The Design and Implementation of a Stellar Gyroscope for Accurate Angular Rate Estimation on CubeSats

by

Nico Calitz

Thesis presented in partial fulfilment of the requirements for the degree of
Master of Engineering
at Stellenbosch University

Supervisor:

Prof W.H. Steyn

Department of Electrical and Electronic Engineering

December 2015

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

December 2015

Abstract

Until recently, small form factor satellites (such as CubeSats) relied almost exclusively on micro electromechanical system (MEMS) gyroscopes for attitude propagation purposes. Unfortunately, the nature of MEMS gyros is such that they exhibit a measure of bias drift. This drift must be compensated for, a task for which stellar gyros have proved to be exceptionally useful.

Stellar gyros are satellite subsystems capable of inferring three-axis attitude propagation based on the displacement of a series of stars between successive image frames. Their design is analogous to that of star trackers, using many of the same hardware designs and algorithms. When used in combination with MEMS solutions, stellar gyros provide not only a means for drift compensation, but also a measure of functional redundancy with regard to attitude propagation.

This thesis presents the design and implementation of stellar gyroscope algorithms capable of operating alongside existing orientation algorithms on traditional star tracker hardware. The *CubeStar* star tracker module is used as development platform. The proposed stellar gyro solution retains CubeStar's existing star extraction algorithms, while investigating alternative methods for star centroiding in addition to the existing centre of gravity (CoG) approach. A dynamic proximity based matching algorithm is suggested to determine star correspondence between image frames. Finally, various well established estimation algorithms are considered for the purpose of rate determination, including singular value decomposition (SVD), Davenport's q-Method and weighted least-squares (WLS).

An initial evaluation of the proposed algorithms is made based on simulations in the MATLAB environment. Simulation results are confirmed through means of practical tests, performed on a simulated night sky in a controlled environment. With a focus on low angular rates, results suggest reliable operation up to ±1 deg/s in all three axes of rotation. As expected for stellar imaging solutions, angular rates estimated in both cross-boresight axes are almost an order of magnitude more accurate than the corresponding estimates in the boresight axis itself.

Opsomming

Mikrosatelliete, soos *CubeSats*, het tot onlangs byna uitsluitlik op mikro elektromeganiese (MEMS) vibrerende struktuur giroskope staatgemaak vir die meet van hoeksnelhede. Ongelukkig is die aard van MEMS giroskope sodanig dat hierdie metings afsette toon wat al hoe verder van hul werklike waardes verskuif. Daar moet gekompenseer word vir hierdie verskuiwing, 'n taak waarvoor stergiroskope besonder geskik is.

Sterrebeeld gebaseerde giroskope (of bloot gewoon stergiroskope) is satelliet substelsels wat daartoe in staat is om 'n satelliet se oriëntasie in drie dimensies te propageer deur gebruik te maak van die verplasing van 'n reeks sterre tussen twee opeenvolgende beelde. Hulle ontwerp in terme van beide hardeware en algoritmes is soortgelyk aan dié van stervolger kameras. Stergiroskope kan ook saam met MEMS toestelle gebruik word. Hulle verskaf beide 'n metode om te kompenseer vir verskuiwings in MEMS metings sowel as 'n funksionele alternatief met betrekking tot hoekafskatting.

Hierdie tesis beskryf die ontwerp en implementering van ster giroskoop algoritmes wat in staat is om hand-in-hand met bestaande oriëntasie algoritmes op tradisionele ster volger hardeware te funksioneer. Die *CubeStar* stervolger module is as ontwikkelings platform gebruik. Die beoogde stergiroskoop ontwerp behou *CubeStar* se bestaande ster ontginnings algoritmes. Verskeie metodes benewens die bestaande swaartepunt benadering word wel ondersoek vir die bepaling van ster sentroïedes. Die korrespondensie tussen opeenvolgende sterbeelde word bepaal deur middel van 'n dinamiese nabyheid gebaseerde passings algoritme. Ten slotte word verskeie algoritmes oorweeg vir die afskatting van 'n satelliet se hoeksnelhede. Dit sluit in enkelvoud waarde ontbinding (SVD), Davenport se q-metode en 'n geweegte kleinste kwadraat (WLS) benadering.

Die voorgestelde algoritmes is ge-evalueer op grond van simulasies in die MATLAB omgewing. Praktiese toetse is uitgevoer op 'n gesimuleerde sterrebeeld om simulasie resultate te bevestig. Met 'n fokus op lae hoeksnelhede dui resultate op betroubare afskatting teen hoeksnelhede van tot ± 1 grade/s rondom al drie rotasie-asse. Soos verwag van ster kameras is die hoekafskattings rondom die transversale asse 'n orde meer akkuraat as die ooreenstemmende afskattings rondom die optiese as.

