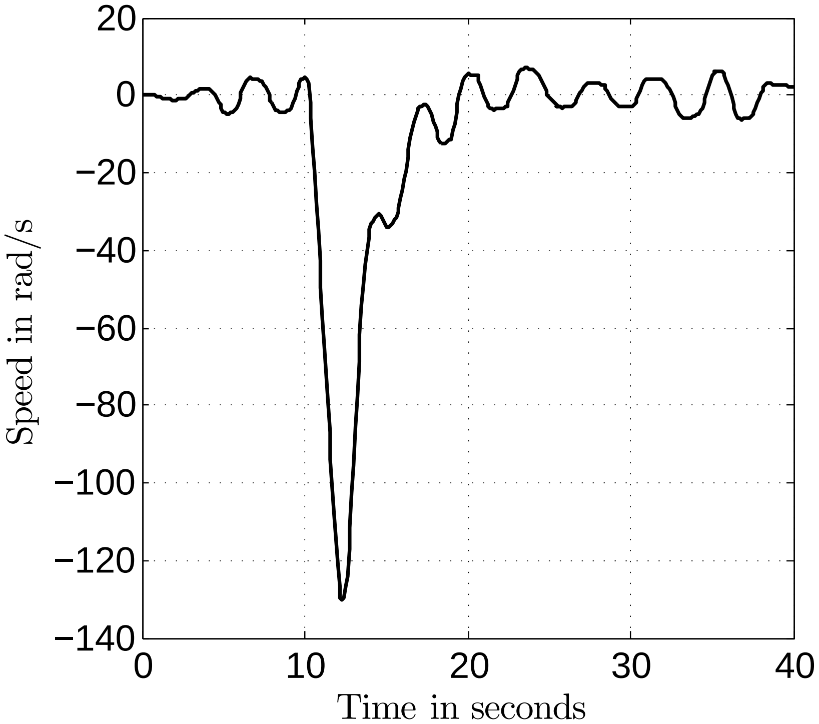


Figure 9.31: Z_b-*Axis RW Speed*

Although, the step time is larger than the specified settling time, the limited power of the actuators makes this time frame acceptable. Therefore, the system responds as fast as what is allowed by the actuators. The next section demonstrates a 90 degree yaw step to simulate what happens when the camera turns off.

9.3.1.2 90 Degree Yaw Step

The LEDs will move out of the camera's FoV for a 90 degree step. The yaw step is shown in Figure 9.32. The step is done with a settling time of 15.96 s. The step seems to be more accurate due to the CSS being less sensitive than the camera. The enlarged view of the steady state response is shown in Figure 9.33, and shows an error of less than 0.1 degrees. However, the error between the actual attitude and estimated attitude is larger than the error obtained with the 10 degree step which is shown in Figure 9.24. The error for the 90 degree step is as large as 1.5 degrees, as presented in Figure 9.34.

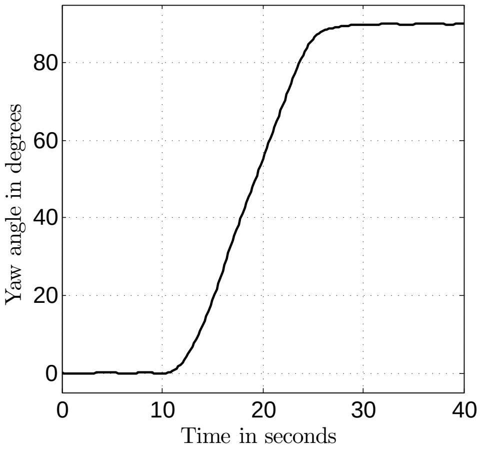


Figure 9.32: *Yaw Angle*

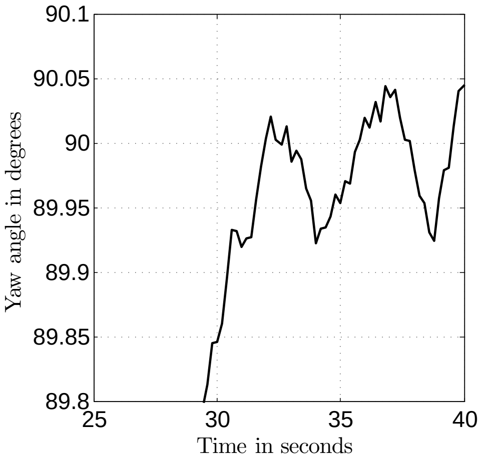


Figure 9.33: *Enlarged view of Settling Yaw Angle*

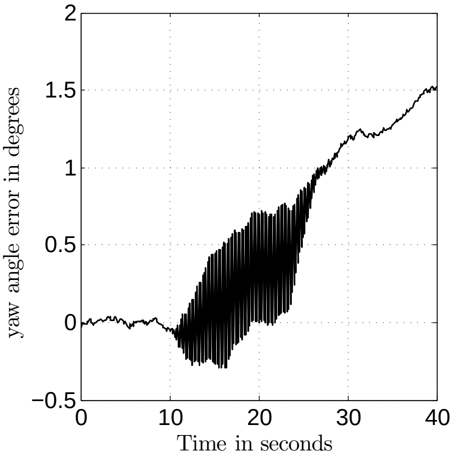


Figure 9.34: *Yaw Angle Error*

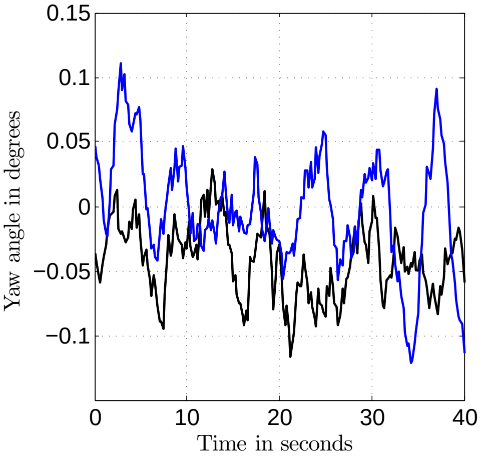


Figure 9.35: *Roll And Pitch Angles*

The torque commands for the 90 degree yaw step of each RW is shown in Figure 9.36 to 9.37. The X_b- and Y_b-axis RWs' torques are the same for the 10 degree step and the 90 degree step. The Z_b-axis RW shows a saturated torque command at 10 s which drops to a zero torque command, at 14 s the maximum RW speed is reached and as a result zero more torque can be generated.

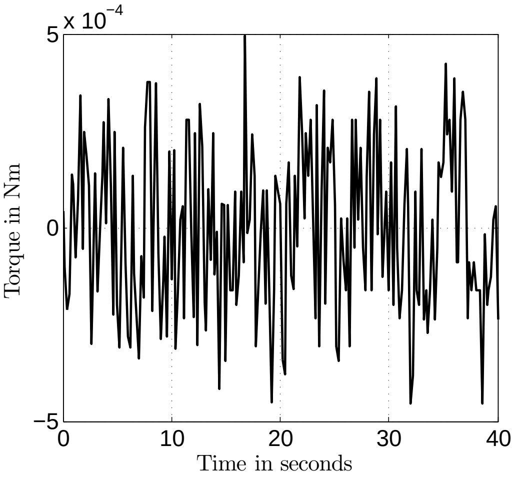


Figure 9.36: *X_b-Axis RW Torque*

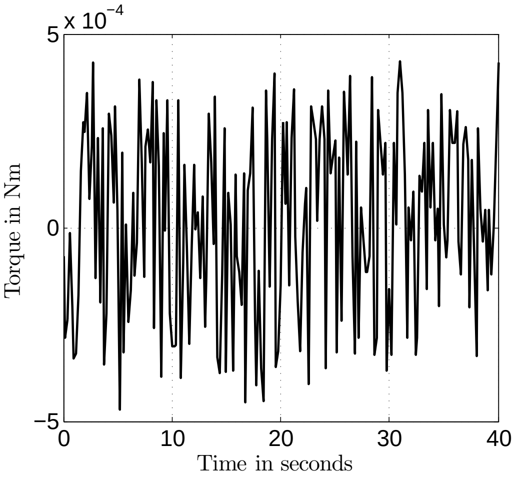


Figure 9.37: *Y_b-Axis RW Torque*

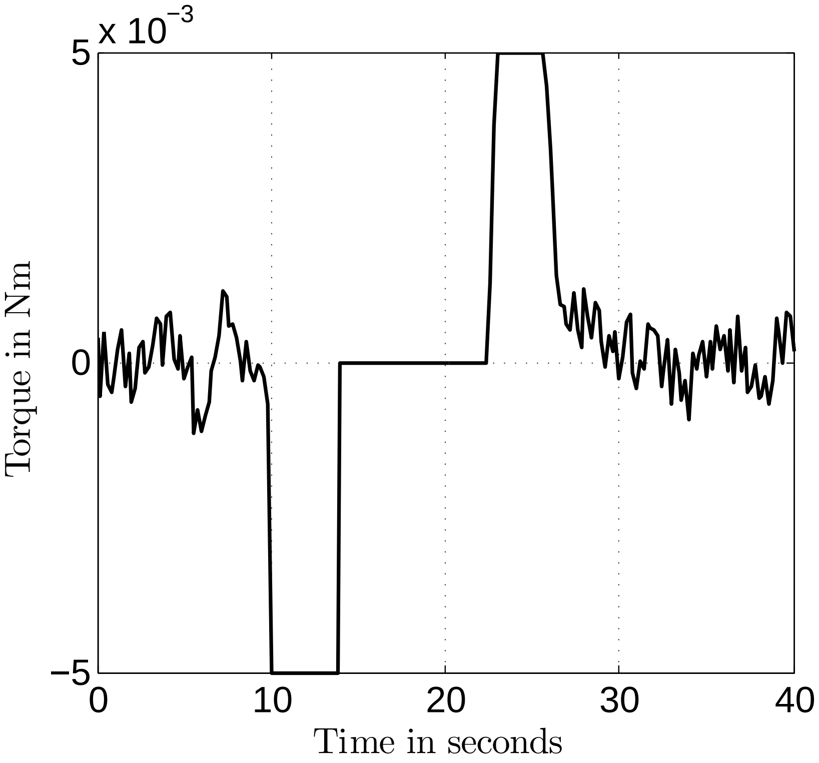


Figure 9.38: *Z_b-Axis RW Torque*

The speeds of the RWs are shown in Figure 9.39 to 9.41. The top speed limit is reached by the Z_b-axis RW speed as shown in Figure 9.41.

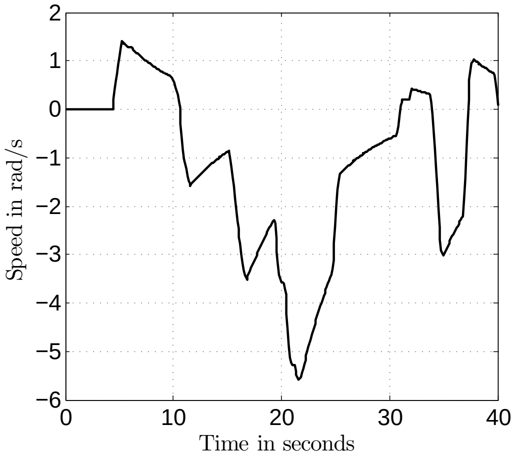


Figure 9.39: X_b*-Axis RW Speed*

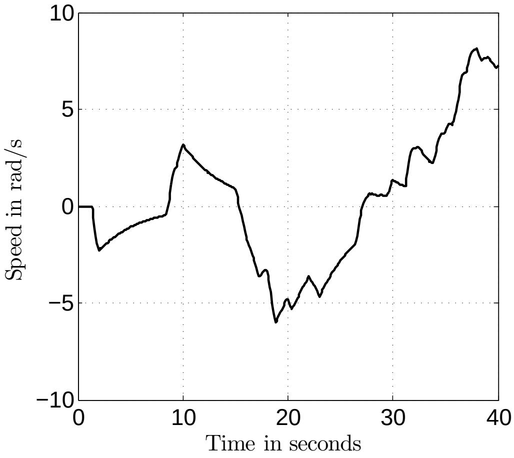


Figure 9.40: Y_b*-Axis RW Speed*

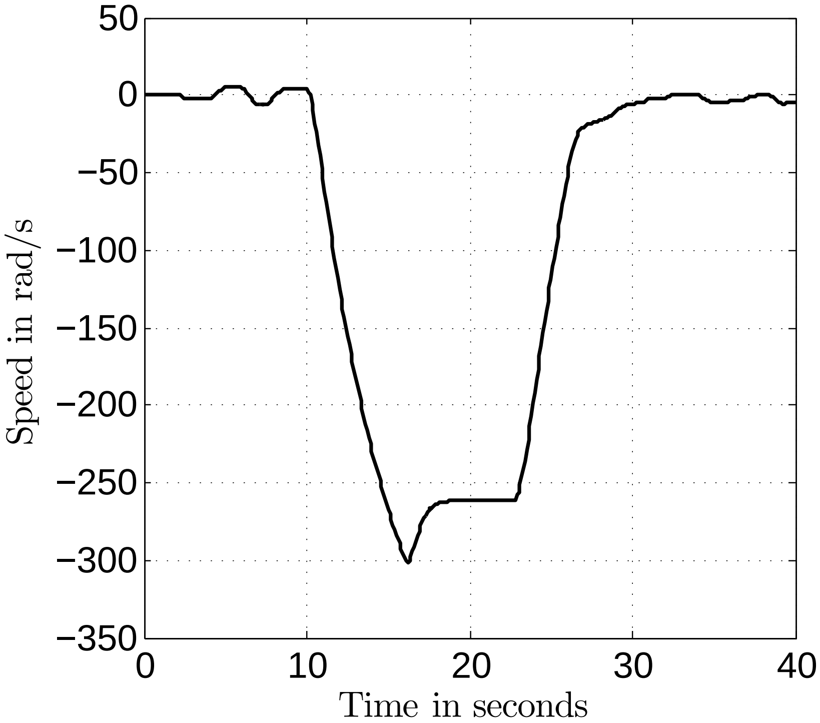


Figure 9.41: Z_b*-Axis RW Speed*

9.4 Simulation Summary

The simulation displayed promising results. The algorithms discussed in Chapter 6 and Chapter 7 were confirmed to work as expected. In addition, the simulation gave an idea of how the real system would respond. After the simulation all the algorithms on the ADCS_V2 module need to be implemented and the system tested on the air bearing.

Chapter 10

HIL Tests and Results

10.1 Attitude Determination HIL

The first HIL test was conducted to confirm that the attitude determination algorithms implemented on the MCU were performing the same as the algorithms implemented in the MATLAB Simulink simulation. The sensor data generated by the sensor model block in MATLAB was sent to the MCU on the ADCS_V2 PCB via the wireless link. The sensor data was used as input to the attitude determination algorithms. Thereafter the output of the algorithms were sent back over the wireless link and could be compared with the output of the algorithms implemented in MATLAB. The results of the two algorithms were almost identical. This confirmed that the simulated and implemented algorithms on the MCU were the same and had the same performance. Figure 10.1 demonstrates this setup in block diagram form.

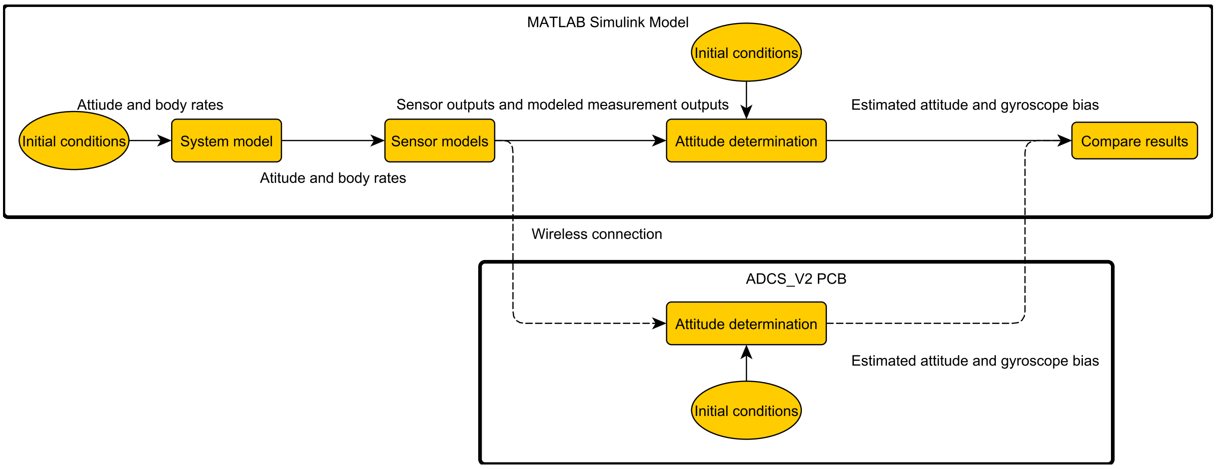


Figure 10.1: *HIL Data Flow*

10.1.1 ADCS HIL

The setup for the air bearing test is shown in Figure 10.2. EyasSAT was placed on the air bearing and turned on. EyasSAT needs to be rotated through 360 degrees in order for all the photo-diodes to go through a maximum and minimum light exposure to allow for calibration of the CSS. Therefore, the minimum and maximum output values can be stored and used for calibration as discussed in Section 3.1.2. EyasSAT is then placed at a desired rest or zero attitude position. The current position of the LEDs, light and gravity vector is stored as the IRF reference when the attitude determination algorithm is started. These vectors act as the modelled sensor measurements as discussed in Chapter 6. The attitude data is transmitted to a MATLAB Simulink interface and displayed. A signal build block in Simulink can be used to command the attitude of EyasSAT. A number of different attitude commands were given and the sections below show figures and provide discussions of relevant findings.

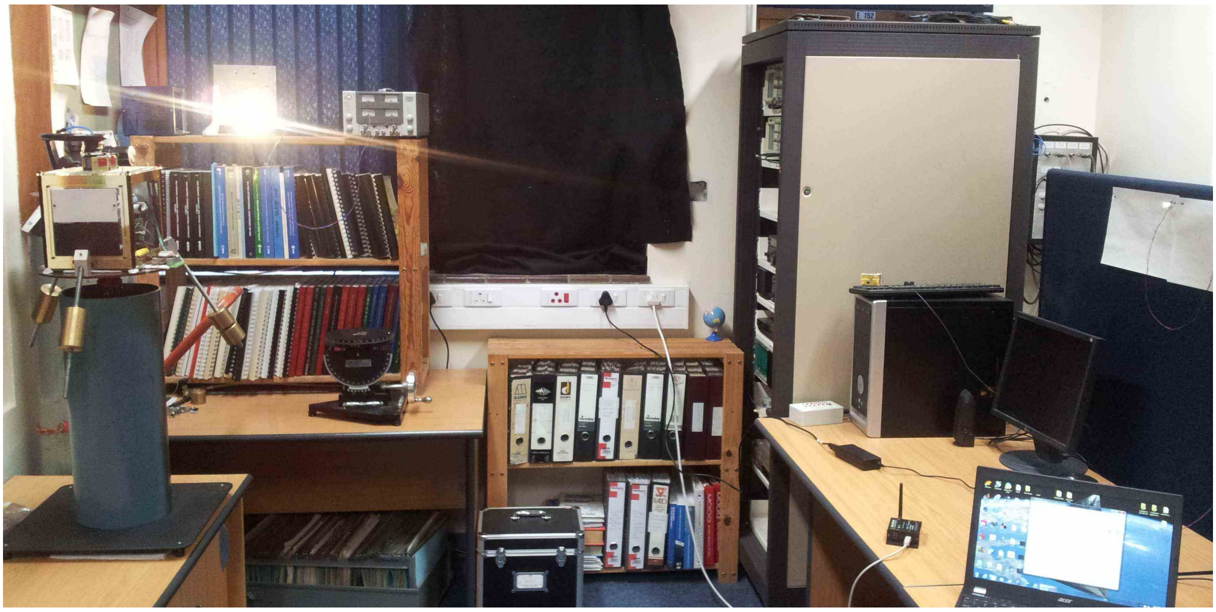


Figure 10.2: *Air Bearing Test Setup*

10.2 10 Degree Yaw Step

The control is not active for the first 10 seconds in the HIL test. Thereafter, the control is activated and given a zero roll, pitch and yaw command. A 10 degree yaw step command is given at 20 s. This attitude is maintained for 40 s before a zero yaw angle is commanded. Figure 10.3 to 10.6 shows the roll, pitch and yaw angel of EyasSAT.

The black signal represents the estimated roll, pitch and yaw angle. The green signal is the estimated roll and yaw angle obtained from the camera. The magenta signal is the estimated roll and pitch angel obtained from the accelerometer. The blue signal is the attitude command signal.

