Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE-3590 Diepenbeek
Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE-3500 Hasselt

2017•2018

Faculteit Industriële ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

FPGA-based Active Pointing Correction of Optical Instruments on Small Satellites

PROMOTOR :

Prof. dr. ir. Luc CLAESEN

PROMOTOR :

De Heer Bram VANDOREN

Tom Mladenov

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

2017•2018

Faculteit Industriële ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

FPGA-based Active Pointing Correction of Optical Instruments on Small Satellites

PROMOTOR :

Prof. dr. ir. Luc CLAESEN

PROMOTOR :

De Heer Bram VANDOREN

Tom Mladenov

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

Preface

This thesis is the result of the crossing of two trajectories, my engineering studies at the University of Hasselt and an extracurricular project at the KU Leuven Institute of Astronomy. Applied technology for space and astronomy are my biggest passions. Due to several space related projects that came across my path, a future career in the space industry became more and more certain. Therefore I did not hesitate when the question arose if I was interested in taking up a CubeSat related master's thesis at KU Leuven.

The researchers at the KU Leuven Institute of Astronomy are amongst the most passionate I have ever met and there was always a pleasant and comfortable atmosphere at the department. In general I want to thank the institute for giving me the chance to work on both this master's thesis as well as the CubeSat groundstation. This allowed me to further develop my professional skills in a highly international and space-related environment.

I would like to thank Mr. Bram Vandoren for being supportive throughout the course of the thesis and for his advice related to the electronics but also for his company during the Active Optics conference at ESA/ESTEC in the Netherlands. Also dr. ir. Tjorven Delabie provided both theoretical and logistic assistance throughout the thesis, for which I'd like to thank him. Thank you Mr. Wim de Munter for the support and also for featuring part of my research at the 4S Symposium in Italy. Furthermore, I would like to thank dr. Bart Vandenbussche for bringing me into direct contact with several industry professionals within the scope of a further career path. Thank you dr. Gert Raskin for illustrating part of my work in the CUBESPEC SPIE paper but also for the advice related to the optics. It was a true honour working with all of you. Finally I would like to thank prof. dr. ir. Luc Claesen, not only during this thesis but also with various other subjects and situations Luc provided great assistance and learned me to never stop chasing my passion, which is what I will continue to do.

I sincerely hope this work is a valuable contribution to the preliminary development of the CUBESPEC mission. I further wish dr. Gert Raskin and his instrumentation team all the best and godspeed in this unique challenge.

Tom Mladenov

Contents

Abstract

CubeSats are mini-satellites that provide the scientific community a standardised platform to perform experiments in space. CUBESPEC is a 6U CubeSat mission concept of the KU Leuven Institute of Astronomy. The payload on CUBESPEC requires a high pointing resolution in the arcsecond-range. However, the Attitude Determination and Control System (ADCS) of the satellite only provides coarse pointing in the order of a 100 arcseconds. This thesis covers the design of a pointing system within the satellite that aims for an accuracy in the arcsecond range.

Firstly, a case study has been carried out that covers previously implemented technology on other CubeSats. Secondly, a proof-of-concept optical setup has been built that represents the configuration of CUBESPEC. The two key elements in the setup are a fine steering mirror and a fine guidance sensor. A part of the control loop has been implemented in Python and OpenCV making it cross-platform. The performance of the system has been assessed by applying disturbances with various frequencies.

Fine control over the steering mirror has been realised using low-noise 16-bit digital-to-analogue converters, interfaced through an FPGA. By using a low-noise piezo amplifier the centroiding error only increased marginally. Because of the 16-bit DACs, the steering resolution is more than sufficient for this application. With the PI-controller settings, the CUBESPEC pointing requirement can be met if the main frequency component in the ADCS drift is below 0.05 Hz.

Abstract

CubeSats zijn kleine satellieten die wetenschappers in staat stellen om experimenten uit te voeren in de ruimte. CUBESPEC is een 6U CubeSat concept door het KU Leuven Instituut voor Sterrenkunde. CUBESPEC vereist een hoge richtnauwkeurigheid in de orde van enkele boogseconden. De Attitude Determination and Control System (ADCS) kan de positie van de satelliet echter tot op 100 boogseconden nauwkeurig regelen. Het doel van deze thesis is het implementeren van een platform dat een richtnauwkeurigheid van enkele boogseconden nastreeft.

Ten eerste werd er een literatuurstudie uitgevoerd naar reeds geïmplementeerde technologie. Ten tweede werd er een optisch systeem gebouwd dat de CUBESPEC opstelling voorstelt. De twee hoofdelementen in het systeem zijn een elektronisch gestuurde spiegel en een beeldsensor. Een deel van de controlelus werd geïmplementeerd in Python en OpenCV die het systeem cross-platform maken. De performantie van de controlelus werd nagegaan door het aanleggen van storingen met verschillende frequenties.

Precieze controle over de elektronische spiegel werd bekomen via 16-bit digitaal-naar-analoog convertoren, geïnterfaced via een FPGA. Het lage ruisniveau van de convertoren resulteert in een minimale toename in de fout bij de positiebepaling. Door de 16-bit nauwkeurigheid is de stuurresolutie meer dan voldoende. Met de gebruikte PI-instelparameters kan de CUBESPEC richtnauwkeurigheid gehaald worden indien de belangrijkste frequentiecomponent in de onzekerheid van het ADCS beneden de 0.05 Hz ligt.

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

1U	1-Unit
AA-MST	AeroAstro Miniature Star Tracker
ADCS	Attitude Determination and Control System
AO	Active Optics
AP	Amplified Piezoelectric
API	Application Program Interface
ASTERIA	Arcsecond Space Telescope Enabling Research in Astrophysics
BPP	Bits Per Pixel
BRITE	BRIte Target Explorer
BST	Berlin Space Technologies
CAD	Computer Aided Design
Cal Poly	California Polytechnic State University
CANX	Canadian Advanced Nanospace eXperiment
CCD	Charge Coupled Device
CDS	CubeSat Design Standard
CMOS	Complementary Metal Oxide Semiconductor
COTS	Commercially Off The Shelf
CSV	Comma Separated Values
DAC	Digital-to-Analogue Converter
DMA	Direct Memory Access
ECI	Earth-Centered Inertial
EOF	End of Frame

Abbreviations Continued

FFC	Flat-Flex Cable
FGS	Fine Guidance Sensor
FMC	FPGA Mezzanine Card
FPGA	Field Programmable Gate Array
FSM	Fine Steering Mirror
GNB	Generic Nano-satellite Bus
GPIO	General Purpose Input-Output
HPPP	High Precision Pointing Platform
ILA	Integrated Logic Analyser
IO	Input/Output
IP	Intellectual Property
ISS	International Space Station
LEO	Low-Earth Orbit
LESIA	Laboratoire d'études spatiales et d'instrumentation en astrophysique
LOS	Line-of-Sight
LPC	Low Pin Count
LSB	Least Significant Bit
LVDS	Low Voltage Differential signalling
MISO	Master In Slave Out
MOSI	Master Out Slave In
OBC	On-Board Computer
OPAMP	OPerational AMPlifier
PCB	Printed Circuit Board
PFP	Picsat Fibered Photometer
PI	Proportional-Integral
PI	Physik Instrumente
PLL	Phase Locked Loop
PMOD	Peripheral MODule
PPOD	Poly Picosatellite Orbital Deployer
PSD	Power Spectrum Density
PSF	Point Spread Function
RMS	Root Mean Square
ROI	Region Of Interest

Abbreviations Continued

SFL	Space Flight Lab
SMF	Single Mode Fiber
SoC	System-on-Chip
SOF	Start of Frame
SPAD	Single-Photon Avalanche Diode
SPI	Serial Peripheral Interface
SS	Slave Select
SSDL	Space Systems Development Laboratory
UART	Universal Asynchronous Receive and Transmit
UHF	Ultra-High Frequency
UI	User Interface
UTIAS	University of Toronto Institute of Aerospace Studies
VI	Virtual Instrument
XiAPI	Ximea Application Program Interface

Symbols

$R^x(\phi)$	Rotation matrix describing the roll
$R^y(\theta)$	Rotation matrix describing the pitch
$R^z(\psi)$	Rotation matrix describing the yaw
ϕ	roll
θ	pitch
ψ	yaw
x, y, z_{SF}	Coordinate system of the fixed spacecraft
x, y, z_{ECI}	Coordinate system of the reference
$e_{x\theta}, e_{x\phi}$	Error on target plane in function of pitch and roll
e_1	Total pointing error on target plane
s	Linear separation of image on detector
u	Angular separation of target
$\frac{u}{s}$	Platescale of the telescope
$s_{actuator}$	Stroke range of piezoelectric actuator
θ_{mirror}	Tilt range of the mirror
ϕ_{mirror}	Beam deviation of mirror
d_{sensor}	Steering range on the sensor
R_{FSM}	Steering resolution
DAC_0, DAC_1	16-bit value of the actuators
M	Transformation matrix linking centroid domain to actuator domain
(x_e, y_e)	Estimated centroid of a target star
(x_t, y_t)	True centroid of a target star
I_{ij}	Intensity of a pixel at detector position (i, j)
x_{ij}, y_{ij}	x- and y-position of a pixel at detector position (i, j)
e_k	Total centroid error on frame k
x_e^k, y_e^k	Calculated x- and y-centroid on frame k
PSD_{sys}	PSD of the system to be measured
$PSD_{sys+baseline}$	PSD of the measured signal
$PSD_{baseline}$	PSD of the baseline recording equipment
K_p	K-factor of the Proportional action
T_i	T-factor of the integral action

Symbols Continued

A_{dist}	Amplitude of the disturbance
f_{dist}	Frequency of the disturbance
A_{error}	Amplitude of the control-loop error
a_{loop}	Disturbance suppression of the loop
t_{req}	Time percentage the pointing requirement is met

Chapter 1

Introduction

The first chapter gives an introduction into the CubeSat concept and its importance to the scientific community. Furthermore the standard will be illustrated and how this affects the design of systems that interact with CubeSats.

1.1 What is a CubeSat?

In 1999 a collaboration between the California Polytechnic State University (Cal Poly) and the Space Systems Development Laboratory (SSDL) at the University of Stanford led to the introduction of a universal satellite standard [2]. The introduction of this standard suddenly made space significantly more accessible. Currently, CubeSat development is a hot topic in the academic, private and corporate sectors. CubeSats are miniature satellites that allow for on-orbit testing of miniaturised payloads with a wide spectrum of applications and with the benefit of small launch costs due to the low mass of the satellite. Unlike conventional large and expensive satellites, most CubeSats reside in the 1 to 8 *kg* weight class. Due to the limited size, the payload or scientific part of the satellite is often concentrated on a single task. Figure 1.1 illustrates an artist's impression of several CubeSats in orbit.

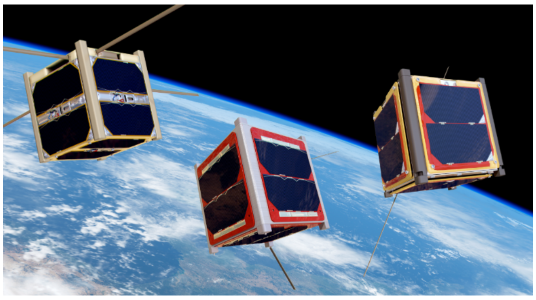


Figure 1.1: Artist's impression of CubeSats orbiting Earth [15]

CubeSat are also extensively used by universities to give students hands-on experience in the form a small scale space-project in collaboration with real industry partners. Nevertheless, the commercialisation of cubesats has also began to take place, primarily in the earth observation sector. Figure 1.2 illustrates the amount of cubesats launched between the year 2000 and 2015, categorised by user. Figure 1.3 depicts a photograph of the development team behind PicSat, a CubeSat dedicated to astronomy launched on the 12th of January 2018 [6].

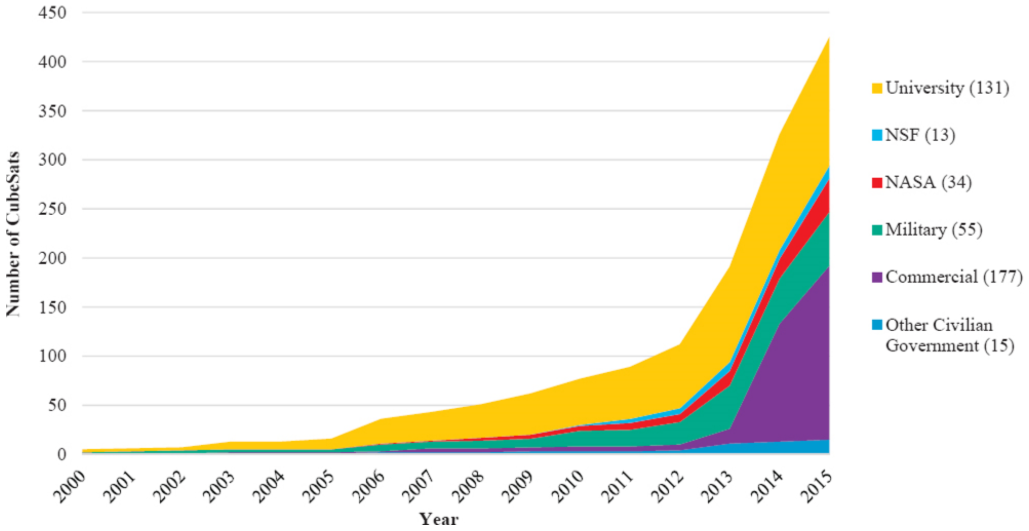


Figure 1.2: Number of cubesats launched between 2000 and 2015 [25]

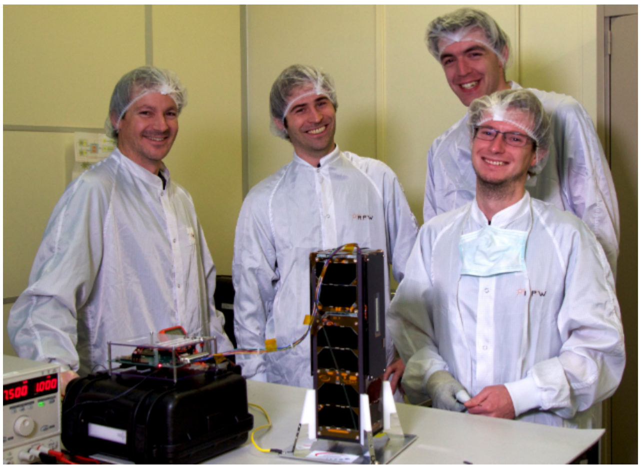


Figure 1.3: Team picture of the PicSat mission [36]

1.2 The CubeSat Standard

In order to provide the possibility of testing larger payloads on CubeSats, several CubeSat Design Specification (CDS) standards have been introduced [22]. A series of different CubeSat standards are displayed in figure 1.4. Having a standard allows for CubeSats to be deployed from the same deployer.

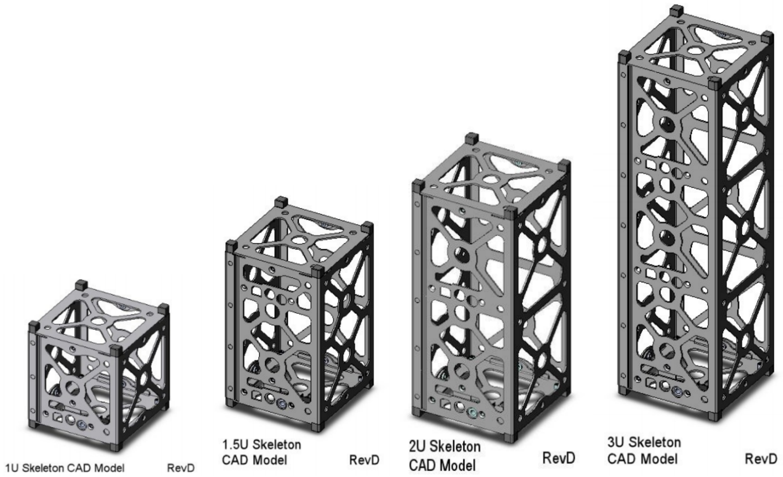


Figure 1.4: Computer-aided Design (CAD) skeleton models of CubeSat size standards ranging from 1U (left) to 3U (right) [22]

A 1U CubeSat is the smallest standard. One Unit corresponds to a 10 cm cube with a maximum mass of 1.33 *kg* as illustrated by figure 1.5 [2]. All standards require the CubeSat design to incorporate rails on the corners. The rails facilitate the ejection of the CubeSat from an orbital deployer. The orbital deployer is further discussed in 1.3.

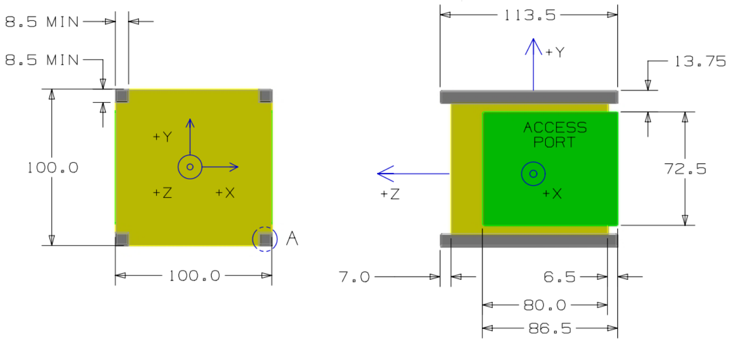


Figure 1.5: Dimensions of a 1U CubeSat [2]

The full technical drawing of a 1U CubeSat can be found in appendix A.

1.3 CubeSat Orbital Deployer

Orbital deployers are used to eject a CubeSat into orbit. One of the first deployers is the Poly Picosatellite Orbital Deployer (P-POD) developed and introduced by Cal Poly. A CAD drawing of the P-POD is illustrated in Figure 1.6

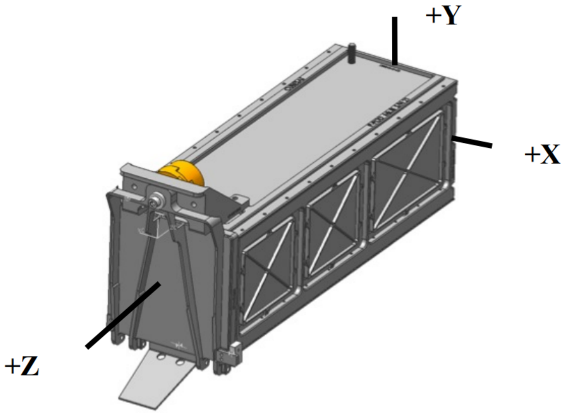
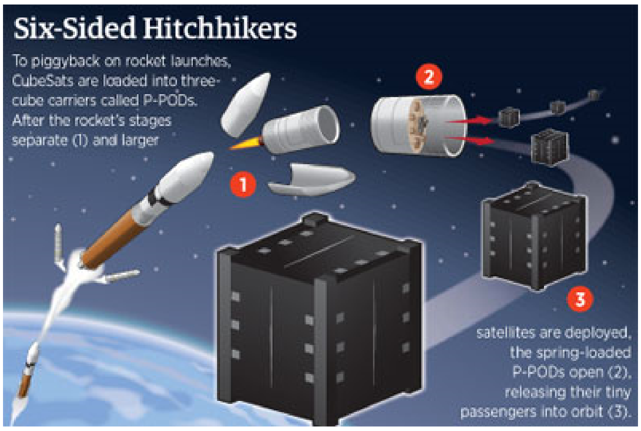


Figure 1.6: P-POD Mk. III Rev. E [21]

The P-POD Mk. III Rev. E User Guide describes the P-POD as follows:

> *The P-POD is capable of carrying three standard 1U CubeSats and serves as the interface between the CubeSats and Launch Vehicle (LV). The P-POD is a rectangular box, made out of alodined aluminum, with a door and a spring mechanism. Once the release mechanism of the P-POD door is actuated, a set of torsion springs at the door hinge swing the door open, and the CubeSats are deployed by the main spring, gliding on the P-POD's rails* [21, p. 9].

The full technical drawing of the P-POD can be found in appendix B. The deployers are interfaced with the launch vehicle. Typically several CubeSats are a secondary or tertiary payload during a mission in which the main payload is a much larger satellite. This configuration is often called a ’hitchhike’ or a ’piggyback’ ride to space. To prevent collision between the satellites, the main satellite is ejected first. Afterwards the smaller CubeSats are deployed into their corresponding orbits. This mission concept is illustrated in figure 1.7.



Six-Sided Hitchhikers

To piggyback on rocket launches, CubeSats are loaded into three-cube carriers called P-PODs. After the rocket's stages separate (1) and larger satellites are deployed, the spring-loaded P-PODs open (2), releasing their tiny passengers into orbit (3).

Figure 1.7: Launch sequence of deploying CubeSats during a larger commercial mission [16]

CubeSats are also regularly deployed from the International Space Station (ISS). In that case the CubeSats are brought to the space station using the commercial resupply missions that provide the astronauts with supplies, food and experiments. Several CubeSats are then placed into the deployer and deployed into orbit from the outside of the space station. The deployment of two CubeSats from the ISS is illustrated in figure 1.8.

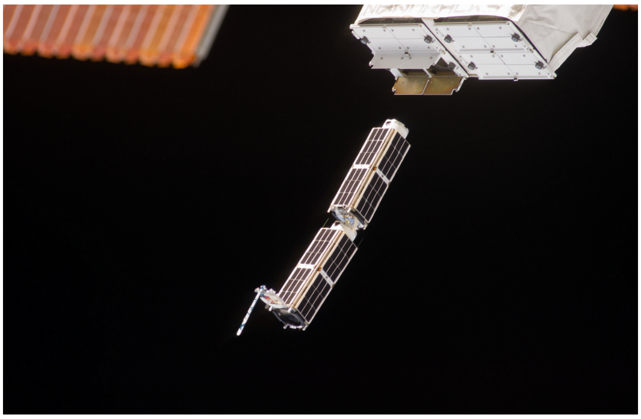


Figure 1.8: Two CubeSats being deployed from the ISS's NanoRacks deployer [23]

1.4 Attitude of a Satellite

Once deployed into orbit, a CubeSat's mission can begin. For some payloads, especially optical, precise attitude control is required. The attitude of a satellite refers to the way the satellite is oriented in space compared to a fixed reference. Both a fixed reference coordinate system and the coordinate system of the fixed body, in this case the satellite, are defined. Figure 1.9 illustrates an example coordinate referencing scene. In this case the earth-centred inertial (ECI) reference is used. This reference is graphically illustrated in figure 1.10 (a).

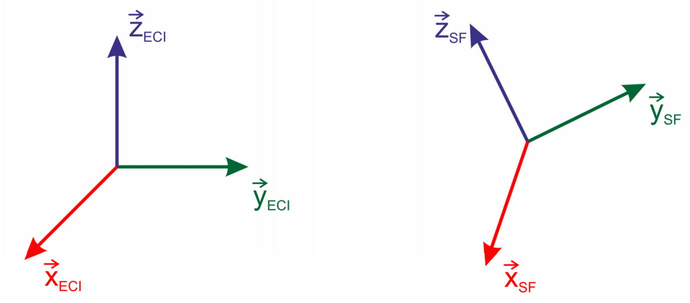


Figure 1.9: The reference coordinate system (left) and the coordinate system of the satellite (right)

A possible way of describing the attitude of the satellite relative to the reference coordinate system is by using Euler angles. Euler angles consist of three angles: roll (ϕ), pitch (θ) and yaw (ψ). The attitude of the satellite is the result of three consecutive rotation matrices that are given by [4]:

$$R^x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & sin(\phi) \\ 0 & -sin(\phi) & cos(\phi) \end{pmatrix}, \tag{1.1}$$

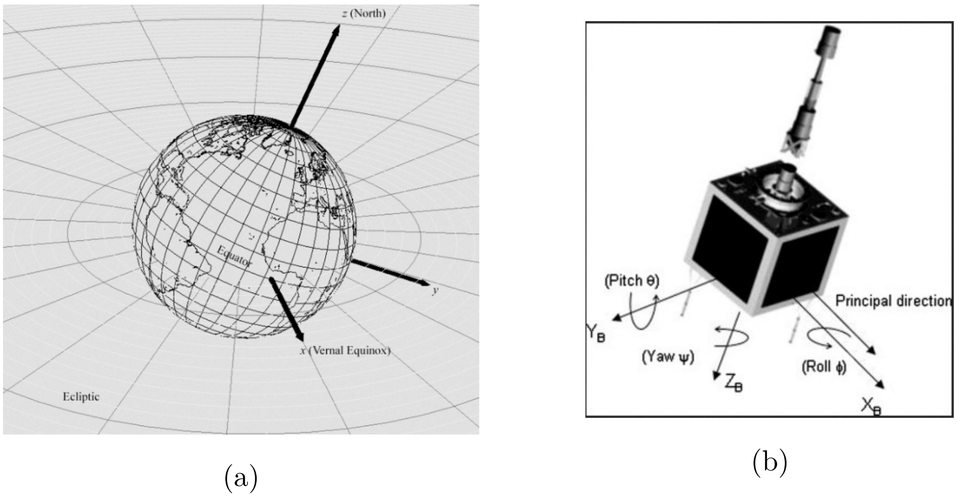
$$R^y(\theta) = \begin{pmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{pmatrix}, \tag{1.2}$$

$$R^z(\psi) = \begin{pmatrix} cos(\psi) & sin(\psi) & 0 \\ -sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{1.3}$$

The attitude of the satellite can therefore be described as:

$$\begin{pmatrix} x_{SF} \\ y_{SF} \\ z_{SF} \end{pmatrix} = R^x(\phi)R^y(\theta)R^z(\psi) \begin{pmatrix} x_{ECI} \\ y_{ECI} \\ z_{ECI} \end{pmatrix}. \tag{1.4}$$

Where (x_{SF}, y_{SF}, z_{SF}) represents the fixed spacecraft system and $(x_{ECI}, y_{ECI}, z_{ECI})$ the reference coordinate system. Figure 1.10 (b) illustrates the attitude determination of a satellite using Euler angles.



(a)

(b)

Figure 1.10: Earth-Centered Intertial (ECI) reference (a) and attitude of a satellite represented by Euler angles (b) [4]

There are other ways of attitude representation that are more computationally efficient, however these are not discussed in this thesis.

Chapter 2

The CUBESPEC Concept

This chapter introduces the problem statement and proposed solution as well as its connection to the CUBESPEC mission concept of the KU Leuven Institute of Astronomy.

2.1 Science Goal

The primary goal of the CUBESPEC mission is to provide an on-orbit turnkey spectroscopy solution for the astronomical community. More specifically, the optical payload will provide the possibility of carrying out the following scientific astronomical studies by means of spectral and intensity monitoring [20]:

- Study the internal structure of massive stars
- Study the properties of low-intermediate mass stars
- Detect exoplanets and study the properties of the host star

Exoplanets can be detected by using the transit photometry method. In this method, the flux output of a host star is closely observed. Depending on the size of the star and the exoplanet, the observed flux from the star will decrease when a transit occurs. Figure 2.1 illustrates this principle.

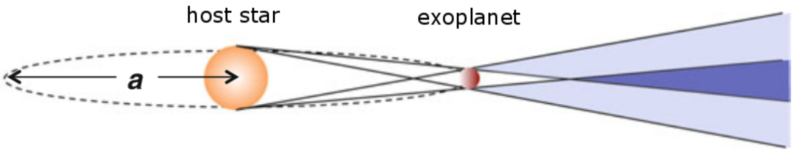


Figure 2.1: Transit of an exoplanet in front of a star [1]

Two events are possible when monitoring the flux output from a star with an orbiting planet. The transit phase causes the largest drop in flux output. The second possible event is occultation in which the planet is located behind the star, resulting in a small drop in the flux since the planet is not reflecting part of the star's flux output. Figure 2.2 depicts these phases.

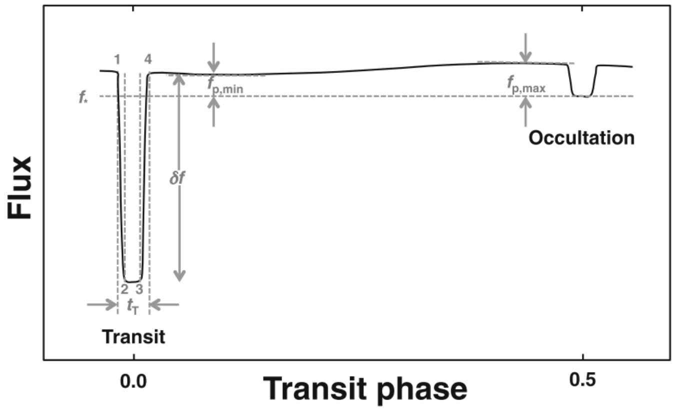


Figure 2.2: Possible phases and influence on flux [1]

In reality the contrast that is observed during transit and occultation phases is relatively small. Figure 2.3 illustrates an actual set of photometry measurements from HD 189733. In this set of data, the occultation phase causes a 0,1 % drop in intensity or 1000 ppm. On the other hand, the transit phase causes a more easily detectable 2,5 % decrease in flux or 25 000 ppm.

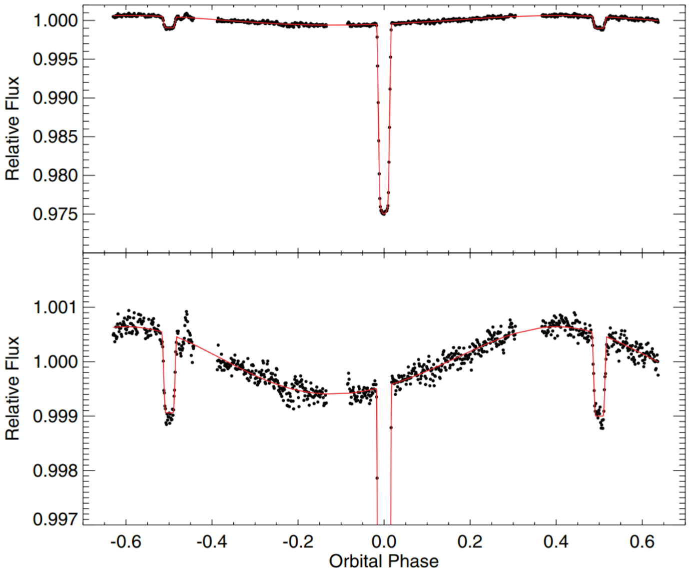


Figure 2.3: Transit measurements from extrasolar planet HD 189733b [19]

2.2 Configuration

In order to maximise the light gathered by the optical payload, the CUBESPEC satellite is based on a 6U CubeSat frame. The optical payload takes up approximately 4U of the total body volume while the electronics and instrumentation take up the remaining 2U. The telescope consists out of two mirrors M1 and M2 that form an offset Cassegrain telescope with a focal length of 1600 mm. Transceivers and antennas are used for communication with the ground. An on-board computer (OBC) provides coordination between all the subsystems. Figure 2.4 illustrates the CUBESPEC configuration.