Contents

List of Figures

List of Tables

Abbreviations and Acronyms

A/D	Analogue to Digital
BCE	Before Common Era
CCD	Charge-Coupled Device
CFA	Colour Filter Array
CMOS	Complementary Metal-Oxide-Semiconductor
CMY	Cyan-Magenta-Yellow
CoG	Centre-of-Gravity
COTS	Commercial Off-The-Shelf
DCM	Direction Cosine Matrix
ECI	Earth-Centred Inertial
ESL	Electronic Systems Laboratory
FOV	Field of View
FPGA	Field Programmable Gate Array
GLS	General Least Squares
ID	Identification
LEO	Low-Earth Orbit
MEMS	Micro-Electromechanical System
MCU	Micro-Controller Unit
NASA	National Aeronautics and Space Administration
PSF	Point Spread Function
RANSAC	Random Sample Consensus
RGB	Red-Green-Blue
ROI	Region of Interest
SNR	Signal-to-Noise Ratio

SVD	Singular Value Decomposition
WCoG	Weighted Centre of Gravity
WLS	Weighted Least-Squares

Acknowledgements

Firstly, I would like to express my gratitude to my supervisor Prof WH Steyn for his invaluable advice, guidance and support throughout the course of my studies.

I thank Pietro Petzer and his team for providing the testing facilities, as well as their assistance in manufacturing the test setup.

I thank my brother, James, for his assistance with many of the illustrations included in my dissertation.

I thank my colleagues at the Electronic Systems Laboratory, especially Douw Steyn, Nico Rossouw and Muhammad Junaid, for providing both an entertaining and intellectually stimulating working environment.

A special thanks to Willem Jordaan for his constant willingness to provide advice, support and encouragement.

My gratitude extends to SANSA for their financial support, which made my studies possible.

Finally I thank my mother, Anieta, and father, Coenie, for their encouragement and unwavering support throughout the duration of my studies. Without them I wouldn't be where I am today.

Chapter 1

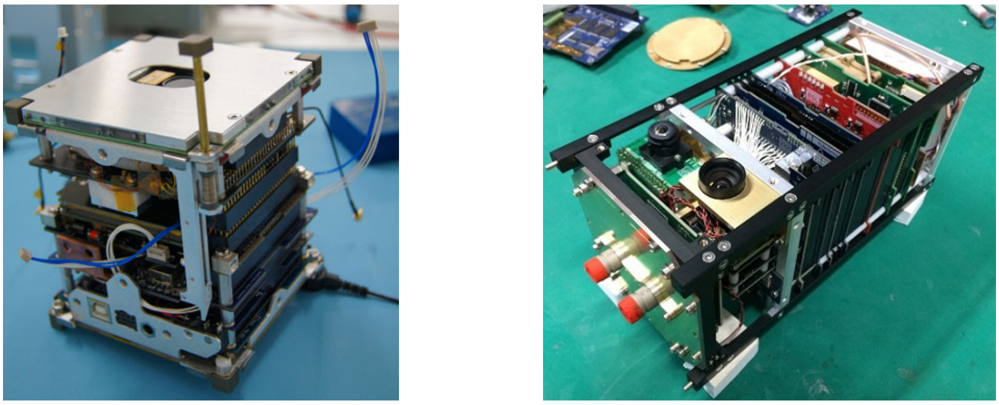
Introduction

This chapter serves as introduction to the relevant concepts that motivate the work presented in this thesis. This is followed by an overview of the aims and objectives for the proposed stellar gyro project.

1.1 CubeSat Design Standard

The CubeSat Project, introduced in 1999 through a collaborative effort between Prof. Jordi Puig-Suari at California Polytechnic State University and Prof. Bob Twiggs at Stanford University's Space Systems Development Laboratory (SSDL), presents a design standard for small scale satellites aimed primarily at reducing development time and costs while increasing accessibility to space [1].

The most commonly used CubeSat form factor, namely a 1U, is essentially defined as a miniature satellite that exhibits dimensions of exactly $(10 \times 10 \times 10)\,\mathrm{cm}^3$ and a mass not exceeding 1.33 kg. Figure 1.1(a) shows an image of ZACUBE-1, South Africa's first 1U CubeSat launched in November 2013 [2].



(a) ZACUBE-1 (1U) [2]

(b) ZA-AeroSat (2U) [3]

Figure 1.1 – Two South African satellites based on the CubeSat design standard

1.1.1 Relevance of CubeSats

The original motivation behind the CubeSat project was to facilitate university-level hands-on space education. A gradual international acceptance of the CubeSat design standard not only provided such educational opportunities, but it also allowed commercial and research institutes from developing countries to participate in space exploration. A reliance on build redundancy, inexpensive commercial of-the-shelf (COTS) components and a standardized launch interface greatly simplified the design process while also reducing development and manufacturing time and costs. Furthermore, the design standardization of such satellites facilitate international collaboration and provide a support structure for less experienced and financially capable participants.

1.1.2 QB50 Launch Mission

The upcoming QB50 mission [4] epitomizes the significance of the CubeSat standard with respect to international collaboration. It will oversee the launch of 50 CubeSats from 27 different countries from all over the world with the goal of achieving affordable and sustainable access to space for small-scale space research and planetary exploration missions. The project also has scientific significance, as it aims to carry out atmospheric research and analysis within the lower thermosphere, the least explored layer of the atmosphere to date. Figure 1.1(b) shows an image of ZA-AeroSat, a 2U CubeSat $[(10 \times 10 \times 20)\,\mathrm{cm}^3]$ designed and manufactured at the University of Stellenbosch that embodies Africa's contribution to the QB50 mission.

1.2 CubeSat Subsystems

The international success of the CubeSat design standard places a strong emphasis on the miniaturization of satellite subsystems. Unfortunately, such miniaturization introduces strict limitations with respect to power consumption and physical volume, which highlights the importance of component integration as well as harnessing the full potential of each of these subsystems. Considering the flawed nature of existing micro-electromechanical system (MEMS) solutions (due to bias drift, as detailed in section 1.2.1), the miniaturization of image based rate determination solutions seem particularly enticing.