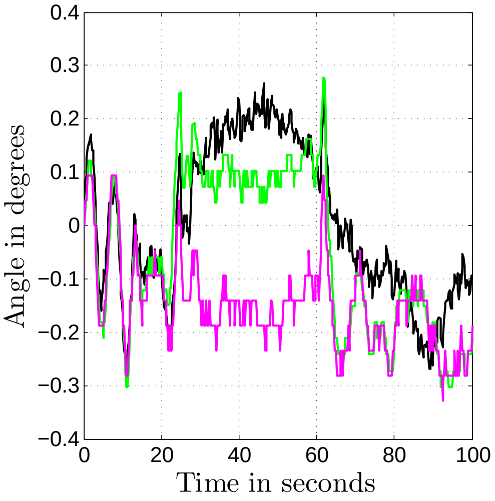


Figure 10.3: *Roll Angle*

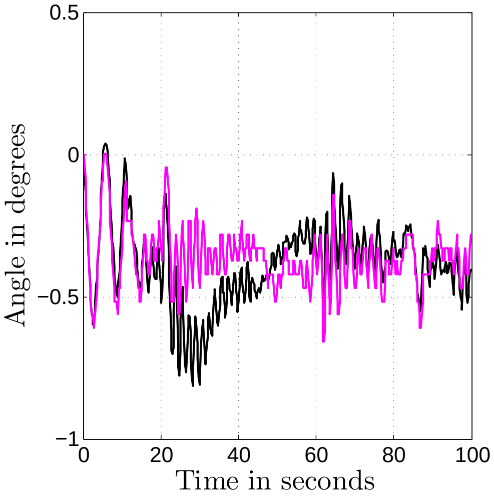


Figure 10.4: *Pitch Angle*

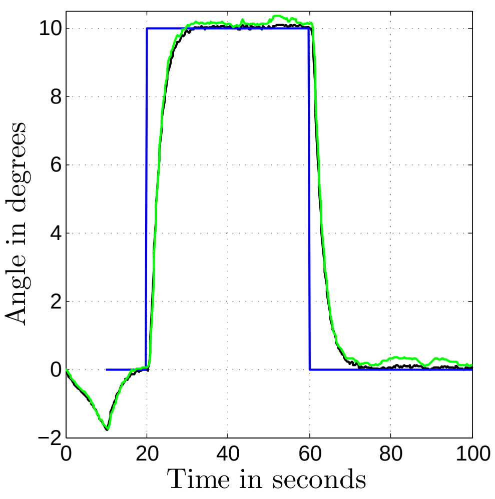


Figure 10.5: *Yaw Angle*

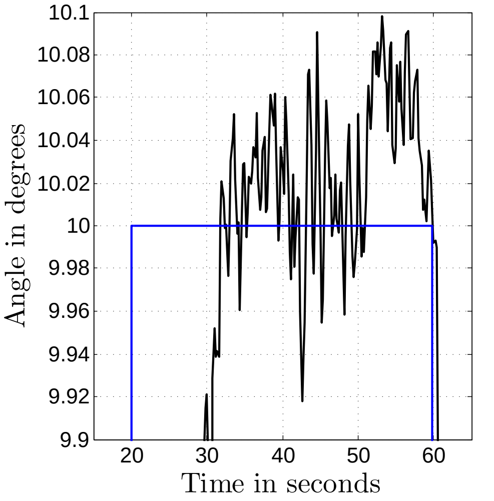


Figure 10.6: *Yaw Angle Enlarged*

The yaw step response has a settling time of 9.45 seconds. The settling time obtained in the simulation is slightly less than this settling time. This is because air friction was not modelled in the RW simulation. The speed of the RW is compared with the simulated RW speed, as demonstrated in Figure 10.9. The simulated RW speeds up to 130 rad/s whereas the real RW only reaches about 85 rad/s. Although, the steady state error is kept below 0.1 degrees, the simulation results indicate that the error is slightly less.

The roll and pitch axes are not angle controlled but rate controlled in order to keep the air bearing surface stable. The air bearing surface might not have been at a zero roll and pitch angle when the EKF was initialised. Thus the estimated attitude will settle or oscillate around offset angles. The roll angle contains a 0.4 degree peak-to-peak angle oscillation and the pitch contains a 0.5 degree peak-to-peak angle oscillation before the 10 s time mark. These oscillations were damped. As a result, the roll amplitude was reduced to an offset angle of -0.15 degree and the pitch amplitude was reduced to an offset angle of -0.4 degree.

The speed responses of the RWs are shown in Figure 10.7 to 10.9. The X_b- and Y_b-axis RWs remain at low speed whilst attempting to damp the roll and pitch body rates. The Z_b-axis RW speed seems to be slowly decreasing over time in order to counter the disturbance torque generated by the aerodynamic drag on the air bearing.

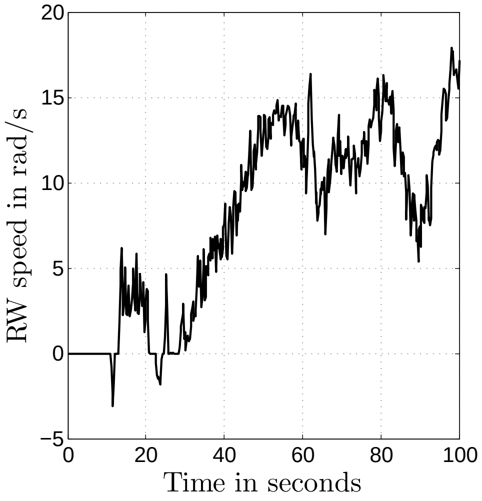


Figure 10.7: *X_b-Axis RW Speed*

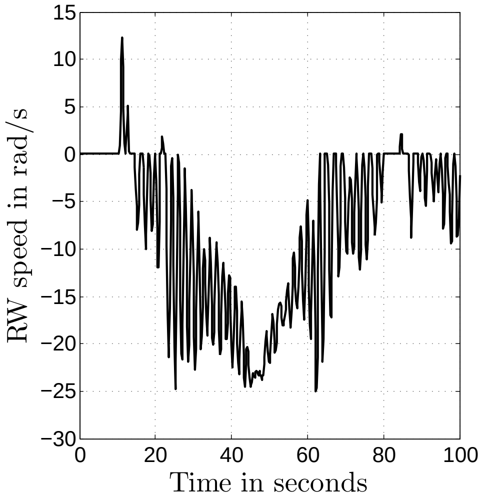


Figure 10.8: *Y_b-Axis RW Speed*

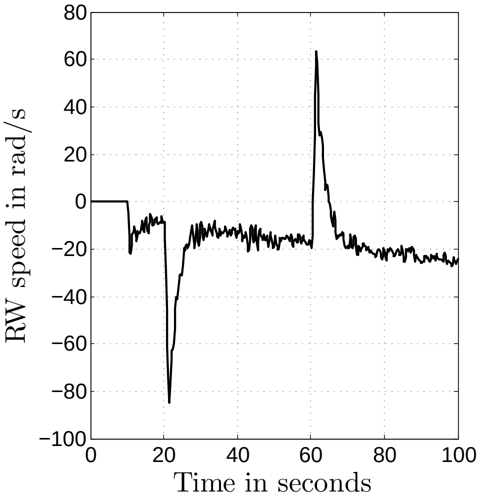


Figure 10.9: *Z_b-Axis RW Speed*

The bias corrected body rates, in rad/s, are shown in Figure 10.10 to 10.12. The roll rate is damped after 10 s but experiences an increase in amplitude after the yaw step is initiated. However, this is corrected and the roll rate is kept to below 5 mrad/s. The pitch rate increased at the 20 and 60 seconds mark, where the yaw steps are performed. However, the rates are damped once at a steady state to below 5 mrad/s.

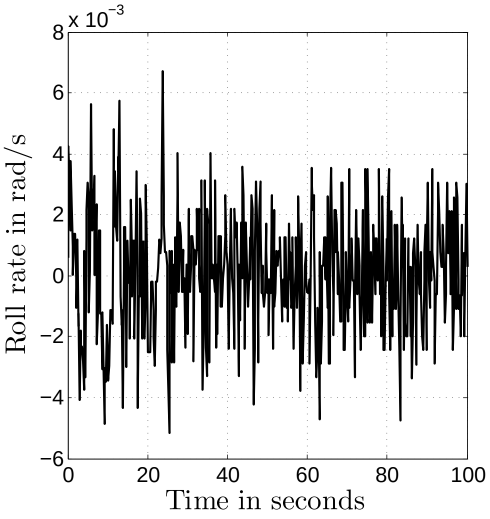


Figure 10.10: *Roll Rate*

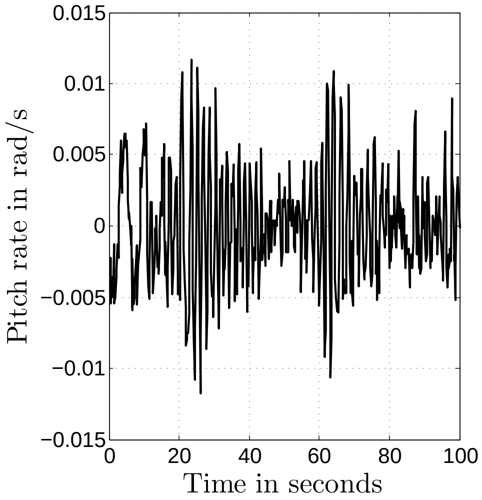


Figure 10.11: *Pitch Rate*

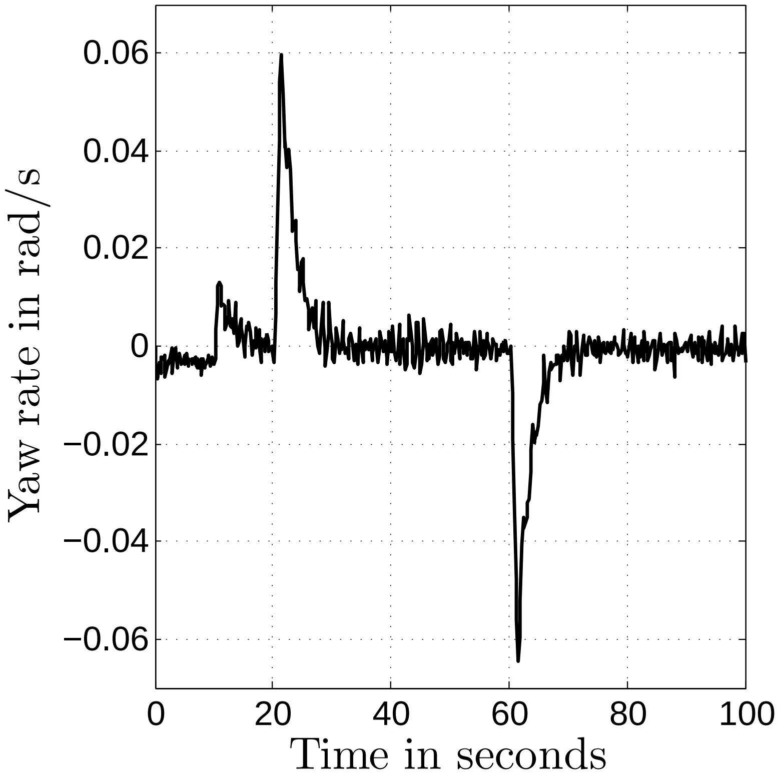


Figure 10.12: *Yaw Rate*

The ADCS performance is further tested by commanding a 90 degree step, which is discussed in the next section.

10.3 90 Degree Yaw Step

Similar to the 10 degree yaw step example, the control is activated at 10 seconds and the step is initiated at 20 seconds. A zero yaw angle step is initiated at the 70 second mark.

The roll angle is depicted in Figure 10.13 with the black signal indicating the estimated roll angle. The blue signal is the estimated roll angle obtained from the accelerometer. The green signal is the estimated roll angle from the camera. The camera turns off when the LEDs moves out of the FoV and a zero signal is indicated. In Figure 10.13, the camera measurement contains an offset. This causes a steady signal at 0.36 degrees and this part of the signal can be ignored. Figure 10.14 shows the EKF estimated pitch angle, which is indicated by the black signal and the blue signal describing the accelerometer estimated angle. Roll and pitch angles are only rate controlled as implemented with the 10 degree yaw step.

Figure 10.15 shows the yaw angle. The yaw angle is measured by the camera where the LEDs are in the FoV and is indicated by the green signal. The yaw angle measured by the CSS is shown by the magenta signal. Furthermore, the accumulated angle from the IMU is indicated by the red signal. The drift contained in the accumulated angle was discussed in Section 3.3.3. The black signal represents the EKF estimated yaw angle. A settling time of 13.34 seconds was indicated in the step response, which is faster than the 15.96 second settling time predicted by the simulation. The steady state error shows a 0.05 degree deviation from the 90 degree mark. This seems more accurate than the 10 degree step, but as seen in the simulation, this is as a result of the less sensitive CSS.

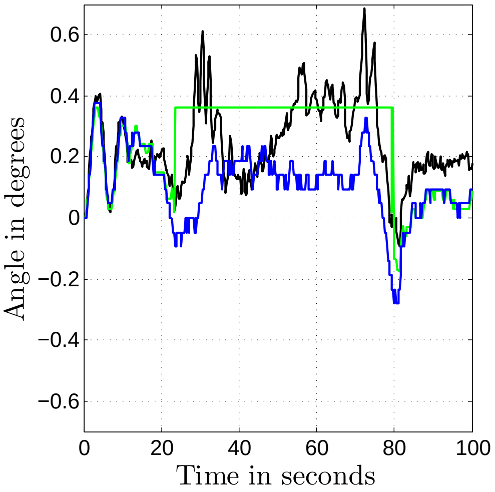


Figure 10.13: *Roll Angle*

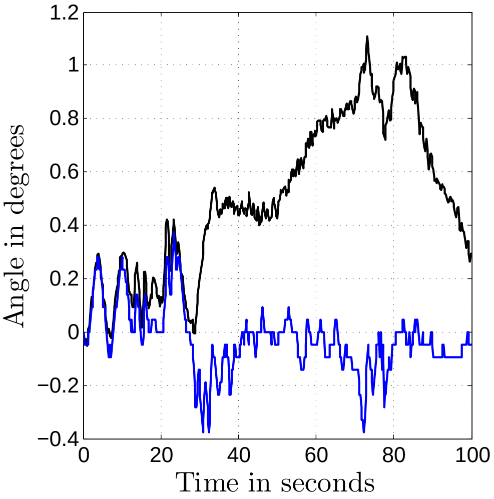


Figure 10.14: *Pitch Angle*

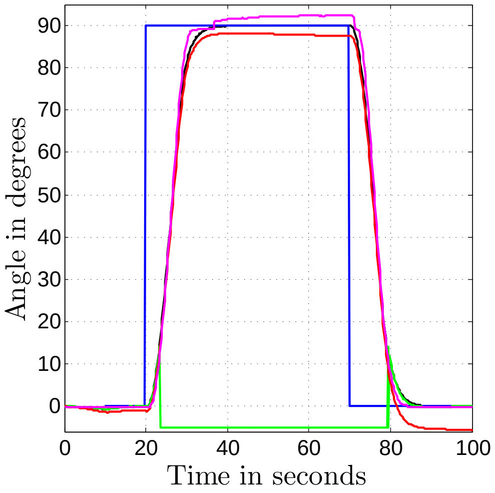


Figure 10.15: *Yaw Angle*

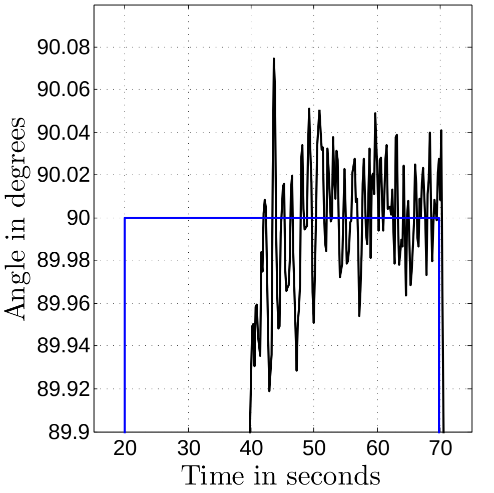


Figure 10.16: *Yaw Angle Enlarged*

The speed of the X_b- and Y_b-axis RWs are shown in Figure 10.17 and 10.18. The speed of the RWs remains relatively low to compensate for the roll and pitch body rates.

The Z_b-axis RW speed is shown in Figure 10.19. The RW accelerates to its maximum speed at the 10 s and 70 s mark to execute the 90 degree yaw angle step. The speed step response of the RW is better than the response shown in the simulation, as the real response is seen to contain less overshoot.

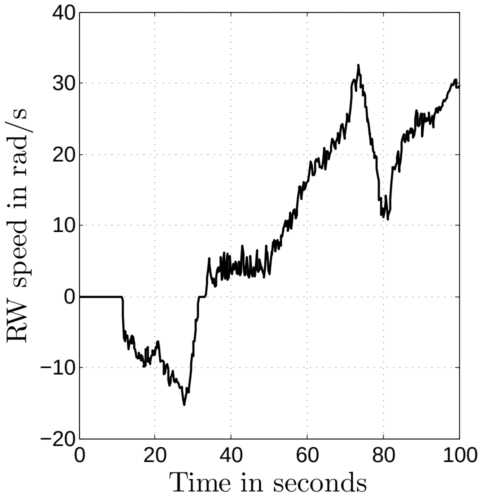


Figure 10.17: *X_b-Axis RW Speed*

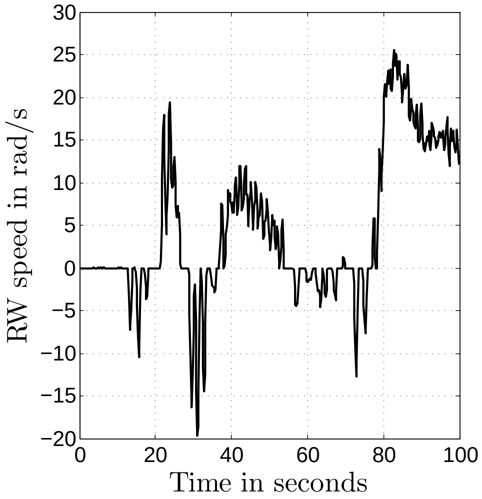


Figure 10.18: *Y_b-Axis RW Speed*

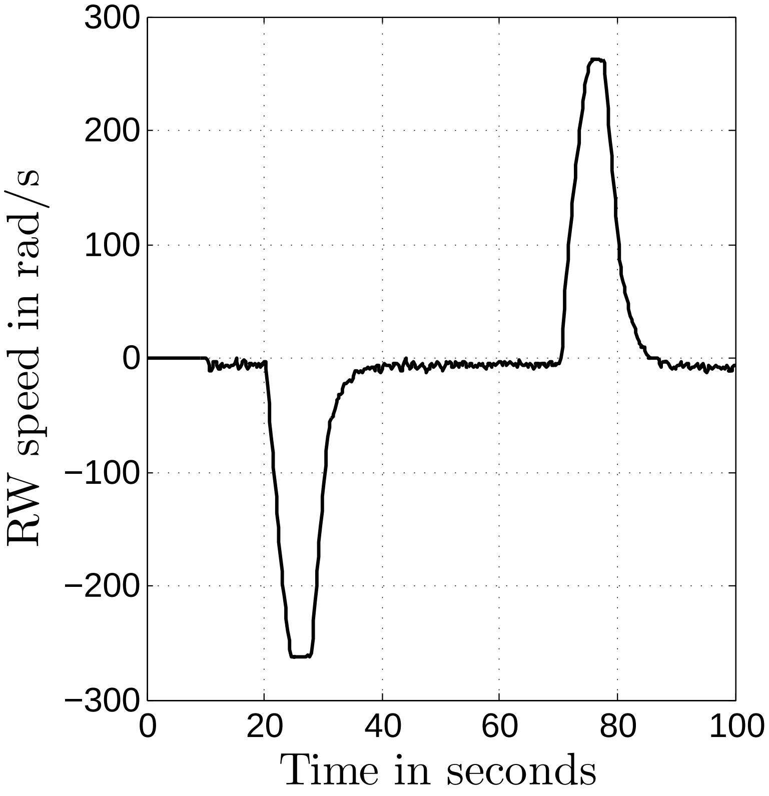


Figure 10.19: *Z_b-Axis RW Speed*

Figures 10.20 to 10.22 show the body rate of EyasSAT through the 90 degree yaw steps. The roll and pitch body rates are damped to maintain a stable roll and pitch angle. The roll rate was damped below 6 mrad/s and the pitch rate below 5 mrad/s in steady state. The yaw rate indicates that a maximum angular rate of about 0.2 rad/s is achieved during the yaw step and at steady state is kept below 8 mrad/s.