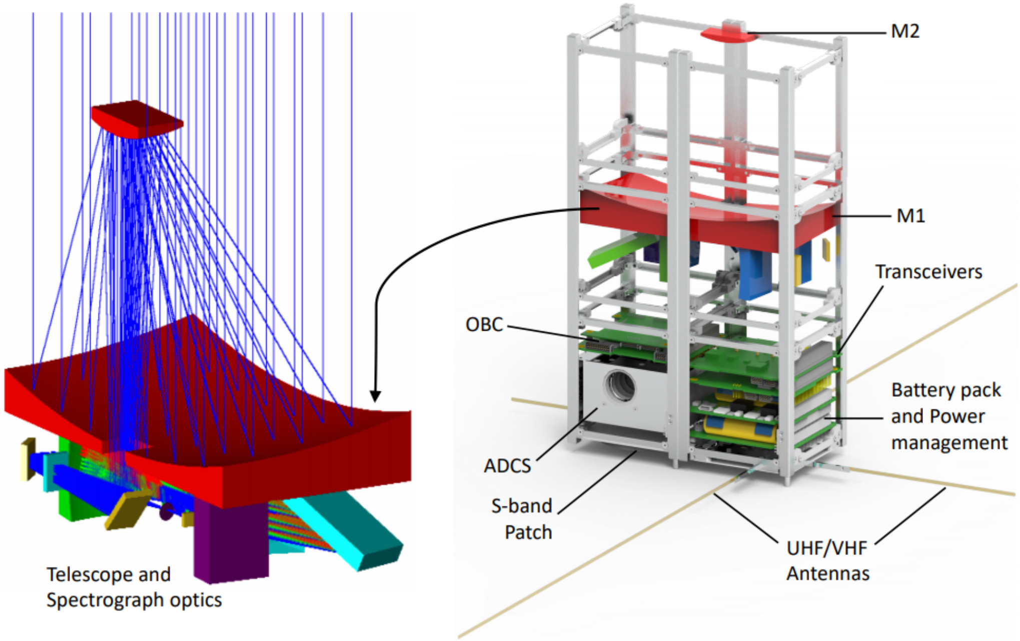

M2
M1
Transceivers
OBC
Battery pack and Power management
ADCS
S-band Patch
UHF/VHF Antennas
Telescope and Spectrograph optics

Figure 2.4: Detailed view of the optics (left) and overview of the subsystems in the 6U cubesat structure (right) [28]

This thesis mainly focuses on the pointing aspects of the satellite and therefore only the ADCS and pointing platform will be further discussed. Figure 2.5 illustrates a more detailed diagram of the optics that are mounted on the back of the primary mirror (M1). The secondary mirror (M2) reflects the light towards the Beam Steering Mirror. The Steering Mirror passes the light towards a Beam Splitter where approximately 5% of the total light is passed to the Fine Guidance Sensor (FGS) via a fixed mirror in the corner of the satellite. The fine guidance sensor and the beam steering mirror are the key components to the pointing platform of CUBESPEC. The spectrograph requires the star to be positioned over a slit with a corresponding angular width of 3.6 arcseconds. During observations, which can range from 2 to 15 minutes, the target star needs to be positioned over the slit [28].

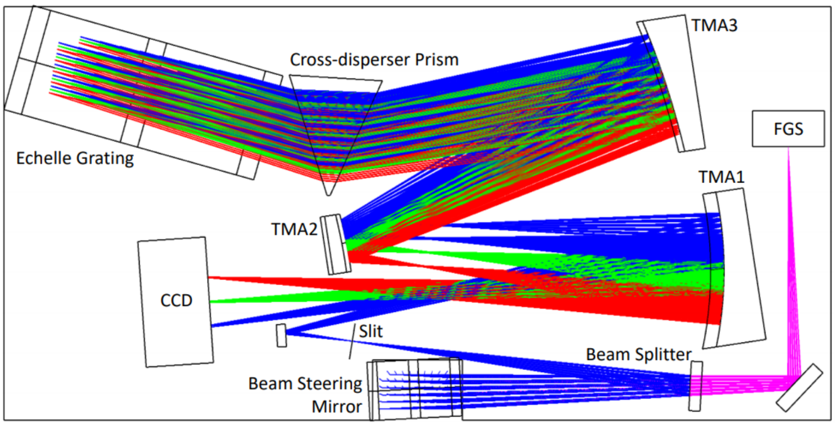


Figure 2.5: Optical configuration of CUBESPEC [28]

2.3 Problem Statement

The scientific instruments on CUBESPEC require accurate pointing towards the target. A first measure of pointing or attitude control is the attitude determination and control system (ADCS) unit which is in development by KU Leuven. Determination implies that the system uses sensors to measure the current attitude of the satellite. Control indicates that the ADCS uses actuators to correct the attitude of the satellite to the desired value. Figure 2.6 illustrates the KU Leuven ADCS.

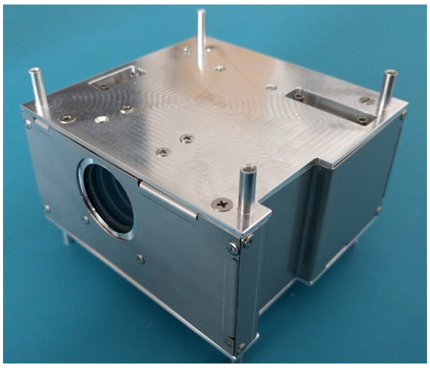


Figure 2.6: The 0.5U ADCS developed by KU Leuven

The KU Leuven ADCS relies on several control loops to work. Sensors measure the attitude of the satellite and a controller together with a desired position are used to correct the attitude of the CubeSat to a desired value. The most important and accurate sensor for attitude determination is a star tracker. The star tracker is a miniature scientific camera that images stellar constellations and by using an internal catalogue, can determine the attitude of the satellite with high stability and accuracy. The viewing port for the star tracker is a circular window on the side of the ADCS including a baffle protecting the sensor from stray light sources. The actuators are reaction wheels that are capable of storing momentum and precisely control the angular rate of the satellite. When the reaction wheels reach maximum speed they can not offer any more momentum storage. Therefore, extra magnetorquers are used for desaturisation. The control-loop diagram of the KU Leuven ADCS is illustrated in figure 2.7 [7].

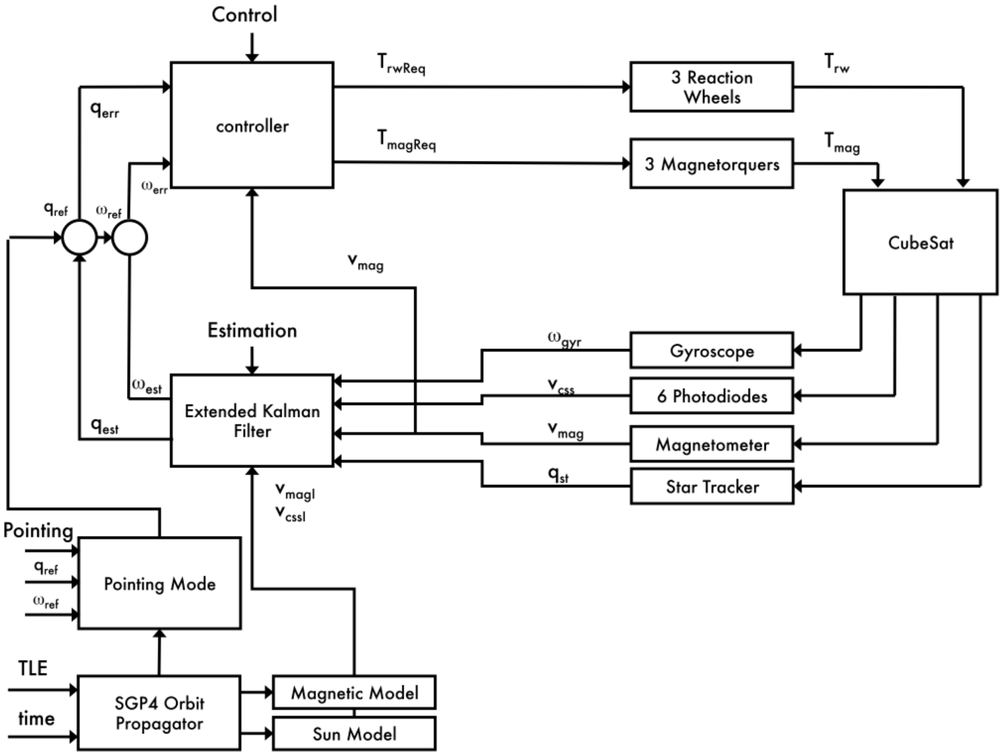


Figure 2.7: Control-loop diagram of the KU Leuven ADCS [7]

The sensors consist of gyroscopes that measure the angular rate of the satellite while six photodiodes act together as a sunsensor. Additionally a magnetometer is used as an extra sensor that together with a magnetic model can provide coarse attitude estimation. The gyroscope, sunsensor and magnetometer are less accurate when compared to the star tracker, however they can provide data at a much higher samplerate.

The ADCS is capable of determining the attitude of the spacecraft with an accuracy of down to 10 arcseconds and controlling the attitude to within 100 arcseconds. However a pointing accuracy of atleast 3.6 arcseconds is needed to allow the star to be kept over the slit in figure 2.5 and in turn, accurate measurements to be made [28].

Figure 2.8 illustrates a general satellite pointing scene. The attitude uncertainty of the ADCS translates into rotational errors around the x-, y- and z-axis. The rotational errors around the x- and y-axis cause a deviation on the target plane indicated by e_x and e_y.

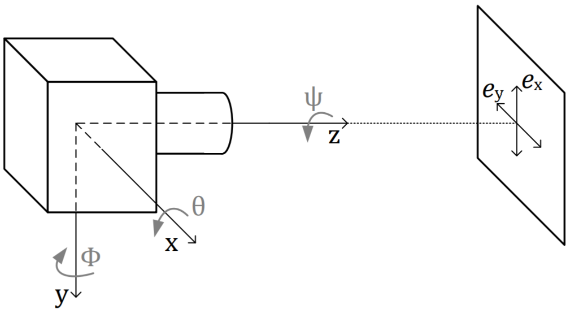


Figure 2.8: Satellite pointing scene [12]

The total Line Of Sight (LOS) or half-cone error e_1 on the target plane is given by:

$$e_1 = \begin{bmatrix} e_{x\theta} \\ e_{y\phi} \end{bmatrix} \quad \text{with} \quad |e_1| = \sqrt{e_{x\theta}^2 + e_{y\phi}^2} \tag{2.1}$$

Due to the lack of accurate pointing, the target star will drift on the image sensor. The amount of drift on the detector is dependent on the optical configuration and the plate scale of the telescope. The plate scale links the angular separation to a linear separation on the detector and is expressed in arcseconds/pixel. For the same attitude uncertainty, the drift of a target star will be less apparent on a large plate scale as opposed to a small plate scale detector. This is further discussed in the next chapter.

2.4 Solution

Since the attitude control accuracy of the current ADCS does not satisfy the CUBESPEC pointing requirement a secondary system is needed. This secondary system is an internal high precision pointing platform that uses a fine guidance sensor to measure the deviation of the target and to correct the pointing using a fine steering mirror as mentioned before. Essentially the pointing accuracy and stability is now determined by two control loops that complement each other.

Figure 2.9 illustrates the double control loop diagram. The ADCS loop (depicted in blue) provides coarse pointing to within 100 arcseconds. The last residual errors are corrected for by the internal High Precision Pointing Platform (HPPP), depicted in orange. An important difference between the two control loops is the fact that the high-frequency inner control loop does not control the attitude of the satellite, it merely controls an internal element in the optical train of the payload. The outer ADCS loop on the other hand controls the actual attitude of the entire satellite.

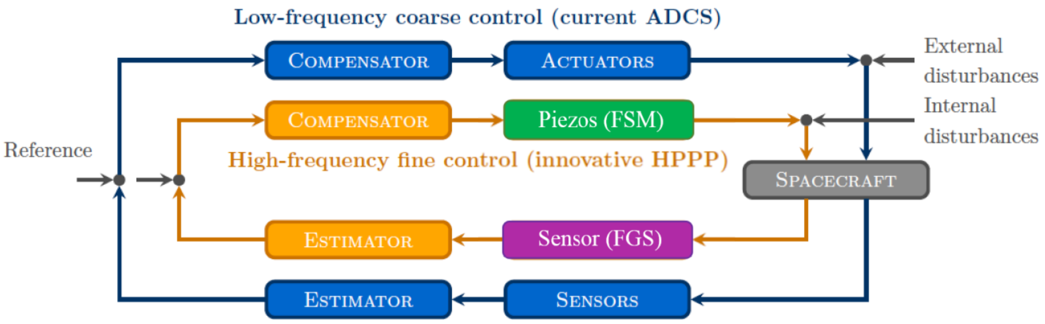


Figure 2.9: Satellite control loop including fine pointing system

The key components of the corrective system are the fine steering mirror (FSM) and the fine guidance sensor (FGS). These are indicated in green and purple respectively on figure 2.9. The key elements are also depicted in the corresponding colours on the 3D rendering of the telescope on CUBESPEC illustrated by figure 2.10.

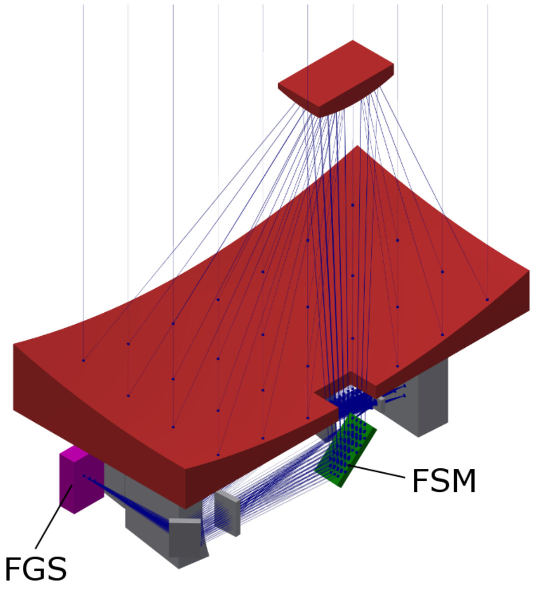


Figure 2.10: CUBESPEC fine pointing configuration consisting of the steering mirror (green) and the guidance sensor (purple)

In order to get a realistic setup that represents the configuration of CUBESPEC, the payload configuration needs to be replicated. This thesis is only concerned with the pointing platform of CUBESPEC, therefore the optics of the payload are not discussed. In the current configuration, a beam steering mirror (green) and guidance sensor (purple) form the steering platform. An optical system has been built that resembles the CUBESPEC configuration. The system consists of two major subsystems, the electro-optics and the electromechanics. The electro-optics are concerned with imaging aspects of the setup. The driving of the steering mirror is performed by the electromechanics subsystem. Figure 2.11 gives a graphical representation of the setup.

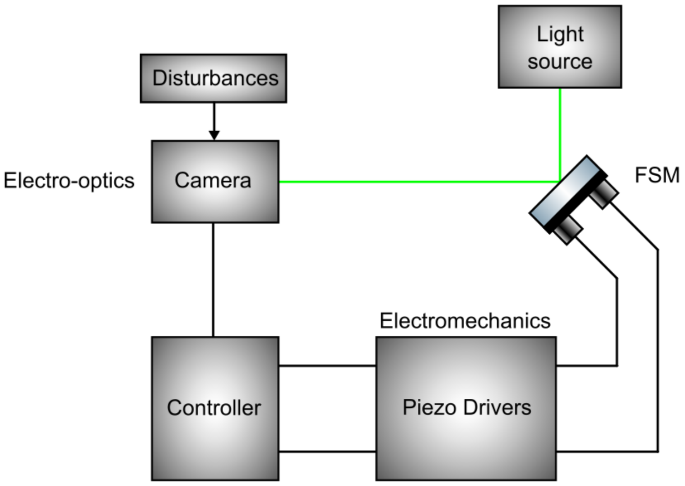


Figure 2.11: System overview

A light source is used to act as an artificial star. The light is reflected towards the camera by the fine steering mirror. The camera in turn detects the position of the point-source and compares this to the setpoint. In case of an error, the position of the steering mirror will be corrected. During this process, disturbances are applied to the camera to simulate the inaccuracy of the ADCS. The system performance is assessed by the ability to keep the star centred on a setpoint for a range of different disturbances applied to the guidance sensor.

Chapter 3

Case Study

In this chapter, three case studies on similar missions are carried out covering aspects and results that are relevant to this thesis. The discussed missions require high photometric resolution and attitude control and are therefore a good basis to reflect upon. Only the optical and attitude aspects of these missions are discussed since they are the most relevant to this thesis. At the end of the chapter a conclusion is formulated.

3.1 BRITE

The BRIght Target Explorer (BRITE) CubeSats are part of a constellation of five nanosatellites in Low Earth Orbit (LEO). The nanosatellites are a result of the development under the Canadian Advanced Nanospace eXperiment (CANX) program which is run by the Space Flight Lab (SFL) of the University of Toronto Institute of Aerospace Studies (UTIAS) [37].

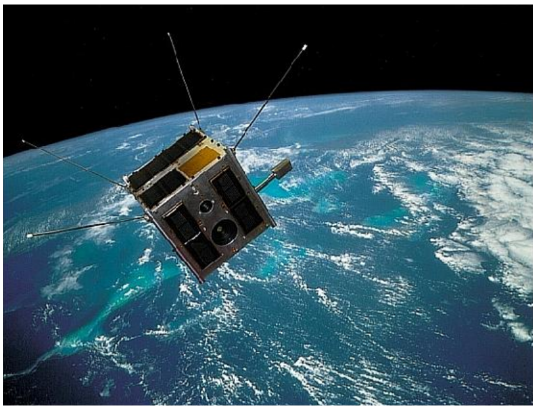


Figure 3.1: Artist's impression of a BRITE satellite in orbit around Earth [13]

3.1.1 Spacecraft Configuration

In contrary to the CubeSat design standards mentioned in 1.2, the BRITE nanosatellites are based on the Generic Nano-satellite Bus (GNB) developed by UTIAS-SFL which is defined as a cube measuring 20 cm and a total mass less than 8 kg [37]. The BRITE spacecraft configuration is illustrated in figure 3.2.

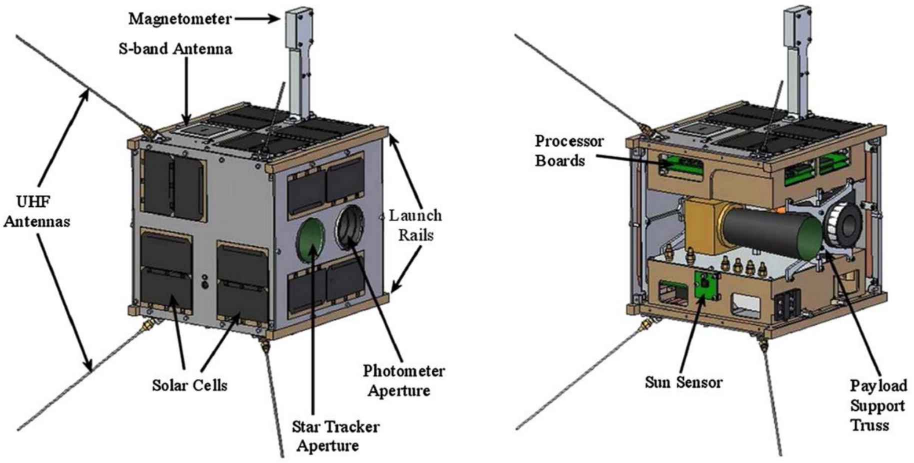


Figure 3.2: Configuration of the BRITE CubeSat [8]

The main payload of BRITE consists of a fixed focal length telescope. Next to the telescope a star tracker is mounted to provide attitude determination. As can be seen in the right section of figure 3.2, the star tracker features a long black tube called a baffle reducing the impact of stray light sources. An S-band and ultra-high frequency (UHF) antenna provide communications with the ground. To run the processor boards and the payload, solar cells are mounted on the outside of the satellite.

3.1.2 Optical Assembly

The main optical instrument of the BRITE CubeSat is a refractive telescope consisting of 3 modules. The first module in the optical train is the baffle containing the optical filter. Secondly the optical cell contains the lenses and spacers and finally the header tray contains the Charge Coupled Device (CCD) imaging sensor, focusing mechanism and thermal control electronics [30]. A cutaway model of the assembly is illustrated in figure 3.3.

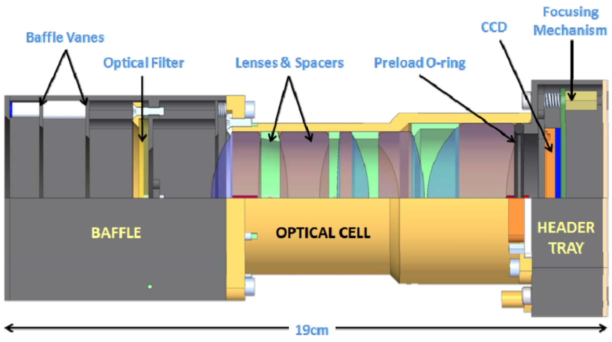


Figure 3.3: Optical Assembly of the BRITE Telescope [30]

.

Table 3.1 summarises the parameters of the telescope and the detector.

Table 3.1: BRITE: Telescope and Detector Parameters [13]

Telescope		Detector	
Aperture	30 mm	Type	Kodak KAI-11002 CCD
Focal length	70 mm	Format	4008 x 2672
F-number	2.33	Pixel pitch	9 μm
FOV	24 deg.	Plate scale	26 arcsec/pix
Lens elements	5		

The plate scale or angular resolution of the detector can be calculated as follows:

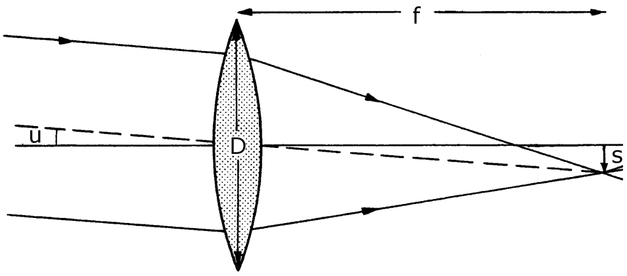


Figure 3.4: Calculation of the plate scale of the telescope [29]

.

The plate scale $\frac{u}{s}$ can be calculated with [1]:

$$\frac{u}{s} = 3600 \times \frac{180}{\pi} \frac{1}{f} \frac{arcsec}{mm}. \tag{3.1}$$

Given that the focal length f of the telescope is equal to 70 mm this becomes:

$$\frac{u}{s} = 3600 \times \frac{180}{\pi} \frac{1}{70mm} = 2946.64 \frac{arcsec}{mm}, \tag{3.2}$$

considering a pixel size of 9 μm this translates into:

$$9 \times 10^{-3} \frac{mm}{pix} \times 2946.64 \frac{arcsec}{mm} = 26.5 \frac{arcsec}{pix}. \tag{3.3}$$

3.1.3 Attitude Control

The BRITE satellite bus uses 3 orthogonally placed Sinclair-SFL 30 mNms reaction wheels. With this configuration, the attitude of the satellite can be controlled with an accuracy better than 1.0 degrees of the target and a stability better than 60 arcseconds. The first generation of BRITE satellites feature a ComTech/AeroAstro MST (Miniature Star Tracker). The star tracker is the most accurate attitude determination sensor of the satellite. The AA-MST however has a limited sampling rate of 0.5 Hz due to the long exposure time that is needed, therefore limiting the control loop bandwidth [13]. The reaction wheels and star tracker are illustrated in figure 3.5 and 3.6.

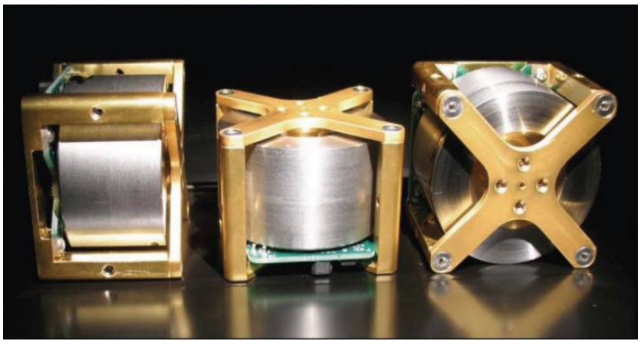


Figure 3.5: Three Sinclair-SFL 30 mNms reaction wheels [39]

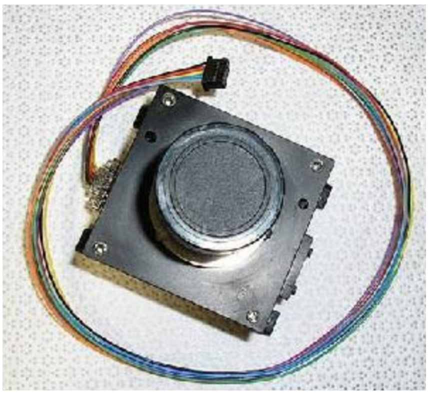
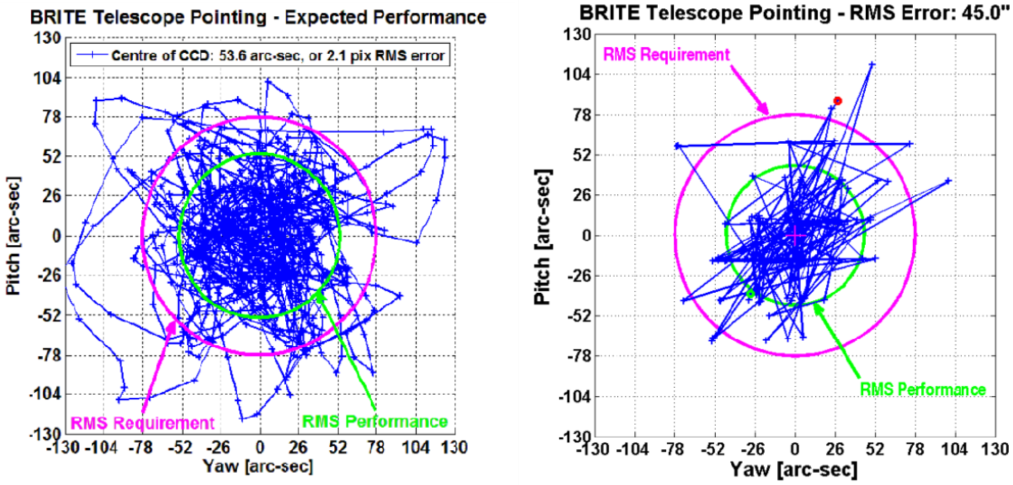


Figure 3.6: The ComTech/AeroAstro-MST (Miniature Star Tracker) [13]

3.1.4 Pointing Performance

Since the telescope is fixed within the satellite body, the pointing accuracy and stability are directly related to the accuracy and stability with which the attitude of the satellite can be determined and controlled. Figure 3.7 depicts the expected (a) and on-orbit performance (b) of the BRITE Telescope using the Sinclair-SFL 30 reaction wheels and the AA-MST star tracker.



(a)

(b)

Figure 3.7: Expected pointing performance of BRITE (a) and on-orbit pointing performance (b) [30]

An initial RMS pointing requirement of 78 arcseconds (3 pixels) was set (depicted in purple). The displayed data is the result of a 15 min measurement in which the centroid is calculated of a stellar point spread function (PSF) resulting in sub-pixel accuracy [30]. A simulation indicates that the expected residual RMS error is 53.6 arcseconds or 2.1 pixels and therefore meeting the requirement. Actual on-orbit results confirm this, indication a residual RMS pointing error of 45 arcseconds (1.73 pixels).

Later models are equipped with a different star tracker. The ST-16 star tracker supports higher samplerates of 2 Hz as opposed to the 0.5 Hz of the old AA-MST star tracker. The higher samplerate leads to a significant increase in control bandwidth evident by the on-orbit results of BRITE-Toronto indicating an RMS pointing error of 11.8 arcseconds or 0.45 pixels [13]. Figure 3.8 illustrates the enhanced pointing performance. The new ST-16 star tracker is illustrated in figure 3.9.

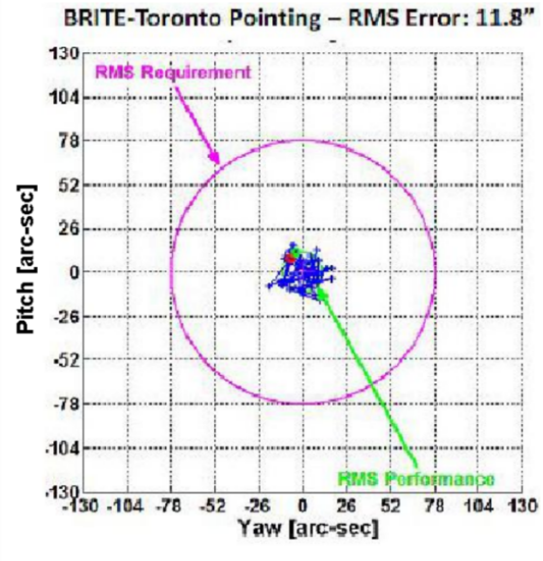


Figure 3.8: On-orbit pointing performance of BRITE after switching to an ST-16 star tracker [13]

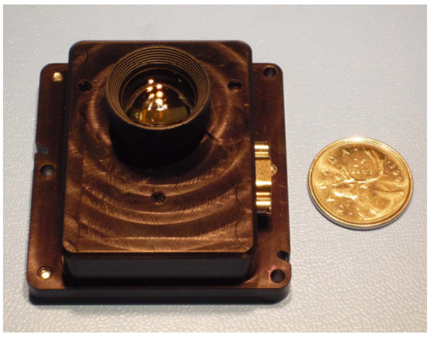


Figure 3.9: ST-16 miniature star tracker [11]

3.2 ExoplanetSat

A similar mission in which an optical payload is used to study stars and exoplanets is the ExoplanetSat mission, later called ASTERIA (Arcsecond Space Telescope Enabling Research in Astrophysics). The optical configuration on ExoplanetSat is fundamentally different to that of CUBESPEC, but nevertheless both missions require a high pointing precision. As opposed to BRITE, ExoplanetSat uses an active correction system to achieve high-resolution pointing. Figure 3.10 depicts a CAD-model of the CubeSat with deployed solar arrays.