1.2.1 Physical Gyroscopes

Satellite attitude propagation typically relies on the integration of gyroscopic data. Gyroscopic sensors (commonly referred to simply as gyroscopes or gyros) employ various operating principles, such as fibre-optics, solid-state laser rings and vibrating structures (MEMS). Even though remarkably accurate, the fibre-optic and laser-ring varieties tend to be large and expensive (the latter, for example, being used on the International Space Station). The MEMS variety, on the other hand, is relatively compact and affordable. Until recently, small satellites relied almost exclusively on MEMS sensors to determine three-axis angular rates.

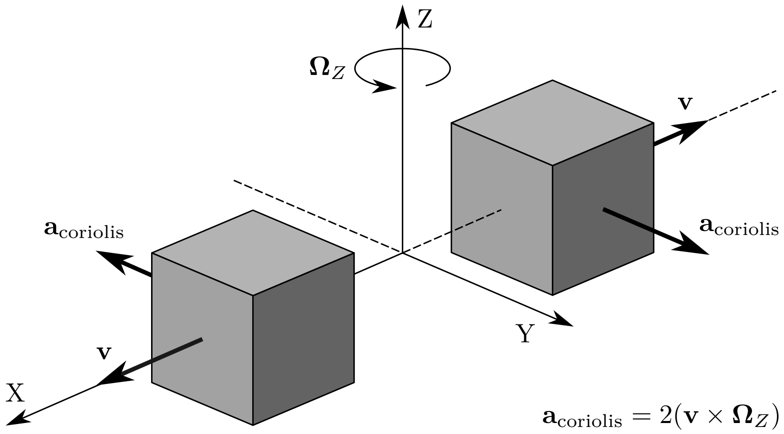

Z
$\mathbf{\Omega}_Z$
v
$\mathbf{a}_{\text{coriolis}}$
$\mathbf{a}_{\text{coriolis}}$
Y
v
X
$\mathbf{a}_{\text{coriolis}} = 2(\mathbf{v} \times \mathbf{\Omega}_Z)$

Figure 1.2 – Illustration of a tuning fork vibrating structure (MEMS) gyro

Figure 1.2 illustrates the basic theory of operation of a tuning fork[1] vibrating structure (MEMS) gyroscope. Consider two proof masses, constantly oscillating and moving in opposite directions (with a velocity of $\mathbf{v}$). When an angular velocity ($\mathbf{\Omega}_Z$) is applied, the Coriolis effect induces an acceleration in the proof masses equal to $\mathbf{a}_{\text{coriolis}} = 2(\mathbf{v} \times \mathbf{\Omega}_Z)$. The resultant displacements of these masses are then read from a capacitive sensing structure.

One of the main drawbacks of using MEMS sensors is bias drift. Inertial sensors, such as MEMS gyroscopes, exhibit a non-zero output even in the absence of an input signal. This output offset, also referred to as the measurement bias, is added to the measured signal. Furthermore, this bias offset drifts over time due to temperature and other stochastic factors. The uncertainty of these factors complicate the process of characterizing and compensating for the observed drift. Furthermore, the dead reckoning nature of MEMS sensors mean that errors are cumulative. The angular rates obtained through integration of MEMS sensor measurements will therefore drift further from its actual value as time progresses [5]. In order to retain an acceptable level of accuracy, drift compensation must be performed.

1.2.2 Star Trackers

Star trackers are three-axis attitude determination devices that utilize the combination of a sensitive camera connected to a micro-controller unit (MCU) and a list of celestially referenced star locations (commonly referred to as a star catalogue) to determine the orientation of a satellite relative to inertial space.

The attitude determination process is initiated through star imaging, after which various complex algorithms are employed to extract star locations from the raw image data. Pattern recognition is performed to match the extracted star locations to an on-board star catalogue, which contains the locations of various stars as referenced to the celestial sphere. If the captured stars are successfully identified, it enables

[1] other vibrating structure gyroscope implementations include piezoelectric, hemispherical resonator, cylindrical resonator and vibrating wheel designs

the star tracker to determine its orientation relative to the celestial sphere. Such orientation is usually determined in the form of an attitude quaternion.

Star trackers have several advantages over alternative attitude sensors (such as sun and horizon sensors). They are

- up to two orders of magnitude more accurate [6],
- capable of reliable operation in both sun-lit (requires baffle) and eclipse portions of an orbit,
- autonomous, capable of determining the attitude quaternion without any external processing.

Due to recent advancements in image sensor technology, CubeSats now frequently resort to star trackers for attitude determination purposes.

1.2.3 Stellar Gyroscopes

Stellar gyroscopes are based on the principle of attitude propagation through a combination of image processing and visual analysis. More specifically, the stellar gyro concept describes the process of inferring three-axis attitude propagation based on the displacement of a series of stars between successive image frames (i.e., image based rate determination).

The physical design of stellar gyros are analogous to that of star trackers. They use similar optical solutions and hardware designs, consequently also exhibiting similar size, weight and power consumption characteristics. Furthermore, stellar gyros also utilize many of the same algorithms for star extraction and centroiding. The main difference being that star trackers apply the determined star vectors towards attitude *determination*, whereas in stellar gyros they are applied towards attitude *propagation*.