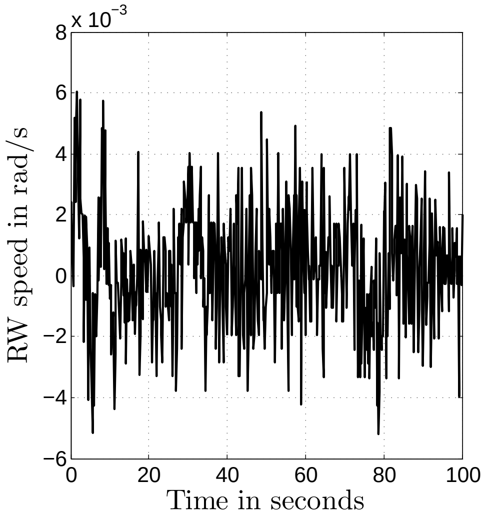


Figure 10.20: *Roll Rate*

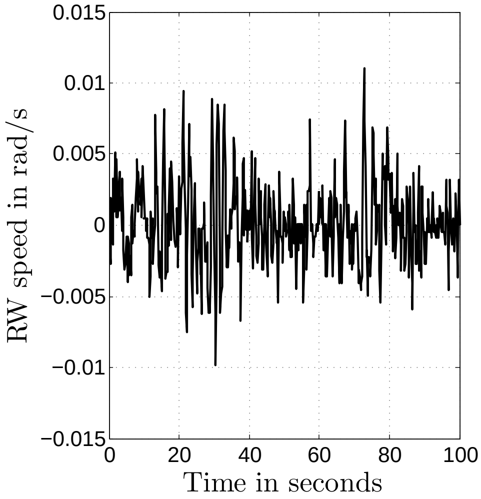


Figure 10.21: *Pitch Rate*

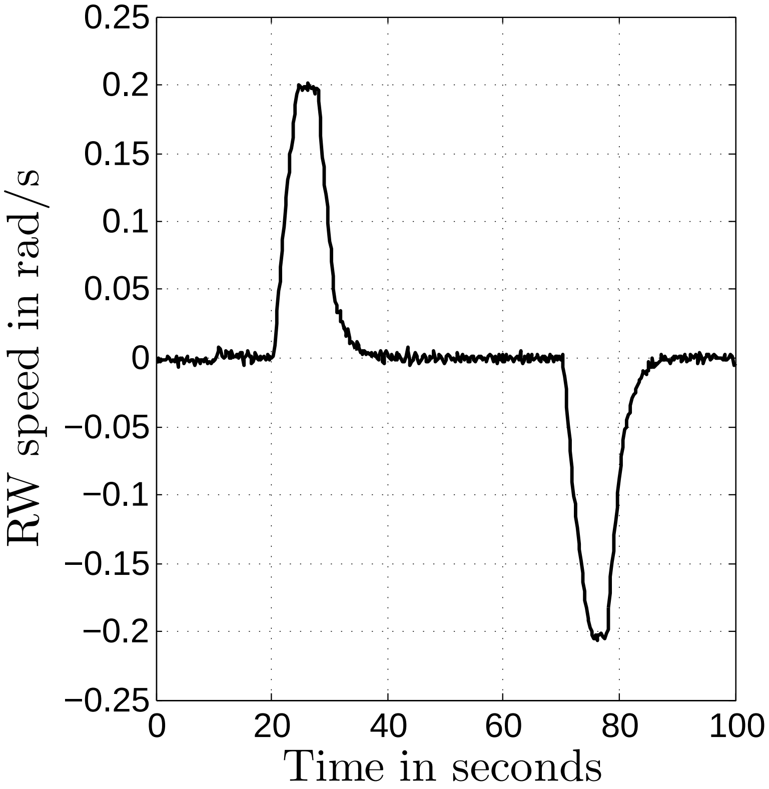


Figure 10.22: *Yaw Rate*

10.4 HIL Test and Results Summary

The air bearing test performed as expected. The simulations provided a good approximation of the real system. The ADCS system implemented was able to perform yaw steps on the air bearing while the roll and pitch rates were damped.

Chapter 11

Conclusion and Recommendations

11.1 Conclusion

Nano-satellites especially CubeSats have become very popular with universities for research purposes. The affordability of these small satellites allow engineering students to obtain hands on experience with the design, build and validation of satellite subsystems.

The goal of this project was to create an ADCS demonstrator and test platform as described in the introduction. Research was executed on ADCS and existing low cost ADCS testing and validation facilities. These facilities were found to use air bearing platforms to validate their ADCS at low cost. An air bearing platform implements near frictionless three DoF rotations.

The convenient small size of nano-satellites allows the entire satellite to be mounted on an air bearing, allowing ADCS validation and testing with the complete satellite. In addition this forms an excellent demonstration platform for illustrating the capabilities and functionality of an ADCS to engineering students.

EyasSAT was used as a design base for a new ADCS module. EyasSAT was developed by the USAFA for the purpose of teaching engineering students the different subsystems found on satellites. An ADCS module is one of the subsystems in EyasSAT. However, the ADCS module is limited, only allowing the demonstration of sensors and actuators. The original ADCS module was replaced by the new ADCS_V2 module designed during this thesis.

The thesis begins by establishing the theoretical background required for the ADCS. The reference frames used are defined along with the methods to describe the satellite's attitude. The dynamic equations of attitude motion was also discussed establishing the dynamic and kinematic equations. Furthermore the disturbance torques experienced with the air bearing setup were described. The theoretical background was first established as the basis for definitions used in the following chapters.

The sensor design is presented first in the hardware section of the thesis. It was found that the *Yaw Attitude Sensor*, later referred to as the CSS, of EyasSAT required an accuracy improvement. This was achieved by the installation of new photo-diodes. The CSS accuracy improved to maximum errors of 8.6 degrees in the yaw axis. A camera sensor was added to allow for increased accuracy during attitude determination when tracking LED light sources. Attitude information with a minimum accuracy of 0.54 degrees in the roll and yaw axis could be supplied by the camera. A new IMU was also used to supply more accurate gyroscope and accelerometer measurements. The accelerometer could supply attitude information in the roll and pitch axis. The accelerometer has an accuracy of 0.046 degrees, given the 16-bit resolution with 0.8192 mg/LSB, and contained measurement noise with an amplitude of 0.046 degrees. The gyroscope has a resolution of 0.01311 deg/s per LSB and measurement noise with an peak amplitude of up to 0.078 deg/s.

The sensor design was followed by a description of the ADCS actuators. EyasSAT originally contained a RW in the Z_b-axis along with magnetorquers in the X_b- and Y_b-axis. To supply EyasSAT with three-axis control capabilities the Z_b-axis RW was removed and replaced with three new RWs, one in each body axis. The speed control design implemented on the RWs is presented. The speed of the RWs could be controlled with an error less than 2.4 rpm. A third magnetorquer was installed in the Z_b axis to allow for three-axis magnetic control.

After the completion of the sensor and actuator design and manufacturing, the hardware section of the project was concluded with the design of a new PCB. The PCB along with the sensors and actuators formed the new ADCS_V2 module. The new module was build to be integrated into EyasSAT with the aid of accurate CAD models. After some adaptations to EyasSAT the ADCS_V2 module was installed. With the integration finalized the hardware section of the project was completed and the software design could commence.

The attitude determination algorithms made use of the sensors previously designed to estimate the attitude of EyasSAT. The algorithms discussed were the Triad, Quest and the EKF algorithms

The attitude control algorithms made use of the RWs. The attitude control algorithm implemented uses a quaternion feedback law to generate torque commands for the RWs. The attitude control was designed so that the yaw angle of EyasSAT, on the air bearing, could be controlled while the roll and pitch angles were only rate controlled to stabilize the air bearing platform. A second control mode was also implemented to allow EyasSAT to align the boresight of the camera with an LED light source pointed by a user. This control mode allows for the demonstration and testing of the camera and the RWs.

The ADCS algorithms were implemented on the MCU. The program flow used to read the sensors, execute attitude determination and control algorithms and output commands to the actuators was discussed in the software development section.

The ADCS was simulated using MATLAB Simulink, which validated the algorithms. The simulation also supplied information about the accuracy and performance capabilities of the ADCS.

The ADCS_V2 module was further tested using HIL techniques to ensure that the algorithms implemented on the module performed as expected. The ADCS_V2 module along with EyasSAT was tested on the air bearing. The ADCS performed as expected given the simulation results. The attitude information was transmitted via a wireless link in real-time to a PC during air bearing tests and demonstrations. Yaw attitude commands could also be sent to EyasSAT wirelessly from the PC.

The ADCS_V2 with EyasSAT forms an excellent ADCS demonstrator for satellite systems engineering students. By completing the ADCS_V2 module and demonstrating the ADCS capabilities the goals of the project was met. However, there are some improvements that can be made and recommendations for future work, which is discussed in the following section.

11.2 Recommendations

11.2.1 Sensors

To increase the accuracy of the photo-diodes the solution for increasing the FoV of the yaw sensors, as discussed in Appendix B.1, can be implemented. Photo-diodes can also be placed on the top and bottom of EyasSAT to supply pitch and roll angle information and not just yaw angle information.

The camera capabilities can be improved by using a wider FoV lens to increase the measurement range. The programming on the FPGA can be made more efficient to allow for the centroid information of both LED light sources on the image sensor to be transmitted to the MCU for centroid calculation of both light points. This will allow sub-pixel accuracy and make the camera sensor more accurate.

A magnetometer can be added to EyasSAT to be used as an additional attitude sensor and to accurately determine the magnetorquer vector.

11.2.2 Actuators

The RWs used brushed DC motors that have more friction than brushless DC motors and as a result can not be speed controlled as accurately. By exchanging the motors with brushless motors, the attitude control capabilities will improve.

By upgrading EyasSAT's batteries the power delivered to the motors can be increased allowing the momentum exchange capability of the RWs to be increased. The RWs can also be run from separate batteries if the batteries in EyasSAT cannot be upgraded.

The magnetic encoder can be upgraded to a newer version in order to encode the motor position in a 14-bit format, allowing the RW speed to be calculated more accurately. The speed measurement of the RWs can also be improved by lowering the angular position sampling frequency. Figure 4.13 showed that three or more samples are taken in one revolution with a voltage input of 10 V to the motors. It was found that the voltage supplied to the motors is closer to 9 V. This decreases the motor speed and ensures that there will always by four or more samples made in one revolution. The sampling period can therefore be doubled as only two samples per revolution is required. Therefore a sampling period of 10 ms can be used. This will allow the speed signal to be filtered more effectively and the noise on the signal will decrease.

11.2.3 PCB

Certain changes can be made to improve the ADCS_V2 PCB. To allow SPI communication with the data handling module on EyasSAT, 3.3V to 5V buffers are required. The EyaBus uses 5V logic signals and the MCU on the ADCS_V2 PCB uses 3.3V logic signals. The buffers were not included in the design because of limited space. The power supply circuit is however not entirely needed as the PCB can receive regulated power from the EyaBus. By using regulated power through the EyaBus and removing the power regulators on the ADCS_V2 PCB, will allow the placement of the 3.3V to 5V voltage translating buffers. The programming headers and RW connection headers can also be placed more strategically.

11.2.4 Air Bearing

The air bearing used does not have an auto balancing feature. The balancing of EyasSAT was done by hand using the three counter weights. As a result EyasSAT will always have some pendulum motion body rates and a roll and pitch offset angle. For control stability reasons the CoM is currently placed below the CoP. By using automated balancing procedures the CoM can be moved toward the CoP more accurately.

A Helmholtz cage can also be added to the air bearing setup to simulate an in-orbit geomagnetic field. The generated magnetic field will allow the testing of magnetic control using magnetorquers and a magnetometer.

Bibliography

[1] *EyasSAT Classroom Satellite User's Manual V4.2*, 4th edn, December 2009.

[2] Ritchey, O.N., Barnhart, D.J., Sellers, J.J., White, J.W., White, T.J. and Clark, H.B.: Eyassat: Creating a progressive space workforce- today. In: *2nd Respnsive Space Conference, Los angeles,CA*. April 2004.

[3] *ADIS16375: Low Profile, Low Noise, Six degrees of Freedom Inertial Sensor*. Analog Devices, One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A., c edn, 2012. Available at: http://www.analog.com/en/mems-sensors/mems-inertial-measurement-units/adis16375/products/product.html.

[4] *Maxon A-max 22 Precious Metal Brushes 3.5 Watt motor*. Maxon Motors, May 2012. Available at: `http://www.maxonmotor.com/maxon/view/product/motor/dcmotor/amax/amax22/110138`

[5] Barnhart, D.J., Sellers, J.J., Bishop, C.A., Gossner, J.R., White, J.J. and Clark, J.C.: Eyassat: A revolution in teaching and learning space systems engineering. In: *AIAA Space Systems Eng. Conf.*, GT-SSEC.A.5. November 2005.

[6] Ivanov, D.S., Karpenko, S.O., Ovchinnikov, M.Y., Roldugin, D.S. and Tkachev, S.S.: Testing of attitude control algorithms for microsatellite "chibis - m" at laboratory facility. *Journal of Computer and Systems Sciences International*, vol. 51 No 1, pp. 106–125, 2012.

[7] Prado, J., Bisiacchi, G., Reyes, L., Vicente, E., Contreras, F., Mesinas, M. and Juarez, A.: Three-axis air-bearing based platform for small satellite attitude determination and control sumulation. *Journal of Applied Research and Technology*, vol. 3, no. 3, pp. 222–237, December 2005.

[8] Kim, B., Velenis, E., Kriengsiri, P. and Tsiotras, P.: A spacecraft simulator for research and education (aas 01-367). In: *ADVANCES IN THE ASTRONAUTICAL SCIENCES*, pp. 897–914. American Astronautical Society, 2001.

[9] Li, J., Post, M., Wright, T. and Lee, R.: Design of attitude control systems for cubesat-class nanosatellite. *Journal of Control Scinece and Engineering*, vol. 2013, April 2013.

[10] Al-Majed, M.I. and Alsuwaidan, B.N.: A new testing platform for attitude determination and control subsystems: Design and applications. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. 2009.

[11] Steyn, W.H. and Al-Dakkan, D.K.: KACST Air Bearing Commissioning results, 2006.

[12] Schwartz, J.L., Peck, M.A. and Hall, C.D.: Historical review of spacecraft simulators. *Journal of Guidance, Control, and Dynamics*, vol. 26, No 4, pp. 513–522, 2003.

[13] Auret, J.: *Design of an Aerodynamic Attiude Control System for a Cubesat.* Master's thesis, The University of Stellenbosch, March 2012.

[14] Wertz, J.R. (ed.): *Spacecraft Attitude Determination and Control.* Kluwer Academic Publishers, 1991.

[15] Steyn, W.H.: *A Multi-mode Attitude Determination and Control System for Small Satellites.* Ph.D. thesis, The Univeristy of Stellenbosch, December 1995.

[16] Liu, Y., Zhou, J., Chen, H. and Mu, X.: Experimental research for flexible satellite dynamic simulation on three-axis air-bearing table. *Journal of Aerospace Engineering*, p. 227:369, January 2013.
Available at: `http://pig.sagepub.com/content/227/2/369`

[17] Everett, J.R.W.R.W. and Puschell, J.J. (eds.): *Space Mission Engineering: The New SMAD.* Microcosm Pess, 2011.

[18] *Solderable Planar Photodiode.* Silonex Inc., 2012.
Available at: `http://www.farnell.com/datasheets/125186.pdf`

[19] Loubser, H.E.: *The development of Sun and Nadir sensor for a solar sail CubeSat.* Master's thesis, The University of Stellenbosch, March 2011.

[20] Kearney, M.-A.: *An Attitude Control System for the Deployment and Stabilisation of a Tethered Dual CubeSat Mission.* Master's thesis, The University of Stellenbosch, March 2013.

[21] Bouguet, J.-Y.: Camera calibration toolbox for matlab. July 2010. Computer Vision Research Group, Dept. of Electrical Engineering, California Institute of Technology.
Available at: `http://www.vision.caltech.edu/bouguetj/calib_doc/`

[22] Open CV camera model. 2013.
Available at: `http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html`

[23] Motor driver drv8800. November 2013.
Available at: `http://www.ti.com/product/drv8800`

[24] *AS5046: Programmable 12-bit 360 dgree magnetic angle encoder with absolute 2-wire serial and analog interfaces.* Austria Micro Systems, 1st edn, 2007. Available at: http://www.ams.com/eng/Products/Position-Sensors/Rotary-Magnetic-Position-Sensor. NOTE: This specific model is no longer available, however new improved versions are available and can be viewed by folowing the provided link.

[25] Gopal, M.: *Digital Control and State Variable Methods, Conventional and Neuro-Fuzzy Control Systems.* 2nd edn. McGraw Hill, 2004.

[26] Barnhart, D.J.: *EyasSAT Classroom Satellite Owner's Manual.* U.S. Air Force Academy and Colorado Satellite Services, 4th edn, 12 December 2009.

[27] *Giant Gecko 280: EFM32GG280 F1024.* Energy Micro, 2012. Available at: http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32GG280.pdf.

[28] Botma, P.J.: *The design and development of an ADCS OBC for a CubeSat.* Master's thesis, The University of Stellenbosch, December 2001.

[29] Hall, C.: Spacecraft attitude dynamics and control, 2012. Class notes.
Available at: `http://www.dept.aoe.vt.edu/~cdhall/courses/aoe4140/`

[30] Steyn, W.H.: A view finder control system for an earth observation satellite. *Aerospace Science and Technology*, vol. 10, pp. 248–255, January 2006.

[31] Wie, B., Weiss, H. and Arapostathis, A.: Quaternion feedback reqgulator for spacecraft eigenaxis rotations. *AIAA Journal of Guidance, Control and Dynamics*, vol. 12, no. 3, pp. 375–380, May-June 1989.

[32] Steyn, W.H.: An attitude control system for sumbandilasat an earth observation satellite. In: *The 4S (Small Satellite Systems and Services) Symposium, ESA Proceedings SP-600.* Rhodes Greece, 26-30 May 2008.

[33] Jordaan, H.W.: *Solar sail CubeSat: The Deployment and Control of a Spinning Solar Sail CubeSat.* Master's thesis, Stellenbosch University, March 2013.

[34] Franklin, G.F., Powell, J.D. and Emami-Naeini, A.: *Feedback Control of Dynamic Systems.* Pearson Education, 2009.

Appendices

Appendix A

Centre of Mass and Inertia Calculations

A.1 Pendulum Motion

The natural frequency of a compound pendulum is given in the equation (A.1.1) [34]

$$\omega_n = \sqrt{\frac{mgr}{I_p}} \tag{A.1.1}$$

where m is the mass of the object, g is the acceleration of gravity, and r is the distance from the pivot point to the CoM of the object.

The inertia about the pivot point can be determined using equation (A.1.2) by measuring the natural frequency of oscillation.

$$I_p = \frac{mgr}{\omega_n^2} \tag{A.1.2}$$

EyasSAT is mounted onto the air bearing surface with counter weights positioned so that EyasSAT balances on the air bearing. In this state the CoM is below the air bearing pivot point so that the system will have a stable pendulum motion. This causes the CoM to always return to its steady state position directly below the air bearing pivot point because of the gravitational force on the CoM causing a restoring gravity torque. The CoM can be seen as the tethered weight to the pivot as with a normal pendulum. By disturbing the balanced state the system will oscillation and by measuring this oscillation frequency will allow for the determination of the moment of inertia of the system.

A.2 Inertia Measurement Strategy

A.3 Mass Calculations

The two unknowns in equation (A.1.2) are the distance of the CoM from the CoP (r) and the inertia (I_p) of the system. Two measurements are required in order to calculate the two unknowns, with one of the unknowns remaining constant.

EyasSAT and the mounting surface was balanced with the counter weights adjusted to the highest possible points while still maintaining balance, to obtain the required measurements. This is the point where the system will have a minimum moment of inertia as the weights will be closest to the CoM. A measurement of the oscillation frequency is taken in this state.

Thereafter, each of the weights are adjusted 5 mm down the rods. This shifts the CoM down a predictable amount and forms the work point where the systems inertia is calculated. A second measurement is taken in this state. Due to the change in position of weights, the inertia of the system also changed. The assumption is made that the inertia stays constant due to the small change in position of the weights. This assumption allows for the calculation of the distance of the CoM from the CoP.

The weights are then adjusted downward another 5 mm and another measurement is taken. The distance of the CoM from the CoP can be calculated again between the work-point and this position. The average of the two calculated distances around the work-point will give an accurate estimation of the distance at the work point.

Another assumption made is that the CoM is directly below the CoP because the system is balanced. The distance vector only has a component in the Z_b-axis.

The amount the CoM position changes with the change in the position of the weights needs to be calculated. Therefore the amount the weights contribute to the CoM needs to be calculated. The position of the CoM can be calculated with equation (A.3.1).

$$COM_z = \frac{1}{m_T}\int r\ dm \tag{A.3.1}$$

In order to simplify the calculations needed, the CoM of each significant unit, or part, in the system will be estimated. These estimations will be added together using equation (A.3.2) to calculate the CoM

$$COM_z = \frac{1}{m_T}\sum r_i m_i \tag{A.3.2}$$

where m_T is the total mass of the system and r_i is the Z_b-axis position of the mass element m_i from the CoP.

Thus the different components were weighed, their distances from the CoP calculated and the results are shown in table A.1.

Description	**Mass**	**position**
Mounting Surface	747 g(M_S)	10 mm(r_S)
EyasSAT	4000 g(M_E)	130 mm(r_E)
Rods	440.88 g(M_R)	-129.904 mm(r_R)
IMU	30 g(M_I)	18.65 mm(r_I)
Counter weights	2990.27 g(M_W)	r_w
Total:	8208.15 g(M_T)	

Table A.1: *Mass of Components*

The CoM is now calculated using equation (A.3.2):

$$COM_z = \frac{1}{M_T}(M_S r_S + M_E r_E + M_R r_R + M_I r_I + M_W r_w) \tag{A.3.3}$$

$$= 0.091 r_S + 0.487 r_E + 0.0537 r_R + 0.00366 r_I + 0.36431 r_w \tag{A.3.4}$$

$$= Constant + 0.36431 r_w \tag{A.3.5}$$

Equation (A.3.5) states that the CoM changes with 36.43% of the change in position of the balancing weights in the Z_b-axis. This means that if the counter weights moves by 10mm, vertically, then the CoM moves by 3.643 mm in the same direction. The rods are mounted at a 30 degree angle and if the weights are moved a set distance d, the distance of the CoM will move:

$$\Delta r = d\cos(30)(0.36431) \tag{A.3.6}$$

A.4 Frequency Measurements and CoM Position Estimation

The frequency measurements were made with weights at the three different positions, which were described in Section A.2. The results are shown in the Table A.2. The change in distance of the CoM (Δr) is given as:

$$\Delta r = 5\cos(30)(0.36431) = 1.57751 \text{ mm} \tag{A.4.1}$$

In the table the position r represents the unknown distance of the CoM below the pivot point or CoP.