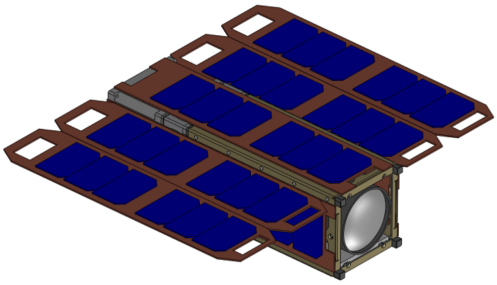


Figure 3.10: Rendering of ExoplanetSat [34]

3.2.1 Spacecraft Configuration

ExoplanetSat is based on a 3U cubesat and features three major subsystems. Firstly the optical payload which is the core of the scientific mission. Secondly the ADCS module located towards the back of the satellite and finally the avionics in the middle. Figure 3.11 illustrates the composition of the CubeSat.

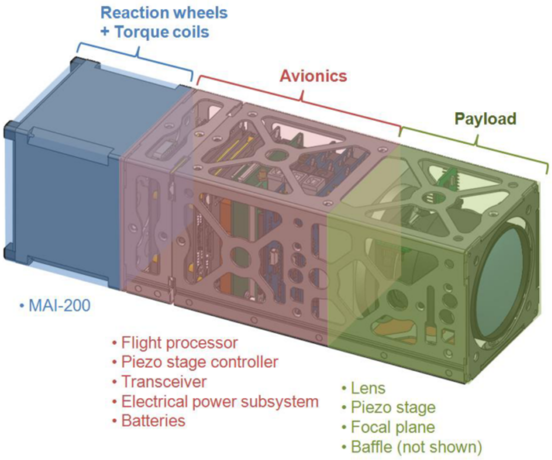


Figure 3.11: Configuration of ExoplanetSat [33]

The avionics section of ExoplanetSat houses the batteries, transceiver and flight processor. These form the basic systems that every satellite relies upon. It also houses the piezo stage controller that interfaces with the positioning stage on which the lens is mounted [33].

3.2.2 Optical Assembly

The telescope on ExoplanetSat consists of a 60 mm diameter fixed focal length lens attached to a piezoelectric stage and a focal plane. The piezoelectric stage offers active correction with an actuating range of ± 50 μm in the X- and Y-direction. The positioning stage essentially translates the focal plane assembly relative to the spacecraft-fixed lens.

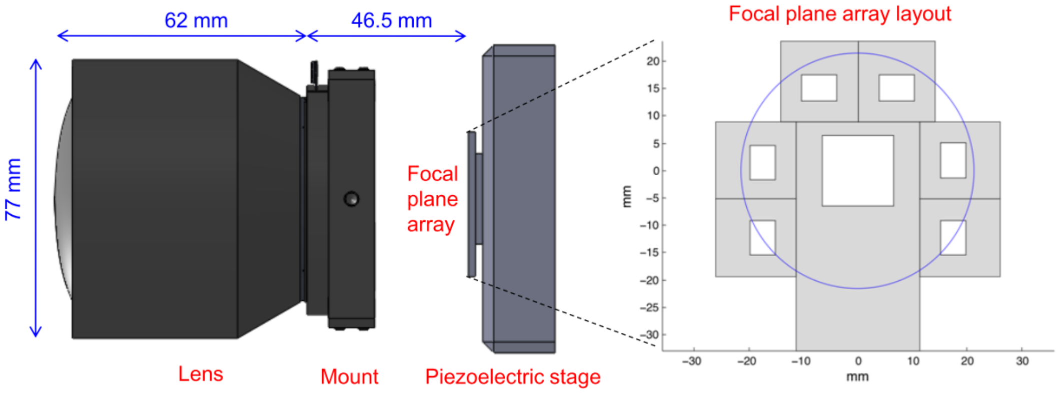


Figure 3.12: ExoplanetSat: Optical Assembly (left) and focal plane imaging configuration (right) [18]

The main imager is centred in the field of view of the optics. Surrounding the main detector, several CMOS sensors have been placed that act as star trackers and fine guidance sensors. To provide fine control, the pixel pitch of the guide sensors is smaller than that of the main detector.

Table 3.2 summarises the parameters of the telescope and the main detector.

Table 3.2: ExoplanetSat: Telescope and main Detector Parameters [33]

Telescope		Detector	
Aperture	60 mm	Format	1024 x 1024
Focal length	85 mm	Pixel pitch	18 μm
F-number	1.4	Plate scale	43.7 arcsec/pix
FOV	17,3 deg.		
Lens elements	6		

3.2.3 Attitude Control

ExoplanetSat uses a MAI-200 which is a commercially of the shelf (COTS) ADCS unit built by Maryland Aerospace Inc. In the standard configuration, the MAI-200 provides attitude control to within 1 degree using the supplied sensors which consist of a magnetometer and a sun sensor. The accuracy can however be substantially improved by interfacing extra sensors such as a star tracker through the serial interface. Using the extra sensors, the MAI-200 provides attitude control to within 60 arcseconds (3σ) [33]. Figure 3.13 illustrates the MAI-200. The unit can also be seen on the far left of figure 3.11.

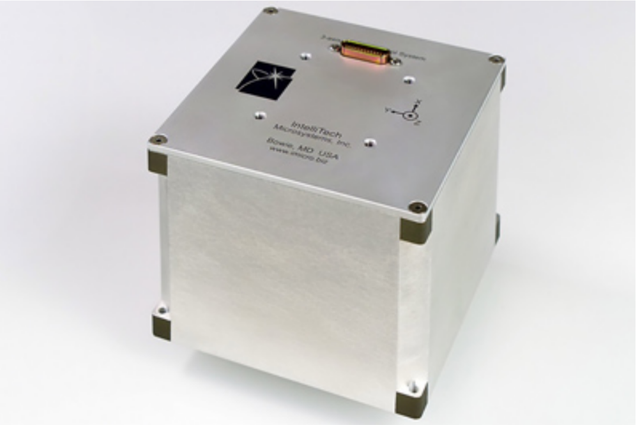


Figure 3.13: MAI-200 - Miniature 3-Axis Reaction Wheel Attitude Determination and Control System for CubeSat Kit Nanosatellites by Maryland Aerospace Inc. [5]

3.2.4 Pointing Performance

A hardware in-the-loop test has been carried out on an optical bench which represents the optics on ExoplanetSat. The lens, together with the piezoelectric stage and focal plane are pointed towards a star field emulator. The star field emulator is a perforated plate, back-illuminated by a light source. A second piezoelectric stage is used to move the star field in the X- and Y-direction, to simulate the ADCS drift. Precomputed trajectory motion is then inserted into the star field emulator. The test-setup is illustrated in figure 3.14.

In the first test (figure 3.15 a), the piezo stage on the spacecraft emulator is disabled. There are two dominant frequency components present in the star-field emulator injected movement. The low frequency component is due to the drift of the ADCS, in other words the ADCS is not capable of controlling the attitude precisely enough. The high frequency jitter is due to the imbalances of the reaction wheels of the MAI-200 ADCS and the centroiding error. The control loop is able to attenuate the low frequency drift, but the reaction wheel imbalances as well as the centroiding errors remain.

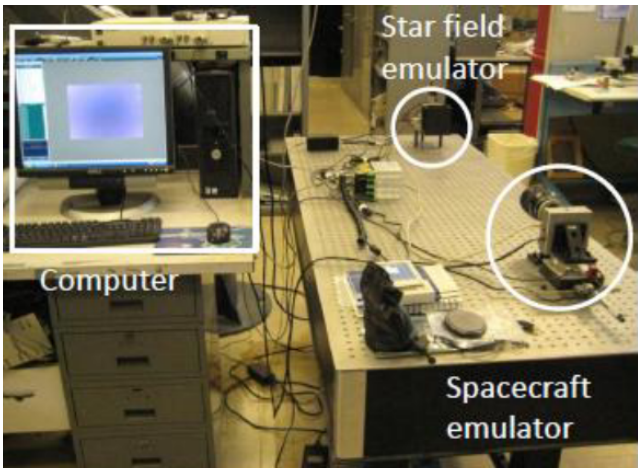
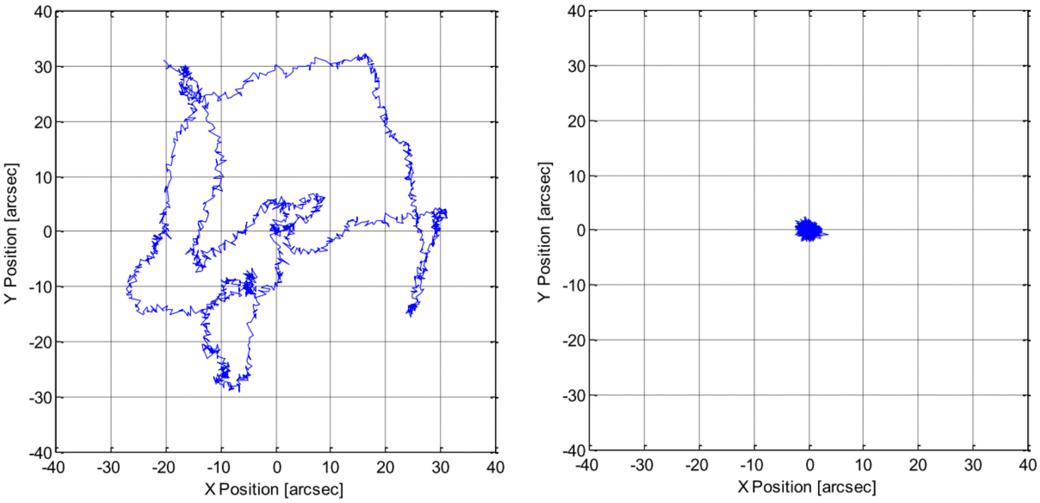


Figure 3.14: Test setup of the fine pointing system) [33]

During the second test (figure 3.15 b), the piezoelectric stage on the spacecraft emulator is enabled. The guide imager now detects the movement of the star field and actively corrects for the errors. The residual pointing error with the stage enabled is 2.3 arcseconds (3σ) [33].



(a)

(b)

Figure 3.15: Pointing resolution of ExoplanetSat, fine pointing stage disabled (a), fine pointing stage enabled (b) [33]

3.3 PicSat

PicSat is a French CubeSat dedicated to unravelling the mysteries of a particular stellar system, named the Beta Pictoris System. PicSat has been developed by a small team at the Laboratoire d'études spatiales et d'instrumentation en astrophysique (LESIA) part of the Paris Observatory [6]. Figure 3.16 illustrates a rendering of the nanosatellite.

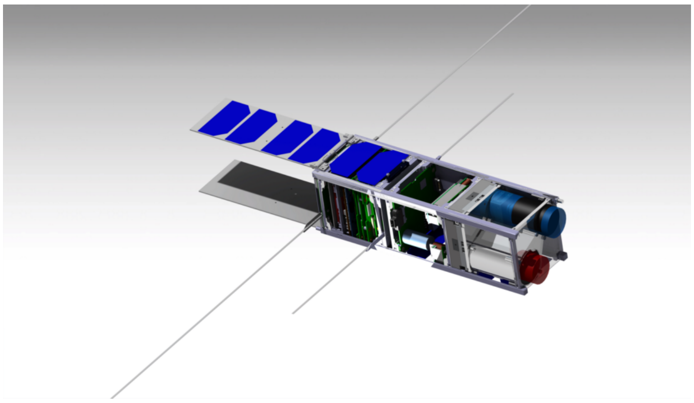


Figure 3.16: Rendering of PicSat [24]

3.3.1 Spacecraft Configuration

PicSat is based on a 3U CubeSat bus. Figure 3.17 illustrates an overview of the configuration of PicSat. The payload (depicted in red) consists of 3 major parts. The telescope (1) gathers the light and focusses it onto a fiber that is attached to a positioning stage (2). Finally the fiber is coupled to a photodiode (3) in order to measure the intensity. The ADCS (depicted in green) consists of a star tracker that is placed towards the front of the satellite and the module with the reaction wheels located in the middle of the satellite. The avionics (depicted in blue) are located towards the back of the satellite and are responsible for data processing, power regulation and communications.

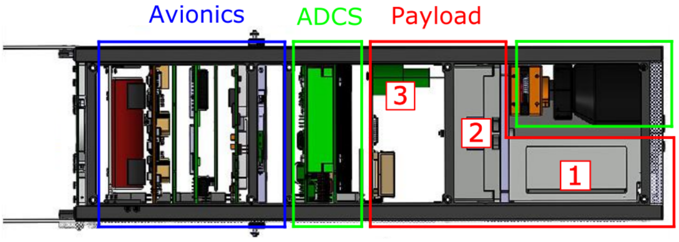


Figure 3.17: Configuration of the PicSat 3U satellite bus [14]

3.3.2 Optical Assembly

The optics on the PicSat payload consist of a telescope (mirrors M1 and M2), a piezo positioning stage and a photodiode. The light gathered by the telescope is coupled into a single mode fiber (SMF) that is attached to a Cedrat XY400M piezoelectric positioning stage. The positioning stage control loop runs at 100 Hz. The fiber couples the light to a single-photon avalanche diode (SPAD). Together these are called the PicSat Fibered Photometer (PFP) [24].

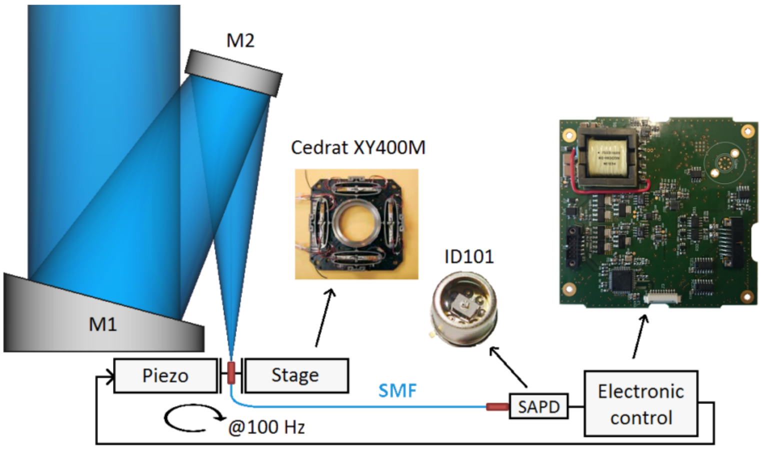
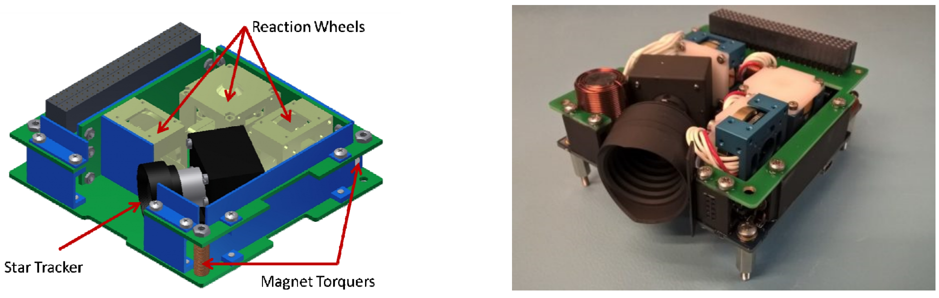


Figure 3.18: Configuration of the picsat optical payload [24]

This optical configuration is unique in the way that only a single photodiode is used to measure the intensity. Essentially the photodiode acts as a single-pixel detector to perform photometry measurements instead of an array of pixels.

3.3.3 Attitude Control

The attitude of PicSat is controlled by the iADCS-100 developed by Berlin Space Technologies (BST). The iADCS-100 uses an ST-200 star tracker that can detect stars up to magnitude 6, determine the attitude with an accuracy of 30 arcsec (3-σ) and provides control to within 1 degree [32].



(a) (b)

Figure 3.19: iADCS-100 CAD-drawing (a) and real unit (b) [32] [35]

3.3.4 Pointing Performance

Figure 3.20 illustrates a simulation of the expected pointing performance of PicSat. The centroids are plotted in function of the distance from the photodiode. With the piezo stage disabled (3.20 left), a 1-σ variation of 20 μm is present. However upon enabling the XY-stage (3.20 right), the variation decreases to 0.7 μm (1-σ). Since a distance of 3 μm on the focal plane corresponds to approximately 2 arcseconds, the pointing performances translate into 13.3 and 0.47 arcseconds respectively (1-σ).

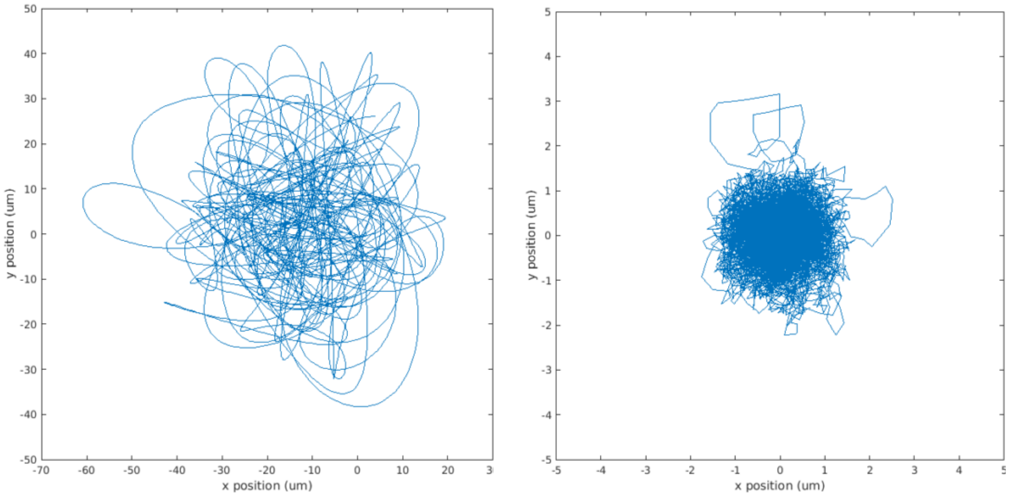


Figure 3.20: Result of a MATLAB/Simulink simulation of the pointing performance of PicSat, piezo stage disabled (left) and piezo stage enabled (right), [24]

3.4 Conclusion

Table 3.3 gives a summary of the case study. The satellites are listed together with the active optics (AO) parameter as well as the ADCS and the final pointing performance of the payload. There is a clear trend towards active optics if arcsecond-level pointing is required. This is complemented by better ADCS subsystems that feature ever more accurate attitude determination and faster sampling of the star tracker due to improved imagers with increased sensitivity. The usage of innovative optics systems such as fibred photometers with a single diode detector are also gaining in popularity. Taking into account these results, as well as the fact that CUBESPEC will feature a custom ADCS by KU Leuven and feature an internal active pointing platform, a 3.6 arcsecond pointing requirement appears very feasible.

Table 3.3: Case study summary

CubeSat	A.O.	ADCS	Pointing Perf. (arcsec)
UniBRITE	No	SFL30 + AA-MST	45.0 (rms)
BRITE-Toronto	No	SFL30 + ST-16	11.8 (rms)
ExoplanetSat	Yes	MAI-200	2.3 (3-σ)[1]
PicSat	Yes	iADCS-100	1.41 (3-σ)[1]

[1] Expected results obtained through simulations and/or bench tests

Chapter 4

Electro-optics

The bench tests and system development illustrated in chapters 6 and 7 are performed with a Ximea USB 3.0 camera together with Python and OpenCV. However, preliminary work has also been done on interfacing a CMOSIS CMV2000 sensor to a Zynq-7000 FPGA since this will be very similar to the actual electronics configuration for CUBESPEC.

4.1 Hardware Overview

Figure 4.1 illustrates a block diagram of the electro-optics subsystems. In the following paragraphs, each of the systems will be looked at in detail, as well as the communication between the blocks and how this has been implemented. Later on, the test results are shown and discussed.

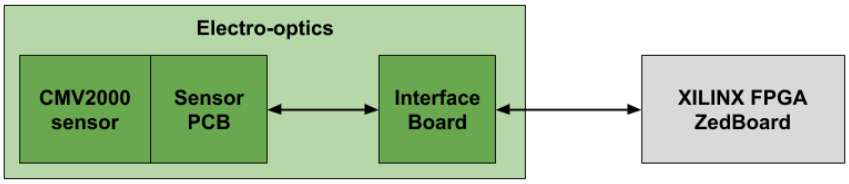
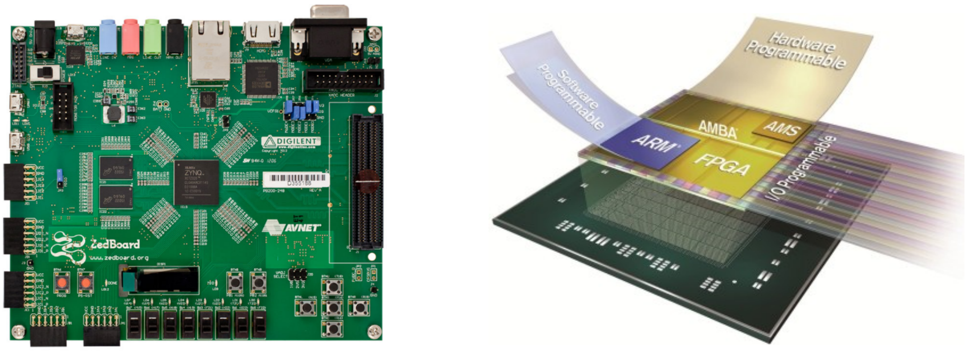


Figure 4.1: Overview of the electro-optics subsystems

4.1.1 ZedBoard

A Xilinx ZedBoard is used as a controller since it has a variety of Input/Output (I/O) interfaces that are suitable for interfacing with other devices. The core of the ZedBoard consists of a Zynq-7000 all Programmable System on Chip (SoC). The ZedBoard and Zynq-7000 SoC are illustrated in figure 4.2.
The Zynq-7000 SoC consists of a configurable part in the form of a field-programmable gate array (FPGA) and a fixed processor architecture, in this case ARM. This



(a)

(b)

Figure 4.2: Xilinx ZedBoard [40]

combines the software configurability of a microprocessor with the hardware configurability of an FPGA, allowing flexible hardware- and software-design.

4.1.2 CMV2000 Image Sensor

In the final implementation on the satellite, the CMOSIS CMV4000 image sensor will be used as a Fine Guidance Sensor (FGS) to provide optical feedback to the system. During the initial prototyping stage however, the 2.2 mega pixel CMV2000 image sensor by CMOSIS will be used. Both sensors feature identical communication standards and data transfer schemes, making the implemented systems also usable for the CMV4000 with minor changes. The appearance and most important parameters of the sensor are illustrated in figure 4.3 and table 4.1.

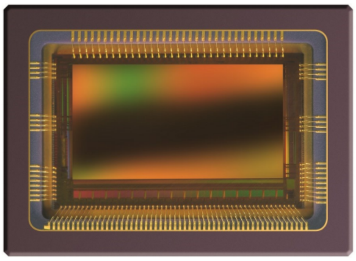


Figure 4.3: CMOSIS CMV2000 image sensor [3]

Parameter	Typical	unit
Format	2048x1088	pixels
Type	mono	-
Resolution	10 and 12	bits
Pixel pitch	5.5	μm
Power consumption	550 to 1200	mW
Framerate	340	FPS
Interface	LVDS	-
Channels	16	-

Table 4.1: CMV2000 specifications [3]

A block diagram of the sensor is illustrated in figure 4.4. The sensor features two main interfaces. A serial peripheral interface (SPI) is used to control the sensor's functions while a low-voltage differential signalling (LVDS) interface is used for the high-speed readout of the sensor.
The hardware for interacting with the sensor's interfaces is elaborated in section 4.2.

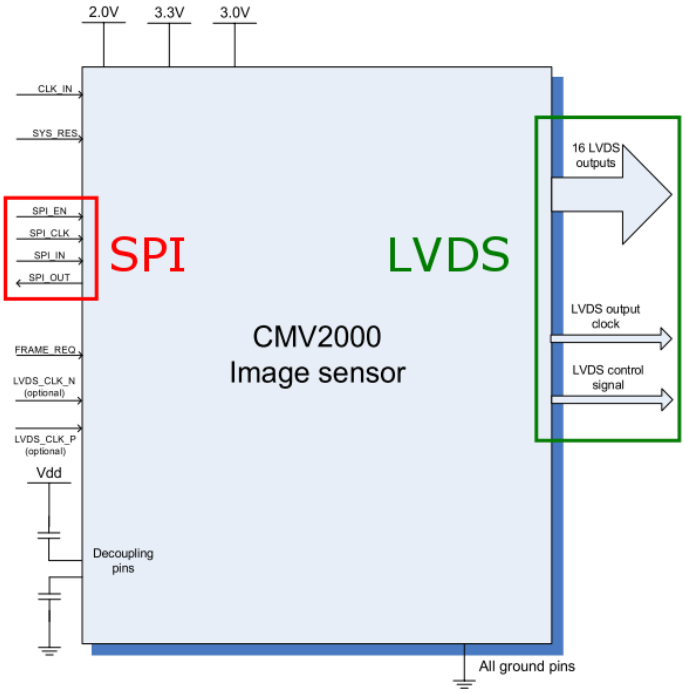


Figure 4.4: CMOSIS CMV2000 image sensor schematic overview [3]

4.1.3 Sensor PCB

The sensor is mounted on a custom PCB that supplies it with the correct signals and voltages. The PCB has been designed in-house by KU Leuven. Two flat-flex cables (FFC) can be attached to the PCB. These facilitate the frame transfer as well as control over the SPI-interface. Figure 4.5 illustrates the backside of the sensor PCB.

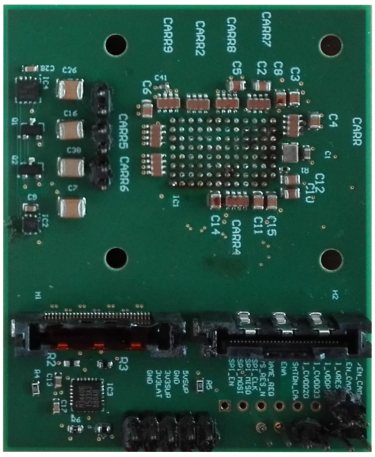


Figure 4.5: Back-view of the sensor PCB

4.1.4 Interface Board

In order to interface the Sensor PCB with the FPGA board, an adapter board has been designed. The adapter board is compatible with the FPGA Mezzanine Card Low Pin Count (FMC-LPC) connector on the ZedBoard and allows interfacing with the FFC connectors going to the sensor board. A jumper is used to select either 3V3 power from the FPGA board or an external supply. Figure 4.6 depicts the appearance of the interface board. The multilayer PCB has been designed using Altium Designer.

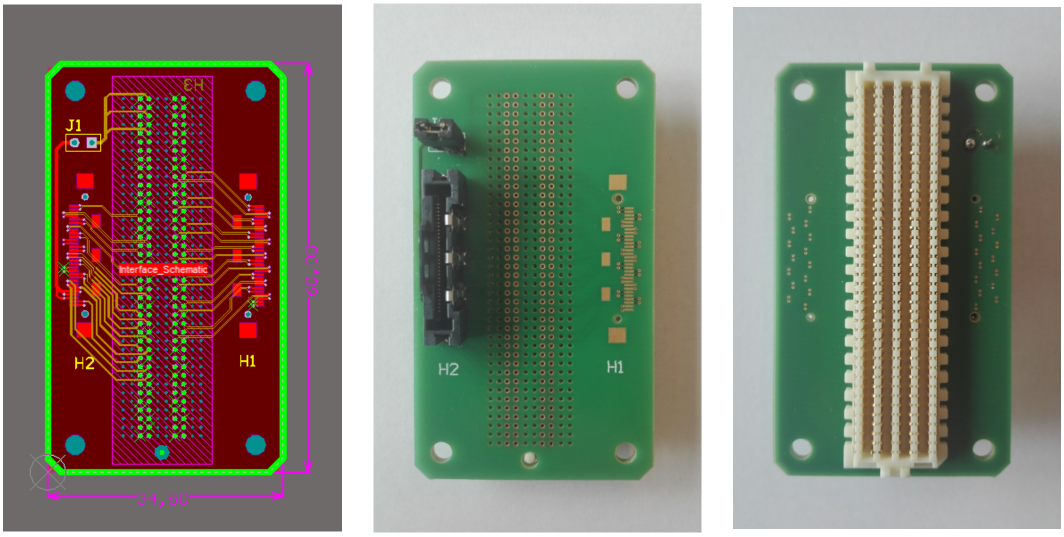


Figure 4.6: Interface PCB - Altium drawing (left), top view of finished PCB (middle) and bottom view of PCB (right)

During the design process, attention has been paid to the impedance of the differential traces, which is approximately 100 ohms. PCB traces that carry larger currents are made wider to keep Joule-losses to a minimum. Figure 4.7 illustrates the sensor-PCB connected to the FPGA via the interface PCB. If the full 16-channel interface is required, another FFC-cable can be connected.

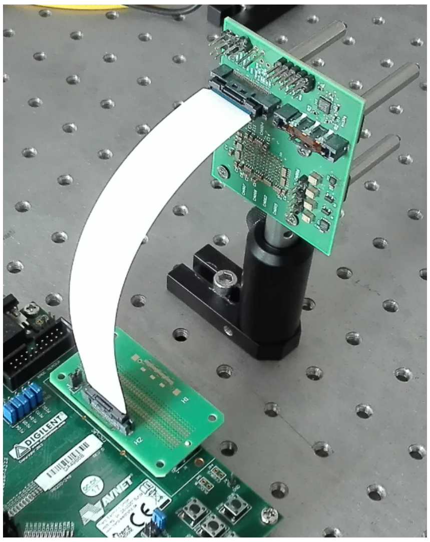


Figure 4.7: Sensor PCB connected to the FPGA via the designed interface board

4.2 Hardware Design

In order to interface with the sensor, an SPI controller implemented in hardware is used. Vivado (Xilinx) is used to design the hardware necessary to interact with the image sensor. Commonly used hardware is integrated into blocks called intellectual property blocks (IP Blocks) and allow rapid design of a hardware system on the Zynq FPGA. Communication between these blocks is provided by a databus standard called AXI.