This analogy is of particular interest with regard to application, since it allows stellar gyro algorithms to be implemented alongside existing orientation algorithms on star tracker solutions. This adds a substantial measure of functionality to an existing subsystem without the need for any additional hardware. Furthermore, since such algorithms largely repurpose existing data, they require very little additional power and execution time.

It is also worth noting that, since angular rates are estimated based solely on the displacement of stars between consecutive image frames, rate estimates are independent of one another. This eliminates the possibility of a cumulative error as observed for its MEMS counterpart (see section 1.2.1). When used in combination with MEMS solutions, stellar gyros not only provide a measure of functional redundancy with regard to attitude propagation, but they can also be used to perform drift compensation for their MEMS counterparts.

The implementation of stellar gyro solutions on CubeSats would therefore not only reduce overall hardware costs, but also increase the availability of resources for the satellite payload.

1.3 Project Aims

For accentuation purposes, the project aims are presented in list format:

- Develop and implement an accurate and reliable stellar gyro solution capable of operating on CubeStar hardware.
- Stellar gyro algorithms should be capable of operating interchangeably with existing orientation algorithms.
- With a focus on low angular rates, the stellar gyro solution should be capable of reliably estimating angular rates up to ±1 deg/s in all three axes of rotation.

The implementation of such stellar gyro algorithms is expected to improve the existing CubeStar subsystem by

- extending functionality without the need for any additional hardware (and very little additional power and execution time requirements),
- providing a measure of functional redundancy with regard to attitude propagation when used in conjunction with MEMS gyroscopes, and
- enabling CubeStar to perform drift compensation for traditional MEMS solutions.

Chapter 2

Background

This chapter provides the necessary theoretical background related to the work presented in this thesis. Topics include an overview of *CubeStar*, relevant astronomy concepts, coordinate systems, attitude representations as well as review of related work.

2.1 CubeStar

CubeStar (see fig. 2.1) is a CubeSat compatible star tracker camera designed by Alex Erlank at the University of Stellenbosch's Electronic Systems Laboratory (ESL) [7]. Among various other aspects, the ESL specialises in the development of CubeSat compatible subsystems, which include space flown units such as CubeSense and CubeComputer [8, 9, 10]. Due to its infancy, CubeStar unfortunately does not yet boast the same flight heritage as the aforementioned subsystems. It does however use many of the same hardware designs, which not only sped up development time, but also provided confirmation as to the applicability of the chosen components. The CubeStar module is used as development platform for the stellar gyro algorithms proposed by this thesis.

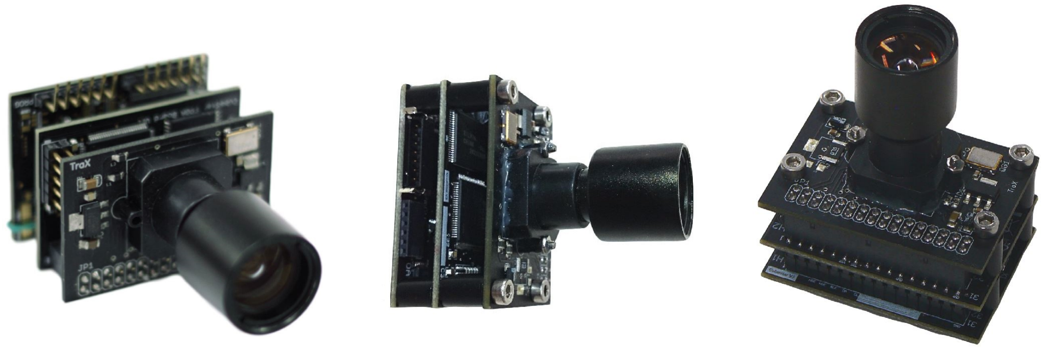


Figure 2.1 – Images of the CubeStar star tracker module

2.1.1 Specifications and Performance

The CubeStar module is a relatively small and lightweight package, measuring in at $50.1 \times 35.2 \times 64\,\mathrm{mm}^3$ with a weight of 56 g. It features nominal and peak power consumption values of 320 mW and 500 mW respectively. The implementation of both I2C and UART data interfaces provide a measure of communication flexibility. One of CubeStar's most unique features is its exceptionally large field of view, covering an area of $52° \times 27°$. The main advantage of such a large FOV is the ability to achieve full sky coverage (see section 3.5) with a less light sensitive, and therefore also less expensive, image sensor.

Despite CubeStar's comparatively low sensitivity (capable of detecting stars up to $M_V = 4$), performance claims are impressive, with 1σ accuracies of 0.01° and 0.03° in the cross-boresight and roll axes respectively.

2.1.2 Hardware Design

CubeStar optics consist of the combination of a Melexis MLX75412 monochrome CMOS image sensor and Lensation BL6012 lens. Key features of the MLX75412 include a 1024×512 pixel resolution, low power rolling shutter, 8/10/12 bit parallel data output, I2C communications and a temperature operating range of $-40\,°\mathrm{C}$ to $115\,°\mathrm{C}$. The main drawbacks of the chosen image sensor are its 2:1 aspect ratio, which results in a large component of the incoming light falling outside the image sensor area, as well as the implementation of a rolling shutter as opposed to the better suited global shutter solution. Fortunately, these drawbacks are easily overshadowed by the low unit cost, flexible parallel data output functionality and large temperature operating range. The Lensation BL6012 was chosen due to its compatibility with the Melexis image sensor, featuring a 1/3 inch optical format and S-Mount mount type. The most distinct feature of the Lensation lens is its exceptionally low focal ratio (f-stop of 1.2), which maximises the amount of light passing through the lens aperture to reach the image sensor.