Position	**Oscillation period** [s]	**Oscillation Frequency** [rad/s]
X-Axis:		
Top ($r = r + \Delta r_t$)	12.325	0.5098 ($w_{x\ top}$)
(Work point) Mid ($r = r$)	7.4667	0.8415 ($w_{x\ mid}$)
Bottom ($r = r + \Delta r_b$)	5.55	1.13211 ($w_{x\ bottom}$)
Y-Axis:		
Top ($r = r + \Delta r_t$)	11.63	0.54 ($w_{y\ top}$)
(Work point) Mid ($r = r$)	7	0.8976 ($w_{y\ mid}$)
Bottom ($r = r + \Delta r_b$)	5.267	1.193 ($w_{y\ bottom}$)

Table A.2: *Frequency Measurements*

The weights at the top most positions equation (A.1.2) is given as:

$$I = \frac{mg(r + \Delta r_t)}{\omega_{top}} \tag{A.4.2}$$

Furthermore, the weights at the work point or middle positions equation (A.1.2) is given as:

$$I = \frac{mgr}{\omega_{mid}} \tag{A.4.3}$$

In addition, the weights at the bottom position equation (A.1.2) is given as:

$$I = \frac{mg(r + \Delta r_b)}{\omega_{bottom}} \tag{A.4.4}$$

If the assumption that inertia remains the same for the small changes in the weights' positions equation (A.4.2) to (A.4.4) can be used to solve r.

A.4.1 X_b-axis calculations

Firstly the middle position and top position are set equal to each other and the distance r is estimated.

$$\frac{mgr}{\omega_{x\ mid}^2} = \frac{mg(r+\Delta r_t)}{\omega_{x\ top}^2} \tag{A.4.5}$$

$$\frac{r}{\omega_{x\ mid}^2} = \frac{(r+\Delta r_t)}{\omega_{x\ top}^2} \tag{A.4.6}$$

$$r\omega_{x\ top}^2 = (r+\Delta r_t)\omega_{x\ mid}^2 \tag{A.4.7}$$

$$r(\omega_{x\ top}^2 - \omega_{x\ mid}^2) = \Delta r_t \omega_{x\ mid}^2 \tag{A.4.8}$$

$$r = \frac{\Delta r_t \omega_{x\ mid}^2}{(\omega_{x\ top}^2 - \omega_{x\ mid}^2)} \tag{A.4.9}$$

$$r = \frac{(-1.57751)(0.8415)^2}{(0.5098^2 - 0.8415^2)} \tag{A.4.10}$$

$$r = 2.4922 \text{ mm} \tag{A.4.11}$$

Secondly, the middle position and bottom position are set equal to each other and the distance r is estimated.

$$\frac{mgr}{\omega_{x\ mid}^2} = \frac{mg(r+\Delta r_b)}{\omega_{x\ bottom}^2} \tag{A.4.12}$$

$$\frac{r}{\omega_{x\ mid}^2} = \frac{(r+\Delta r_b)}{\omega_{x\ bottom}^2} \tag{A.4.13}$$

$$r\omega_{x\ bottom}^2 = (r+\Delta r_b)\omega_{x\ mid}^2 \tag{A.4.14}$$

$$r(\omega_{x\ bottom}^2 - \omega_{x\ mid}^2) = \Delta r_b \omega_{x\ mid}^2 \tag{A.4.15}$$

$$r = \frac{\Delta r_b \omega_{x\ mid}^2}{(\omega_{x\ bottom}^2 - \omega_{x\ mid}^2)} \tag{A.4.16}$$

$$r = \frac{(1.57751)(0.8415)^2}{(1.13211^2 - 0.8415^2)} \tag{A.4.17}$$

$$r = 1.94764 \text{ mm} \tag{A.4.18}$$

The average of the two estimations is taken to give a good approximation of the distance r at the midpoint or work point.

$$r = \frac{(2.4922) + (1.94764)}{2} = 2.21922 \text{ mm} \tag{A.4.19}$$

The inertia can now be calculated with (A.1.2) using the above calculated CoM position r.

$$I_{xx} = \frac{mgr}{\omega_{x\ mid}^2} \tag{A.4.20}$$

$$= \frac{(8.20815)(9.81)(2.21922 \times 10^{-3})}{0.8415^2} \tag{A.4.21}$$

$$= 0.2523518 \text{ kg.m}^2 \tag{A.4.22}$$

A.4.2 Y_b-axis calculations

Firstly the middle position and top position are set equal to each other and then the distance r is estimated.

$$\frac{mgr}{\omega^2_{y\ mid}} = \frac{mg(r+\Delta r_t)}{\omega^2_{y\ top}} \tag{A.4.23}$$

$$\frac{r}{\omega^2_{y\ mid}} = \frac{(r+\Delta r_t)}{\omega^2_{y\ top}} \tag{A.4.24}$$

$$r\omega^2_{y\ top} = (r+\Delta r_t)\omega^2_{y\ mid} \tag{A.4.25}$$

$$r(\omega^2_{y\ top} - \omega^2_{y\ mid}) = \Delta r_t \omega^2_{y\ mid} \tag{A.4.26}$$

$$r = \frac{\Delta r_t \omega^2_{y\ mid}}{(\omega^2_{y\ top} - \omega^2_{y\ mid})} \tag{A.4.27}$$

$$r = \frac{(-1.57751)(0.8976)^2}{(0.54^2 - 0.8976^2)} \tag{A.4.28}$$

$$r = 2.47231 \text{ mm} \tag{A.4.29}$$

Secondly, the middle position and bottom position are set equal to each other and then the distance r is estimated.

$$\frac{mgr}{\omega^2_{y\ mid}} = \frac{mg(r+\Delta r_b)}{\omega^2_{y\ bottom}} \tag{A.4.30}$$

$$\frac{r}{\omega^2_{y\ mid}} = \frac{(r+\Delta r_b)}{\omega^2_{y\ bottom}} \tag{A.4.31}$$

$$r\omega^2_{y\ bottom} = (r+\Delta r_b)\omega^2_{y\ mid} \tag{A.4.32}$$

$$r(\omega^2_{y\ bottom} - \omega^2_{y\ mid}) = \Delta r_b \omega^2_{y\ mid} \tag{A.4.33}$$

$$r = \frac{\Delta r_b \omega^2_{y\ mid}}{(\omega^2_{y\ bottom} - \omega^2_{y\ mid})} \tag{A.4.34}$$

$$r = \frac{(1.57751)(0.8976)^2}{(1.193^2 - 0.8976^2)} \tag{A.4.35}$$

$$r = 2.05805 \text{ mm} \tag{A.4.36}$$

The average of the two estimations is calculated to give a good approximation of the distance r at the midpoint or work point.

$$r = \frac{(2.47231) + (2.05805)}{2} = 2.26518 \text{ mm} \tag{A.4.37}$$

The inertia can now be calculated with (A.1.2) using the above calculated CoM position r.This result can be compared to the result obtained in (A.4.19), and the results are seen to be very similar. This is expected because the position is the same for both, however, differences could be due to slightly inaccurate readings and difficulties in perfectly balancing the system during tests.

$$I_{yy} = \frac{mgr}{\omega_{y\ mid}^2} \tag{A.4.38}$$

$$= \frac{(8.20815)(9.81)(2.26518 \times 10^{-3})}{0.8976^2} \tag{A.4.39}$$

$$= 0.22638692 \text{ kg.m}^2 \tag{A.4.40}$$

A.4.3 Z_b-axis

The inertia of the Z_b-axis is determined using the principal of momentum conservation. The Z_b-axis reaction wheel was spun up to a known speed and the angular rate of the system measured with the MEMS gyroscope. The equation describing the momentum conservation is shown in equation A.4.41 and A.4.42. The result in equation A.4.43 shows that when the RW speed changes, the system angular speed changes and given the known RW inertia, then the inertia of the system can be calculated [11].

$$M_{RW} = M_{system} \tag{A.4.41}$$

$$I_{RW}\Delta\omega_{RW} = I_{zz}\Delta\omega_z \tag{A.4.42}$$

$$Izz = \frac{I_{RW}\Delta\omega_{RW}}{\Delta\omega_z} \tag{A.4.43}$$

Figure A.1 shows the response of the system angular rate to a RW speed change of 1000 rpm.

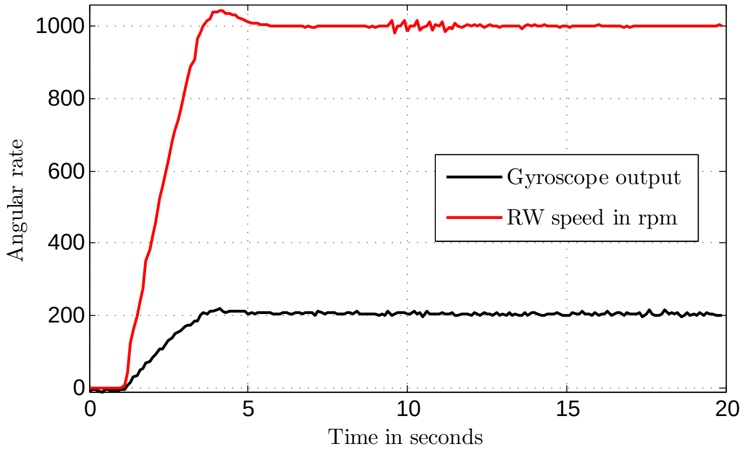


Figure A.1: *Reaction Wheel Speed and Angular Rate of EyasSAT*

The inertia can now be calculated as:

$$I_{zz} = \frac{(7.613 \times 10^{-5})(1000 \times \frac{360}{60})}{((206+6) \times 0.013108)} \tag{A.4.44}$$

$$= 164.37463 \times 10^{-3} kg.m^2 \tag{A.4.45}$$

$$= 0.16437463 \text{ kg.m}^2 \tag{A.4.46}$$

Table A.3 shows the results of a number of different tests for calculating the Z_b-axis inertia.

$\Delta\omega_{RW}$	$\Delta\omega_z$	**Inertia** kg.m^2
104.6902 rad/s	0.0488 rad/s	163.321×10^{-3}
104.6897 rad/s	0.0493 rad/s	161.66383×10^{-3}
-198.371 rad/s	-0.0934 rad/s	161.65759×10^{-3}

Table A.3: *Test Results*

The average of the previous results is a good estimate for the Z_b-axis inertia, which is set at 162.7543×10^{-3} kg.m^2.

A.5 Summary

The inertia values that will be used are shown in Table A.4.

I_{xx}	252.3518×10^{-3} kg.m^2
I_{yy}	226.3869×10^{-3} kg.m^2
I_{zz}	162.7543×10^{-3} kg.m^2

Table A.4: *System Inertia Values*

Appendix B

EyasSAT Yaw Sensors

B.1 Increasing FoV Solution

Two photocells in one axis can be used as a method for increasing the FoV of the photocells and aid light detection at large angles of α [14]. This concept is illustrated in Figure B.1. The photocell's FoV may be physically reduced with apertures that will not detect light at large angles of α. However the combination of reduced FoV of both photocell's can increase the total FoV.

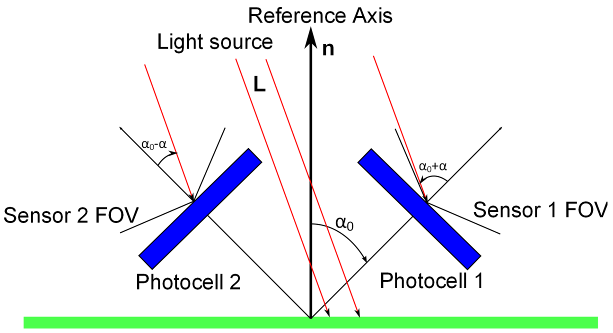


Figure B.1: *Two Detectors for a Larger FoV*

In this configuration the output on each detector is given as

$$I_1 = I_0 cos(\alpha_0 - \alpha) \tag{B.1.1}$$
$$I_2 = I_0 cos(\alpha_0 + \alpha) \tag{B.1.2}$$

and taking the difference between each detector results in equation (B.1.5).

$$\begin{aligned} \Delta I &= I_1 - I_2 && \text{(B.1.3)} \\ &= 2I_0 \sin(\alpha_0) \sin(\alpha) && \text{(B.1.4)} \\ &= Constant \sin(\alpha) && \text{(B.1.5)} \end{aligned}$$

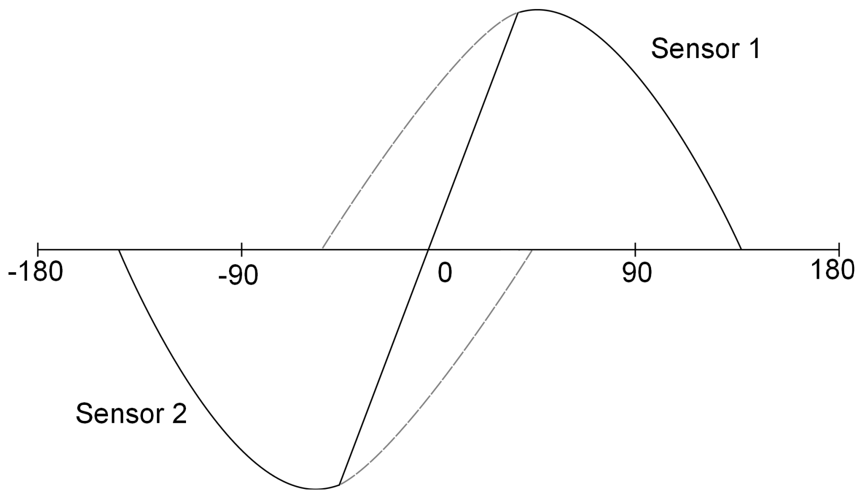


Figure B.2: *Summed Output of the Two Sensors*

The output of the two detectors, using equation B.1.3, is shown in Figure B.2 where the dashed line is the output from each sensor and the solid line is the combined output.

However, this solution was not implemented on EyasSAT due to time constraints and the limited number of cells. The following section discusses the performance of the solar cells on EyasSAT.

B.2 Yaw Attitude Sensor Evaluation

The sensor test is done by balancing EyasSAT on the air bearing. A light source is placed at a 90 degree yaw angle in relation to the BRF. Then EyasSAT is rotated through 360 degrees so that the solar cells' minimum and maximum values can be stored for calibration. The Z_b-axis reaction wheel is given a speed step command resulting in EyasSAT spinning about the Z_b-axis at a fixed speed. The solar cells outputs are monitored via the wireless data link and a MATLAB Simulink interface is used to capture all the data.

The solar cells' voltage outputs are measured over resistors. The solar cells outputs are sampled using an analogue to digital converter, which outputs a 16 bit value every 100 ms. The initial position of the light source is at 90 degrees relative to EyasSAT in the BRF. Figure B.3 shows the output of each cell when EyasSAT does a positive Z_b-axis or yaw rotations:

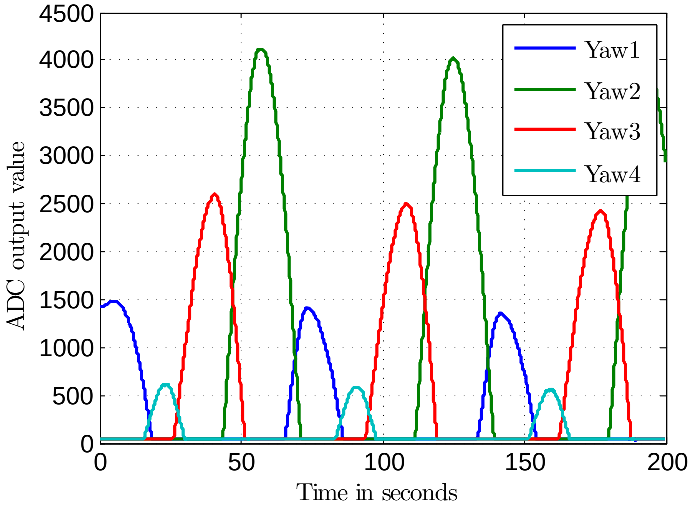


Figure B.3: *Solar Cell Measurements when EyasSAT is Rotating*

Figure B.3 shows that the gain of each sensor is very different. Ideally, when the one cell output decreases from the maximum, the next solar cell will immediately begin to increase. This is due to the light that will begin to fall on that cell because of the 180 degree FoV. However, in Figure B.3 this does not happen. The next cell output only begins to increase when the decreasing output of the solar cell is as low as 50% of its maximum value, due to the reduced FoV. The unit vector will not change unless two cells' outputs change. Therefore, the unit vector will remain constant until the next cell rotates towards the light and begins to react to the light. The effect of this can be seen in Figure B.4 and Figure B.5 as the areas where the unit vector remain constant and the light position remains constant.

The unit vector generated is shown in n Figure B.4. The solar cell outputs in Figure B.3 were used to generate this unit vector figure.

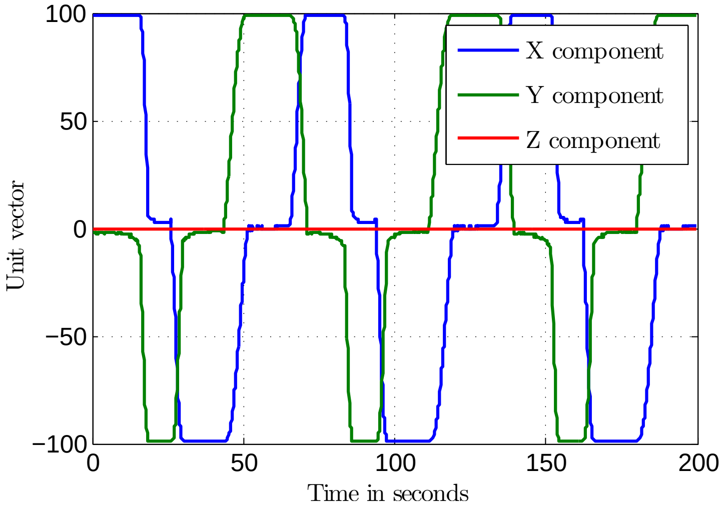


Figure B.4: *Solar Cell Generated Unit Vector Output*

The angle of the light source can be obtained by calculating the arctan of the X_b and Y_b components of the unit vector. The yaw angle of the light source is shown in Figure B.5. The angle of the light source is not very accurate. Although EyasSAT is rotating at a constant angular speed the figure shows areas where the angle remains constant. This is due to the non-ideal behaviour and reduced FoV of the solar cells discussed previously.

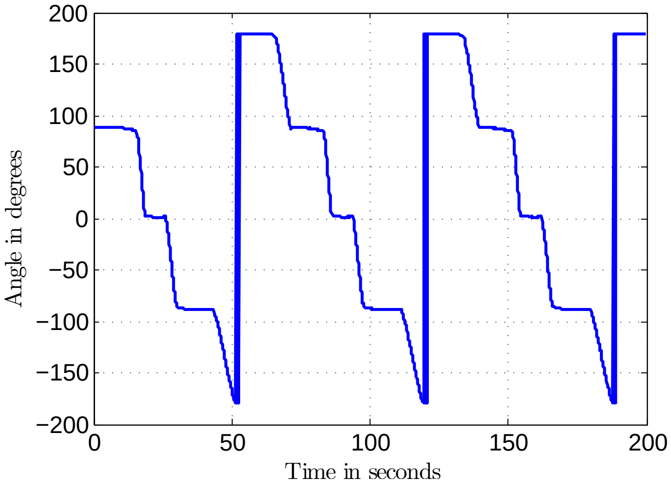


Figure B.5: *Angle Output of Light Source from Yaw Solar Cell Sensors*

Photo-diodes were used To improve the accuracy of the CSS.

B.3 Photo-diode Evaluation

The same test setup and test procedure used with the solar cells was followed. The output of each photo diode is shown in Figure B.6.

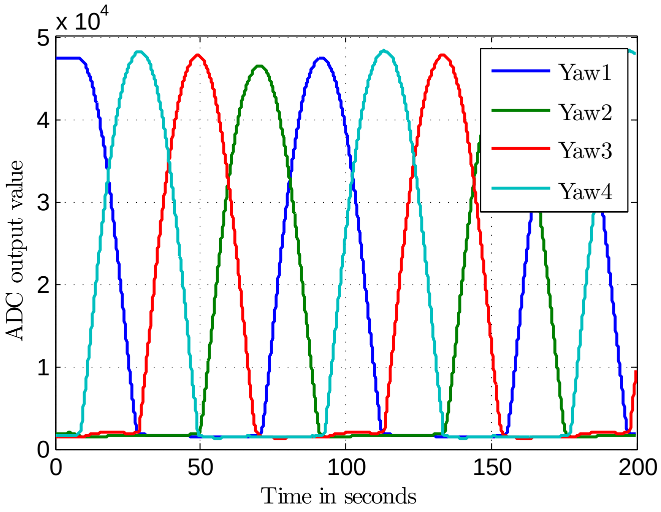


Figure B.6: *Photo-diode Measurements while EyasSAT is Rotating*

The output of photo-diodes detection of light is much more effective and gives greater accuracy than the output of the solar cells. This can be seen by the generated unit vector in Figure B.7.