4.2.1 Sensor Control Hardware

As mentioned before, the CMV2000 image sensor can be controlled through the use of SPI. The SPI protocol involves four signals that are present on the sensor:

- SPI_EN: Active high signal indicating activity, also called slave select (SS)
- SPI_CLK: A clock signal which is used to sample the data
- SPI_IN: Data to write to the image sensor (master to slave)
- SPI_OUT: Data to be read from the sensor (slave to master)

The SPI_IN signal on the image sensor is also commonly referred to as master-out-slave-in (MOSI). Similarly, the SPI_OUT signal is often referred to as master-in-slave-out (MISO). In this case the sensor is the slave and the FPGA is the master. The settings of the sensor can be controlled by writing the appropriate values in the sensor's registers. Each register in the sensor is assigned a 7-bit address to which data can be written or read from. Figure 4.8 illustrates the SPI read timing as described by the datasheet of the image sensor.

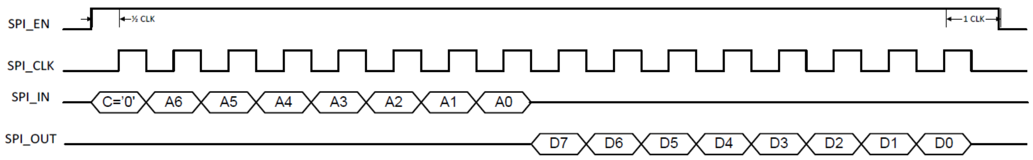


Figure 4.8: SPI read timing [3]

When reading the contents of a register, the 7-bit source-address (A6 to A0) needs to be sent preceded by a control bit C, set to '0' indicating a read operation. Immediately after sending the last address bit, the sensor responds with the 8-bit register data (D7 to D0). All data sent across the SPI interface is sent MSB first.
A write operation requires the sent address to be preceded by a control bit C set to '1'. Immediately following the 7-bit address (A6 to A0), the data to be written into the register needs to be sent (D7 to D0). Figure 4.9 illustrates a write sequence to the image sensor.

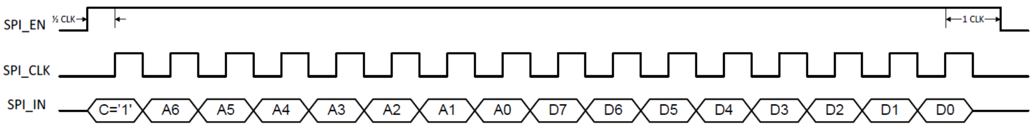
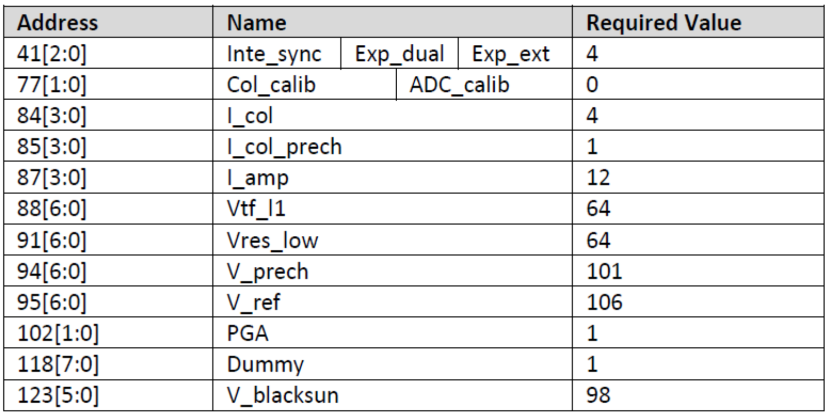


Figure 4.9: SPI write timing [3]

Upon initialisation of the CMOS sensor, a number of register need to be programmed with values that are different from their default start up value. This recommended 'register map' is illustrated in figure 4.10.



Address	Name	Required Value
41[2:0]	Inte_sync Exp_dual Exp_ext	4
77[1:0]	Col_calib ADC_calib	0
84[3:0]	I_col	4
85[3:0]	I_col_prech	1
87[3:0]	I_amp	12
88[6:0]	Vtf_l1	64
91[6:0]	Vres_low	64
94[6:0]	V_prech	101
95[6:0]	V_ref	106
102[1:0]	PGA	1
118[7:0]	Dummy	1
123[5:0]	V_blacksun	98

Figure 4.10: Recommended register values upon startup [3]

To facilitate SPI-communications from the FPGA, An AXI Quad SPI IP-block is used. The normal SPI protocol indicates that SPI_EN needs to be low (active low) whenever a read or write action is being carried out. However the CMV2000 image sensor requires SPI_EN to be high during data transfer (active high). This is solved by adding an inverter to the SPI_EN or slave select (SS) signal coming from the AXI Quad SPI block. The block diagram in Vivado is depicted in figure 4.11.

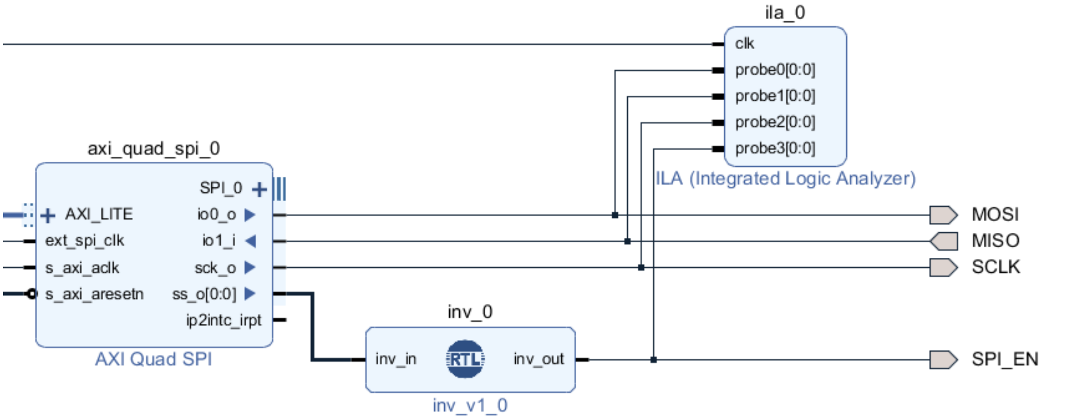


Figure 4.11: AXI Quad SPI block with external signals to the CMV2000

As seen in figure 4.8, both the transmitted address and the response by the sensor are 8-bit wide. Therefore the transaction width is set to 8 in the IP-block settings. Master mode is enabled since the image sensor is the slave, and the FPGA is the master. An integrated logic analyser (ILA) has been added to the block design for debugging purposes.

4.2.2 Sensor Readout Hardware

The pixel data of the sensor is transmitted via an LVDS-interface. This type of interface allows high data rates while still maintaining a good resilience to noise. The good performance in noisy conditions is provided by the differential nature of the transmission lines. Noise causes the same disturbance on both data lines. Since the receiver measures the voltage differential on these lines, noise has little impact on the received signal. Figure 4.12 depicts how the influence of noise is reduced to a minimum.

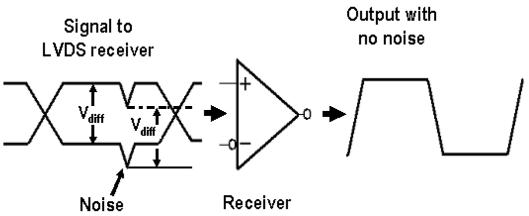


Figure 4.12: Influence of noise on a LVDS transmission pair [17]

In total there are 18 LVDS output pairs on the sensor, where 16 pairs are pixel data outputs and 2 pairs are clock and control signals. The clock output allows sampling of the pixel data on the FPGA while the control output can be used to synchronize the pixel data. In the current PCB and cable configuration, 4-output LVDS is used since the 4 LVDS channels and SPI are available on one FFC connection. In this mode the image data is sent using only 4 LVDS channels. Figure 4.13 depicts the channelisation in 4-channel mode. Each channel is used to transfer a 512 pixel wide portion of the frame. The pixel remapping for a readout of one row is illustrated in figure 4.14.

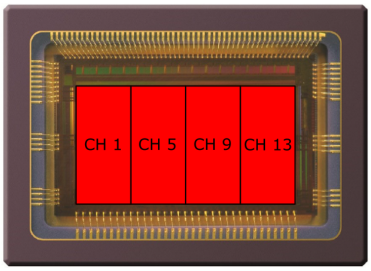


Figure 4.13: Channelization of the frame in 4-output mode

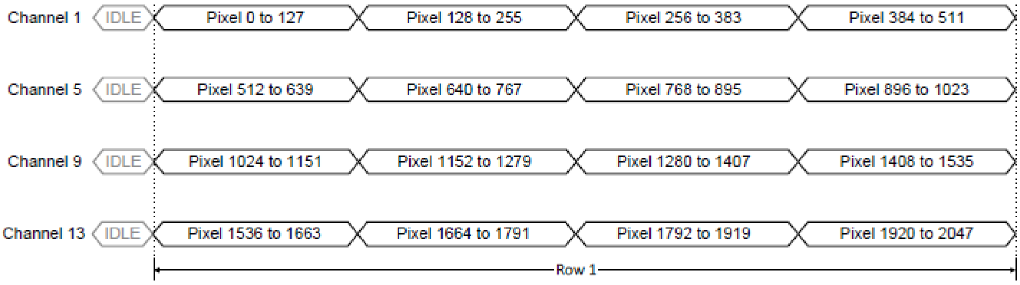


Figure 4.14: Pixel remapping in 4-output mode [3]

Differential receivers in hardware are used to interface with the LVDS channels coming from the sensor. More specifically this is done by utilising the Utility Buffer in Vivado and configuring it for differential input signals, meaning a positive signal (indicated by P) and an inverted signal (indicated by N). This is illustrated for CH5 and CH9 as well as the CLK and CTR channel in figure 4.15.

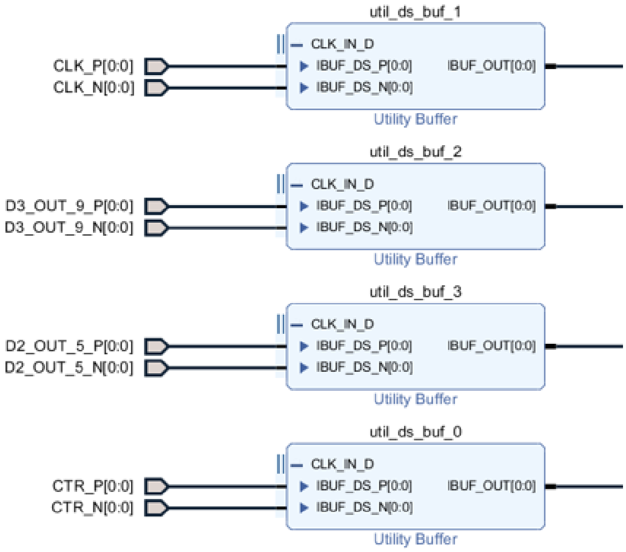


Figure 4.15: LVDS receivers

Once the serial stream is properly received, the original pixel and framing data needs to be recovered. This process is called deserialisation or synchronisation. The sensor facilitates this process by outputting a training pattern when there is no active image data on the outputs. This allows hardware to synchronise on the datastream and remain in-lock. However, prior to synchronisation, proper sampling of the serial data must be achieved.

The sampling clock that is provided by the CMOS sensor (CLK) can not be used directly to sample the signals (DATA), since the transition of the datasignals occurs on a rising or falling edge of the sampling clock. To achieve reliable data sampling, the clock must be delayed in phase by 90 degrees by means of a phase locked loop (PLL). The delayed clock now provides transitions when the data is in a stable state. To achieve data sampling on every rising edge of the clock, the phase-delayed clock (CLK-90) must further be multiplied by a factor of two. This process is illustrated in figure 4.16.

Since the data is now properly sampled, the synchronisation can be implemented. As mentioned before, the sensor outputs a training pattern that can be used to synchronise with. A training pattern is a fixed sequence of bits that is known. The default 10-bit training pattern on the LVDS-lines is 00 0101 0101_2. When no active pixel data is being sent, this pattern is present on the output of the sensor. A synchronisation block has been written in Verilog that uses the sampling clock that was generated using the method in figure 4.16 as well as a slower pixelclock. The sensor sampling resolution in this case is 10 bits per pixel (BPP). Therefore, the rate of the pixelclock is 10 times slower than that of the bitclock coming from the sensor.

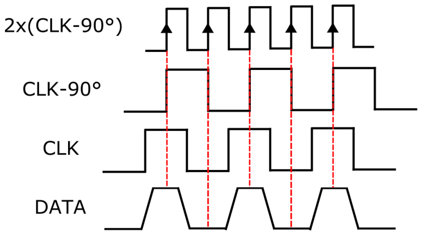


Figure 4.16: Achieving correct data sampling by using a PLL

Figure 4.17 illustrates the synchronisation block. The block requires a sampling clock (SAMPCLK) and a slower pixel clock (PIXCLK). The synchronisation block converts the serial stream to a 10-bit parallel stream. Initially, the block checks for the training pattern on the LVDS-lines, once the pattern is found, a counter is started. Every time the counter is reset, a check is done to verify if there is still data-lock. In the case of a loss of lock, the counter is halted until a new training pattern is found.

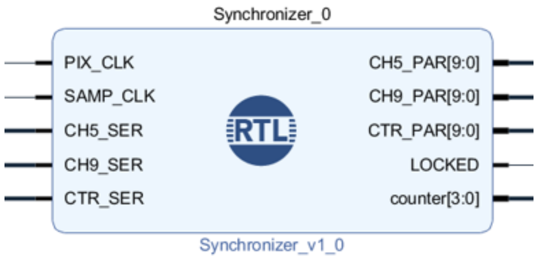


Figure 4.17: Synchronisation block implemented in Verilog

To allow frame buffering and manipulation, the frames need to be transferred to a sufficiently large storage area. The DDR memory on the ZedBoard (512 MB) is more than large enough to store one frame. To interface with the on-board DDR memory, direct memory access (DMA) must be used. A Xilinx AXI4 to DMA block can be used to fulfil this purpose. However this block requires the data to be formatted to AXI-4 stream format.

The last step is to convert the data format of the sensor to the AXI-4 format. This can be done via the Video-in to AXI-4 stream block. The Video-in to AXI-4 block requires the video data as well as timing signals. The timings signals are generated using a custom block illustrated in figure 4.18.
The generated timing signals together with the video-data are sent to the AXI DMA block providing access for a stream to a memory mapped (S2MM) interface.
Figure 4.19 illustrates the entire chain.

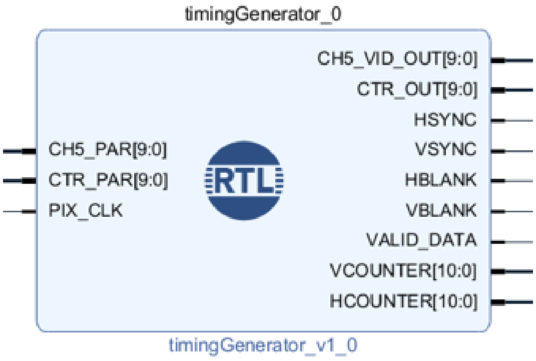


Figure 4.18: Timing generator block implemented in Verilog

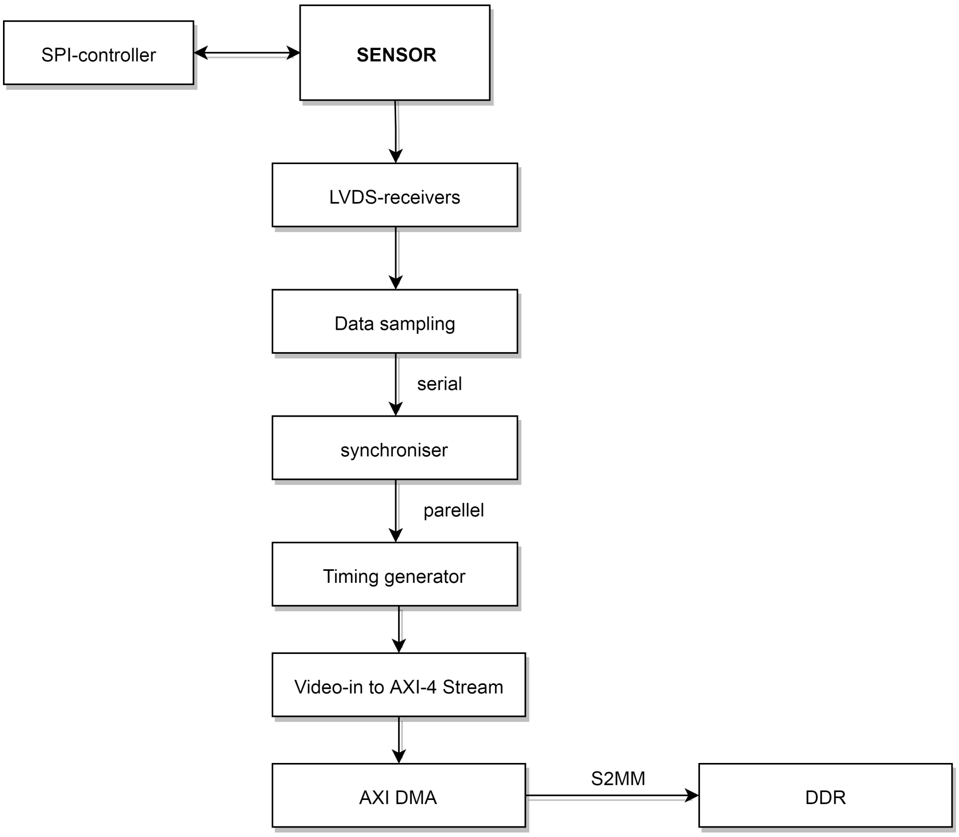


Figure 4.19: The entire sensor data processing pipeline

4.3 Software Design

4.3.1 Sensor Control Software

When the hardware design is finished it has to be synthesised and implemented on the ZYNQ chip. After implementation it is possible to export a bitstream and program the FPGA to the desired architecture. For initial testing, a standalone operating system is used to run the bare C-code on the integrated ARM processor. A board support package (BSP) is generated from the exported hardware that is developed in Vivado. From the software side it is possible to interact with the hardware through the AXI-bus and the appropriate libraries that follow from the board support package. When the sensor initialised, the required registers are programmed and read back to verify successful transactions, the results are discussed in section 4.4.1.

4.3.2 Sensor Readout Software

When successful frame transfer to the DDR is achieved, a lightweight Linux distribution running on the Zynq-7000 can be configured to allow frame reading from a fixed memory space. A device driver and API collection called Video4Linux can be used to read the frames from memory and interact with other applications such as OpenCV [38]. Figure 4.20 illustrates how the frames could be read in Linux for further work.

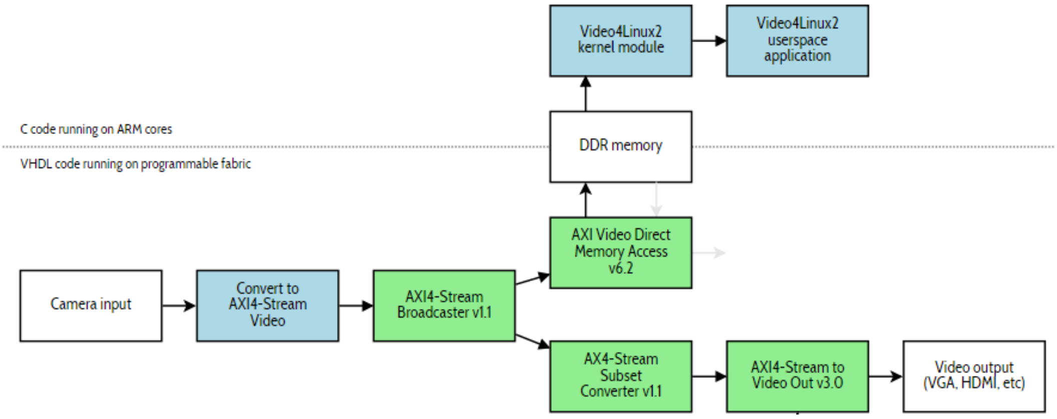

Video4Linux2 kernel module
Video4Linux2 userspace application
C code running on ARM cores
DDR memory
VHDL code running on programmable fabric
AXI Video Direct Memory Access v6.2
Camera input
Convert to AXI4-Stream Video
AXI4-Stream Broadcaster v1.1
AX4-Stream Subset Converter v1.1
AXI4-Stream to Video Out v3.0
Video output (VGA, HDMI, etc)

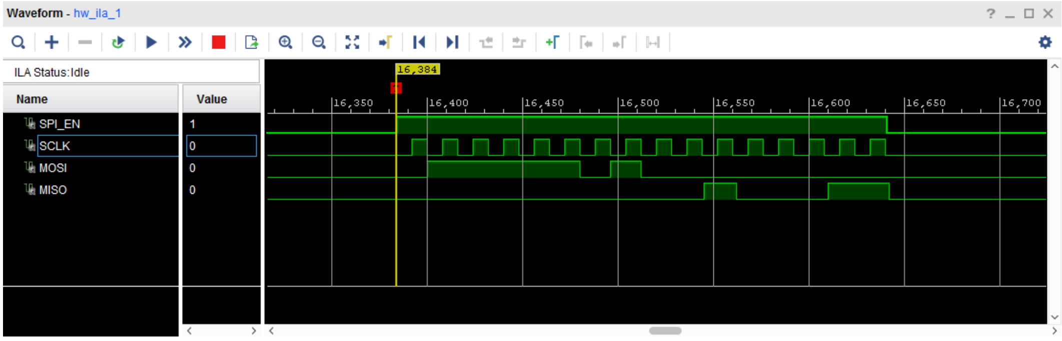
Figure 4.20: Utilising the S2MM channel of the VDMA for video capture [38]

4.4 Test Results

4.4.1 Sensor Control Tests

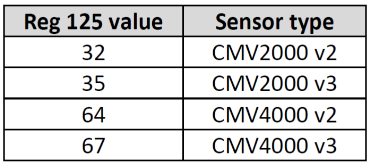
Initially a test has been done to asses if there is correct communications between the CMOS sensor and the FPGA through the SPI-interface. A script in C has been written that reads the values of several registers and outputs this data over the UART port. Besides the serial output in the terminal, the lower level signals can be inspected by using the ILA that was implemented in the Vivado block diagram.

A trigger was has been on the rising edge of SPI_EN and the C script was run. Figure 4.21 illustrates a read action as seen on the waveform viewer. The displayed sequence is a read operation since the first bit prior to the sent address is a logic 0. In this case the sent address is 111 1101_2 or 125. By reading register 125 it is possible to determine the type and version of the image sensor as illustrated in figure 4.22. In this case the 8-bit response of the sensor is 0010 0011_2 or 35. This indicates the sensor is of the type CMV2000 version 3 [3].



Name	Value
SPI_EN	1
SCLK	0
MOSI	0
MISO	0

Figure 4.21: Reading register data from the CMOS image sensor



Reg 125 value	Sensor type
32	CMV2000 v2
35	CMV2000 v3
64	CMV4000 v2
67	CMV4000 v3

Figure 4.22: Sensor version in function of contents of register 125 [3]

Besides reading from the registers of the CMOS sensor it is also possible to write data. A number of registers need to be written with the correct values upon start-up since their default value is different from the recommended value. A write sequence is displayed in figure 4.23.

Unlike the read operation, the first bit is now a logic 1 indicating a write operation. The control bit is followed by the 7-bit address of the register that needs to be written. This is immediately followed by the 8-bit data that needs to be written to that register. In this case a value of 4 is written into register with address 010 1001_2

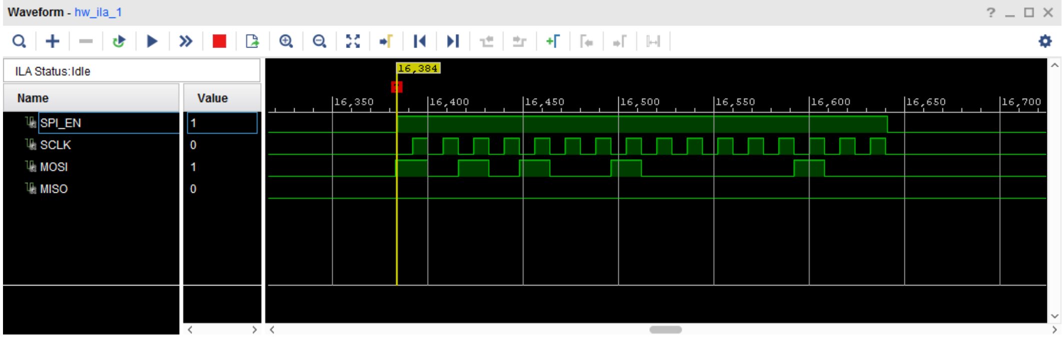
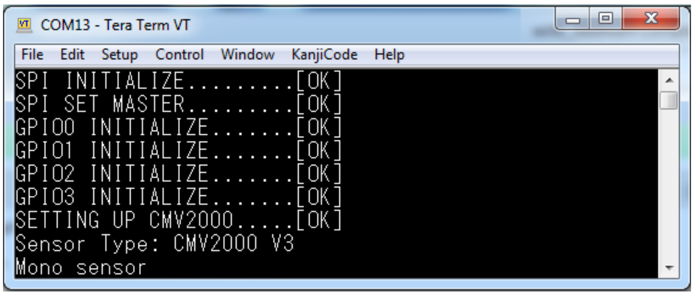


Figure 4.23: Writing data to the CMOS image sensor

or 41. Upon writing a certain value into a register, the same register is afterwards evaluated to make sure the write operation was successful. The initialisation of the SPI and general purpose input/output (GPIO) pins is carried out first. Afterwards the sensor type is read as a test and the info is displayed on-screen.



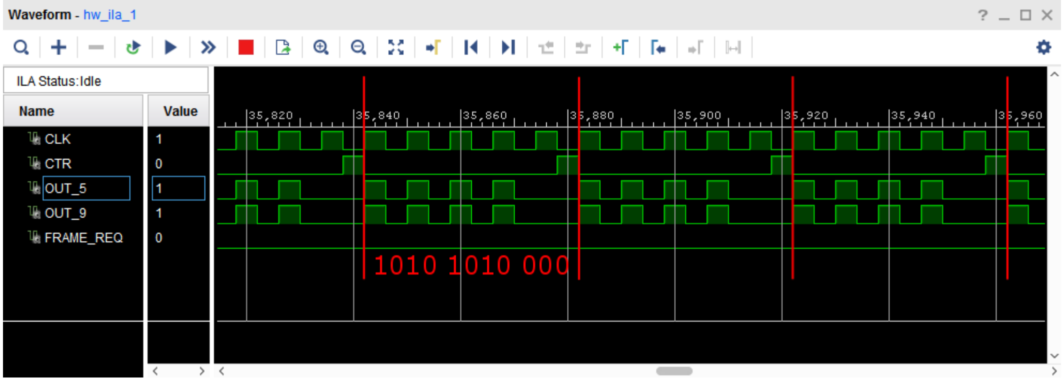
```
SPI INITIALIZE.........[OK]
SPI SET MASTER.........[OK]
GPIO0 INITIALIZE.......[OK]
GPIO1 INITIALIZE.......[OK]
GPIO2 INITIALIZE.......[OK]
GPIO3 INITIALIZE.......[OK]
SETTING UP CMV2000.....[OK]
Sensor Type: CMV2000 V3
Mono sensor
```

Figure 4.24: Data viewed on the serial terminal

4.4.2 Sensor Readout Tests

Figure 4.25 illustrates the output of the LVDS channels. The bottom two channels are the LVDS outputs from the sensor. The training pattern is clearly visible since no active imagery is sent. Since the data on this interface is sent Least Significant Bit (LSB) first, the training pattern (00 0101 0101_2) is seen in reverse order. The top channel represents the sampling clock being sent by the sensor. The second channel from the top is the control channel that can be used to determine the state of the sensor. The MSB and MSB-1 bit of the control channel are always '1' and '0' respectively. These two bits are also used to periodically check if there is data-lock. In this case all the other bits are zero which indicates the sensor is in idling mode. The separate pixels are indicated by the red vertical separators.

The implemented synchronisation block and timing generator have been tested by triggering the sensor using the FRAMEREQ pin to start a frame dump. Figure 4.26 illustrates the start of a frame (SOF). The top channel represents the pixel-data.



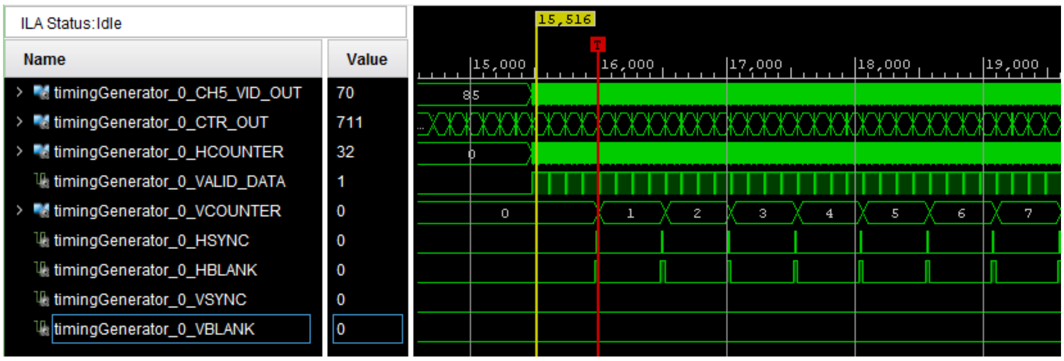
Waveform - hw_ila_1

ILA Status:Idle

Name	Value
CLK	1
CTR	0
OUT_5	1
OUT_9	1
FRAME_REQ	0

Figure 4.25: Idling data output seen on the LVDS-channels

Initially the pixel data has a value of 85, which matches the decimal value of the training pattern. Once the frame transfer is started, the HCOUNTER counts every time the pixel clock has a rising edge. In 4.26, as illustrated by the VCOUNTER, the readout of the first 8 lines is shown. The readout of one section of the sensor (512 pixels in 4-CH mode) line is split up in four transfers of 128 pixels. The validity of each 128 pixel block is indicated by the VALIDDATA signal. Between two 128 pixel blocks, the training pattern is present on the output, and therefore is not valid image data.