At its core, CubeStar uses a 32bit Energy Micro ARM Cortex M3 processor (EFM32GG280F1024), featuring 128 kB of RAM and 1024 kB of flash memory. The EFM32GG was chosen due to its exceptionally low power consumption claims and high performance capabilities. Since the EFM32GG was previously used on-board CubeComputer, various libraries and code snippets could easily be reapplied to CubeStar.

Images captured using the Melexis sensor are 512 kB in size, rendering the internal 128 kB RAM insufficient for storing such images. An external memory mapped 1024 kB Alliance Semiconductor SRAM module is therefore used. An Actel IGLOO NANO AGLN030 FPGA acts as buffer between the micro controller and image sensor.

2.1.3 Algorithms

Star trackers use a variety of algorithms to determine orientation. Since this research is focussed on the design of a stellar gyroscope capable of operating in parallel with existing orientation algorithms, some of the algorithms written for CubeStar were reapplied. To promote a general understanding, a brief overview of the orientation process is given below.

The process is initiated through imaging. Once an image is captured, stars must be extracted from the raw image data. This is done by assessing whether the value of certain pixels exceed a predetermined minimum threshold required for it to be part of a star. The specific pixels to be assessed are determined such as to minimise the required number of computations that will guarantee the extraction of each and every star. If a pixel is identified as exceeding this threshold, that pixel is subjected to a recursive growing algorithm responsible for extracting all adjacent pixels also exceeding the threshold. If the extracted pixel group exceeds the minimum number of pixels required for it to be classified as a star, the pixel group is subjected to a centre of gravity (CoG) centroiding algorithm in order to determine the star centroid. The determined centroid location is then corrected for lens distortion. This process is repeated until all stars have been extracted from the captured image. Finally, the centroid locations are subjected to a geometric voting algorithm to determining the star tracker's orientation. Even though the voting algorithm is of little use in stellar gyroscope applications, the remaining algorithms can easily be reapplied.

2.2 Relevant Astronomy Concepts

The nature of this thesis is such that it relies heavily on various astronomical concepts. Since many of these concepts are often unfamiliar to engineers, a brief description is provided to promote a general understanding of the work contained throughout.

2.2.1 Stellar Parallax

Stellar parallax describes the apparent shift in position of celestial bodies relative to one another as seen by an observer from two separate points. Figure 2.2 depicts the maximum parallax angle case for an earth based observer, also referred to as the annual parallax.

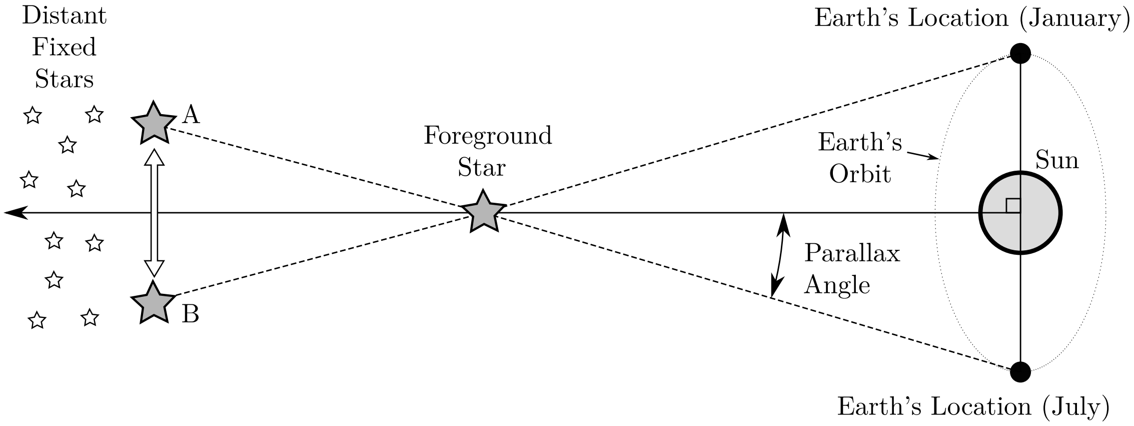


Figure 2.2 – Annual stellar parallax relative to an earth-based observer

It is observed that the location of the foreground star relative to distant "fixed" stars will differ depending on the earth's position in orbit. Note that the distant stars will also exhibit some form of parallax, but the effect will be much less pronounced than that of relatively close stars. Even though stellar parallax will have a more significant impact on orientation algorithms than their rate estimation counterparts, a general analysis is performed here in the interest of meticulosity.