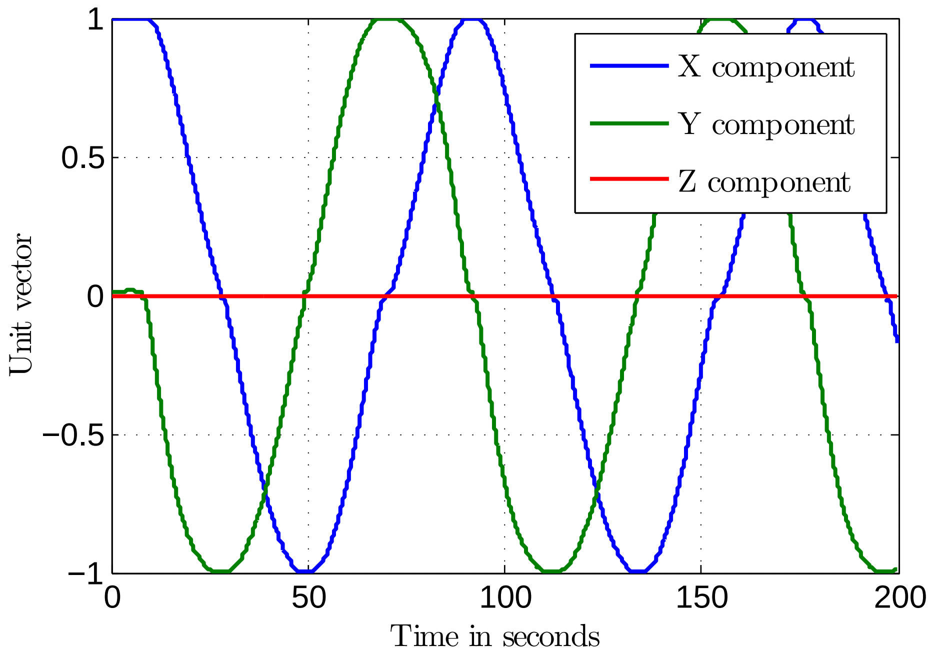


Figure B.7: *Unit Vector Generated Using Photo-diodes*

The yaw angle is also much more accurately represented as shown in figure B.8.

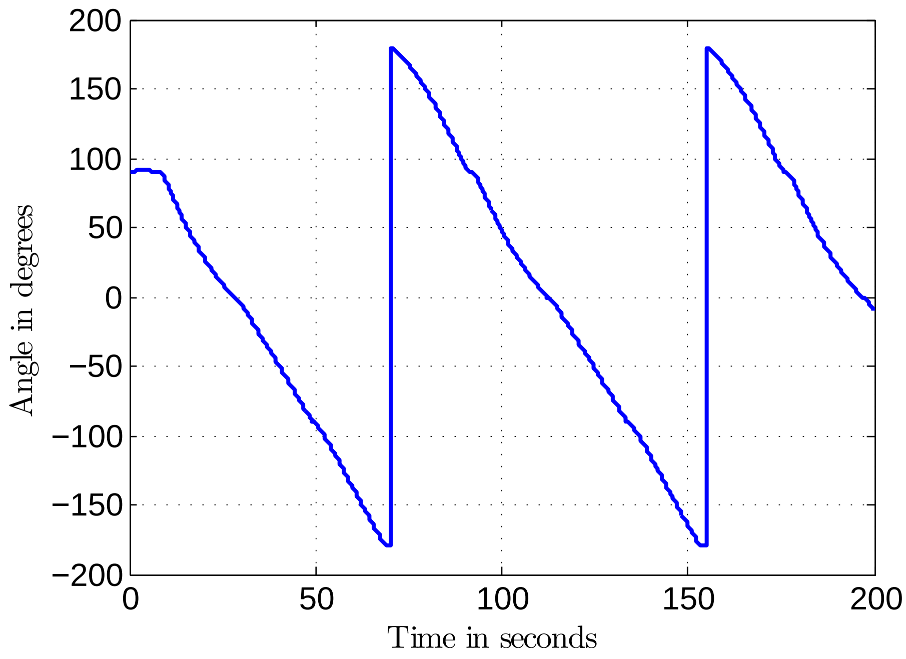


Figure B.8: *Yaw Angle of Light Source*

Each sensor's FoV was tested by rotating the photo-diode through 180 degrees with regards to a fixed light source in order to test the response and accuracy of the photo-diodes. A two-axis rotation platform was used to accurately rotate EyasSAT with regards to the light source. The rotation station is shown in Figure B.9 to B.11.

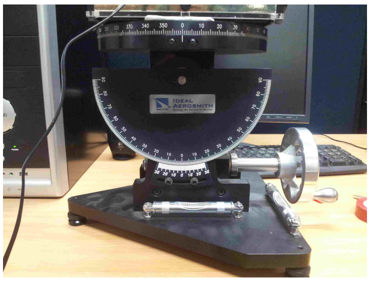


Figure B.9: *Rotation Platform Side View*

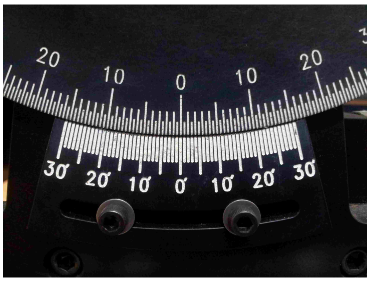


Figure B.10: *X-axis Degree Marker*

Figure B.11: *Z-axis Degree Marker*

The output of each photo-diode is shown below with the error made at each degree step. The blue signals represent the ideal response, the black exemplifies the true response and the red denotes the error in degrees between the two.

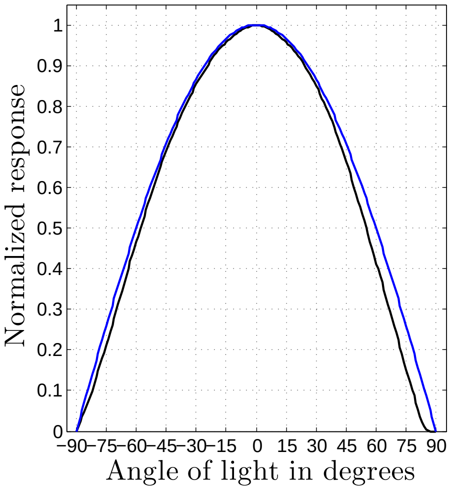


Figure B.12: *Sensor Yaw1 Response*

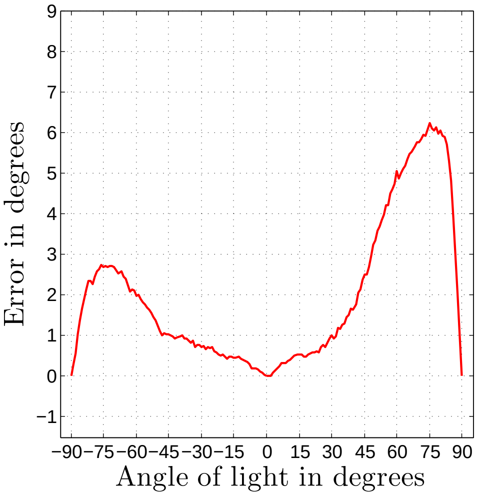


Figure B.13: *Sensor Yaw1 Error*

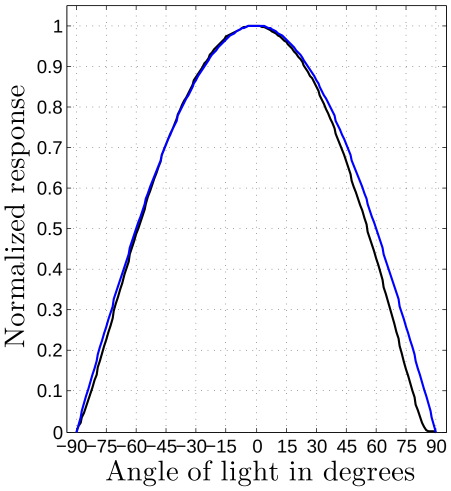


Figure B.14: *Sensor Yaw2 Response*

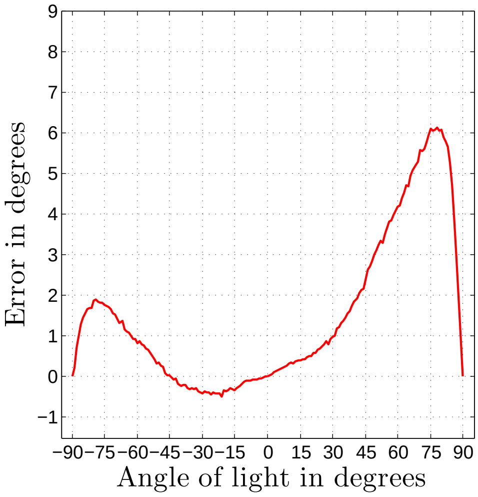


Figure B.15: *Sensor Yaw2 Error*

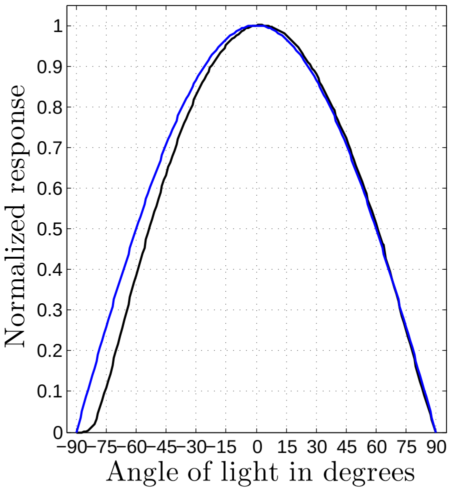


Figure B.16: *Sensor Yaw3 Response*

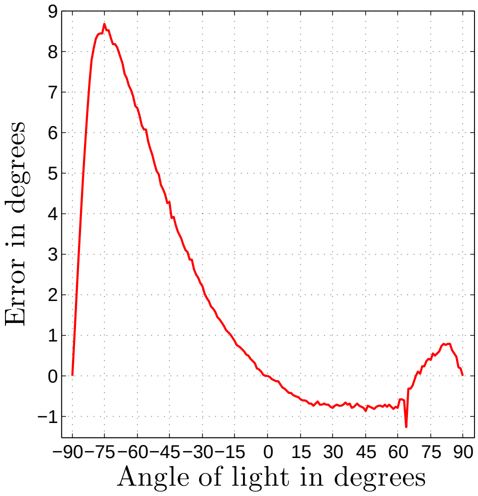


Figure B.17: *Sensor Yaw3 Error*

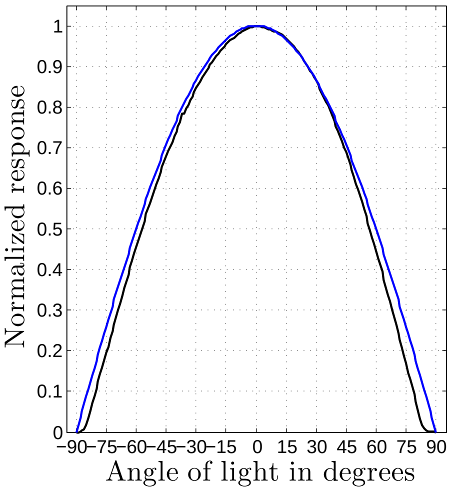


Figure B.18: *Sensor Yaw4 Response*

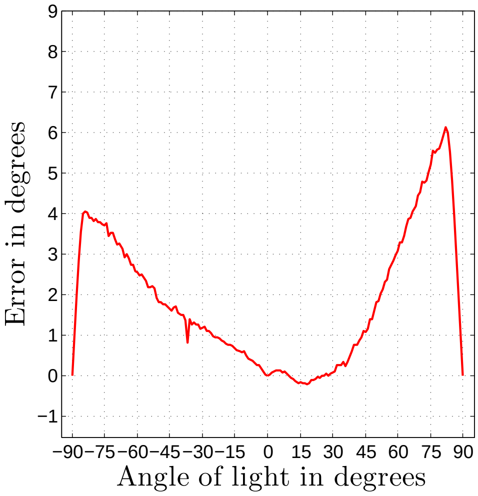


Figure B.19: *Sensor Yaw4 Error*

Appendix C

Reaction Wheel

C.1 Dimensions of each Flywheel

The dimensions of each flywheel vary slightly. Table C.1 shows the dimensions of each flywheel in their respective body axis using the dimension definitions in figure C.1.

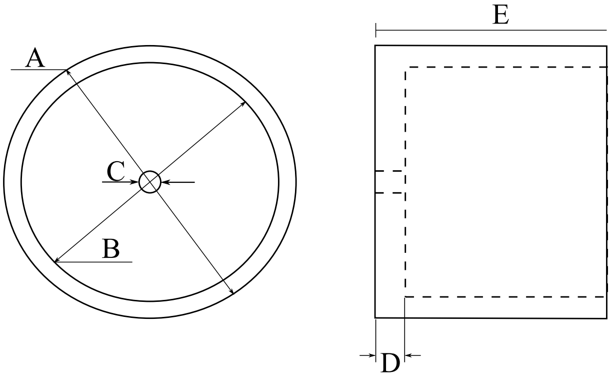


Figure C.1: *Flywheel Dimensions*

Axis	**A**	**B**	**C**	**D**	**E**
Design specs:	45.00mm	32.00mm	2mm	5.00mm	30.00mm
X_b	45.73mm	34.56mm	2mm	5.18mm	30.10mm
Y_b	44.09mm	31.94mm	2mm	4.95mm	29.91mm
Z_b	44.09mm	31.72mm	2mm	4.86mm	30.03mm

Table C.1: *All Flywheel Dimensions*

C.2 RW Inertia Calculations

The total inertia of each RW needs to be calculated. Thus the inertia of the motor, flywheel and the magnet with the magnet holder are summed together to calculate the RW and the motor axis inertia.

C.2.1 Inertia of Flywheel

In order to calculate the inertia of the flywheel, the inertia of three separate parts need to be calculated. These are the disc, the ring and the hole for the axis. The disc is the part of the flywheel that is connected to the axis and forms the inner diameter of the flywheel. The ring is defined as the part that extends from the inner diameter to the outer diameter of the flywheel and extends to the entire height of the flywheel. The total inertia can be calculated by adding the inertia of the disc and ring and subtracting the inertia of the hole. In order to calculate the inertia of each part, the mass has to be known, which in turn means that the density and volume specifications are needed. Mass can be measured and volume calculated, therefore the only unknown is density. The density was calculated using equation (C.2.1) as 8324.1 kg/m^3.

$$Mass = Density \times Volume \tag{C.2.1}$$

C.2.1.1 Inertia of Disc

The volume of a disk can be calculated using (C.2.2)

$$V_{disk} = \pi r^2 h \tag{C.2.2}$$

where r is the radius of the disc and h is the height.

The inertia and mass of the disc can be calculated with equation (C.2.3),

$$I_{disk} = 0.5mr^2 \tag{C.2.3}$$

where m is the mass of the disc.

The inertia of the disc can be calculated by combining the density of the brass and equations (C.2.1) to (C.2.3) together to obtain (C.2.4).

$$I_{disc} = 0.5mr^2 = 0.5(DV) = 0.5(D\pi r^2 h)r^2 = 0.5D\pi hr^4 \tag{C.2.4}$$

The mass of the disc is given as:

$$m_{disk} = D\pi hr^2 \tag{C.2.5}$$

The Table C.2 shows the calculated mass and inertia of each disc in each axis.

Axis	**Mass**	**Inertia** [kg.m^2]
X_b	0.0404 kg	6.0389×10^{-6}
Y_b	0.0330 kg	4.21×10^{-6}
Z_b	0.032 kg	4.0207×10^{-6}

Table C.2: *Inertia and Mass of each Flywheel Disc*

C.2.1.2 Inertia of Ring

The volume of the ring can be calculated as:

$$V_{ring} = V_{disk} - V_{inner-disk} = \pi r^2 h - \pi {r_{inner}}^2 h = \pi h(r^2 - r^2_{inner}) \tag{C.2.6}$$

where r is the radius of the outer radius and r_{inner} is the inner radius.

The inertia of the ring around the axis can be calculated as:

$$I_{ring} = 0.5m(r^2 - r^2_{inner}) \tag{C.2.7}$$

Thus the inertia is calculated using equation (C.2.8).

$$I_{ring} = 0.5m(r^2 - r^2_{inner}) = 0.5[D\pi h_{ring}(r^2 + r^2_{inner})](r^2 - r^2_{inner}) \tag{C.2.8}$$

The mass of the ring is given in equation C.2.9.

$$M_{ring} = D\pi h_{ring}(r^2 + r^2_{inner}) \tag{C.2.9}$$

The inertia and mass of each flywheel was calculated and the results are shown in Table C.3.

Axis	**Mass**	**Inertia** [kg.m^2]
X_b	0.1765 kg	7.2483×10^{-5}
Y_b	0.1841 kg	6.7892×10^{-5}
Z_b	0.1806 kg	6.6927×10^{-5}

Table C.3: *Inertia and Mass of each Flywheel Ring*

C.2.1.3 Inertia of Axis Hole

The inertia of the hole made for the motor axis needs to be calculated and subtracted from the inertia of the disc. The inertia of the hole can be calculated using equation (C.2.4), where the radius used is the radius of the hole (1mm). The mass of the axis hole can be calculated using equation (C.2.5). The inertia and mass of each hole is shown in table C.4.

Axis	**Mass**	**Inertia** [kg.m^2]
X_b	1.3546×10^{-4} kg	6.7731×10^{-11}
Y_b	1.2709×10^{-4} kg	6.3546×10^{-11}
Z_b	1.2945×10^{-4} kg	6.4723×10^{-11}

Table C.4: *Inertia and Mass of each Flywheel Axis Hole*

C.2.1.4 Inertia and Mass of Flywheel

The total inertia and mass of the flywheel can now be calculated using equation C.2.10 for the inertia and equation C.2.11 for mass.

$$I_{fw} = I_{disc} + I_{ring} - I_{hole} \tag{C.2.10}$$

$$M_{fw} = M_{disc} + M_{ring} - M_{hole} \tag{C.2.11}$$

The results are shown in Table C.5.

Axis	**Mass**	**Inertia** (kg.m^2)
X_b	0.2168 kg	7.8522×10^{-5}
Y_b	0.2160 kg	7.1913×10^{-5}
Z_b	0.2135 kg	7.1137×10^{-5}

Table C.5: *Inertia and Mass of each Flywheel*

C.2.2 Inertia of Magnet Holder

The inertia of the magnet holder can be calculated with the same equations used in the previous section. The strategy for calculating the inertia of the magnet holder is shown in equation (C.2.12). The inertia of the holder is calculated by taking the inertia of the spherical body of the holder minus the inertia of the hole for the motor axis and the cavity for the magnet.

$$I_{holder} = I_{body} - I_{axis} - I_{magnet\ hole} \tag{C.2.12}$$

$$I_{body} = 0.5D\pi r_{body}^4 h = 0.5(1474.8)\pi(0.00868/2)^4(0.00596) = 4.8983 \times 10^{-9} \tag{C.2.13}$$

$$I_{axis} = 0.5D\pi r_{axis}^4 h = 0.5(8500)\pi(0.001)^4(0.00249) = 5.7682 \times 10^{-12} \tag{C.2.14}$$

$$I_{magnet\ hole} = 0.5D\pi r^4_{magnet\ hole}h = 0.5(8500)\pi(0.00304)^4(0.00347) = 6.8654 \times 10^{-10} \tag{C.2.15}$$

$$I_{holder} = 4.206 \times 10^{-9}\ kg.m^2 \tag{C.2.16}$$

The inertia of the magnet can also be calculated:

$$I_{magnet} = 0.5Mr^2 = 0.5(0.001)(0.00597/2)^2 = 2.3167 \times 10^{-9} \tag{C.2.17}$$

The total inertia of the holder and magnet:

$$I_{MagnetHolder} = I_{holder} + I_{magnet} = 6.5227 \times 10^{-9}\ kg.m^2 \tag{C.2.18}$$

C.2.3 Inertia of Reaction Wheel

The total inertia of the reaction wheel can now be calculated using the previously calculated values and the inertia of the motor obtained from the datasheet.

$$I_{RW} = I_{FlyWheel} + I_{Motor} + I_{MagnetHolder} \tag{C.2.19}$$

The calculated inertia of each RW is show in table C.6.

Axis	**Inertia** [kg.m^2]
X_b	8.2738×10^{-5}
Y_b	7.5354×10^{-5}
Z_b	7.613×10^{-5}

Table C.6: *Inertia of each RW*

C.3 Speed Estimation

C.3.1 Conversion To RPM

In order to obtain a speed value in revolutions per minute a transformation has to be done. The steps for converting the raw change in angular position over 20 samples to RPM is listed below.

- The average angular position travelled needs to be calculated from the summed value by dividing by the number of samples (which is always 20).

- The encoder values need to be converted to revolutions using the resolution of the encoder which outputs a 10 bit value leading to a resolution of $\frac{1}{1024}$ rev/unit.
- The revolution per sampling period needs to be changed to revolution per second by multiplying by 1000, to go from milliseconds to seconds, and divided by 5 for the 5ms.
- Now the revolutions per second can be converted to RPM by multiplying by 60.