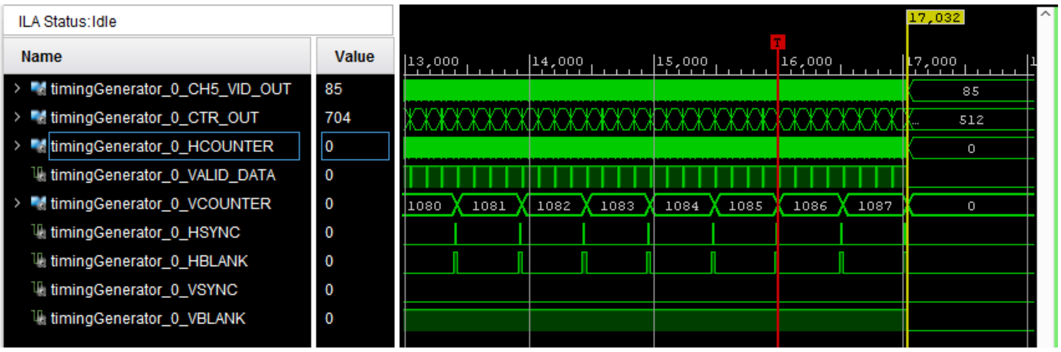


ILA Status:Idle

Name	Value
timingGenerator_0_CH5_VID_OUT	70
timingGenerator_0_CTR_OUT	711
timingGenerator_0_HCOUNTER	32
timingGenerator_0_VALID_DATA	1
timingGenerator_0_VCOUNTER	0
timingGenerator_0_HSYNC	0
timingGenerator_0_HBLANK	0
timingGenerator_0_VSYNC	0
timingGenerator_0_VBLANK	0

Figure 4.26: Start of a frame

Figure 4.27 depicts the end of a frame (EOF). The last line finished transferring when the HCOUNTER reaches 1087, indicating the readout of a total of 1088 lines. After the frame end, the pixeldata reverts to the training pattern (decimal value of 85) and the control channel also reverts to the default value.



ILA Status:Idle

Name	Value
timingGenerator_0_CH5_VID_OUT	85
timingGenerator_0_CTR_OUT	704
timingGenerator_0_HCOUNTER	0
timingGenerator_0_VALID_DATA	0
timingGenerator_0_VCOUNTER	0
timingGenerator_0_HSYNC	0
timingGenerator_0_HBLANK	0
timingGenerator_0_VSYNC	0
timingGenerator_0_VBLANK	0

Figure 4.27: End of a frame

Chapter 5

Electromechanics

This chapter gives an overview of the electromechanical subsystems. The hardware and software design for the specific subsystems is also elaborated. At the end of the chapter, the results of the individual tests are illustrated and discussed.

5.1 Hardware Overview

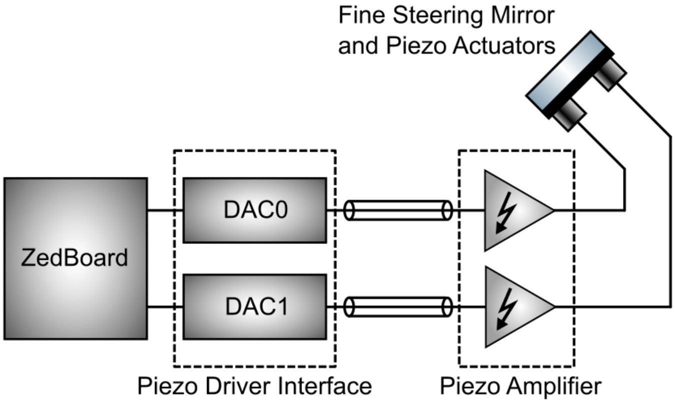
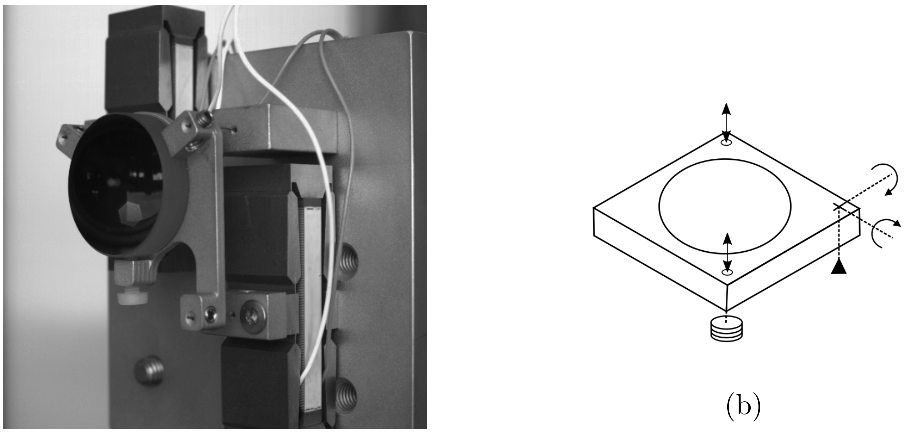


Figure 5.1: Electromechanics subsystem

5.1.1 Fine Steering Mirror and Actuators

The key component of the electromechanics is a fine steering mirror (FSM). The FSM consists of a mirror assembly that is connected to piezoelectric actuators. The mirror assembly consists of a tip-tilt configuration. Two actuators and a fixed pivot point serve as the base for the circular mirror. Figure 5.2 illustrates the configuration of the steering mirror.

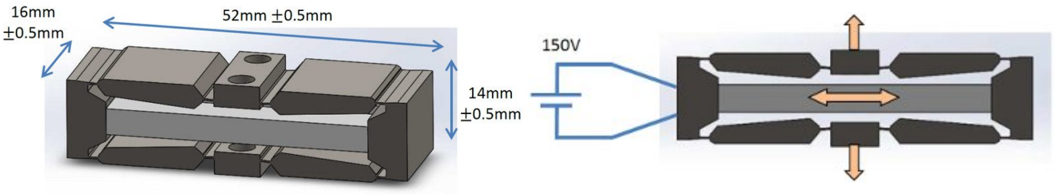


(a)

(b)

Figure 5.2: Fine steering mirror assembly (a) and conceptual drawing (b)

The two actuators are amplified piezoelectric (AP) stack actuators that accept DC voltages of up to 150 V yielding a displacement of 830 μm. The specifications of the AP830 piezoelectric actuators are given by table 5.1.



(a)

(b)

Figure 5.3: 3D-drawing of the amplified stack actuator (a) and direction of travel (b) [26]

Table 5.1: AP830 piezoelectric actuator specifications [26]

Parameter	Typical	unit
Displacement	>830	μm
Voltage	-15 to +150	V
Resonance	230	Hz
Force	90	N
Capacitance	8.3	μF

5.1.2 Piezo Amplifier

The piezoelectric actuators of the steering mirror require high voltages in order to obtain a reasonable stroke. Therefore, suitable amplifiers are used to interface with the actuators. The amplifiers boost an analog input signal to a sufficiently high voltage level to drive the amplified piezo stacks. The piezo amplifier uses low noise solid state amplifiers making it very suitable for this application. In this case only two of the three channels are used since there are only two actuators present on the mirror assembly. Figure 5.4 depicts the amplifier module and the corresponding diagram on figure 5.5.

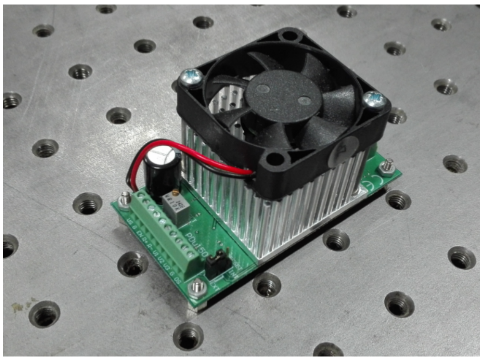


Figure 5.4: PDU150 Piezo Amplifier module from PiezoDrive

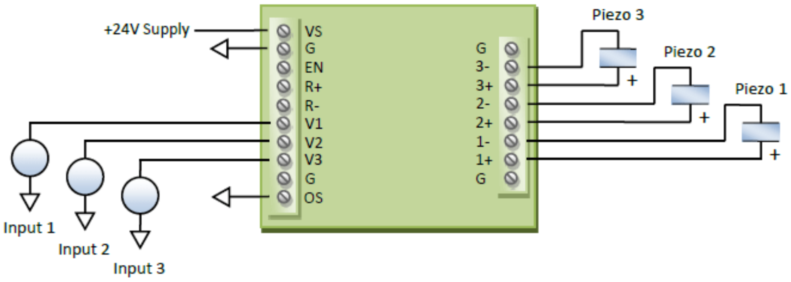


Figure 5.5: Connection diagram of the PDU150 [27]

5.1.3 Piezo Driver Interface

The final component of the electromechanics is the piezo driver interface. The interface consists out of digital-to-analogue converters (DACs) that convert a digital to an analog signal that can be fed to the amplifier. The DACs allow the steering mirror to be controlled through the FPGA. The PmodDA3 is a low-noise 16-bit single channel digtial-to-analogue converter (DAC) that is based on the Analog Devices (AD) AD5541A [9]. No external buffers are needed since the output of the AD5541A is already buffered by an on-board buffer (AD8605). An AD441 provides 2.5 V to the DAC as a precise voltage reference. SMA connectors and coax cables are used to carry the analog voltage to the piezo amplifiers. This reduces the impact of noise on the system. Table 5.2 specifies the output characteristics of the DAC module. The DACs connected to the ZedBoard are illustrated in figure 5.6.

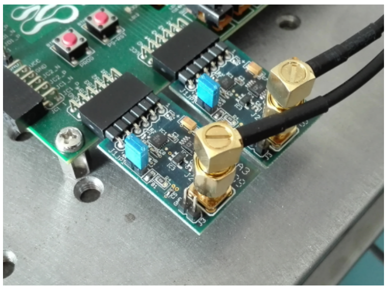


Figure 5.6: 2x PMOD DA3

Parameter	Typical	unit
Resolution	16	bits
Settling time	1	μs
Output noise	0.134	μV_{p-p}
Maximum slew rate	17	$V/\mu s$

Table 5.2: PMOD DA3 specifications [9]

5.2 Hardware Design

5.2.1 DAC Control Hardware

The DACs require a 3-wire SPI interface and an LDAC pin to update the register contents of the output. The SPI- and GPIO-controllers that drive the corresponding pins are implemented in hardware by means of the Vivadeo IP-integrator blocks. The AXI databus is used to make the SPI and GPIO interfaces addressable from the integrated ARM core. Figure 5.7 illustrates how the individual blocks are connected.

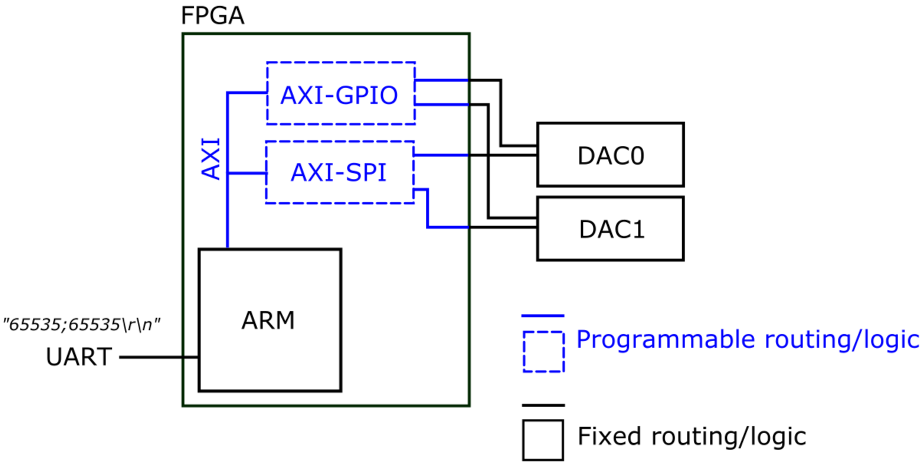


Figure 5.7: Interconnection of the subsystems on the ZedBoard

Since no data is transmitted from the DACs to the FPGA, only a SCLK, CS and MOSI pin are required for communications. An LDAC pin is used to simultaneously update the output of the X-DAC and the Y-DAC, even if the values have been written sequentially in the DAC registers. An AXI Quad SPI block is used for SPI communications and a GPIO IP-block for driving the LDAC pin. Figure 5.6

illustrates a more detailed hardware design in Xilinx Vivado. The final step is the routing of the ports to the corresponding pins on the PMOD connectors to which the DACs have been externally connected. An integrated logic analyzer has also been inserted to provide troubleshooting capabilities.

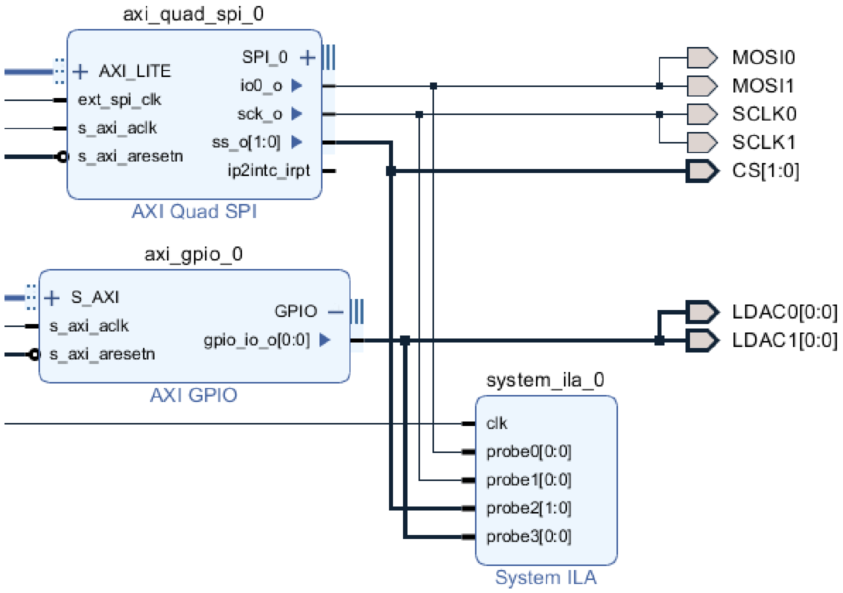


Figure 5.8: Vivado Block Design: DAC control hardware

5.3 Software Design

5.3.1 C-code

The SPI controllers that are implemented in hardware are interfaced with the ARM processor through the AXI data bus. The SPI and GPIO hardware is now software-addressable. When a command is received over the UART interface, the C-code parses the command to obtain the two desired decimal DAC values, separated by a delimiter in the command. The individual values are sequentially written on the common SPI interface and the registers are synchronously updated by toggling the LDAC pin. This sequence is illustrated in figure 5.9.

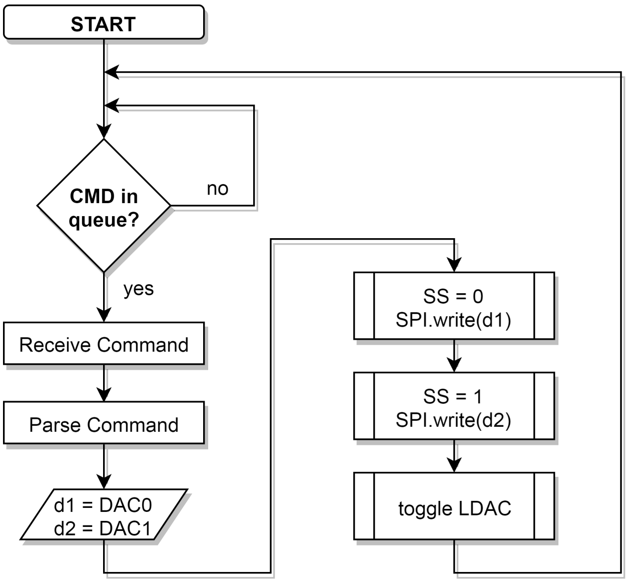


Figure 5.9: Software flow of the electromechanics

5.3.2 LabVIEW Interface

For initial testing purposes, a LabVIEW User Interface (UI) has been developed to easily write new values to the DACs and change the orientation of the steering mirror. The user sets the desired 16-bit values of the DACs and presses the 'LOAD DACs' button. The block diagram of this Virtual Instrument (VI) is illustrated by C.3 which can be found in appendix C. The User Interface (UI) is depicted in figure 5.10. Upon pressing the 'Load DACs' button, a serial command is formatted using the desired values and sent to the ZedBoard.

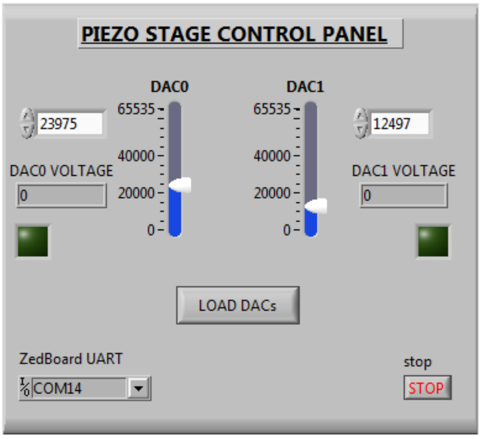


Figure 5.10: LabVIEW user interface

5.4 Test Results

The Integrated Logic Analyser has been used to verify communications with the DACs. As can be seen in figure 5.11, DAC0 and DAC1 are written sequentially and both outputs synchronously updated by driving the LDAC pin low and back to high. The DACs share an SPI bus but each have a separate Slave Select (SS) line to address each slave individually. The CLK line is illustrated in red in figure 5.11. The clocking rate is derived form the main SYSCLK which runs at 100 MHz. The AXI-SPI block uses a frequency divider set at 16 resulting in a clocking transfer rate of 6.25 MHz.

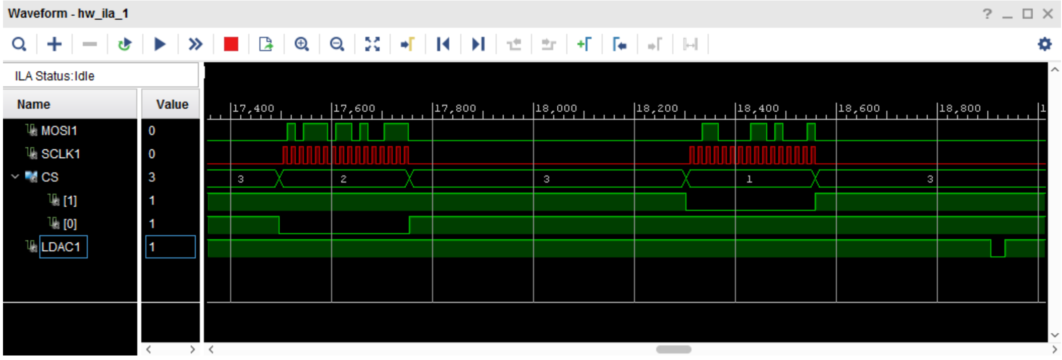


Figure 5.11: SPI writing sequence as viewed by the ILA

Chapter 6

Integration and Setup

The previous chapters illustrate the individual subsystems and the performed tests. This chapter covers the integration of the subsystems in a setup on an optical bench.

6.1 Setup

A setup has been built on an optical table that resembles the fine steering platform configuration of CUBESPEC. A photograph of the setup is illustrated in figure 6.1. A laser coupled fiber (1) and collimator (2) act as an artificial star. The steering mirror (3) is set at 45 degrees to allow the light from the point source to hit the fine guidance sensor (4). The distance between the bare CMV4000 sensor and the steering mirror is 160 mm, the same configuration as in CUBESPEC.

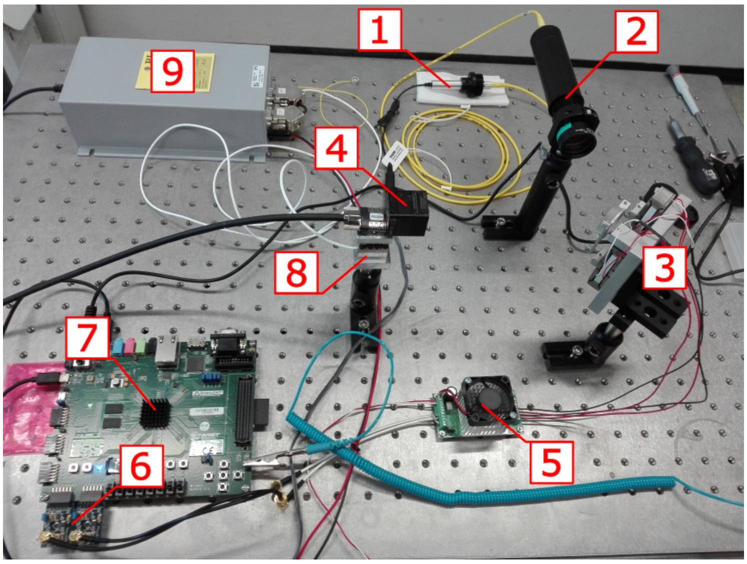


Figure 6.1: Photo of the setup showing the setup on the optical bench

The steering mirror is driven by the piezo amplifier (5) which has a gain of 20. The analogue input of the amplifier is provided by two 16-bit low noise digital-to-analogue converters (6). The DACs are controlled by the ZedBoard which uses a Zynq-7000 FPGA (7). For control-loop bandwidth measurements and disturbance injection, the focal plane is mounted on a translating piezo stage (8). The controller (9) for this stage from Physik Instrumente (PI) can also be seen in the setup. A lens with a focal length of 250 mm is used to converge the light from the fiber to a spot on the detector. The setup is schematically represented by figure 6.2

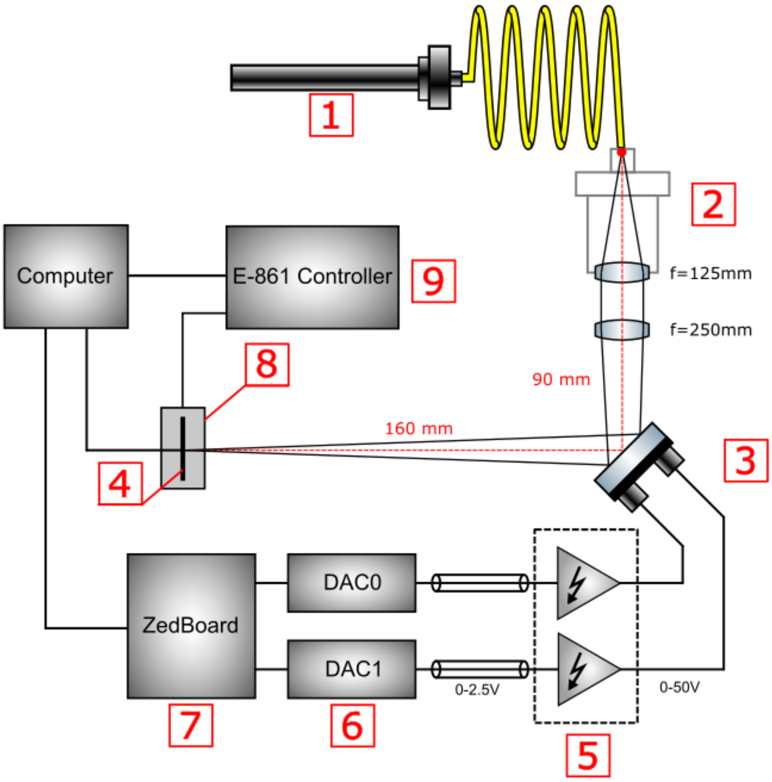


Figure 6.2: Graphical representation of the setup on the optical table

The fine guidance sensor (4) is a Ximea MQ042RG-CM reference camera [42]. The camera is used without a lens since this is the configuration in which it will be used in CUBESPEC. The specifications of the sensor are illustrated in table 6.1.

Figure 6.3: MQ042RG-CM

Parameter	Value	unit
Part No.	MQ042RG-CM	-
Sensor type	CMV4000	-
Sensor size	2048x2048	pixels
Pixel size	5,5	μm

Table 6.1: Camera parameters [42]

6.1.1 Image Scale

In order to link the pixel fractions to angular resolutions, the optical parameters of the collimation lens and camera need to be taken into account. The image scale of an optical system is given by [1]:

$$3600 \times \frac{180}{\pi}\frac{1}{f} = \frac{206265}{f}\frac{arcsec}{mm}. \tag{6.1}$$

Given that the focal length of the setup is equal to 250 mm the image scale of the setup can be calculated as

$$\frac{206265}{250mm} = 825,06\frac{arcsec}{mm}, \tag{6.2}$$

considering a pixel size of 5,5 μm this becomes:

$$5,5 \times 10^{-3}\frac{mm}{pix} \times 825,06\frac{arcsec}{mm} = 4,54\frac{arcsec}{pix}. \tag{6.3}$$

6.1.2 Steering Range

The used digital-to-analogue converters can supply a maximum voltage of 2.5V. The PDU150L has a gain of 20 resulting in a maximum output voltage of 50V. Since the piezo actuators can be supplied up to 150V, only 1/3 of the actuator range is used in this setup. Assuming the piezos are linear across their voltage range, the resulting maximum displacement range $s_{actuator}$ is given by:

$$s_{actuator} = \frac{830\mu m}{3} \times \frac{1}{2} = \pm 138,3\mu m. \tag{6.4}$$

The point at which the actuator is connected to the mirror assembly is located 32,76 mm from the pivot point, the tip-tilt range of the mirror is therefore given by:

$$\theta_{mirror} = \arctan\left(\frac{138,3 \times 10^{-6}m}{32,76 \times 10^{-3}m}\right) = \pm 4,22 \times 10^{-3} radians. \tag{6.5}$$

As illustrated by figure 6.4, the reflected beam will be deflected by the double of the angle that the mirror has been tilted.

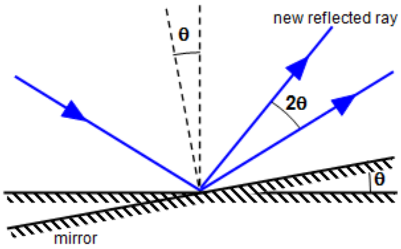


Figure 6.4: Simplified rotating mirror situation [31]

The resultant beam deviation is therefore given by:

$$\phi_{mirror} = 2 \times 4,22 \times 10^{-3} radians = \pm 8,44 \times 10^{-3} radians, \tag{6.6}$$

resulting in a coverage on the image sensor of:

$$d_{sensor} = \tan(8,44 \times 10^{-3} radians) \times 0,16m = \pm 1,35 \times 10^{-3} m. \tag{6.7}$$

The sensor coverage can also be expressed in pixels by using the pixel scale of the CMOS sensor:

$$d_{sensor} = \pm 1,35 \times 10^{-3} m \times \frac{1}{5,5 \times 10^{-6}} \frac{pix}{m} = \pm 245 pix. \tag{6.8}$$

Using the calculated plate scale in 6.3 this can also be expressed in arcseconds:

$$d_{sensor} = \pm 245 pix \times 4,54 \frac{arcsec}{pix} = \pm 1112,3 arcseconds. \tag{6.9}$$

The full steering range is therefore 2224,6 arcseconds which is well above the ADCS attitude uncertainty of 100 arcseconds.

6.1.3 Steering Resolution

The theoretical sensor coverage is in this case 490 by 490 pixels. Considering the DACs have a 16-bit sampled output, the steering resolution R becomes:

$$R_{FSM} = \frac{490 pix}{2^{16}} = 7,5 \times 10^{-3} \frac{pix}{bit}. \tag{6.10}$$

Considering the image scale of the setup calculated in 6.3 this yields:

$$R_{FSM} = 7,5.10^{-3} \frac{pix}{bit} \times 4,54 \frac{arcsec}{pix} = 34 \times 10^{-3} \frac{arcsec}{bit} \tag{6.11}$$

Figure 6.5 illustrates a graphical representation with calculated values of the steering mirror setup.

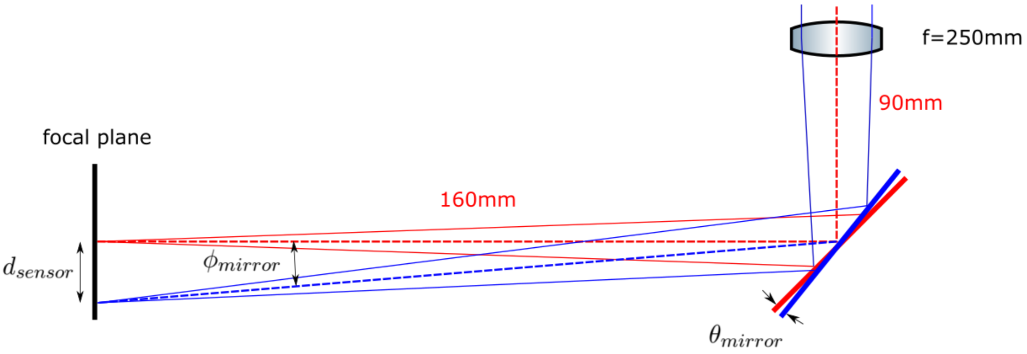
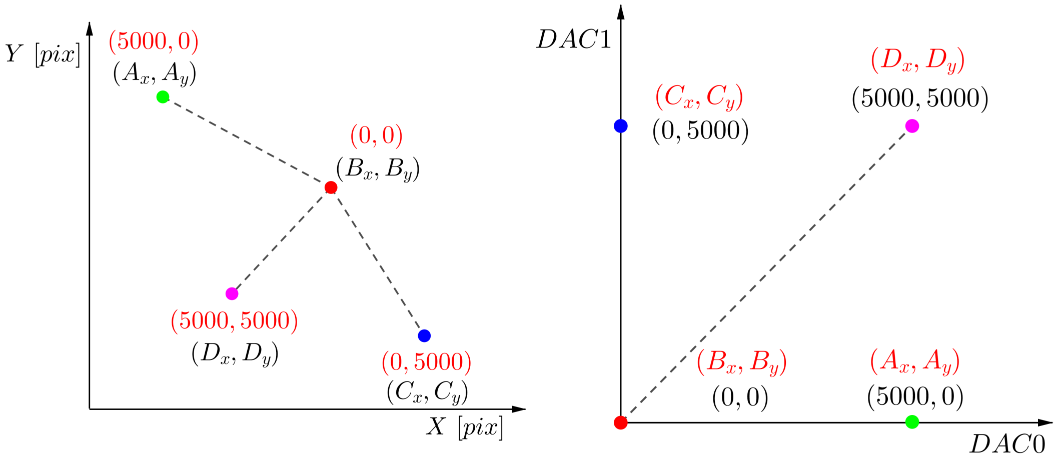


Figure 6.5: Graphical representation of the calculated values

6.1.4 Fine Steering Mirror Calibration

Due to the tip-tilt configuration of the steering mirror, the resulting mirror movement of the actuators is not independent of each other, i.e. activating one of the two actuators will result in a movement that has a component in both the horizontal and the vertical direction on the camera sensor. This can be solved by executing a calibration pattern for known actuator positions and then matching the obtained centroids to a Cartesian actuator grid. The obtained transformation matrix then provides translation from the desired Cartesian pixel coordinates to a linear combination of the 16-bit DAC actuator values. Figure 6.6 (a) illustrates the pixel positions of the centroid for different combinations of actuator values (indicated in red). Figure 6.6 (b) illustrates the transformed pixel positions to a Cartesian grid for the DAC values.