Consider *Proxima Centauri* from the *Alpha Centauri* system, the closest star to our Solar system. The distance between the Sun and *Proxima Centauri* is approximately 4.24 light-years (40.11×10^{12} km). Considering an average orbital radius of 149.6×10^{6} km for the earth's orbit around the Sun, the worst case annual parallax angle (as is determined by the closest star) can be determined as

$$\rho = \tan^{-1}\left(\frac{149.6 \times 10^{6}}{40.11 \times 10^{12}}\right) = 0.000\,214^{\circ} \tag{2.2.1}$$

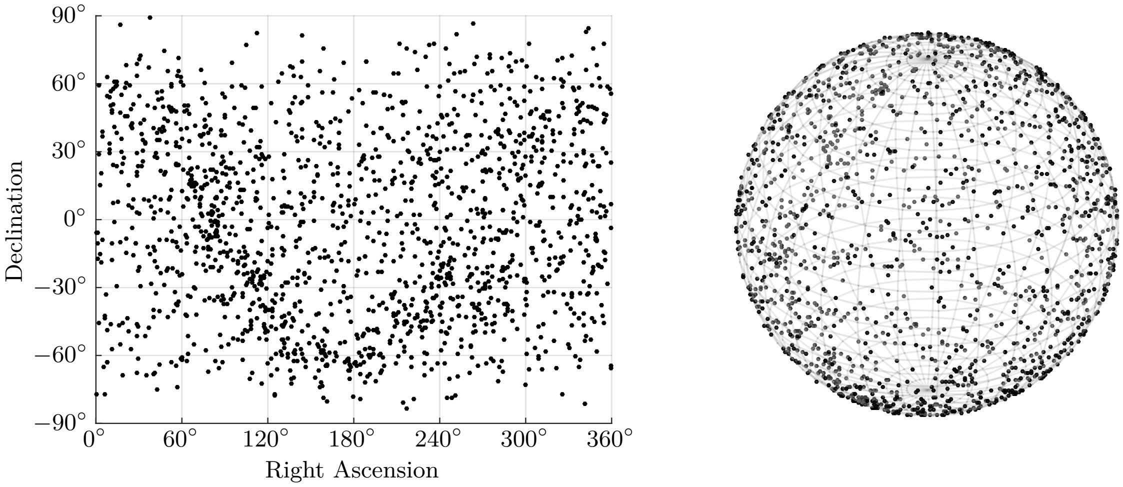
Note that the annual parallax angle calculated above is based on a six month cycle. Since stellar gyroscope algorithms only rely on data from two successive images, usually captured less than a second apart, the effect of stellar parallax is negligible.

2.2.2 Celestial Sphere

The concept of a celestial sphere is often used to describe the position of various celestial bodies relative to a point of observation.

The conclusions made in section 2.2.1 imply that, for the purpose of low earth orbit (LEO) stellar gyroscope applications, the position of distant celestial bodies (with an emphasis on stars) relative to one another is for all intents and purposes "fixed". Describing the location of such celestial bodies relative to an observer in terms of distance therefore carries very little value. A more relevant and meaningful representation can be obtained by projecting all stars onto an imaginary sphere of arbitrary radius with the observer based at its origin, and defining star locations exclusively in terms of separation angles. Such representation, as illustrated by fig. 2.3, is referred to as the celestial sphere.

The applicability of such representation can be bolstered even further by defining a sphere with unit radius, allowing star locations to be described using unit vectors. This offers various advantages both in terms of interpretation and manipulation. The analogy between direction and unit vectors allows for the unique description (independent of surrounding stars) of star locations relative to the point of observation. Furthermore, representing stars as unit vectors accommodate transformation between reference frames using various well established methods [11]. Section 2.3.1 presents an overview of the celestial coordinate system.



(a) 2D Projection Map

(b) 3D Spherical Representation

Figure 2.3 – Illustration of the celestial sphere for stars up to $M_V = 5$

2.2.3 Stellar Magnitude

The magnitude system, devised around 150 BCE by Greek astronomer Hipparchus, is a logarithmic measure of the brightness of celestial objects [12]. Hipparchus originally classified stars based on how bright they appeared to the naked eye. The brightest stars would fall in the first magnitude class, whereas the dimmest stars would fall in the sixth magnitude class. The problem with Hipparchus' representation is twofold: firstly, the human eye is more sensitive to some frequencies than to others and secondly, the classification does not consider the distance of each star from the point of observation. These problems led to the eventual distinction between the apparent and absolute magnitude scales, as well as frequency standardisation through use of the UBV photometric system.

Apparent Magnitude

The apparent magnitude scale resembles Hipparchus' original representation in the sense that it defines star brightness relative to an earth based observer. The simplicity of such representation lies in the fact that it only requires a measure of the apparent brightness of an object in order to determine its magnitude. The apparent magnitude scale therefore forgoes the distinction between the intrinsic brightness of a star and its distance to the point of observation in favour of a more empirical representation. As emphasised in section 2.2.1 through the concept of stellar parallax, a stellar gyro in LEO is for all intents and purposes an earth based observer. The apparent magnitude scale is therefore much better suited for analysing stellar gyro detection capabilities than the absolute scale (see below).

Absolute Magnitude

Some astronomical applications rely on information regarding the intrinsic brightness of stars. This requires some comparative measure capable of assessing star brightness irrespective of its distance from the point of observation. The absolute magnitude scale provides such measure by determining the apparent magnitude each star would have if it were placed at a distance of 10 parsecs (32.616 light years) from the point of observation. The absolute magnitude scale has very little use where LEO stellar gyro applications are concerned.