The total conversion is calculated in equation (C.3.1).

$$v_{rpm} = \frac{D_{avg}}{20} \times \frac{1}{1024} \times \frac{1000}{5} \times 60 = D_{avg} \times 0.58594 \tag{C.3.1}$$

C.3.2 Conversion to Rad/s

The following steps needs to be implemented to obtain a value in radians per second.

- The average in angular position travelled needs to be calculated from the summed value by dividing by the number of samples summed (which is always 20).
- The encoder values need to be converted to revolutions using the resolution of the encoder of $\frac{1}{1024}$ rev/unit.
- Convert revolutions to radians by multiplying with 2π.
- Now the radians per sampling period can be converted to rad/s by multiplying by 1000 and dividing by 5 for the 5ms sampling time.

The total conversion constant is calculated in equation (C.3.2)

$$v_{rad/s} = \frac{D_{avg}}{20} \times \frac{1}{1024} \times 2 * \pi \times \frac{1000}{5} = D_{avg} \times 0.06136 \tag{C.3.2}$$

C.4 Plant Identification

The RW PCB was supplied with 9V power and given different voltage steps using a PWM signal. The speed feedback was monitored and saved. A time constant and gain was then calculated for each response to form a transfer function. The transfer function constants obtained with static friction taken into account are listed in Table C.7 for the Z_b-axis RW.

The transfer functions obtained was given a 4V input step and the step respond plotted in Figure C.2 to show the different responses.

PWM duty cycle	**Voltag input [V]**	τ[s]	K[**A2D/V**]
20%	1.8-0.810	4.8s	857.576
30%	2.7-0.810	4.1s	759.259
40%	3.6-0.810	4s	715.412
50%	4.5-0.810	4s	690.244
60%	5.4-0.810	4s	672.331
70%	6.3-0.810	3.85s	638.434
80%	7.2-0.810	3.7s	620.344
90%	8.1-0.810	3.6s	601.097
100%	9-0.810	3.3s	568.865
Average constants:		3.93s	680.396

Table C.7: *Plant Identification Constants*

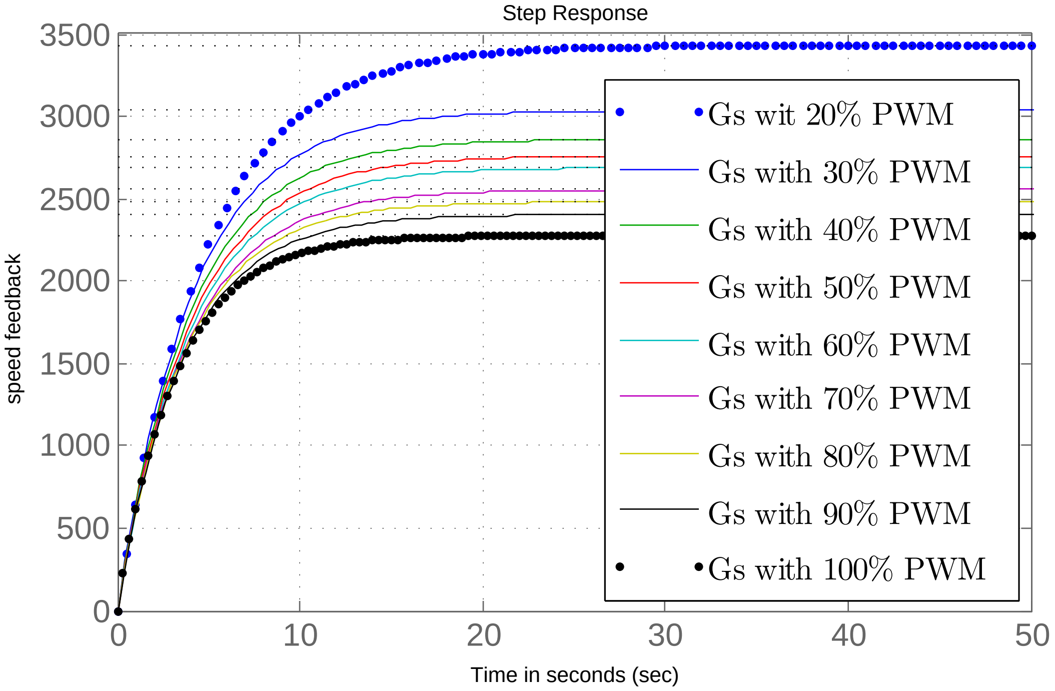


Figure C.2: *Transfer Functions Responses Given a 4 V Step*

As can bee seen in Figure C.2 and Table C.7 the transfer function gains change according to the input voltage and are not constant. The time constant remains more or less the same. However, the gain decreases as the voltage and speed increases. This is caused by the air friction on the flywheel that increases as the speed increases.

One transfer function representing the plant is required to design a controller or compensator. The transfer function with the largest gain was chosen to ensure that the plant will respond as per the design constraints. If the plant has less gain, then the plant will respond more sluggish or damped, but will still be stable. The chosen transfer function for the plant of the Z_b-axis RW is shown in equation (C.4.1).

$$G_{OL}(s) = \frac{A2D}{V} = \frac{KT}{s+T} = \frac{(857.576)(1/4.8)}{s+1/4.8} = \frac{178.7}{s+0.2083} \tag{C.4.1}$$

C.4.1 Control Specifications

The RW speed controller requires a zero steady state error and a 2 s settling time was attempted with an over-damped response. The closed loop poles were calculated from these requirements.

Given the 2% settling time σ was obtained as shown in equation (C.4.2) to (C.4.4).

$$t_{s2\%} = \frac{4}{\sigma} \tag{C.4.2}$$

$$\sigma = \frac{4}{t_{s2\%}} \tag{C.4.3}$$

$$\sigma = 2 \tag{C.4.4}$$

An over damped value of 0.9, as a value for ζ, was chosen after some experimentation.

$$\zeta = 0.9 \tag{C.4.5}$$

The natural frequency is calculated in equation (C.4.6) to (C.4.8):

$$\sigma = \zeta\omega_n \tag{C.4.6}$$

$$\omega_n = \frac{\sigma}{\zeta} \tag{C.4.7}$$

$$\omega_n = 2.222 \text{ rad/s} \tag{C.4.8}$$

The damped frequency is calculated in equation (C.4.9)

$$\omega_d = \omega_n\sqrt{1-\zeta^2} = 0.9686 \text{ rad/s} \tag{C.4.9}$$

The Zero steady state error implies a free integrator pole is required in the closed loop system. The desired closed loop poles are given in equation (C.4.10).

$$s_{CL} = -\sigma \pm j\omega_d = -\zeta\omega_n \pm j\omega_d = -2 \pm j0.9686 \tag{C.4.10}$$

C.4.2 Controller Design

The magnitude condition can be used to obtain the gain K needed to place the closed loop poles on the desired positions on the root locus. The Figure C.3 was used as reference for the root locus design.

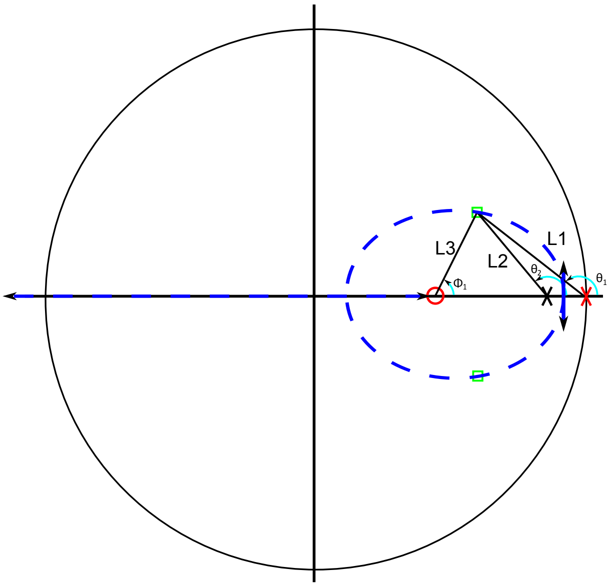


Figure C.3: *Root Locus Design*

Angle Condition

$$\angle(D(z)G(z)) = \pm 180 \text{ deg} \tag{C.4.11}$$

$$\sum \phi_i - \sum \theta_i = \pm 180 \text{ deg} \tag{C.4.12}$$

$$\theta_1 = 180 - arctan(\frac{0.0792}{1 - 0.8149}) = 156.83504 \text{ deg} \tag{C.4.13}$$

$$\theta_2 = 180 - arctan(\frac{0.0792}{0.9794 - 0.8149}) = 154.29109 \text{ deg} \tag{C.4.14}$$

$$\phi_1 = -180 + \sum \theta_i = -180 + 311.12613 = 131.12613 \text{ deg} \tag{C.4.15}$$

$$a = \frac{0.0792}{tan(180 - \phi_1)} + 0.8149 = 0.8840542 \tag{C.4.16}$$

Magnitude Condition

$$|K\ D(z_{cl})G(z_{cl})| = 1 \tag{C.4.17}$$

$$K = \frac{L_1 L_2}{L_3(17.68)} \tag{C.4.18}$$

$$L_1 = \sqrt{0.0792^2 + (1 - 0.8149)^2} = 0.20133219 \tag{C.4.19}$$

$$L_2 = \sqrt{0.0792^2 + (0.9794 - 0.8149)^2} = 0.18257297 \tag{C.4.20}$$

$$L_3 = \sqrt{0.0792^2 + (0.8840542 - 0.8149)^2} = 0.10514249 \tag{C.4.21}$$

$$K = \frac{(0.20133219)(0.18257297)}{(0.10514249)(17.68)} = 0.01977376 \tag{C.4.22}$$

C.4.2.1 Compensator

The compensator or controller is now given as:

$$D(z) = \frac{0.01977376(z - 0.8840542)}{z - 1} \tag{C.4.23}$$

The plant and controller was implemented in MATLAB Sisotool, in order to confirm that the closed loop poles went through the desired points.

The control differential equation that was implemented on the MCU is calculated in equation (C.4.24) to (C.4.26).

$$D(z) = \frac{0.01977376(z - 0.8840542)}{z - 1} = \frac{U(z)}{E(z)} \tag{C.4.24}$$

$$zU(z) - U(z) = z0.01977376E(z) - 0.01748108E(z) \tag{C.4.25}$$

$$u(k) = 0.01977376e(k) - 0.01748108e(k-1) + u(k-1) \tag{C.4.26}$$

The control equation was programmed on the MCU and the control was tested.

C.5 Control Validation

Figure C.4 shows how the RW responds to different speed steps sizes at different speeds and directions.

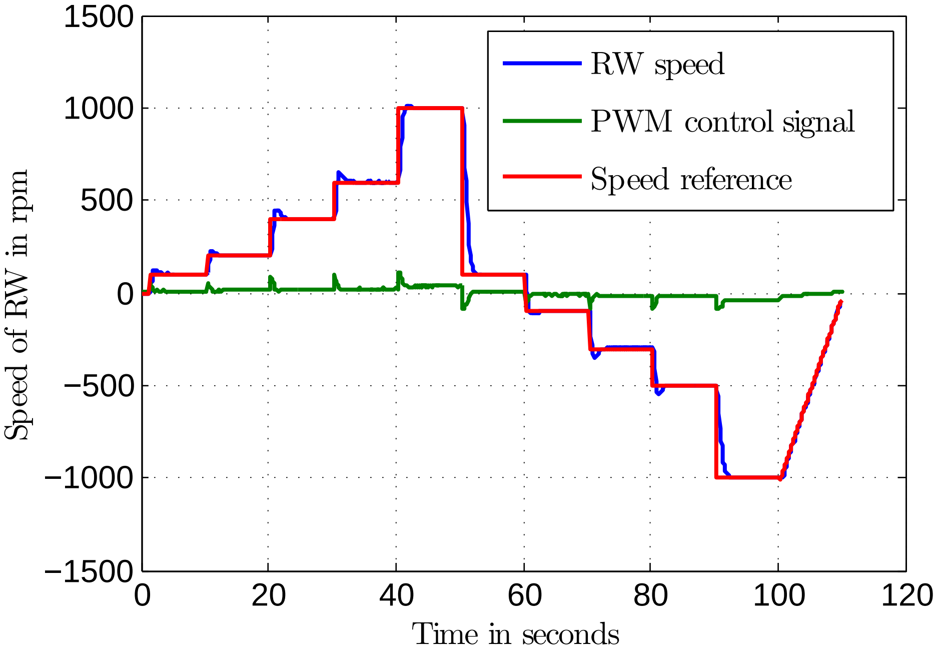


Figure C.4: *Z_b-Axis RW Speed Reference Steps*

Ramp speed references were generated at low speed around zero as shown in Figure C.5, to demonstrate the effect of friction and how well the RW can follow a small speed ramp input.

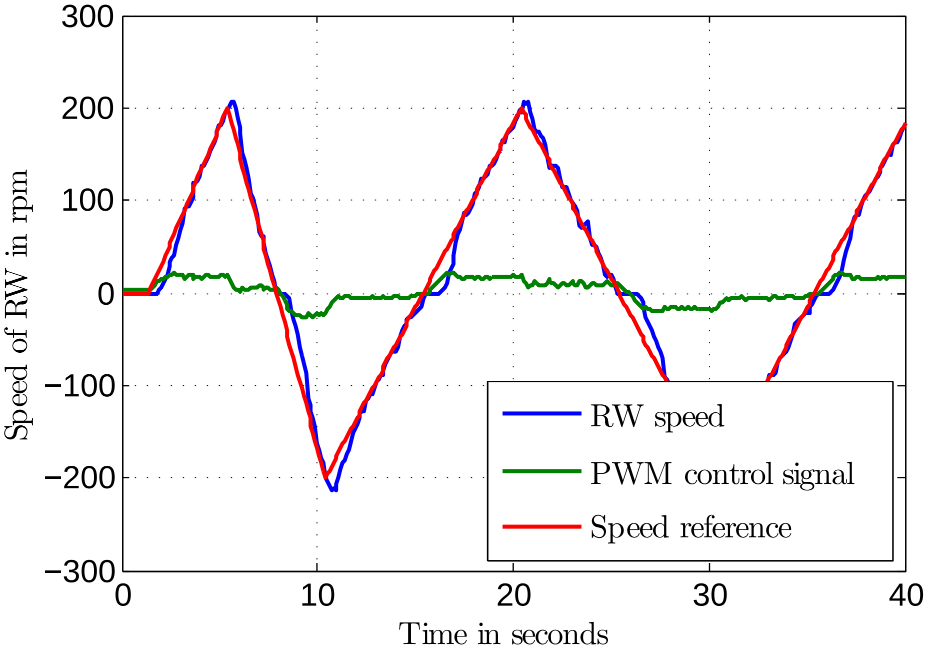


Figure C.5: *Z_b-Axis RW Response to 200 rpm Ramp Inputs around Zero RPM*

The effect of friction can be seen in Figure C.5, where the RW speed is observed to go through zero and struggle to move on. A 1500 rpm ramp input response is shown in Figure C.6. In this figure the effect of friction is less notable as the speed changes faster and the motor can overcome the friction more effectively. The large change in speed at the ramp peaks shows the motor speed overshooting the speed reference.

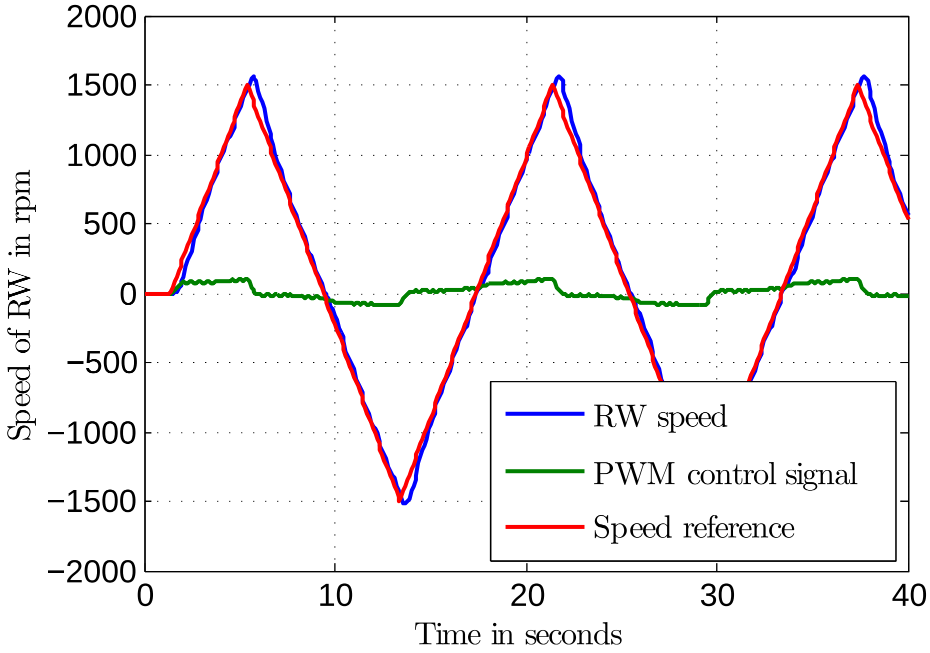


Figure C.6: *Z_b-Axis RW 1500 rpm Ramp Speed Reference around Zero RPM Response*

Figure C.7 shows how the RW responds to a sine wave speed input. Again the effect of friction can be noted where the wheel speed passes through zero.

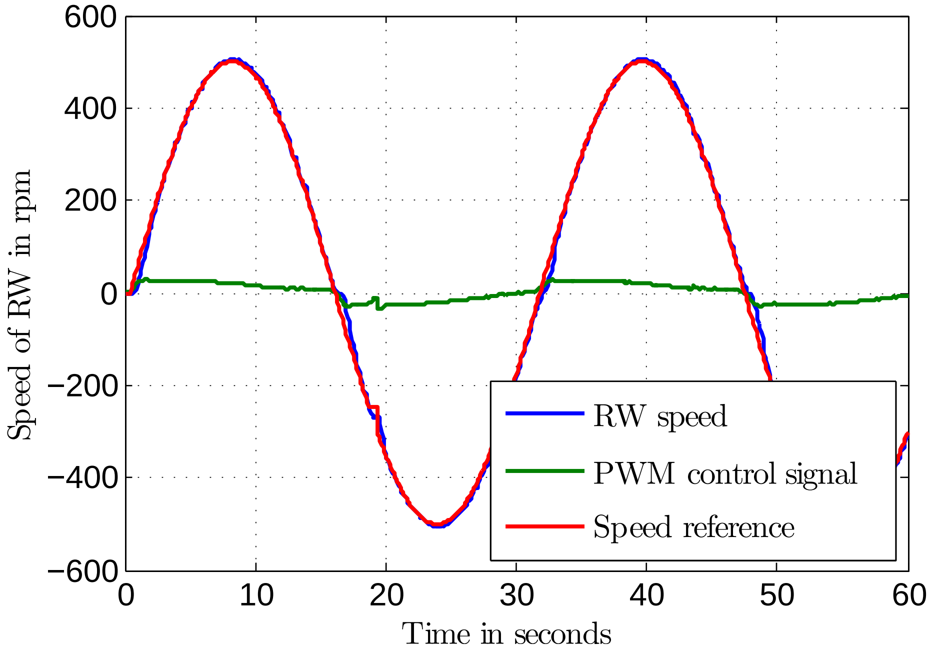


Figure C.7: *Z_b-Axis RW Response to a 500 rpm Amplitude Sine Wave Speed Reference*

Appendix D

Datasheets

D.1 Photo-diode

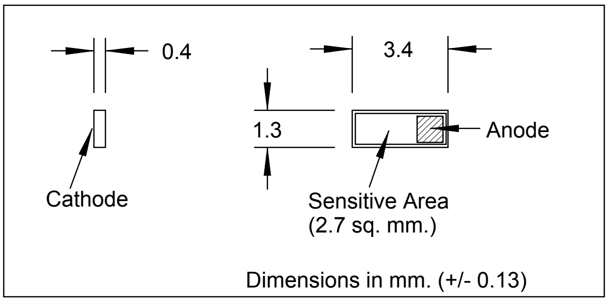
SILONEX inc

SLCD-61N8

Solderable Planar Photodiode

Features

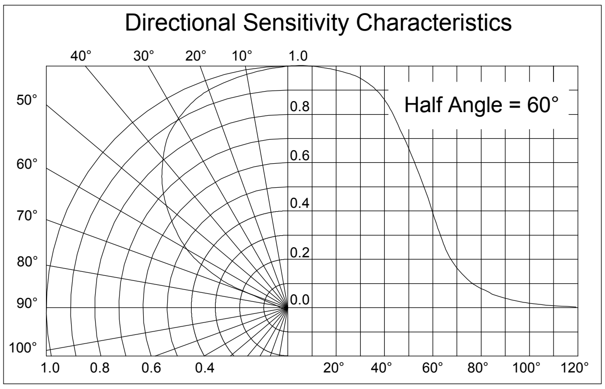
- Visible to IR spectral irradiance range
- High reliability
- Oxide passivation
- Linear short circuit current
- Low capacitance, high speed
- Available in arrays where # indicates number of elements (maximum of 8 elements)



Also available with leads as part number SLSD-71N8

Description

The Silonex series of silicon solderable planar photodiodes feature low cost, high reliability, and linear short circuit current over a wide range of illumination. These devices are widely used for light sensing and power generation because of their stability and high efficiency. They are particularly suited to power conversion applications due to their low internal impedance and relatively high shunt impedances, and stability. These devices also provide a reliable, inexpensive detector for applications such as light beam sensing and instrumentation. The electrical characteristics below are per element. In the multielement arrays the cathodes are common to a all elements.