(a)

(b)

Figure 6.6: Transformation linking the pixel positions to the corresponding values for the DACs (centroid domain and actuator domain)

To find the corresponding DAC values (DAC_0, DAC_1) for a desired Cartesian pixel position (P_x, P_y), the pixel coordinates need to undergo a geometrical transformation as defined by the calibration sequence. The Cartesian pixel position is equal to a transformation matrix multiplied by the steering mirror actuator settings:

$$\begin{bmatrix} P_x \\ P_y \\ 1 \end{bmatrix} = M \begin{bmatrix} DAC_0 \\ DAC_1 \\ 1 \end{bmatrix}. \tag{6.12}$$

In essence, the terms of the transformation matrix define the mechanical setup. To determine the coefficients of the transformation matrix, a calibration pattern is acquired consisting of four centroid positions for which the actuator values are known. The centroid positions are then mapped to control points, in this case the actuator settings for which the centroids have been recorded. Several functions in OpenCV exist to map data points to a set of control points and calculate the optimal rigid transformation. This technique is widely used in image rectification and transformation.

6.1.5 Steering Mirror Frequency Response

The frequency response can be determined by exciting the system (mirror and actuators) with a range of frequencies and measuring the amplitude and phase of the response at the corresponding input frequency. The image sensor could be used to measure the amplitude of the deviation, however this limits the frequency response plot to half of the framerate of the camera due to the Nyquist theorem. An alternative measurement method has been used and is illustrated in figure 6.7.

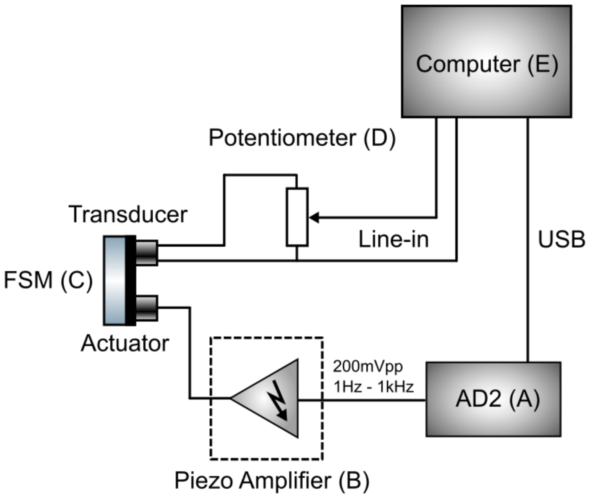


Figure 6.7: Diagram of the setup used to measure the frequency response of the steering mirror

In this setup, one of the actuators is used to excite the steering mirror (C) over a range of frequencies using a sweeping signal generator (A). The sweeping signal generator is connected to the piezo amplifier (B). The other actuator will act as a transducer, converting the vibration of the steering mirror into an electrical signal. The signal from the transducer is sampled by the audio card of a computer (E) at 44.1 kHz, which in turn allows a frequency response measurement of up to 22.1 kHz due to the Shannon-Nyquist theorem.

This method offers high resolution as the sampling uses 24 bits to digitise the transducer signal. Since the transducer is a piezo actuator that is directly attached to the mirror, the sampled signal is therefore directly related to the mechanical movement of the steering mirror. In order to prevent exceeding the maximum voltage of the audio card, the transducer signal is connected to a potentiometer (D) of which the wiper is connected to the line-in of an audio card. This method allows a good characterization with relation to the response flatness and eigenfrequencies of the system. However it does not provide information about the phase or delay of the system. A photograph of the setup is shown in figure 6.8. The results of the frequency response are discussed in the next chapter.

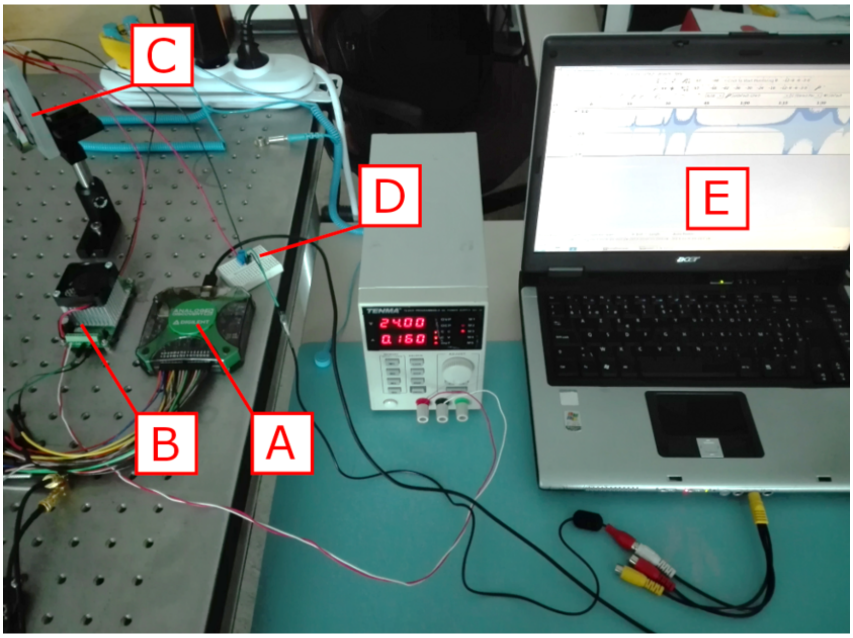


Figure 6.8: Photograph of the frequency response measurement setup showing the sweeping waveform generator (A), piezo amplifier (B), the steering mirror (C), the potentiometer (D) and the laptop with recording software (E)

6.2 Control Loop Implementation

In order to create a control system, the electro-optics need to be able to communicate with the electromechanics. The majority of the processing chain and control-loop has been implemented in Python together with OpenCV due to the open source nature, good support and available documentation. The interfacing to the XIMEA camera is done via the Ximea Python API (xiAPI) [41]. Since the correction system will likely be run on an FPGA together with a lightweight Linux version, it is an advantage that the majority of the system has been implemented in a cross-platform language such as Python. This neglects the need of completely re-writing the code for future applications. As indicated on figure 6.9 the control-loop is implemented in software while the actuators, platform and camera sensor are located in the hardware-domain of the setup.

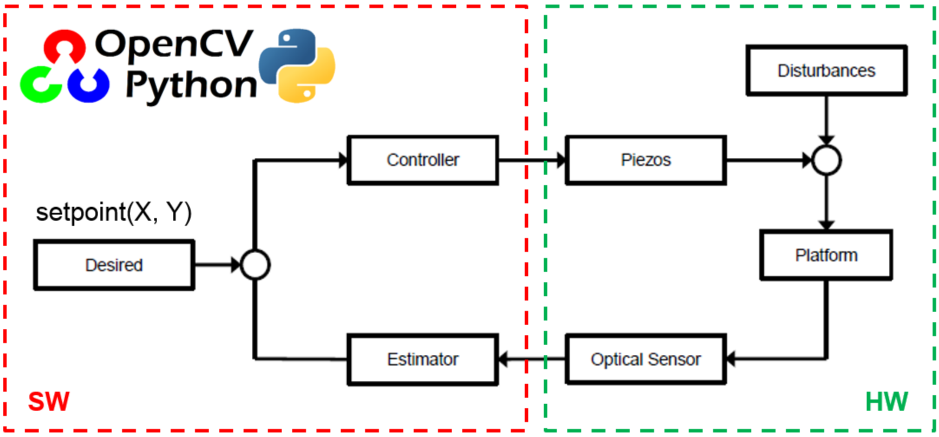


Figure 6.9: Control-loop diagram

The process flow of the program is illustrated in figure 6.10. Firstly the Ximea camera is configured via the XiAPI and a stream is started. The stream aids in setting up and aligning the optics. Focus adjustments can also be monitored live. When the setup and image are satisfactory, the user presses CTRL+C. This initiates the calibration sequence. The calibration sequence moves the steering mirror to four positions and uses the corresponding actuator values to calculate the transformation matrix that is stored in the program. If successful, the user can press enter to start the control loop. During each step of the control-loop, the parameters are logged to a comma-separated values (CSV) file. These parameters include the frame number, the setpoints for X and Y, the X- and Y-response and finally the actuator settings sent to the mirror in each step. This makes analysis of the performance afterwards more convenient. The control-loop parameters are written to the filename of the .csv file to prevent data confusion. The processes illustrated in grey will be further elaborated in sections 6.2.1 and 6.2.2.

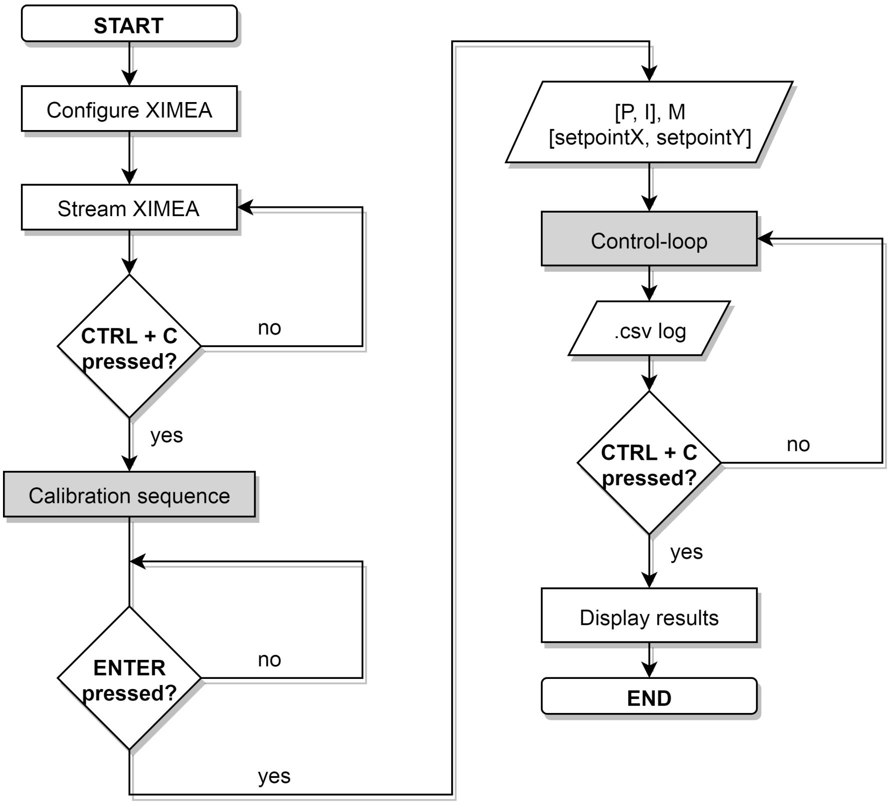


Figure 6.10: Flowchart of the main program

6.2.1 Calibration Sequence

The calibration sequence moves the steering mirror to the four positions and takes a frame on which centroiding is performed. The calculated centroids, together with the known actuator values are then used to calculate the transformation matrix M in OpenCV which is then locally stored. This process is illustrated in a flowchart in figure 6.11.

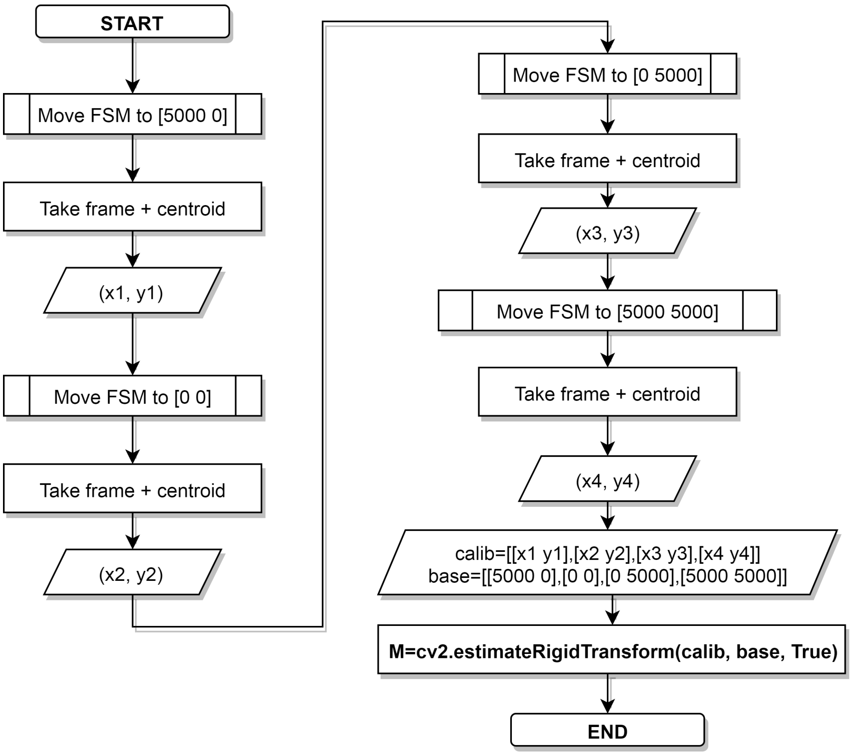


Figure 6.11: Flow of the calibration algorithm

6.2.2 Control-loop

Once the calibration has been completed and the transformation matrix calculated, the control loop can be started. Prior to the start of the control-loop, the software calculates a setpoint based on the steering mirror actuating range to ensure maximum steering range within the ROI. Figure 6.12 illustrates only a single control-loop sequence since the loop is closed in figure 6.10. The frames are first thresholded to reduce the impact of background noise within the ROI. The centroids are then calculated and the errors are determined by subtracting the centroids from the setpoints. The errors are then fed to a discrete PI-controller. The output from the PI-controller in the centroid domain is then converted to the actuator domain via the transformation matrix and the calculated 16-bit values sent to the steering mirror.

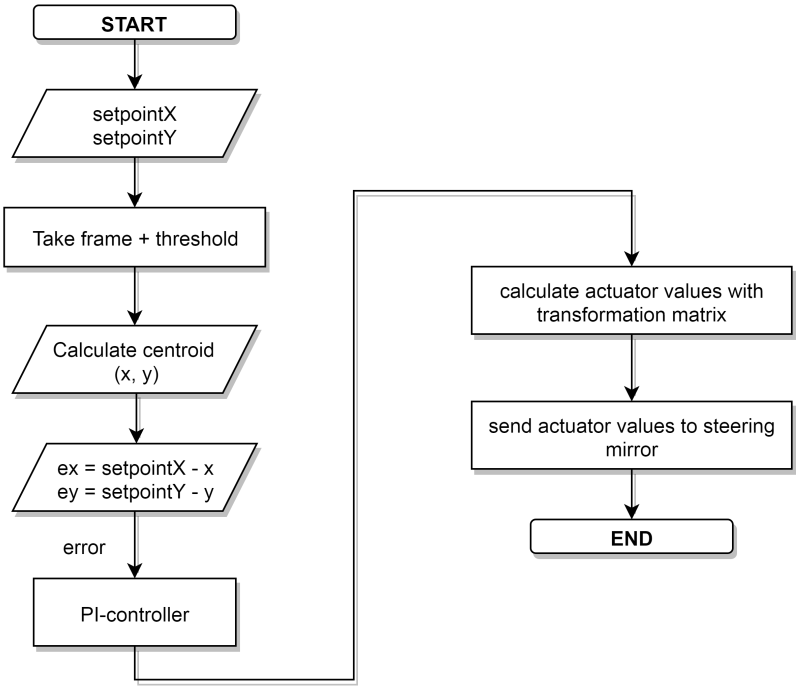


Figure 6.12: Flow of the control

When the control-loop is initiated, the live video stream is started and the control errors are displayed. Figure 6.13 illustrates the control-loop in operation.

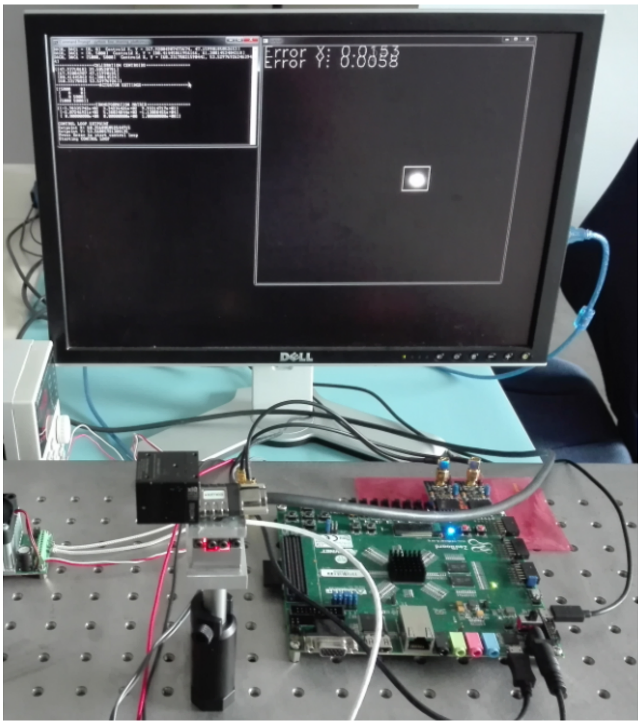
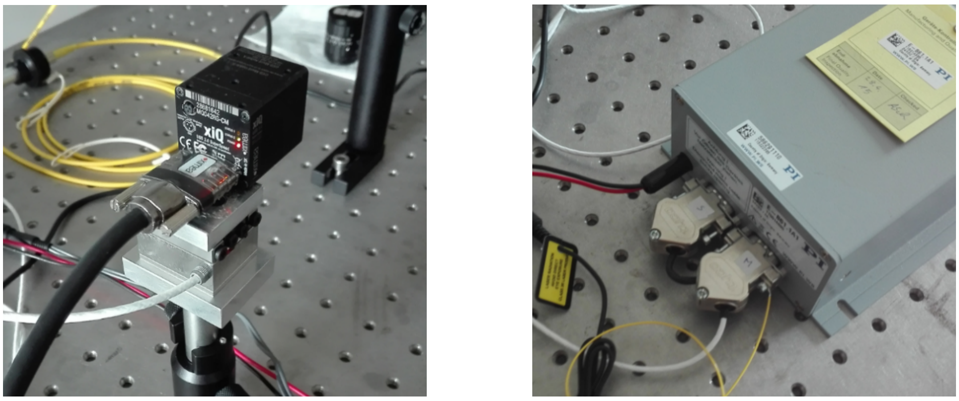


Figure 6.13: Software interface during the control-loop process

6.3 Control Loop Disturbance Injection

The disturbance attenuation for a certain frequency can be measured by applying the disturbance with that particular frequency and certain amplitude, and observing the resulting error in the control loop. The ratio of output amplitude and input disturbance amplitude results in the loop suppression of that particular disturbance. In order to apply a well controllable disturbance to the control loop, another piezo actuator has been used. The used actuator is a linear walking piezo type with a high range yet precise resolution by means of an optical feedback system. The Ximea camera has been mounted to the translational stage as illustrated in figure 6.14 (a). The controller for the piezo stage is illustrated in figure 6.14 (b).

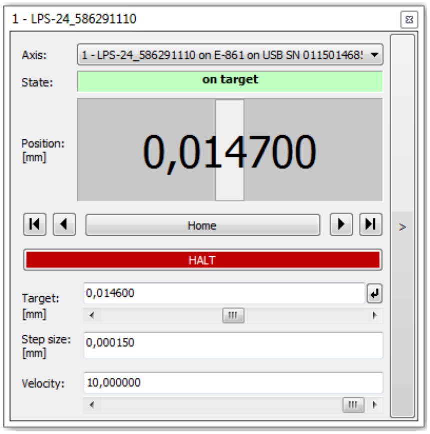
The E-861 controller can be interfaced with the existing PI MikroMove software. The software for the E-861 however does not support the waveform generator function. Therefore a custom macro has been written that can be uploaded to the controller. The macro that produces a sinewave has been generated using a Python script. The script requires the desired frequency, amplitude and number of samples per period. Based on these inputs, the Python script generates a macro file with the corresponding syntax that is compatible with the E-861 controller. Once the macro is uploaded to the controller, it can be run in loop mode resulting in a continuous sinewave motion of the linear stage. For setting an initial reference position, the LPS24 single stage interface illustrated in figure 6.15 can be used.



(a)

(b)

Figure 6.14: Ximea camera mounted to the linear actuator (a) and controller from Physik Instrumente (b)



1 - LPS-24_586291110

Axis: 1 - LPS-24_586291110 on E-861 on USB SN 011501468!

State: on target

Position: [mm] 0,014700

Target: [mm] 0,014600

Step size: [mm] 0,000150

Velocity: 10,000000

Figure 6.15: LPS24 stage control window showing the current relative position in μm units

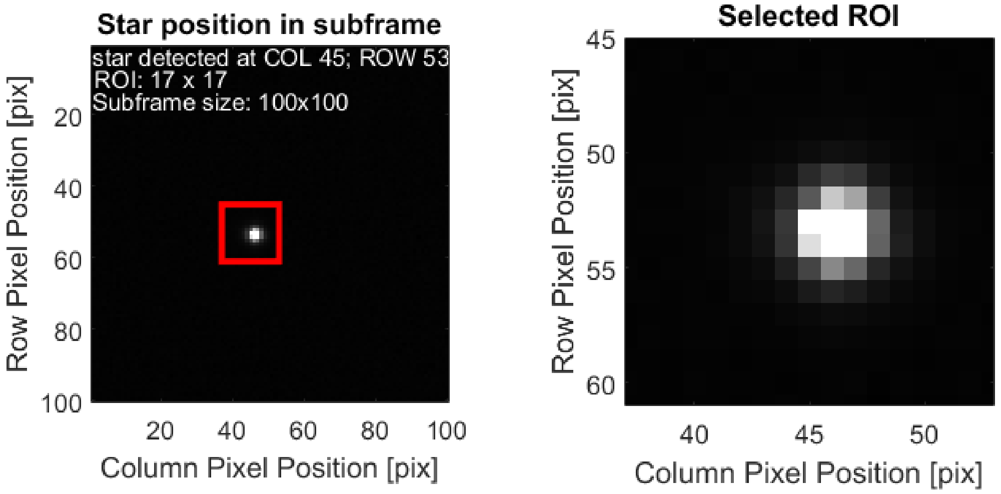
Chapter 7

Results

This chapter presents the obtained data from the various tests of which the setups have been illustrated in the previous chapter.

7.1 Centroiding Error

For this test the frames are captured by the XIMEA camera in the form of uncompressed TIF image stacks. A MATLAB script has been written in order to perform centroiding on the captured frames. The subframes are 100 by 100 pixels in size. The first frame is used to find the pixel with the highest intensity, this is considered to be the star in the frame. Figure 7.1 illustrates the results from this process.



(a)

(b)

Figure 7.1: Detection of the brightest pixel in the subframe (a) and the selected ROI (b)

A region of interest (ROI) of 17 by 17 pixels is selected around the brightest pixel.

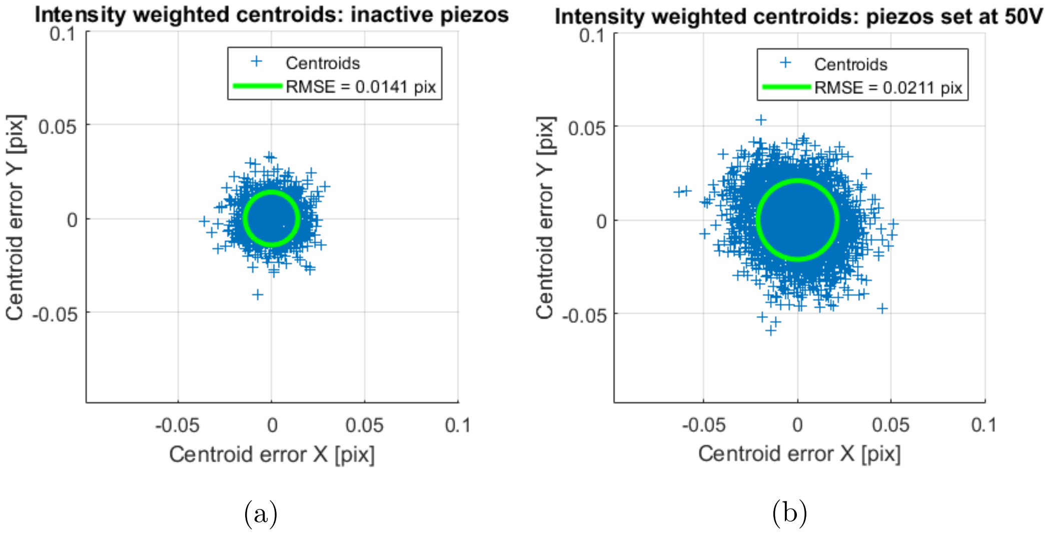
For each frame, the centroiding is performed within the selected ROI. The centroids are calculated via the center of gravity method illustrated by equation 7.1 [7]:

$$(x_e, y_e) = \left(\frac{\sum_{ij} I_{ij} x_{ij}}{\sum_{ij} I_{ij}}, \frac{\sum_{ij} I_{ij} y_{ij}}{\sum_{ij} I_{ij}} \right). \tag{7.1}$$

The true centroid (x_t, y_t) can be estimated by taking the mean of all X-centroids and the mean of all Y-centroids. Relative centroid errors are then calculated by subtracting the true centroid from the absolute centroids. The total centroid error on frame k is then given by the Euclidean distance between the true centroid and the calculated centroid:

$$e_k = \sqrt{(x_e^k - x_t)^2 + (y_e^k - y_t)^2}. \tag{7.2}$$

Figure 7.2 illustrates the centroid errors without and with active piezos. An increase in relative RMS centroiding error can be observed when the piezos are set at a fixed voltage of 50 V. The electrical noise on the amplifier inputs is causing an extra centroiding error to be added to the original error since the steering mirror now is subjected to very small vibrations. Depending on the mechanical orientation of the steering mirror, the extra X and Y-components will have a different contribution to the total centroiding error.



(a)

(b)

Figure 7.2: Relative centroiding error with deactivated piezos (a) indicating 0.014 pix RMSE and with piezos set at 50V (b) indicating 0.021 pix RMSE

The RMS centroiding error with the steering mirror disabled is 0.014 pixels or 0.078 arcseconds. Upon activation of the steering mirror, the centroiding error increases to 0.021 pixels or 0.116 arcseconds.

7.2 Calibration Algorithm

The calibration algorithm moves the steering mirror to 4 positions and records the corresponding centroid position. Figure 7.3 illustrates the recorded calibration centroids (black) with corresponding actuator values (red).

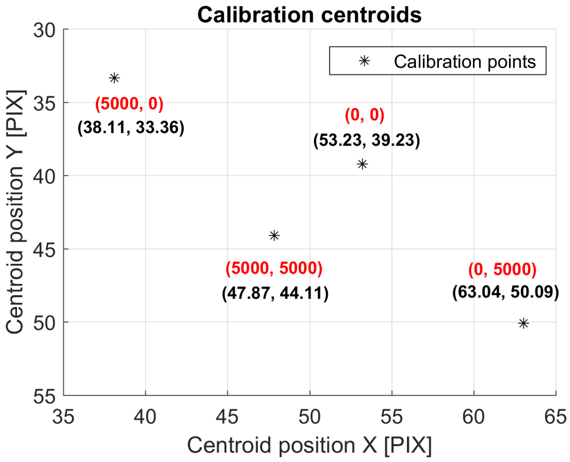
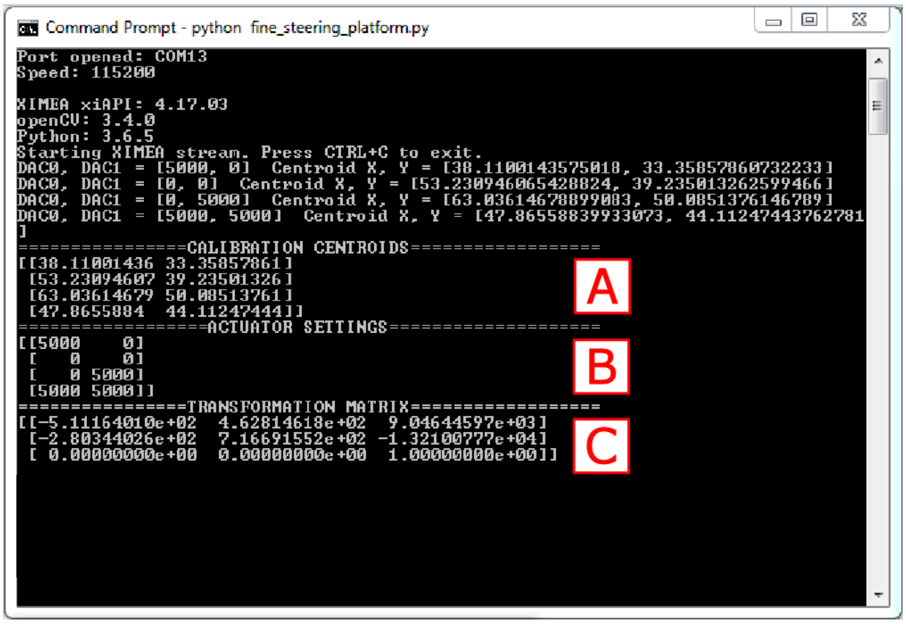


Figure 7.3: Acquired calibration points (black) and corresponding actuator settings (red)

The calibration centroids together with the actuator values are used to calculate the transformation matrix linking the centroid to the actuator domain. In order to reduce hysteresis effects during calibration, the maximum values sent to the DACs is 5000 instead of the full 16-bit value. Figure 7.4 illustrates the software output during calibration. During calibration, the process can be monitored live via the command terminal output. The measured centroids (A) are displayed as the mirror moves to the four positions. The actuator settings (B) are also displayed. The final output after calibration is the transformation matrix M (C).