The UBV Photometric System and Visual Magnitude

One of the main problems of determining stellar magnitudes with the human eye is frequency bias. Human vision is limited to the visible portion of the electromagnetic spectrum (see fig. 2.4). Radiation in this visible portion, commonly referred to simply as light, accounts for wavelengths ranging from about 400 to 700 nm [13]. Furthermore, as illustrated by the Bayer filter, the human eye is most sensitive to yellow-green light, which constitutes the low frequency portion of the visible spectrum. Since many stars emit radiation in the higher frequency (blue and ultraviolet) spectra, brightness analysis via the naked eye creates an unfair bias.

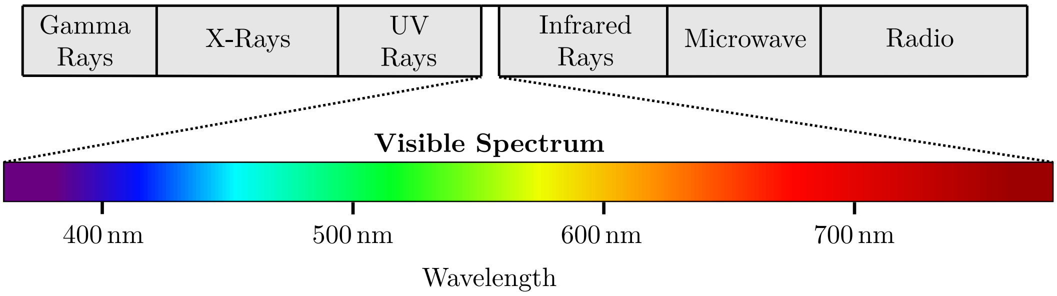


Figure 2.4 – Illustration of the visible frequency spectrum

The UBV photometric system, introduced in 1950 by American astronomers Harold Lester Johnson and William Wilson Morgan, proposed a solution to this problem by defining a photometric system for classifying star magnitude based on its frequency [14]. The UBV system categorises magnitudes in three different wavelength bands: U (centred within the ultraviolet spectrum at 350 nm), B (centred in the blue region of the visible spectrum at 430 nm) and V (centred close to the middle of the human visual range at 550 nm) [14]. Determining the radiation in each of these wavelength bands provide a much more accurate portrayal of star brightness. Seeing as most image sensors mimic the electromagnetic response of the human eye, star tracker operation is concerned mainly with radiation in the visible (V) band. Consequently, star brightness is often described by a **visual magnitude**, which represents the apparent magnitude due to radiation in the visible spectrum. For the remainder of this thesis, the terms *visual magnitude* and *apparent magnitude* are used interchangeably, both referring to the apparent magnitude due to radiation in the visible spectrum.

2.3 Coordinate Systems

Coordinate systems are used to describe the location of objects relative to their surroundings. Four coordinate systems are used throughout this thesis: celestial coordinates, earth-centred inertial coordinates, body-fixed coordinates and image plane coordinates.

2.3.1 Celestial Coordinates

The celestial coordinate system, as illustrated by fig. 2.5, uses spherical coordinates to define the location of objects on the celestial sphere (see section 2.2.2). It is analogous to the geographic coordinate system used on the surface of the earth. Grewal *et al.* [15] defines celestial coordinates as a system for inertial directions referenced to the earth's rotation axis and the vernal equinox.

Considering the aforementioned analogy, the polar celestial coordinates *right ascension* and *declination* can be interpreted as the celestial equivalents of longitude and latitude respectively. The celestial equator, defined to coincide with the earth's equator, serves as the 0° reference for measuring declination. Convention defines positive and negative declination values north and south of the celestial equator respectively (declination values range between $-90°$ to $90°$). Similarly, the vernal equinox serves as 0° reference for measuring right ascension, which is measured in an eastwards direction along the celestial equator (right ascension values range between 0° to 360°). Right ascension can be measured using either degrees (preferred in this thesis), radians or hours (with a 15 °/h conversion factor).

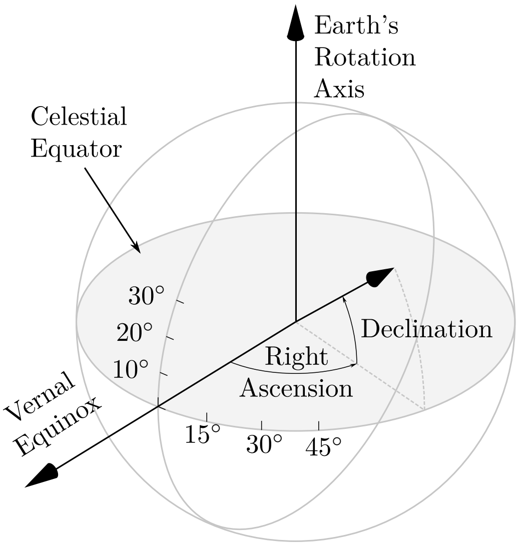


Figure 2.5 – Celestial coordinate system

2.3.2 Earth-Centred Inertial (ECI) Coordinates

Grewal *et al.* [15] defines the earth-centred inertial reference frame such that (a) the origin coincides with the earth's centre of mass; (b) the Z-axis is directed along the earth's rotation axis; (c) the X-axis is directed towards the vernal equinox; and (c) the Y-axis is defined to complete the right handed coordinate system (see fig. 2.6).