Absolute Maximum Ratings

Storage Temperature	-40°C to +125°C
Operating Temperature	-40°C to +125°C

Electrical Characteristics (T$_A$=25°C unless otherwise noted)

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
I_{SC}	Short Circuit Current	100	170		μA	V_R=0V, Ee=25mW/cm^2 (1)
V_{OC}	Open Circuit Voltage		0.40		V	Ee=25mw/cm^2 (1)
I_D	Reverse Dark Current			1.7	μA	V_R=5V, Ee=0
C_J	Junction Capacitance		100		pF	V_R=0V, Ee=0, f=1MHz
S_λ	Spectral Sensitivity		0.55		A/W	λ=940nm
V_{BR}	Reverse Breakdown Voltage	20			V	I_R=100μA
λ_P	Maximum Sensitivity Wavelength		930		nm	
λ_R	Sensitivity Spectral Range	400		1100	nm	
$\theta_{1/2}$	Acceptance Half Angle		60		deg	

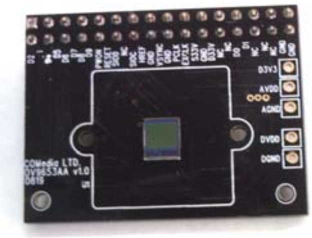
Specifications subject to change without notice

Notes: (1) Ee = light source @ 2854 °K

D.2 Camera Datasheet

AA9653
Sensor Module

General Description

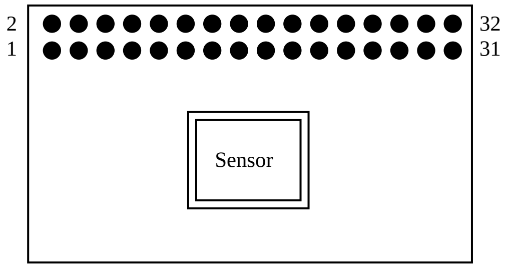


This is a series of digital sensor module using various OmniVision' CMOS image sensor.

The digital video port provides full-frame, sub-sampled or windowed 8/10bit images in a wide range of formats, control thru the serial camera control bus.

Features:

- Mega pixel CMOS image sensor, OV9653
- Sensor array 1300x1028pixels
- lens format: 1/4"
- support SXGA,VGA,QVGA,CIF,QCIF
- VGA 30fps, QVGA 60fps, QCIF 120fps
- Raw RGB, GRB422, YUV422 format
- Small size : 41 x 31 mm
- Built-in Len (Optional)
- Progressive Scan
- Serial Camera Control Bus interface
- Single 3.3V operation
- Low power consumption (<120mW)



PCB Layout (Top view)

Pin Description

1~8	D2-D9	Digital output Bus.
9	PWDN	Power down mode, active high
10	RST	Reset, active high
11	SIOD	Serial data
12	NC	
13	SIOC	Serial clock input
14	HREF	Horizontal window reference output
15	AGND	Analog Ground
16	VSYN	Vertical Sync output
17	AGND	Analog Ground
18	PCLK	Pixel clock output
19	EXCLK	External clock input
20	VCC	Power Supply 3.3VDC
21	AGND	Analog Ground
22	VCC	Power Supply 3.3VDC
23~24	NC	
25~26	D0-D1	Digital output bus
27-30NC		
31~32	GND	Common ground

Lens Information

Focal length	6.5mm
F No	3.2
Distortion	<3%
Focus	2M to infinity
FOV	41 deg diagonal
IR cut filter	650nm

Module Specification

Dimension	41x31mm
Header Pin	16pin x 2 row
Pin pitch	2.54mm
Total height from pcb	13mm

Note: Please refer to the data sheet of OV sensor for detail specification and features.

Advanced Information
Preliminary Datasheet

OV9625 Color CMOS SXGA (1.3 MPixel) CAMERACHIP™
OV9121 B&W CMOS SXGA (1.3 MPixel) CAMERACHIP™

General Description

The OV9625 (color) and OV9121 (black and white) are high-performance 1.3 mega-pixel CAMERACHIPS™ for digital still image and video camera products.

Both devices incorporate a 1280 x 1024 (SXGA) image array and an on-chip 10-bit A/D converter capable of operating at up to 15 frames per second (fps) at full resolution and an improved micro lens design to decrease shading. Proprietary sensor technology utilizes advanced algorithms to cancel Fixed Pattern Noise (FPN), eliminate smearing, and drastically reduce blooming. The control registers allow for flexible control of timing, polarity, and CameraChip operation, which, in turn, allows the engineer a great deal of freedom in product design.

Features

- Optical Black Level Calibration (BLC)
- Improved micro lens design to decrease shading
- Video or snapshot operations
- Programmable/Auto Exposure and Gain Control
- Programmable/Auto White Balance Control
- Horizontal and vertical sub-sampling (4:2 and 4:2)
- Programmable image windowing
- Variable frame rate control
- On-chip R/G/B channel and luminance average counter
- Internal/External frame synchronization
- SCCB slave interface
- Power-on reset and power-down mode

Ordering Information

Product	Package
OV09625-C00A (Color, SXGA, VGA)	CLCC-48
OV09121-C00A (B&W with microlens, SXGA, VGA)	CLCC-48

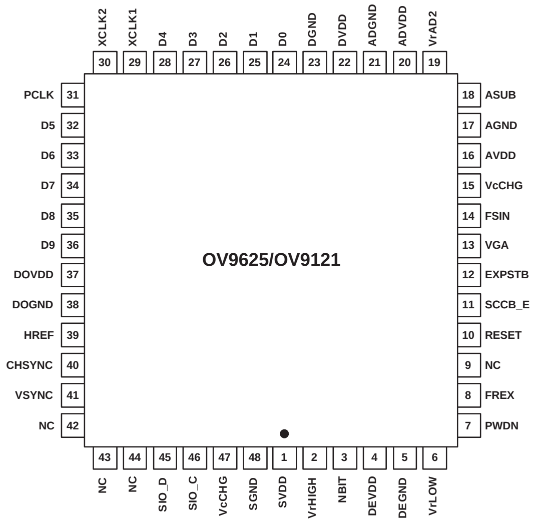
Applications

- Digital still cameras
- PC camera/dual mode
- Video conference applications
- Machine vision
- Security cameras
- Biometrics

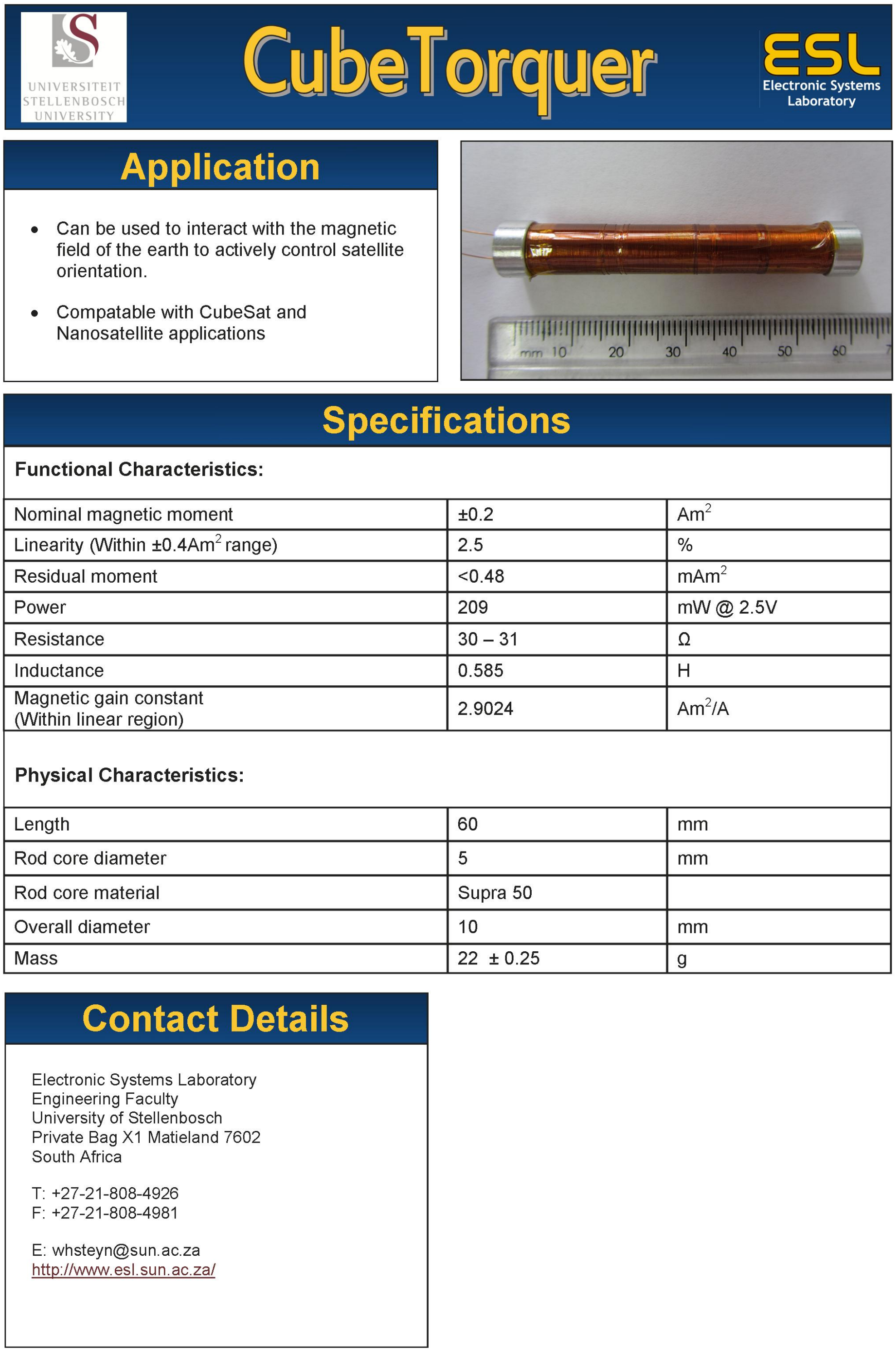
Key Specifications

Array Size	SXGA	1280 x 1024
	VGA	640 x 480
Power Supply	Core	2.5VDC ± 10%
	Analog	3.3VDC ± 10%
	I/O	3.3VDC ± 10%
Power Requirements	Active	< 50 mA
	Standby	< 10 μA
Output Formats (10-bit)		Raw RGB Data
Lens Size		1/2"
Max. Image Transfer Rate	SXGA	15 fps
	VGA	30 fps
Sensitivity		1.0 V/Lux-sec
S/N Ratio		54 dB
Dynamic Range		60 dB (due to ADC limitations)
Scan Mode		Progressive
Gamma Correction		N/A
Electronics Exposure	SXGA	Up to 1050:1
	VGA	Up to 500:1
Pixel Size		5.2 μm x 5.2 μm
Dark Current		28 mV/s
Fixed Pattern Noise		< 0.03% of $V_{PEAK-TO-PEAK}$
Image Area		6.66 mm x 5.32 mm
Package Dimensions		.560 in. x .560 in.

Figure 1 OV9625/OV9121 Pin Diagram



D.3 CubeTorquer Datasheet



UNIVERSITEIT STELLENBOSCH UNIVERSITY

CubeTorquer

ESL Electronic Systems Laboratory

Application

- Can be used to interact with the magnetic field of the earth to actively control satellite orientation.
- Compatable with CubeSat and Nanosatellite applications

Specifications

Functional Characteristics:

Nominal magnetic moment	±0.2	Am^2
Linearity (Within ±0.4Am^2 range)	2.5	%
Residual moment	<0.48	mAm^2
Power	209	mW @ 2.5V
Resistance	30 – 31	Ω
Inductance	0.585	H
Magnetic gain constant (Within linear region)	2.9024	Am^2/A

Physical Characteristics:

Length	60	mm
Rod core diameter	5	mm
Rod core material	Supra 50	
Overall diameter	10	mm
Mass	22 ± 0.25	g

Contact Details

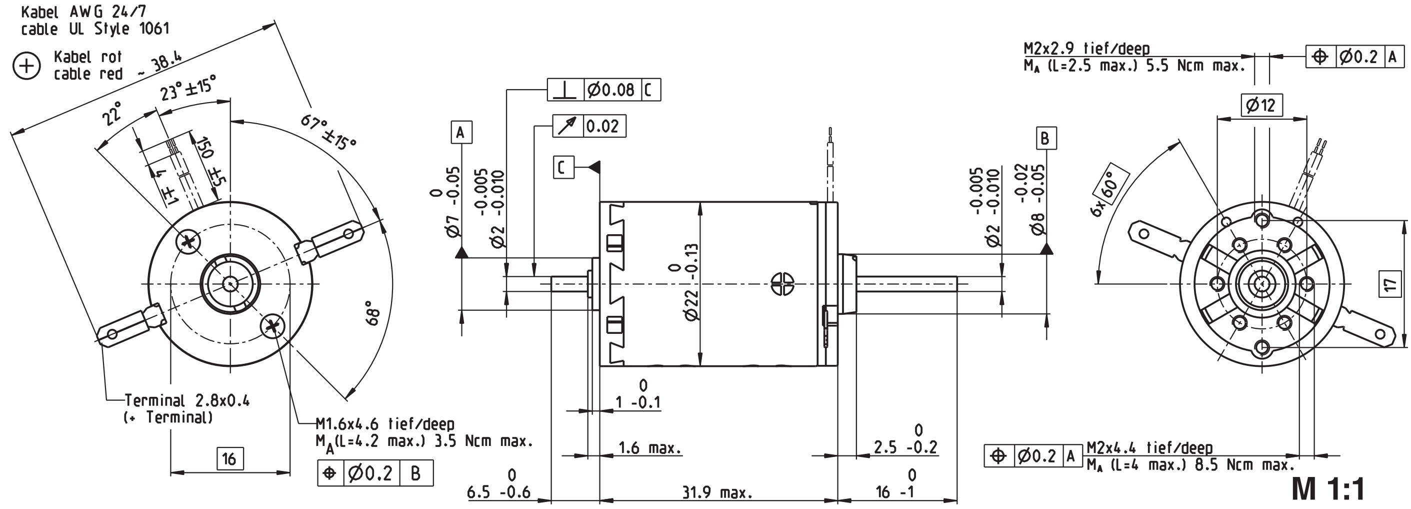
Electronic Systems Laboratory
Engineering Faculty
University of Stellenbosch
Private Bag X1 Matieland 7602
South Africa

T: +27-21-808-4926
F: +27-21-808-4981

E: whsteyn@sun.ac.za
http://www.esl.sun.ac.za/

D.4 Motor Datasheet

A-max 22 ∅22 mm, Precious Metal Brushes CLL, 3.5 Watt, CE approved



🟥 Stock program
□ Standard program
▨ Special program (on request)

Article Numbers

	with terminals	110130	110132	110133	110134	110135	110136	110137	110138	110139	110140	110141	110142
	with cables	139846	352986	352987	352988	352989	352990	352991	352992	352993	352994	352995	352996
Motor Data													
Values at nominal voltage													
1 Nominal voltage	V	4.5	6	7.2	7.2	7.2	9	12	15	18	24	36	42
2 No load speed	rpm	7210	6630	7000	6240	5620	6140	6630	6680	6480	6520	6950	7320
3 No load current	mA	26.7	17.8	16	13.6	11.8	10.6	8.88	7.17	5.73	4.33	3.16	2.92
4 Nominal speed	rpm	4970	4380	4770	3990	3340	3830	4320	4360	4140	4160	4540	4940
5 Nominal torque (max. continuous torque)	mNm	4.82	5.24	5.27	5.27	5.22	5.15	5.1	5.09	5.05	5	4.89	4.92
6 Nominal current (max. continuous current)	A	0.84	0.628	0.555	0.495	0.442	0.381	0.306	0.246	0.197	0.147	0.103	0.0932
7 Stall torque	mNm	15.1	15.3	16.4	14.6	12.8	13.7	14.6	14.7	14.1	13.9	14.2	15.2
8 Starting current	A	2.56	1.79	1.69	1.34	1.06	0.992	0.856	0.693	0.537	0.399	0.29	0.28
9 Max. efficiency	%	81	81	82	81	80	81	81	81	81	81	81	81
Characteristics													
10 Terminal resistance	Ω	1.76	3.36	4.27	5.39	6.78	9.07	14	21.6	33.5	60.1	124	150
11 Terminal inductance	mH	0.106	0.222	0.288	0.362	0.445	0.584	0.89	1.37	2.1	3.68	7.29	8.95
12 Torque constant	mNm/A	5.9	8.55	9.73	10.9	12.1	13.9	17.1	21.2	26.2	34.8	48.9	54.3
13 Speed constant	rpm/V	1620	1120	981	875	790	689	558	450	364	274	195	176
14 Speed / torque gradient	rpm/mNm	482	438	430	432	443	451	458	459	465	474	494	486
15 Mechanical time constant	ms	20.6	19.8	19.7	19.7	19.8	20.2	20.1	20.2	20.3	20.3	20.6	20.4
16 Rotor inertia	gcm^2	4.07	4.32	4.38	4.36	4.26	4.27	4.2	4.21	4.16	4.1	3.97	4.01

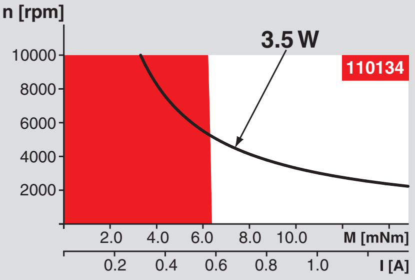
Specifications

Thermal data	
17 Thermal resistance housing-ambient	20 K/W
18 Thermal resistance winding-housing	6.0 K/W
19 Thermal time constant winding	10.2 s
20 Thermal time constant motor	313 s
21 Ambient temperature	-30…+65°C
22 Max. permissible winding temperature	+85°C
Mechanical data (sleeve bearings)	
23 Max. permissible speed	10000 rpm
24 Axial play	0.05 - 0.15 mm
25 Radial play	0.012 mm
26 Max. axial load (dynamic)	1 N
27 Max. force for press fits (static)	80 N
(static, shaft supported)	440 N
28 Max. radial loading, 5 mm from flange	2.8 N
Mechanical data (ball bearings)	
23 Max. permissible speed	10000 rpm
24 Axial play	0.05 - 0.15 mm
25 Radial play	0.025 mm
26 Max. axial load (dynamic)	3.3 N
27 Max. force for press fits (static)	45 N
(static, shaft supported)	440 N
28 Max. radial loading, 5 mm from flange	12.3 N
Other specifications	
29 Number of pole pairs	1
30 Number of commutator segments	9
31 Weight of motor	54 g
CLL = Capacitor Long Life	

Values listed in the table are nominal.
Explanation of the figures on page 49.

Option
Ball bearings in place of sleeve bearings
Without CLL

Operating Range



Comments

🟥 **Continuous operation**
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

□ **Short term operation**
The motor may be briefly overloaded (recurring).

— **Assigned power rating**

maxon Modular System

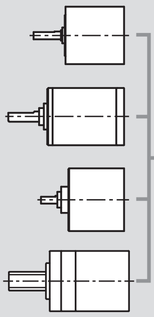
Overview on page 16 - 21

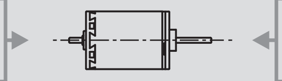
Planetary Gearhead
∅22 mm
0.1 - 0.6 Nm
Page 222/223

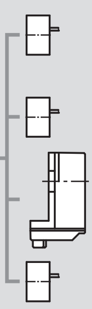
Planetary Gearhead
∅22 mm
0.5 - 2.0 Nm
Page 224/225

Spur Gearhead
∅24 mm
0.1 Nm
Page 230

Spindle Drive
∅22 mm
Page 253/254







Recommended Electronics:

ESCON 36/2 DC	Page 292
ESCON 50/5	292
EPOS2 24/2	312
EPOS2 Module 36/2	312
EPOS3 70/10 EtherCAT	319
Notes	**18**

Encoder MR
32 CPT,
2 / 3 channels
Page 268

Encoder MR
128 / 256 / 512 CPT,
2 / 3 channels
Page 270

Encoder Enc
22 mm
100 CPT, 2 channels
Page 275

Encoder MEnc
∅13 mm
16 CPT, 2 channels
Page 284

D.5 Magnetic Encoder Datasheet

austriamicrosystems

AS5046
PROGRAMMABLE 12-bit 360° MAGNETIC ANGLE ENCODER
WITH ABSOLUTE 2-WIRE SERIAL AND ANALOG INTERFACES

PRELIMINARY DATA SHEET

1 General Description

The AS5046 is a contactless magnetic angle encoder for accurate measurement up to 360°.

It is a system-on-chip, combining integrated Hall elements, analog front end and digital signal processing in a single device.

The AS5046 provides a digital serial 12-bit as well as a programmable 10-bit ratiometric analog output that is directly proportional to the angle of a magnet, rotating over the chip.

In addition, the serial interface enables a user configurable arrangement of the Hall array and allows access to each individual Sensor of the Hall Array.

The AS5046 also provides high resolution information of the magnetic field strength, respectively the vertical distance of the magnet, thus adding excellent state-of-health information of the overall system.