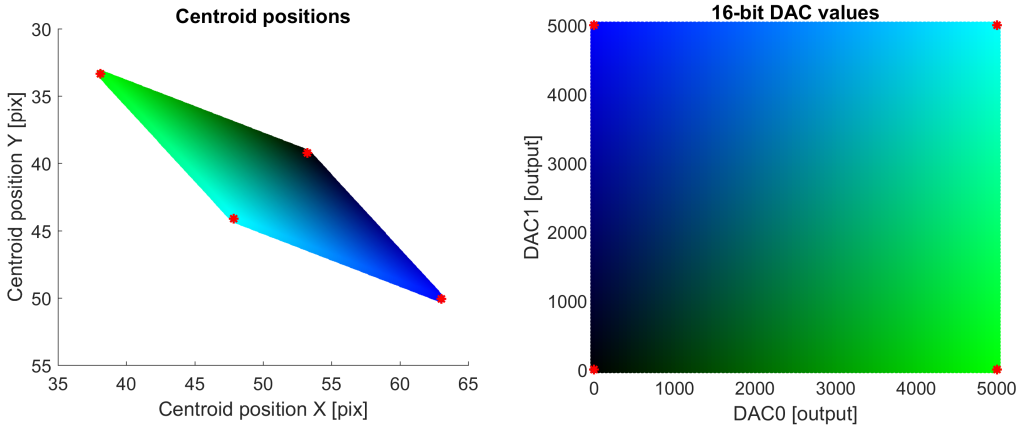


```
Port opened: COM13
Speed: 115200

XIMEA xiAPI: 4.17.03
openCV: 3.4.0
Python: 3.6.5
Starting XIMEA stream. Press CTRL+C to exit.
DAC0, DAC1 = [5000, 0]  Centroid X, Y = [38.1100143575018, 33.35857860732233]
DAC0, DAC1 = [0, 0]  Centroid X, Y = [53.2309460654288824, 39.235013262599466]
DAC0, DAC1 = [0, 5000]  Centroid X, Y = [63.0361467889083, 50.0851376146789]
DAC0, DAC1 = [5000, 5000]  Centroid X, Y = [47.86558839933073, 44.11247443762781
]
================CALIBRATION CENTROIDS==================
[[38.11001436 33.35857861]
 [53.23094607 39.23501326]
 [63.03614679 50.08513761]
 [47.8655884  44.1124744]]
===================ACTUATOR SETTINGS====================
[[5000    0]
 [   0    0]
 [   0 5000]
 [5000 5000]]
================TRANSFORMATION MATRIX==================
[[-5.11164010e+02  4.62814618e+02  9.04644597e+03]
 [-2.80344026e+02  7.16691552e+02 -1.32100777e+04]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

Figure 7.4: Software output during calibration

Figure 7.5 illustrates the result of applying the calculated transformation matrix to the calibration centroids.



(a)

(b)

Figure 7.5: Transformation from pixel domain (a) to actuator domain (b) using the FSM calibration points

The transformation of horizontal and vertical centroid movements is illustrated in figure 7.6.

The transformation of a horizontal path is depicted in green on figure 7.6. For a movement in the X-direction from pixel 1 (49, 43) to pixel 2 (55, 43) for example,

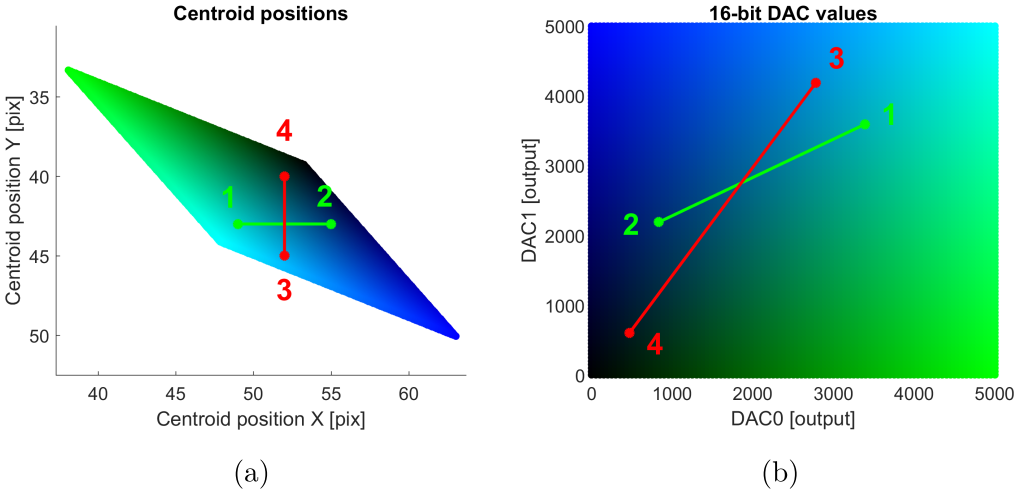


Figure 7.6: Horizontal and vertical movement in the centroid domain (a) linearly transformed to the actuator domain (b)

the transformation matrix indicates how this linear path has to be taken in terms of the steering mirror actuator values. Similarly, for a movement in the Y-direction from pixel 3 (52, 40) to pixel 4 (52, 45) for example, the linear path is illustrated in red on figure 7.6. For any other desired arbitrary path or sequence, the consecutive pixel positions must be transformed and the actuators must be set to each of the corresponding values in the correct order. When the coefficients of the transform have been determined correctly using the calibration sequence, precise Cartesian control over the steering mirror is possible. A test pattern is useful to assess the hysteresis of the actuators and determine the correctness of the transformation. In this case, a square test pattern has been used. Figure 7.7 illustrates the centroids of the test pattern.

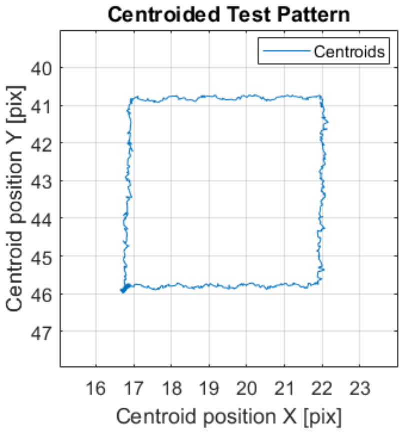


Figure 7.7: Centroided test pattern indicating horizontal and vertical centroid movement

7.3 Steering Mirror Frequency Response

The setup illustrated in 6.1.5 is used to measure the frequency response of the steering mirror. The frequency content of the recorded wav file is analysed using Matlab. A logarithmic frequency plot indicates which frequencies are present in the signal and their corresponding amplitude. In order to get more a reliable measurement, a power spectrum density (PSD) has been acquired without the active sweep, indicating the baseline characteristics of the sampling device. The actual frequency response of the system is determined by subtracting the system baseline from the recorded signal during the sweep as illustrated by formula 7.3. Figure 7.8 illustrates the process of determining the steering mirror frequency response.

$$PSD_{sys} = PSD_{sys+baseline} - PSD_{baseline}. \tag{7.3}$$

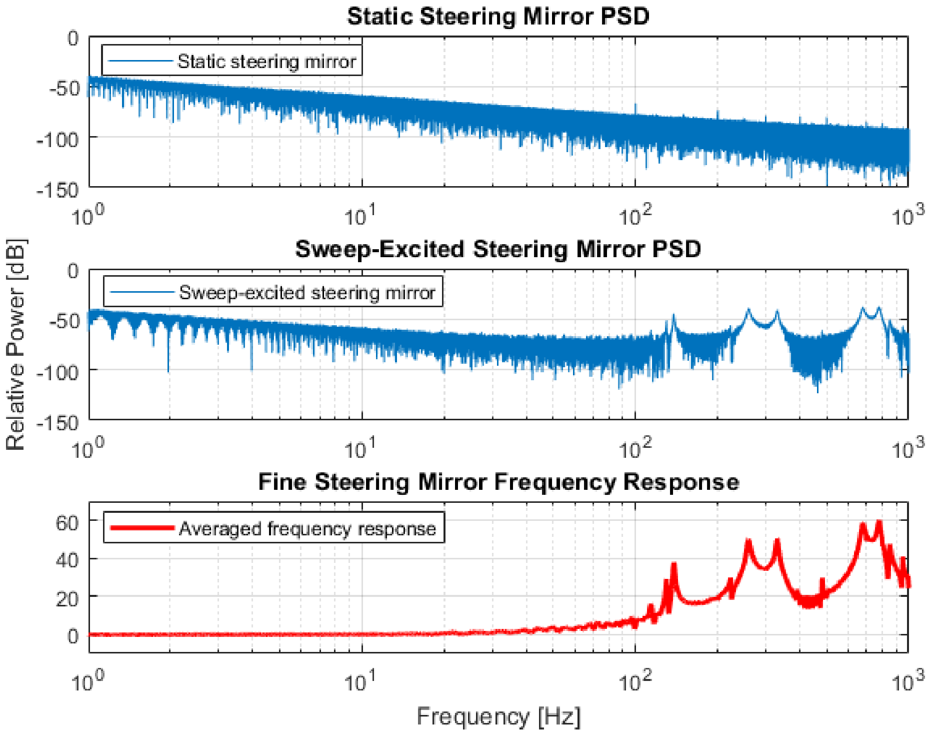


Figure 7.8: Baseline response of the sampling equipment (top), frequency content of the recorded response and baseline response (middle) and resulting steering mirror frequency response (bottom)

A sinusoidal signal with an amplitude of 50 mVpp at the amplifier input has been used to excite the steering mirror over a range of frequencies. The sweeping frequency range was set from 1 Hz to 1 kHz and completed 120 seconds. The sweep duration was set long enough to allow sufficient time for the mirror to adapt to each frequency and provide the corresponding output amplitude. A measurement has been made where one channel acts as the actuator, and the other channel as the transducer. In a second measurement the actuator and the transducer channels were interchanged. Both measurements indicated an identical frequency response. Figure 7.9 illustrates a close-up view of the frequency response. For frequencies beneath 40 Hz, the +3 dB point is not reached. Therefore the mirror is usable until atleast 40 Hz without an excessive response in amplitude.

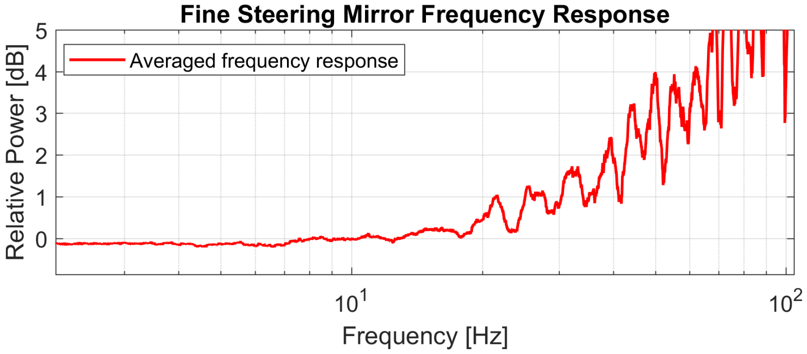
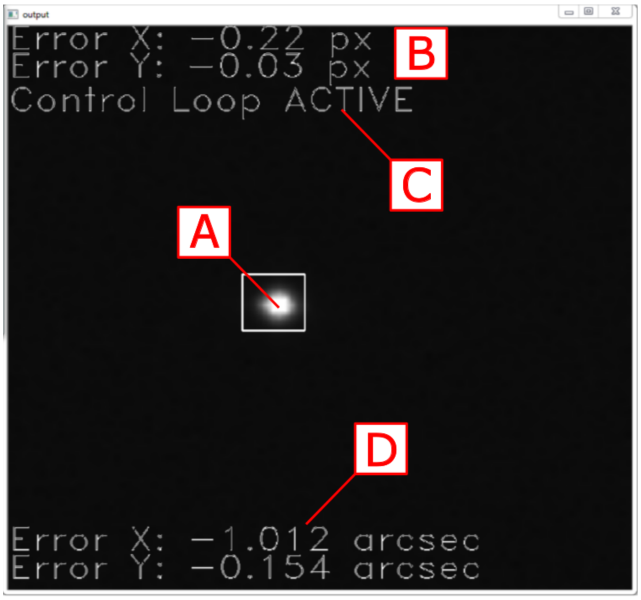


Figure 7.9: Close-up of the steering mirror frequency response

7.4 Control Loop Performance

The control loop performance is assessed by the system's ability to keep the target star (A) on the setpoint. The system features a live interface on which the status of the control-loop can be monitored. Basic information such as the loop status (C) and the loop error in pixels (B) and arcseconds (D) are displayed. The videostream also aids in focusing and aligning the optics. Figure 7.10 illustrates the interface of the control loop.



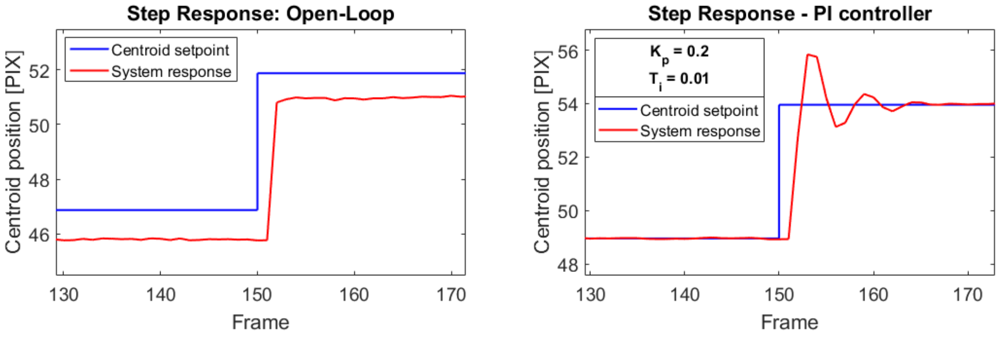
```
Error X: −0.22 px
Error Y: −0.03 px
Control Loop ACTIVE