An internal voltage regulator allows operation of the AS5046 from 3.3V or 5.0V supplies.

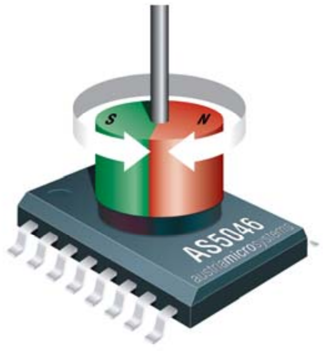


Figure 1: Typical arrangement of AS5046 and magnet

Benefits

- Complete system-on-chip
- High reliability due to non-contact sensing
- Bi-directional 2-wire interface
- Programmable ratiometric analog output
- Ideal for application s in harsh environments
- Robust system, tolerant to magnet misalignment, airgap variations, temperature variations and external magnetic fields
- No calibration required

2 Key Features

- 360° contactless high resolution angular position encoding
- User programmable zero position
- 12-bit 2-wire serial interface
- Versatile analog output
  programmable angular range up to 360°
  programmable ratiometric output voltage range
- High resolution magnet distance indication
  256 steps within recommended range (~0.5 to 1.8mm)
  256 steps over extended range (~0 to 5mm)
- Mode input for optimizing noise vs. speed
- Alignment mode for magnet placement guidance
- Wide temperature range: - 40°C to + 125°C
- Small package: SSOP 16 (5.3mm x 6.2mm)

3 Applications

The AS5046 is ideal for applications that require high resolution, a minimum of wires between controller and sensor and where the vertical distance of the magnet is of importance:

- Remote sensors
- Rotate-and-push manual input devices
- Joysticks
- Applications with extended safety requirements regarding magnet distance

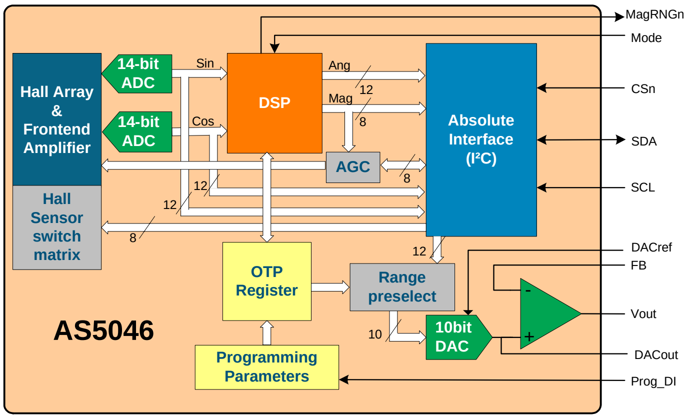


Figure 2: AS5046 block diagram

D.6 EyasSAT Owners Manual

Creating EyasSat Experiments

- The EyasSat Educational Satellite System™ is designed to be expanded with Experiment Modules
  - Each EyasSat™ stack can support up to three Experiment Modules simultaneously
- Experiment Modules can be powered by the EyasSat Power module
- The IHU can "attach" each Experiment Module individually. When the attach command is sent:
  - The module is powered on
  - The IHU polls it for telemetry
    - Experiment modules gather and format their own telemetry
  - Commands directed at Experiment Modules are passed through by the IHU

Experiment Module Requirements

- Electrical. All connections are via the EyaBUSTM
  - Connection to the SPI bus as an SPI slave
    - Chip Select, MISO, MOSI, SCK
  - Connection to power
    - 9V unregulated constant power (only if unswitched high current is needed)
    - 3.3V regulated switched power (choose 3.3V or 5V or both)
    - 5V regulated switched power
    - Ground

- Logical/software
  - Function as an SPI slave
    - Respond to telemetry polls from IHU
    - Respond to commands passed through by IHU

Experiment Module Steps

1) Choose a microcontroller that supports SPI
   1. Atmel® (ATMEGA128 is standard on EyasSat™ modules)
   2. Systronix ™ JStamp ™

    3. Savage Innovations OOPic ™ II+
    4. Parallax ™ Basic Stamp ™ II
    5. PIC
    6. Or choose and FPGA and program it as SPI slave (review SPI timing diagrams if necessary)
    7. Or any other processor that runs from 3.3V or 5V and supports SPI
2) Put it on a PC-104 form factor board
    1. PC104 form factor including the bus connectors and mounting holes
    2. Must **NOT** implement the PC-104 bus standard (Use the EyaBus ™ standard)
    3. Recommend including in-system programming capability

- Connect the microcontroller to the EyaBus ™
    - Power: 9V, 3.3V or 5V switched, ground
    - SPI Pins: MISO, MOSI, SCK
    - Choose the Chip Select for Experiment #1, 2 or 3
- Implement Firmware
    - Respond via SPI to requests for telemetry (see Firmware Overview)
    - Respond to commands via SPI in this format:
        - <Command Character><Value Character> (i.e. t1 or f0)

Important EyasBus™ Pins

- SPI
    - SCK – A11(Connect to your microcontroller's SCK)
    - MOSI – A12 (Connect to your microcontroller's MOSI)
    - MISO – A13 (Connect to your microcontroller's MISO)
    - Chip Select (Pick one)
        - Experiment #1 – A19
        - Experiment #2 – A18
        - Experiment #3 – A17
- Power
    - +9V Raw - A1 (Use only if you need raw +9V)
    - Ground – B1
    - +3.3V Regulated Switched (Pick one if you need +3.3V)
        - Experiment #1 – B8
        - Experiment #2 – B9
        - Experiment #3 – B11
    - +5.0V Regulated Switched (Pick one)

        - Experiment #1 – B12
        - Experiment #2 – A2
        - Experiment #3 – A8
    - **NOTE: EyasSat™ is a 5V system**. 3.3V is provided for internal use in Experiments. All EyasBus™ logic and control signals are 5V. If your experiment runs on 3.3V add level shifters like the MAX3371 for the interface to the EyaBUS'

Firmware Overview

- Minimum required firmware is an SPI interrupt service routine (ISR)
    - Triggered when Data is received via SPI
    - Implements a state machine with three states
        - Idle Mode
            - Waiting to receive a command via SPI (default)
            - When any other mode ends, this mode becomes active
        - Telemetry Mode
            - In process of returning telemetry string to IHU
            - This mode activated when a 'T' is received in Idle Mode
            - This mode is ended when an 'R' is received in Telemetry Mode
        - Command Mode
            - Command received, waiting for command value
            - This mode activated when anything but a 'T' or 'R' is received in Idle Mode
            - This mode is ended when an 'R' is received in Command Mode

Firmware Example (Telemetry Mode)

- Telemetry mode
    - You need to prepare the telemetry string outside the interrupt to maximize performance
        - Main loop of firmware can prepare telemetry string on a regular basis
    - Once in telemetry mode

    - The telemetry string is sent to the IHU one character at a time when polled by the IHU
    - IHU will poll for 120 characters
    - After 120 characters, IHU will send an 'R'

TLM mode SPI Traffic

- Experiment SPI ISR is in idle state, TLM data has been collected and formatted into a character string
- IHU sends a 'T' - Experiment enters TLM mode
- IHU waits 60ms
- IHU sends '0' characters - Experiment sends back the TLM string one character at a time
  - IHU sends '0's until a hex 00 is returned or 120 characters have been received
- IHU sends an 'R' - Experiment returns to ISR idle state
- IHU then sends the TLM string on the downlink

Firmware Example (Command Mode)

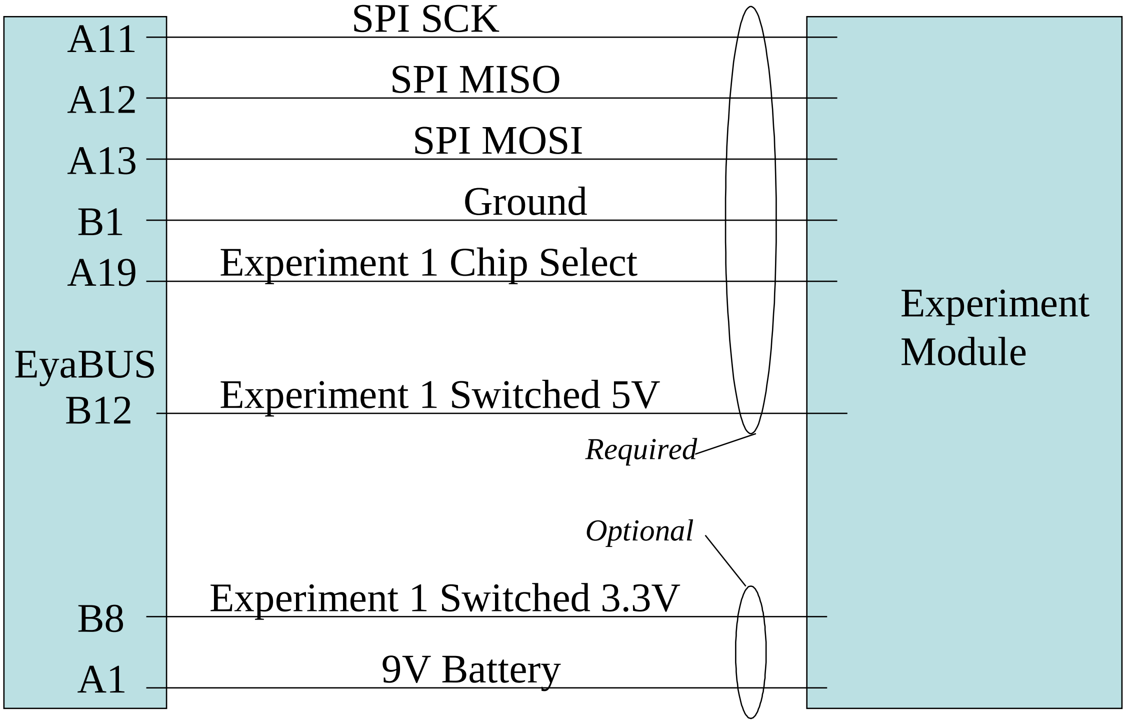
- Commands consist of
  - Command character
    - Any character – indicates command to invoke
  - Value character
    - Any character – parameter to the command
  - Examples
    - l1 – turns your laser on (l0 turns it off)
    - p2 – sets the laser power to 2 (p5 sets it to power level 5)
    - t<0x37> - sets an I/O port to 0x37
  - When in command mode, you have received the command character from the IHU and are waiting for the value character from the IHU
  - Typically, subroutines are called for each command, and passed the parameter

Command Mode SPI Traffic

- IHU passes through commands for experiments. Experiment one in this example
- Example: From radio link 1l1 is entered.
  - IHU passes l1 to experiment one
  - Experiment one takes the l as the command and the 1 as the value for that command to use

- IHU responds on the radio link with a message indicating what was sent to the experiment
- IHU does not look for a response from the experiment

Connections to an Experiment Board



D.7 EyasSAT User Manual

Appendix C (EyaBus Description)

Patent Pending

			Pin	Row A	Row B	Pin
			1	9V Unreg Bat	GND	1
			2	+5 sw 5 (Experiment 2)	+3 Vcc	2
			3	+5 sw 6 (ADCS)	+5 Vcc (DH and Comm)	3
			4	Ambiant temp (PF7)		4
			5	Experiment temp (PF6)		5
			6	Bat temp (PF5)	5V reg temp (PF1)	6
			7		Analog GND	7
Row D	**Row C**	**Pin**	8	+5 sw 7 (Experiment 3)	+3.3 sw 1 (Experiment 1)	8
Exp Interboard	Exp Interboard	0	9	+5 sw 8 (reserved)	+3.3 sw 2 (Experiment 2)	9
Exp Interboard	Exp Interboard	1	10	/SS (PB0)	N/C	10
Solar pnl2 +	Solar pnl2 -	2	11	SCK (PB1)	+3.3 sw 3 (Experiment 3)	11
T/R Z +	T/R Z -	3	12	MISO (PB3)	+5 sw 4 (Experiment 1)	12
T/R X +	T/R X -	4	13	MOSI (PB2)	IHU PWM	13
T/R Y +	T/R Y -	5	14			14
Therm B +	Therm B -	6	15	/CS 6 (PC6)		15
Therm W +	Therm W -	7	16	/CS 5 (PC5)		16
Panel temp +	Panel temp -	8	17	/CS 4 (PC4) (Experiment 3)		17
Solar pnl1 +	Solar pnl1 -	9	18	/CS 3 (PC3) (Experiment 2)		18
Sep SW +	Sep SW -	10	19	/CS 2 (PC2) (Experiment 1)	Gen I/O 7 (PA7)	19
Sun Bottom+	Sun Bottom-	11	20	/CS 1 (PC1) (ADCS board)	Gen I/O 6 (PA6)	20
Sun Top +	Sun Top -	12	21	/CS 0 (PC0) (Power board)	Gen I/O 5 (PA5)	21
Enable +	Enable -	13	22		Gen I/O 4 (PA4)	22
Case reserved	Yaw 1+	14	23		Gen I/O 3 (PA3)	23
Yaw 2 +	Yaw 3 +	15	24	Exp Serial TTL RXD	XStr CTS	24
Yaw 4 +	Yaw -	16	25	Exp Serial TTL TXD	XBee CTS	25
Charge +	Charge +	17	26	RF serial RX TTL (PD2)	RF speed select	26
Charge -	Charge -	18	27	RF serial TX TTL (PD3)		27
Case reserved	N/C	19	28		GPS1PPS	28
			29		+5 Vcc (DH and Comm)	29
			30	I2C data (PD1)	XBee RSSI	30
			31	I2C clk (PD0)	GND	31
			32	GND	GND	32

D.8 Microcontroller Datasheet

EFM®32

... the world's most energy friendly microcontrollers

EFM32GG280 DATASHEET

F1024/F512

Preliminary

- **ARM Cortex-M3 CPU platform**
  - High Performance 32-bit processor @ up to 48 MHz
  - Memory Protection Unit
- **Flexible Energy Management System**
  - 20 nA @ 3 V Shutoff Mode
  - 0.4µA @ 3 V Shutoff Mode with RTC
  - 0.9 µA @ 3 V Stop Mode, including Power-on Reset, Brown-out Detector, RAM and CPU retention
  - 1.1 µA @ 3 V Deep Sleep Mode, including RTC with 32.768 kHz oscillator, Power-on Reset, Brown-out Detector, RAM and CPU retention
  - 50 µA/MHz @ 3 V Sleep Mode
  - 200 µA/MHz @ 3 V Run Mode, with code executed from flash
- **1024/512 KB Flash**
  - Read-while-write support
- **128/128 KB RAM**
- **85 General Purpose I/O pins**
  - Configurable push-pull, open-drain, pull-up/down, input filter, drive strength
  - Configurable peripheral I/O locations
  - 16 asynchronous external interrupts
  - Output state retention and wake-up from Shutoff Mode
- **12 Channel DMA Controller**
- **12 Channel Peripheral Reflex System (PRS) for autonomous inter-peripheral signaling**
- **Hardware AES with 128/256-bit keys in 54/75 cycles**
- **Timers/Counters**
  - 4× 16-bit Timer/Counter
    - 4×3 Compare/Capture/PWM channels
    - Dead-Time Insertion on TIMER0
  - 16-bit Low Energy Timer
  - 1× 24-bit Real-Time Counter and 1× 32-bit Real-Time Counter
  - 3× 16/8-bit Pulse Counter with asynchronous operation
  - Watchdog Timer with dedicated RC oscillator @ 50 nA
- **Backup Power Domain**
  - RTC and retention registers in a separate power domain, available in all energy modes
  - Operation from backup battery when main power drains out
- **External Bus Interface for up to 4x256 MB of external memory mapped space**
  - TFT Controller with Direct Drive
- **Communication interfaces**
  - 3× Universal Synchronous/Asynchronous Receiver/Transmitter
    - UART/SPI/SmartCard (ISO 7816)/IrDA/I2S
  - 2× Universal Asynchronous Receiver/Transmitter
  - 2× Low Energy UART
    - Autonomous operation with DMA in Deep Sleep Mode
  - 2× I^2C Interface with SMBus support
    - Address recognition in Stop Mode
- **Ultra low power precision analog peripherals**
  - 12-bit 1 Msamples/s Analog to Digital Converter
    - 8 single ended channels/4 differential channels
    - On-chip temperature sensor
  - 12-bit 500 ksamples/s Digital to Analog Converter
  - 2× Analog Comparator
    - Capacitive sensing with up to 16 inputs
  - 3× Operational Amplifier
    - 6.1 MHz GBW, Rail-to-rail, Programmable Gain
  - Supply Voltage Comparator
- **Low Energy Sensor Interface (LESENSE)**
  - Autonomous sensor monitoring in Deep Sleep Mode
  - Wide range of sensors supported, including LC sensors and capacitive buttons
- **Ultra efficient Power-on Reset and Brown-Out Detector**
- **Debug Interface**
  - 2-pin Serial Wire Debug interface
    - 1-pin Serial Wire Viewer
  - Embedded Trace Module v3.5 (ETM)
- **Pre-Programmed Serial Bootloader**
- **Temperature range -40 to 85 ºC**
- **Single power supply 1.85 to 3.8 V**
- **LQFP100 package**

32-bit ARM Cortex-M0+, Cortex-M3 and Cortex-M4 microcontrollers for:

- Energy, gas, water and smart metering
- Health and fitness applications
- Smart accessories
- Alarm and security systems
- Industrial and home automation
- www.energymicro.com/gecko

ZERO TINY GECKO LEOPARD GIANT WONDER

ENERGY® micro

Appendix E

PCB

E.1 PC 104 Standard

The standard size and hole locations for the PC104 is shown in figure E.1 [28]

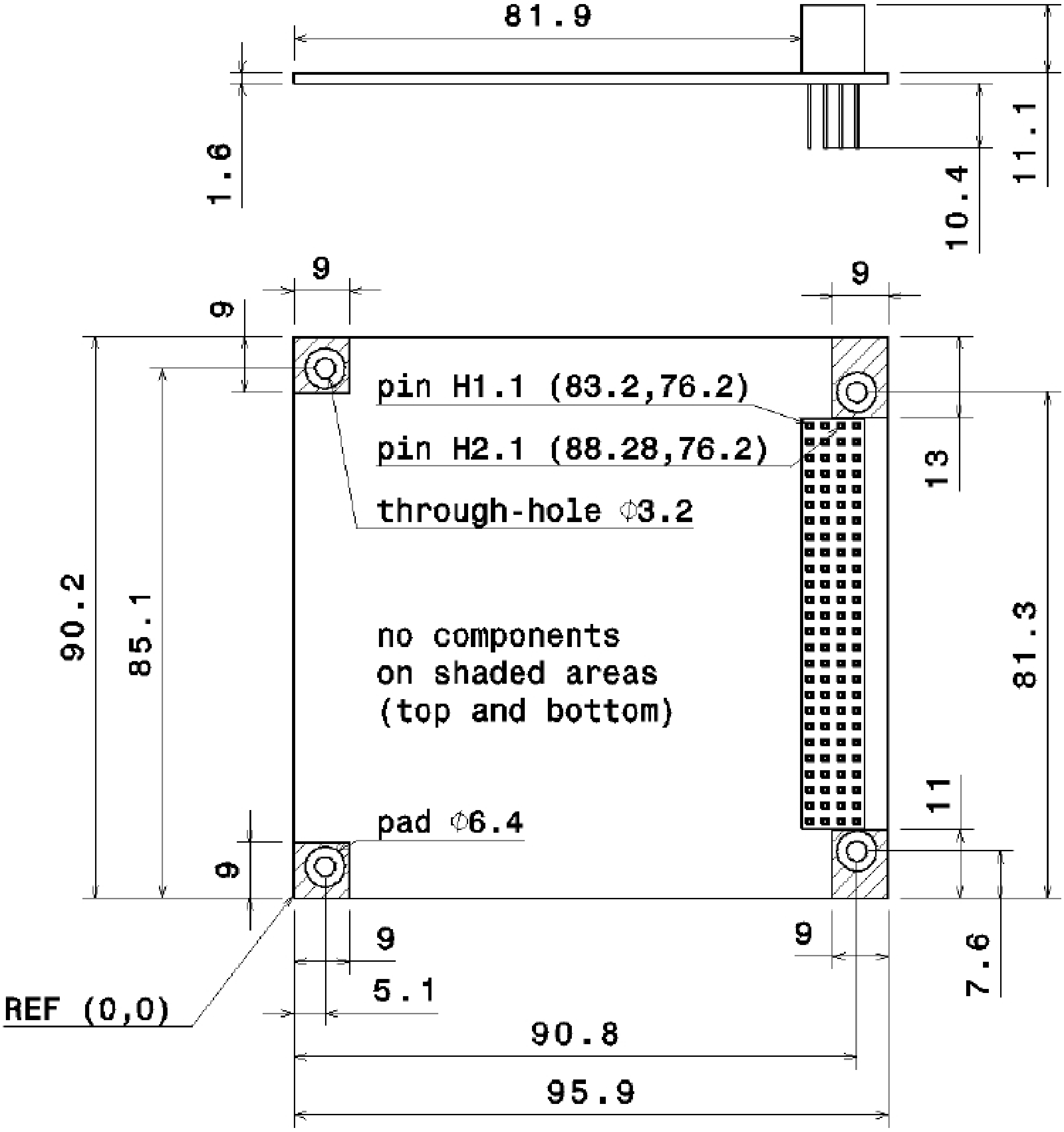


Figure E.1: PC104 Standard Size