Error X: −1.012 arcsec
Error Y: −0.154 arcsec
```

Figure 7.10: Interface of the control-loop showing the target star (A), the control-loop error in pixels (B), the loop status (C) and the loop error in arcseconds (D)

7.4.1 Stability

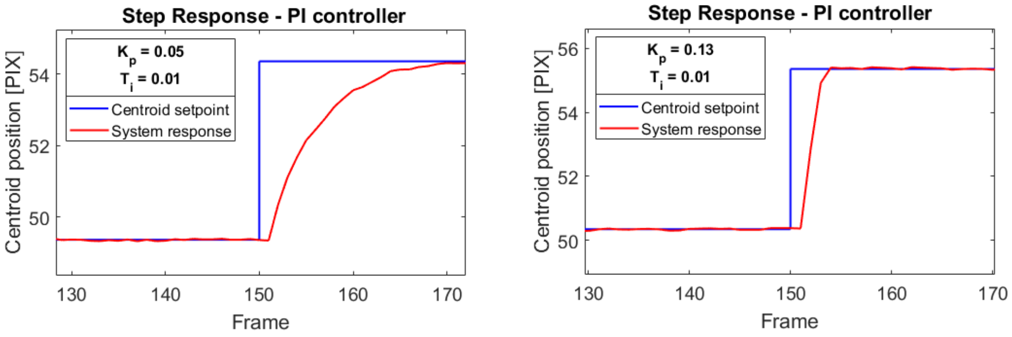
Firstly, the stability of the control loop was assessed using a step response. The step has been implemented in software by incrementing the X-setpoint by 5 pixels at timestep 150. Figure 7.11 (a) illustrates the open-loop step response of the system. In open-loop there is no feedback and this results in a steady-state error that can clearly be seen. In figure 7.11 (b) the loop is closed using a PI-controller set at $K_p = 0.2, T_i = 0.01$. Due the infinite DC-gain caused by the integrator, the steady state error becomes zero. However the gain is too large resulting in a significant amount of overshoot and therefore a long settling time.



(a) (b)

Figure 7.11: Step response in open loop (a) and step response in closed loop with a PI-controller set at $K_p = 0.2$ and $T_i = 0.01$ (b)

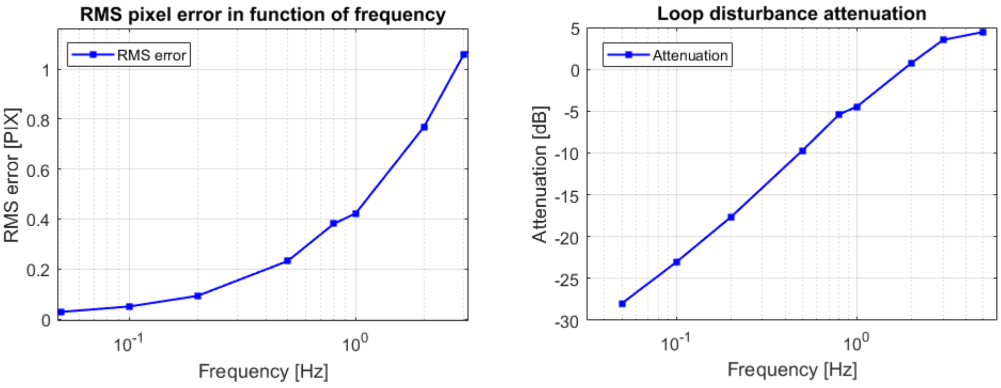
By adjusting the gain or P-action of the PI-controller, the step response can be optimised. Figure 7.12 illustrates the tuning of the P-action. A small amount of overshoot is acceptable since the disturbances will in most cases be characterised by a slow drift instead of sudden changes. Therefore the I-term of the PI-controller is more crucial. A loop sampling rate of 30 Hz has been used throughout all the measurements. During a measurement all the relevant data is logged to a comma-separated values (CSV) file that can later be analysed using Python or Matlab.



(a) (b)

Figure 7.12: Step response with PI controller with parameters $K_p = 0.05$ and $T_i = 0.01$ (a) and with PI-controller parameters $K_p = 0.13$ and $T_i = 0.01$ (b)

Disturbances with an amplitude of 1 pixel have been introduced via the translating piezo to the camera and the corresponding RMS control error on the detector has been measured. Furthermore for various frequencies the loop attenuation has been calculated and plotted logarithmically. The results are illustrated in figure 7.13.



(a)

(b)

Figure 7.13: RMS-error for a disturbance with amplitude 1 pixel in function of frequency with PI-controller parameters $K_p = 0.13$ and $T_i = 0.01$ (a) and disturbance attenuation plot (b)

7.5 CUBESPEC Requirement

The scientific payload on CUBESPEC requires a target star to be kept within a 3.6 arcseconds slit at least 80% of the time. In other words, the resulting relative centroid errors must reside between -1.8 and +1.8 arcseconds of the setpoint for at least 80% of the time. The attitude control uncertainty of the ADCS is in the range of 100 arcseconds. According to equation 6.3 this corresponds to a deviation on the guidance sensor given by:

$$A_{dist} = \frac{\pm 50 arcsec}{4,54 \frac{arcsec}{pix}} = \pm 11 pix. \tag{7.4}$$

To simulate an ADCS drift with an uncertainty of 100 arcseconds a disturbance has to be applied to the camera with an amplitude of 11 pixels. In the first part of the measurement the control-loop is not closed. Therefore the disturbance is directly visible on the camera sensor. Halfway in the measurement, the control-loop is closed and the residual errors are plotted. This is illustrated in figure 7.14.

The closed loop response data in the second half of the measurement are converted to relative errors by subtracting the setpoint from the centroids. The distribution of the closed-loop centroid errors are illustrated using a boxplot in figure 7.15.

Figure 7.15 illustrates that the pointing requirement of 3.6 arcseconds or within -1.8 and +1.8 arcsecond is met for atleast 50% of the time. Using Matlab, the number of

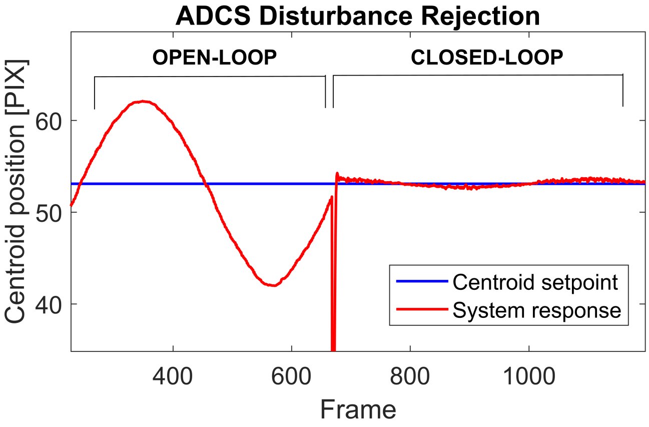


Figure 7.14: Difference between open-loop and closed-loop control when a 0.05 Hz, 22pix p-p disturbance (100 arcseconds) is applied using the translating piezo ($K_p = 0.05$ and $T_i = 0.005$).

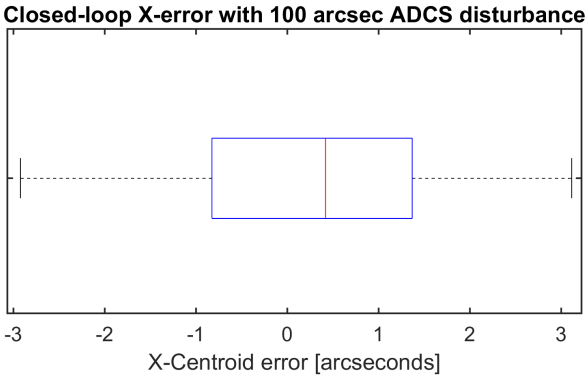


Figure 7.15: Boxplot illustrating the residual errors in arcseconds when a 100 arcsecond disturbance at 0.05 Hz is applied to the guidance sensor. Control-loop parameters: $K_p = 0.05$ and $T_i = 0.005$

X-centroids between -1.8 and + 1.8 arcseconds were counted and compared to the total amount of samples, yielding the time percentage t_{req} the requirement is met. This has been done for various PI-controller parameters. The result is illustrated in table 7.1.

Table 7.1: Percentage of the time the CUBESPEC pointing requirement is met for a 100 arcsecond disturbance at 0.05 Hz in function of PI-control parameters K_p and T_i

PI-parameters		
K_p	T_i	t_{req} (%)
0.13	0.01	71.1
0.08	0.008	74.6
0.05	0.005	82.3

The results indicate that by increasing the P-action of the controller in an effort to make the system faster has a negative effect on the pointing performance. Since the expected disturbances are mostly characterized by a slow drift, the I-action of the PI-controller is more crucial, as evident by the measurements. However the I-action can not be made too small since this will result in a noisy control loop introducing high-frequency jitter. To compensate fore the possible instability, upon decreasing the I-term, the P-term must also be decrased to avoid instability. This makes the system very responsive to drift and steady state errors, however it becomes inherently slow when sudden changes are applied to the control-loop. This is a control compromise that has to be taken into account during further development.

To determine the loop bandwidth, a number of disturbances are applied with a fixed amplitude A_{dist} and varying frequency f_{dist}. For each of the disturbances the amplitude of the control loop error A_{error} is measured. The loop attenuation a_{loop} at frequency f_{dist} is then given by:

$$a_{loop}(f_{dist}) = 20 \times \log\left(\frac{A_{error}(f_{dist})}{A_{dist}}\right) \tag{7.5}$$

Disturbances with an amplitude of 5 pixels have been applied with frequencies between 0.02 Hz and 5 Hz. For each frequency the loop attenuation has been calculated using formula 7.5. When disturbances reach a sufficiently high frequency, the control loop is not capable of suppression anymore and this leads to an amplification in control-error which can be seen in figure 7.16 for frequencies above 2 Hz.

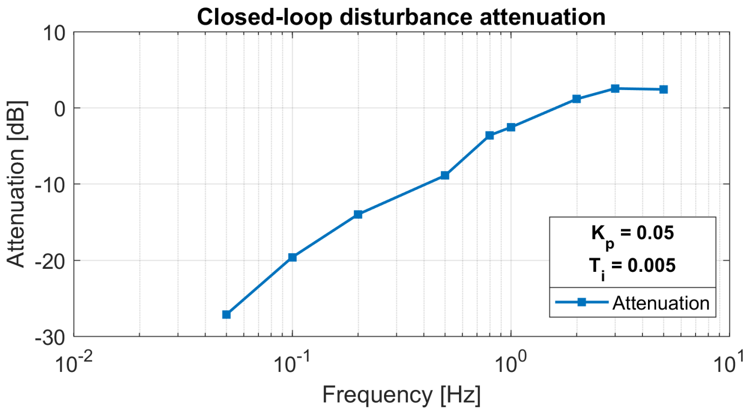


Figure 7.16: Disturbance attenuation for various frequencies, PI-controller set at $K_p = 0.05$ and $T_i = 0.005$

Chapter 8

Conclusion and Outlook

This chapter formulates a conclusion based upon the results illustrated in the previous chapters.

8.1 Conclusion

A proof-of-concept correction system has been implemented and built on an optical bench that resembles the CUBESPEC configuration. For initial verification and testing, the platform was integrated with Matlab and LabVIEW through a serial interface. Afterwards the the frame capturing and centroiding, as well as the control loop have been implemented in Python and openCV. Low noise 16-bit DACs offer good static and dynamic control over a 2-channel fine steering mirror. By executing the developed calibration algorithm, Cartesian control over the mirror is possible. A periodic calibration is highly useful to compensate for long-term effects such as creep. By using coaxial cables for the analog signals, only a minimal increase in centroiding error has been achieved. Due to the 16-bit sampling of the voltage applied to the piezo amplifier, the steering resolution proved to be well below the centroiding error.

To assess the control-loop performance, various sinusoidal disturbances were introduced with a translating piezoelectric stage on which the guidance sensor has been mounted. The measurements indicate that careful control over the I-action is required, and increasing the P-action has negative effect on the performance. The control-loop bandwidth and disturbance suppression was measured indicating that ADCS disturbances with a frequency below 0.05 Hz can be attenuated sufficiently, meeting the CUBESPEC requirement. Therefore good characterisation of the expected ADCS drift is necessary.

Preliminary work has also been done on interfacing a CMV2000 image sensor with a Zynq-7000 SoC development platform. SPI-controllers for interfacing with the sensor as well as LVDS receivers have been successfully implemented and tested. A deserialiser that converts the serial stream into a parallel datastream has also been implemented and tested as well as the conversion to the AXI4-stream format.

A universal testbed has been built for demonstrating and developing active pointing correction systems on small satellites. The testbed features live monitoring and control-loop parameter adjustment to obtain an optimal control-loop bandwidth.

8.2 Outlook

A 3-channel actuated steering mirror is currently in development. This will also allow some degree of electronic focussing since the entire mirror assembly can be moved up or down. The ZedBoard can also support three DACs since there are adequate PMOD connectors available for interfacing. Moreover, the used piezo amplifier also supports three independent channels. In order to utilise the full range of the piezoelectric actuators, low-noise operational amplifiers (OPAMPS) could be placed between the DACs and the piezo amplifier. The steering resolution is well below the centroiding error, loss in resolution due to amplification can therefore be afforded. By amplifying the voltage coming from the DACs with a factor of 3 the full range of the piezos can be utilised while still maintaining a steering resolution that is below the centroiding error. In a later development stage, the setup could also be configured to simulate various stellar magnitudes and assess the corresponding system performance. By placing a continuously adjustable diaphragm in the uncollimated beam, the intensity of the source can be adjusted. The improved setup is illustrated in figure 8.1.

Simulations have been carried out that estimate the centroid error on the fine guidance sensor in function of stellar magnitude (figure 8.2). In these estimations, the light loss due to the optical elements of the telescope have also been taken into account [28]. When the exposure time is too short, the signal-to-noise ratio on the detector will not be sufficient for an accurate centroid determination. A longer exposure time yields a better centroiding accuracy but will limit the loop sampling rate, decreasing the control-bandwidth of the system.

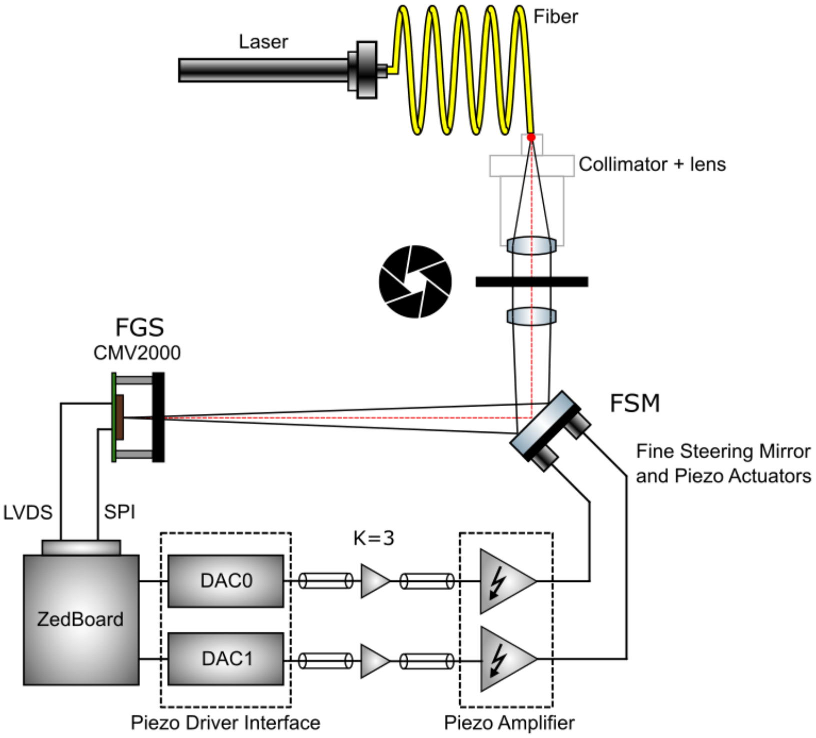


Figure 8.1: Optimised setup with intensity adjustment and full piezo range usage

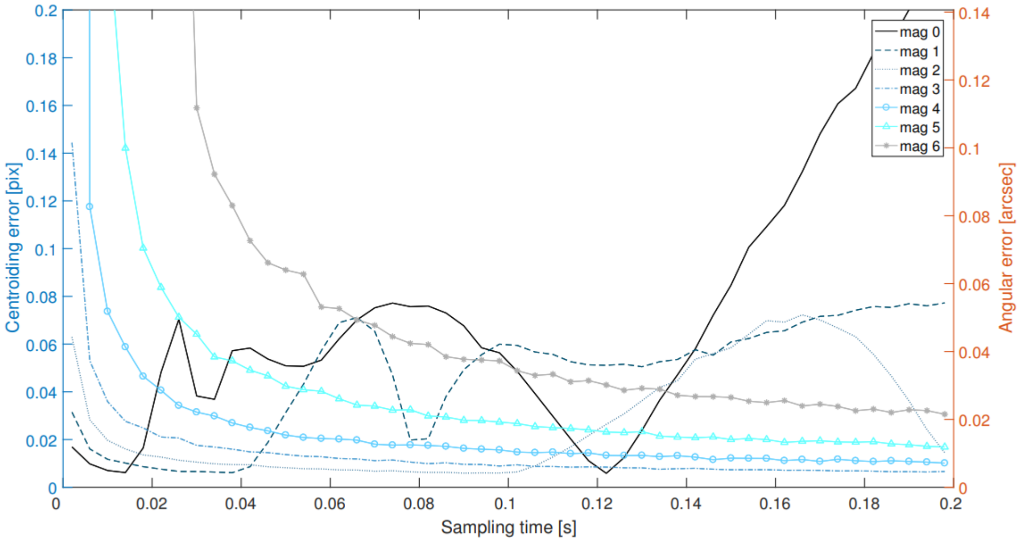
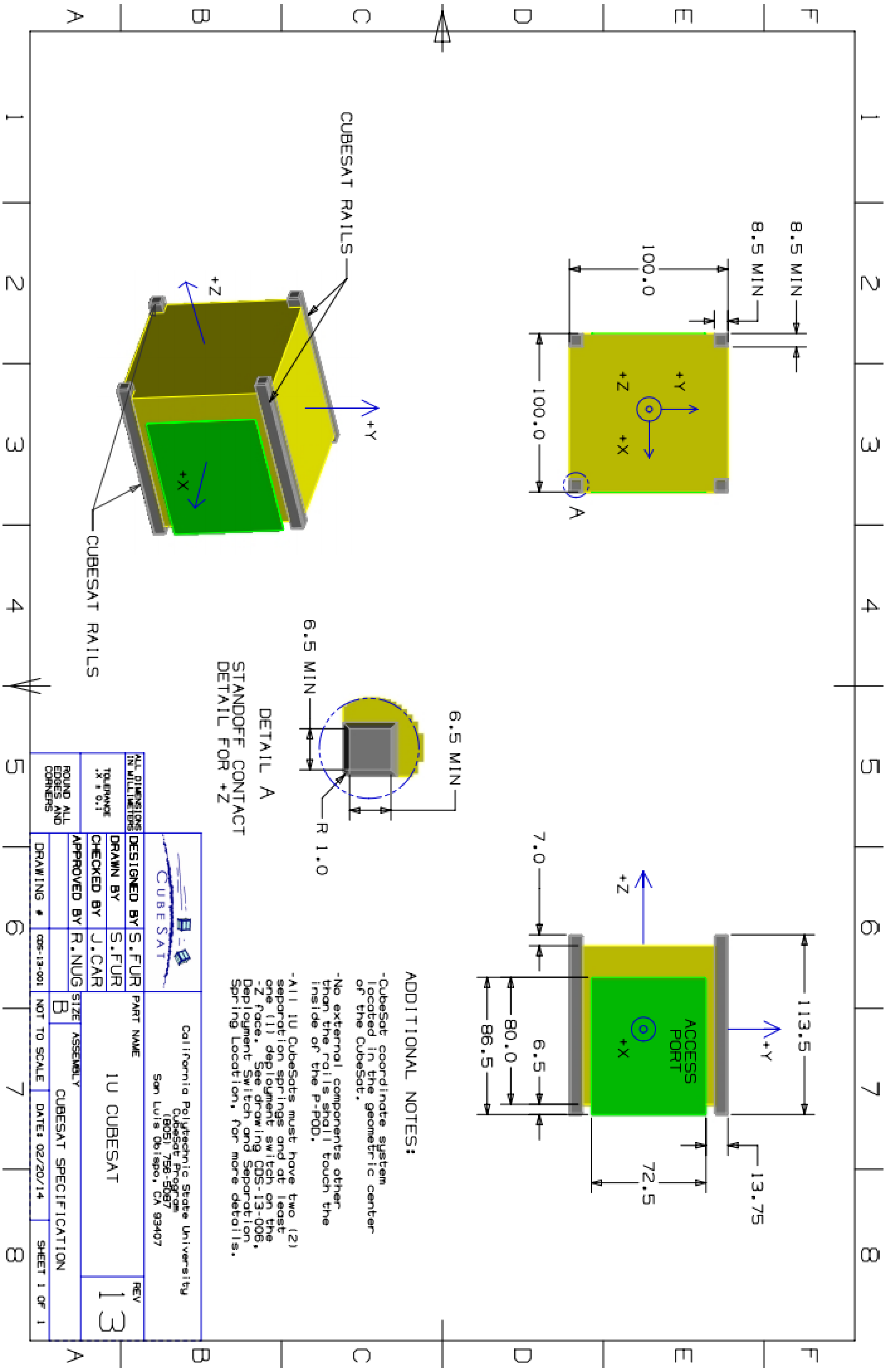


Figure 8.2: Centroiding error in function of guidance sensor sampling time [28]

Appendices

Appendix A

CubeSat Design Standards



ADDITIONAL NOTES:

-CubeSat coordinate system located in the geometric center of the CubeSat.

-No external components other than the rails shall touch the inside of the P-POD.

-All 1U CubeSats must have two (2) separation springs and at least one (1) deployment switch on the -Z face. See drawing CDS-13-006, Deployment Switch and Separation Spring Location, for more details.

Figure A.1: Dimensions of a 1U CubeSat [2].

Appendix B

CubeSat Deployer Standards

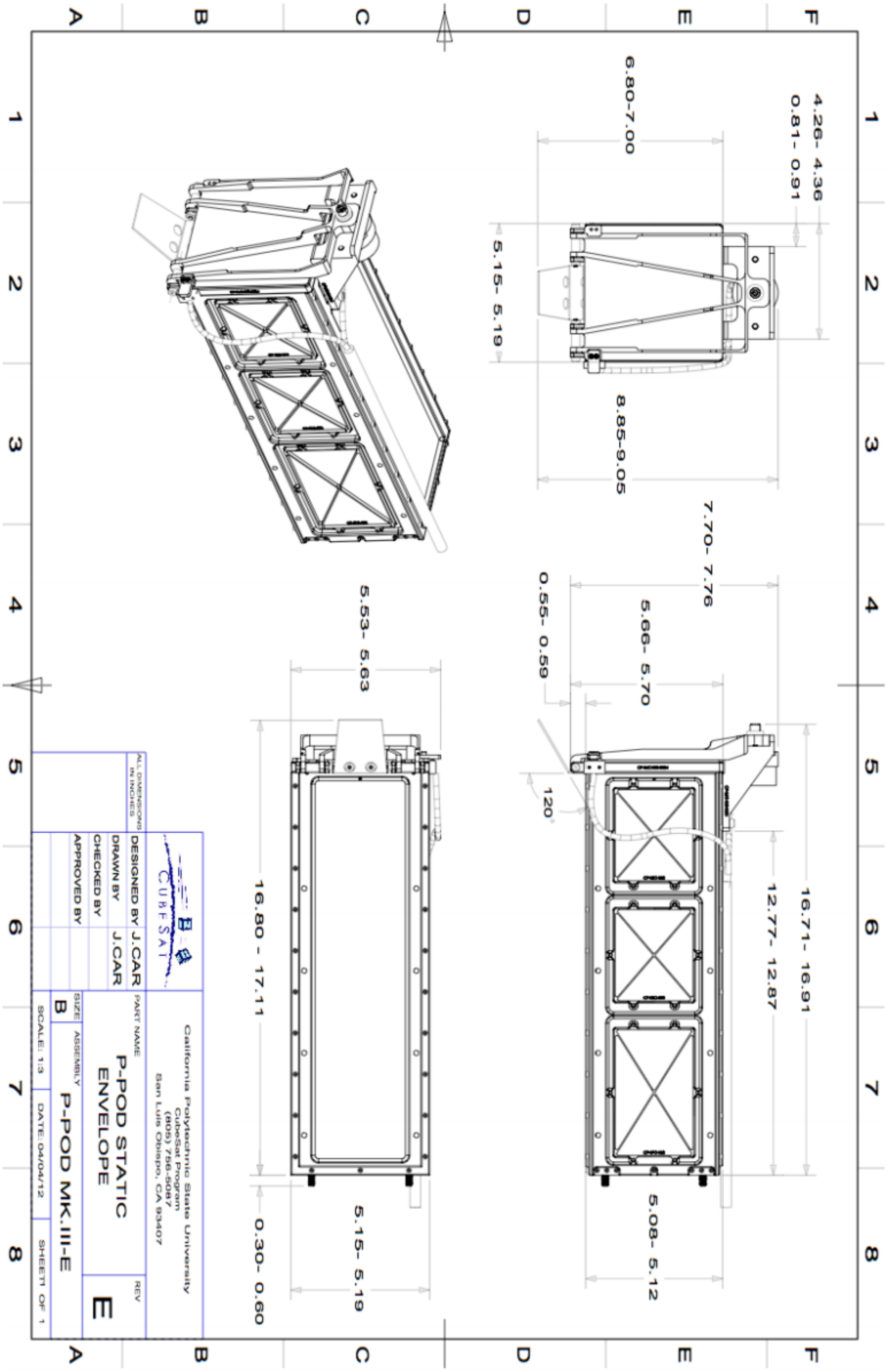


Figure B.1: P-POD Static Envelope Drawing [21].

Appendix C

Schematics

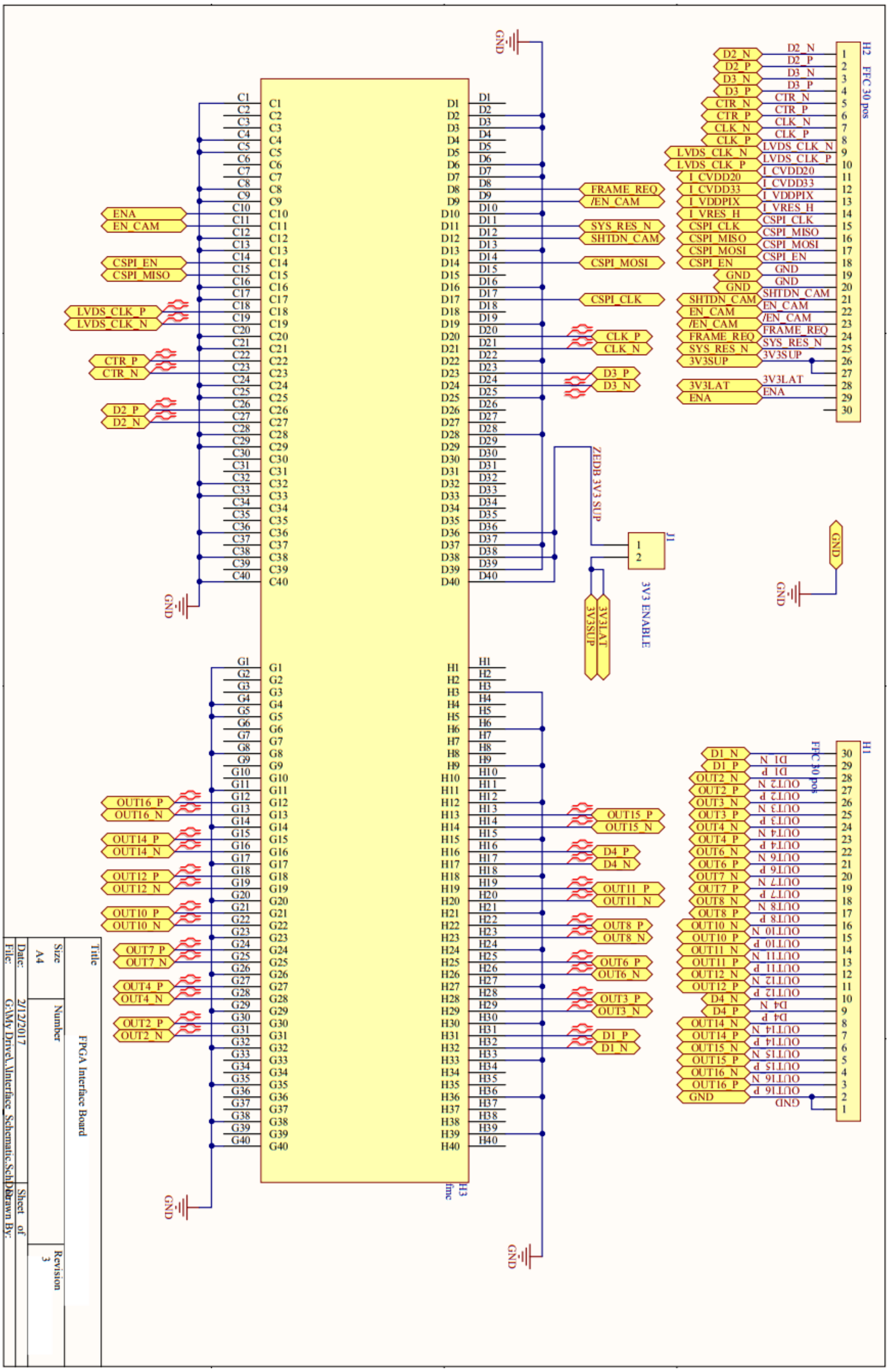
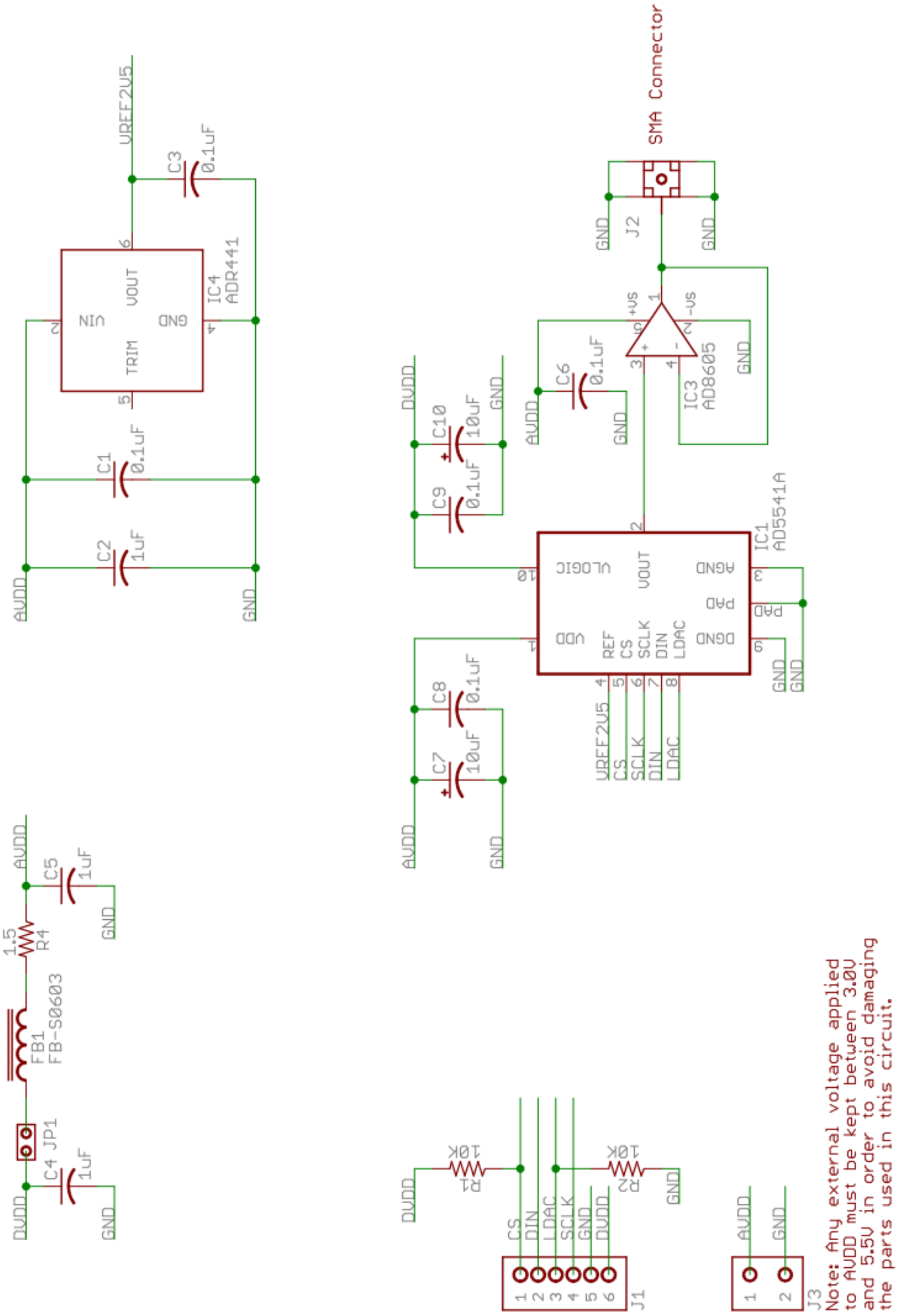


Figure C.1: FPGA Interface Board Schematic



Note: Any external voltage applied to AVDD must be kept between 3.0V and 5.5V in order to avoid damaging the parts used in this circuit.

Figure C.2: Digilent 16-bit DAC board schematic [10]

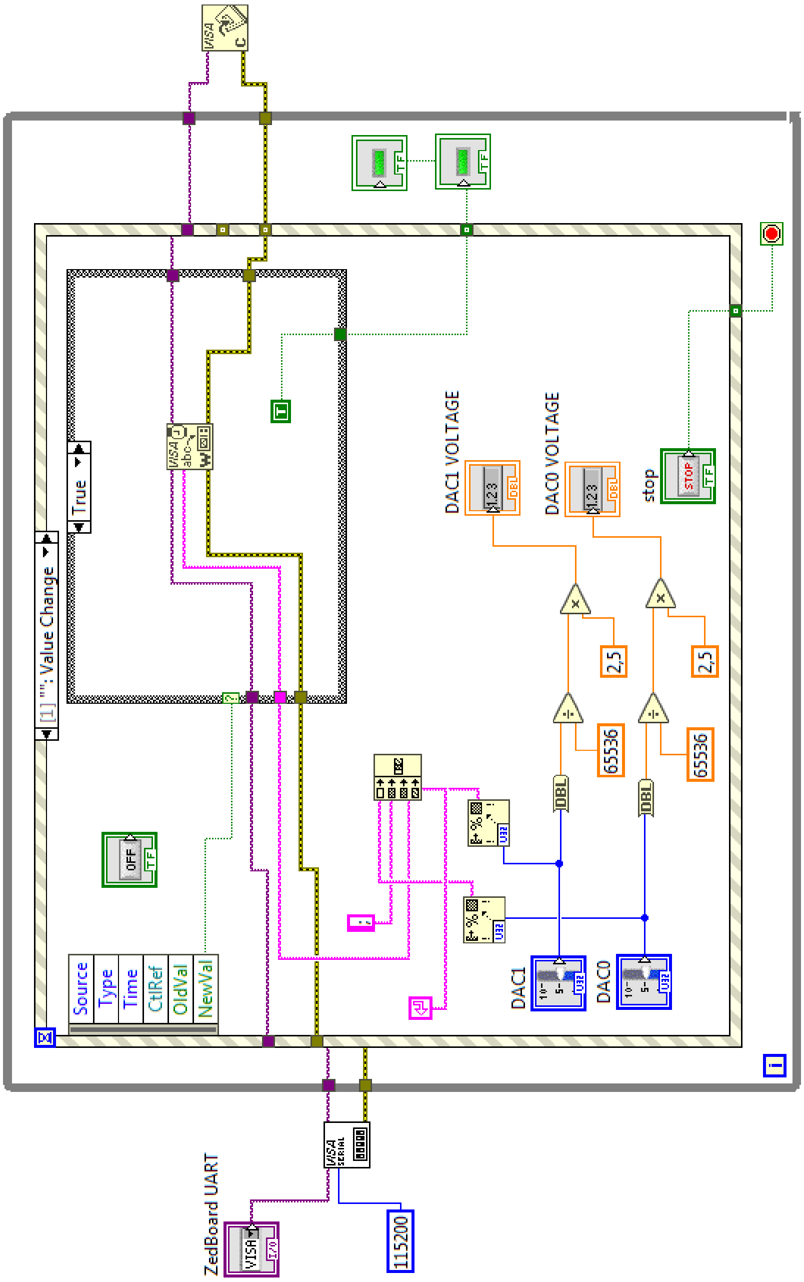


Figure C.3: Piezo DAC LabVIEW user interface diagram

Bibliography

[1] Valerio Bozza, Luigi Mancini, and Alessandro Sozzetti. *Methods of Detecting Exoplanets*, volume 428. 2016.

[2] CalPoly. Cubesat design specification. *The CubeSat Program, California Polytechnic State . . .*, 8651(June 2004):22, 2009.

[3] CMOSIS BVBA. CMV2000 Image Sensor Datasheet - CMOSIS. Technical report, 2015.

[4] Farshid Faal Concordia. Fault Detection , Isolation and Identification of Formation Flying Satellites using Wavelet- Entropy and Neural Networks. *Concordia University Montreal*, (April 2013), 2017.

[5] Cubesatkit.com. Mai-200 Adacs. (April):1–11, 2011.

[6] Observatoire de Paris. Unravelling the Beta Pictoris System. Available at `https://picsat.obspm.fr/home?locale=en`.

[7] Tjorven Delabie. Star Tracker Algorithms and a Low-Cost Attitude Determination and Control System for Space Missions. (January):187, 2016.

[8] Norman C. Deschamps, C. Cordell Grant, Daniel G. Foisy, Robert E. Zee, Anthony F.J. Moffat, and Werner W. Weiss. The BRITE space telescope: Using a nanosatellite constellation to measure stellar variability in the most luminous stars. *Acta Astronautica*, 65(5-6):643–650, 2009.

[9] Digilent. PmodDA3 Reference Manual. pages 2–3, 2015.

[10] a National Instruments Company Digilent. Pmod DA3 Datasheet. Available at `https://reference.digilentinc.com/_media/reference/pmod/pmodda3/pmodda3_sch.pdf`.

[11] Tom Dzamba, John Enright, Doug Sinclair, and Kofi Amankwah. Success by 1000 Improvements: Flight Qualification of the ST-16 Star Tracker. *28th Annual AIAA/USU Conference on Small Satellites*, 5000:1–14, 2014.

[12] ECSS. ESA pointing error engineering handbook ESSB-HB-E-003. *Ecss*, 1 Edition(July):1–72, 2011.

[13] ESA eoPortal Directory. BRITE (BRIght-star Target Explorer) Constellation / BRITE Austria, UniBRITE. Available at https://directory.eoportal.org/web/eoportal/satellite-missions/pag-filter/-/article/brite.

[14] ESA eoPortal Directory. PicSat (Pictoris Satellite) mission. Available at https://directory.eoportal.org/web/eoportal/satellite-missions/p/picsat.

[15] ESA European Space Agency. CubeSats in Space. Available at http://www.esa.int/spaceinimages/Images/2017/10/CubeSats_in_space.

[16] Forbes. Nanosatellites Take Off. Available at https://www.forbes.com/forbes/2010/1122/technology-pumpkin-inc-andrew-kalman-toasters-in-space.html#a46faa51e691.

[17] National Instruments. Understanding LVDS for Digital Test Systems. Available at http://www.ni.com/white-paper/4441/en/.

[18] Rebecca Jensen-clem and Joe Mccarter. Exoplanet Observations with CubeSats Exoplanets & Detection. pages 1–23.

[19] Heather A. Knutson, Nikole Lewis, Jonathan J. Fortney, Adam Burrows, Adam P. Showman, Nicolas B. Cowan, Eric Agol, Suzanne Aigrain, David Charbonneau, Drake Deming, Jean Michel Désert, Gregory W. Henry, Jonathan Langton, and Gregory Laughlin. 3.6 and 4.5 μm phase curves and evidence for non-equilibrium chemistry in the atmosphere of extrasolar planet HD 189733b. *Astrophysical Journal*, 754(1), 2012.

[20] KU Leuven. Enabling spectroscopy of stars from a CUBESAT platform Meeting BELSPO 13-July-2017. Technical report, KU Leuven, 2017.

[21] W. Lan, R. Munakata, R. Nugent, and D. Pignatelli. Poly Picosatellite Orbital Deployer Mk III Rev. E User Guide. pages 1–21, 2014.

[22] Anne Marinan and Kerri Cahoy. From CubeSats to Constellations: Systems Design and Performance Analysis. (September):116, 2013.

[23] NASA. NanoRacks CubeSats Deployed From Space Station. Available at https://www.nasa.gov/content/nanoracks-cubesats-deployed-from-space-station.

[24] M. Nowak, S. Lacour, V. Lapeyrère, L. David, A. Crouzier, C. Dufoing, H. Faiz, T. Lemoult, and P. Trébuchet. Reaching sub-milimag photometric precision on Beta Pictoris with a nanosat: the PicSat mission. 2018:99044L, 2016.

[25] National Academies of Sciences Engineering and Medicine. *Achieving Science with CubeSats: Thinking Inside the Box.* The National Academies Press, Washington, DC, 2016.

[26] PiezoDrive. AP Series Amplified Piezoelectric Actuators.

[27] PiezoDrive. Three Channel, Ultra-low Noise Miniature 150V Piezo Driver. pages 1–4.

[28] Gert Raskin, Tjorven Delabie, Wim De Munter, Tom Mladenov, Bart Vandenbussche, Bram Vandoren, and Dirk Vandepitte. CUBESPEC : Low-cost astronomical spectroscopy from a nano-satellite. *SPIE*, 2018.

[29] Marcia Rieke. Astr 250: Fundamentals of Astronomy. Available at `http://ircamera.as.arizona.edu/astr_250/Lectures/Lec_10sml.htm`.

[30] Karan Sarda, C. Cordell Grant, Monica Chaumont, Seung Yun Choi, Bryan Johnston-Lemke, and Robert E. Zee. On-Orbit Performance of the Bright Target Explorer (BRITE) Nanosatellite Astronomy Constellation. *28th Annual AIAA/USU Conference on Small Satellites*, 2014.

[31] Schoolphysics. REFLECTION (Plane mirrors). Available at `http://www.schoolphysics.co.uk/age16-19/Optics/Reflection/text/Reflection_/index.html`.

[32] Tom Segert, Steven Engelen, Matthias Buhl, and Bert Monna. iADCS-100 - an Autonomous Attitude Determination and Control Subsystem based on Reaction Wheels and Star Tracker in 1/3U Package. *The 4S Symposium 2012*, (1):2–7, 2012.

[33] Matthew W. Smith, Sara Seager, Christopher M. Pong, Matthew W. Knutson, David W. Miller, Timothy C. Henderson, Sungyung Lim, Tye M. Brady, Michael J. Matranga, and Shawn D. Murphy. The ExoplanetSat Mission to Detect Transiting Exoplanets with a CubeSat Space Telescope. *Proceedings of the 25th Annual Small Satellite Conference*, pages 1–9, 2011.

[34] Matthew W. Smith, Sara Seager, Christopher M. Pong, Jesus S. Villaseñor, George R. Ricker, David W. Miller, Mary E. Knapp, Grant T. Farmer, and Rebecca Jensen-Clem. ExoplanetSat: detecting transiting exoplanets using a low-cost CubeSat platform. page 773127, 2010.

[35] Space News Spaceflight 101 and Beyond. PicSat Satellite Overview. Available at `http://spaceflight101.com/pslv-c40/picsat/`.

[36] PicSat Unravelling the Beta Pictoris System. PicSat Images, 2018. Available at `https://picsat.obspm.fr/resources/images?locale=en`.

[37] Space Flight Lab (UTIAS-SFL) University of Toronto Institute of Aerospace Studies. BRITE-Constellation: nano-satellites for astrophysics. Available at `http://www.brite-constellation.at/`.

[38] Lauri Vosandi. Video capture with VDMA. Available at `https://lauri.xn--vsandi-pxa.com/hdl/zynq/xilinx-video-capture.html`.

[39] W. W. Weiss, S. M. Rucinski, A. F. J. Moffat, A. Schwarzenberg-Czerny, O. F. Koudelka, C. C. Grant, R. E. Zee, R. Kuschnig, St. Mochnacki, J. M. Matthews, P. Orleanski, A. Pamyatnykh, A. Pigulski, J. Alves, M. Guedel, G. Handler, G. A. Wade, and K. Zwintz. BRITE-Constellation: Nanosatellites for Precision Photometry of Bright Stars. *Publications of the Astronomical Society of the Pacific*, 126(940):573–585, 2014.

[40] all programmable Xilinx. ZedBoard. Available at `https://www.xilinx.com/products/boards-and-kits/1-8dyf-11.html`.

[41] XIMEA. xiAPI Manual. Available at `https://www.ximea.com/support/wiki/apis/XiAPI_Manual`.

[42] XIMEA. XIMEA - USB3 Vision Standard cameras with USB 3.0 interface based on Onsemi, CMOSIS and e2V. Available at `https://www.ximea.com/en/products/usb3-vision-cameras-xiq-line/mq042rg-cm`.

Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:
FPGA-based Active Pointing Correction of Optical Instruments on Small Satellites

Richting: **master in de industriële wetenschappen: elektronica-ICT**
Jaar: **2018**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

Mladenov, Tom

Datum: **4/06/2018**