Similar to the celestial coordinate system, the ECI coordinate system is fixed relative to the celestial sphere. The main difference between these coordinate systems is that the ECI system uses a three dimensional unit vector representation as opposed to the spherical approach employed by the celestial system. Even though the physical interpretation of such unit vector representation might be less intuitive than its spherical counterpart, its versatility and applicability to existing algorithms identify it as the better suited of the two representations for most stellar imagery applications.

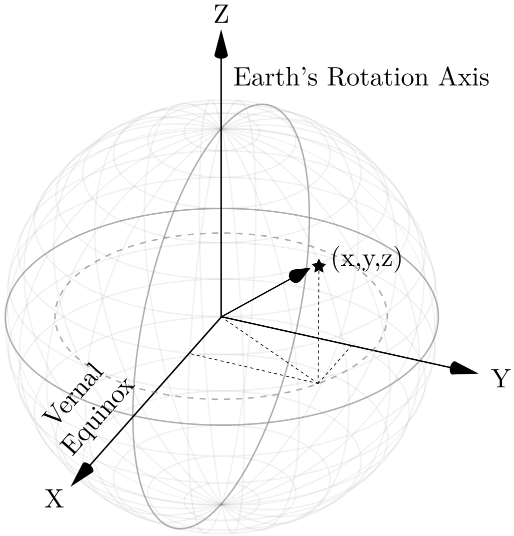


Figure 2.6 – Earth-centred inertial (ECI) coordinate system

2.3.3 Body-Fixed Coordinates

The body-fixed coordinate system is defined relative to the body of an object. However, various formalism exist for describing the location of the origin and orientation of the axes relative to the body itself.

For the purpose of this thesis, the body-fixed coordinate system was defined such that (a) the origin coincides with the optical centre of the lens (an ideal pinhole model is assumed); (b) the Z-axis is directed along the camera boresight; (c) the Y-axis points towards the upwards vertical direction; and (d) the X-axis points towards the transverse direction completing the right handed coordinate system. Figure 2.7 illustrates the orientation of the axes relative to a stellar gyro camera.

Body-fixed coordinates are of particular interest to stellar gyro applications. Stars within the camera FOV are often described using body-fixed unit vector coordinates. Angular rates are also estimated relative to the these axes.

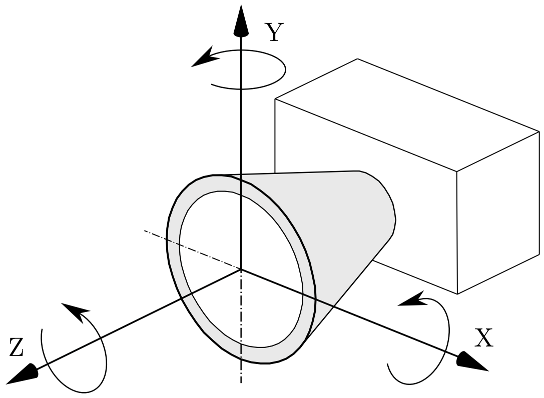


Figure 2.7 – Orientation of the body-fixed axes relative to a stellar gyro camera

2.3.4 Image Plane Coordinates

Image plane coordinates are two dimensional coordinates used to describe a location on the image sensor surface (and therefore also on the image itself). The origin is usually defined as the point where the lens boresight axis intersects the image plane. Coordinates can be expressed using either millimetres or pixels, the later preferred throughout this thesis. Figure 2.8 illustrates the image plane coordinate system using a pinhole lens model.

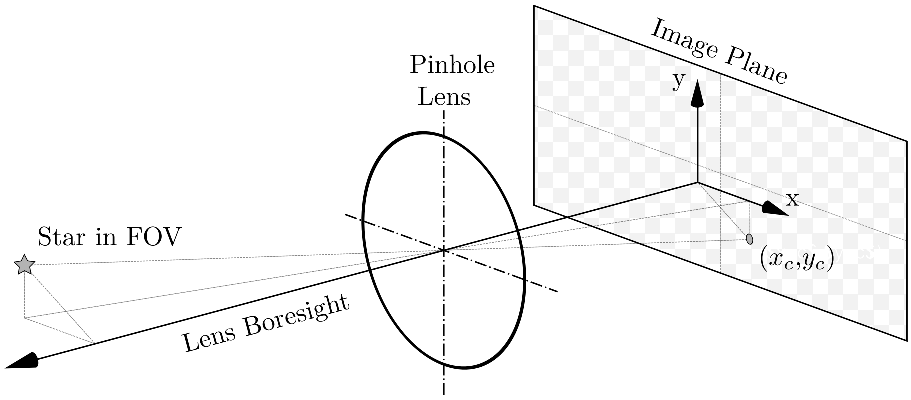


Figure 2.8 – Image plane coordinate system

2.4 Attitude Representations

Various formalisms exist for describing both the orientation and transformation of objects in three dimensional space. These formalisms are particularly useful when describing the attitude of a satellite in orbit. Of interest is a publication by Shuster [11] in which he performs a comprehensive survey and analysis of such formalisms with respect to satellite attitude determination. This section presents a brief overview of the attitude representations used throughout this thesis.

2.4.1 Axis-Angle

Euler's rotation theorem, in its most fundamental form, states that any rotation in three dimensional space can be expressed as a single rotation about a fixed axis. This axis is unique in the sense that it remains unchanged by the rotation. Figure 2.9 illustrates such rotation about fixed axis $\hat{\mathbf{n}}$ through angle θ.

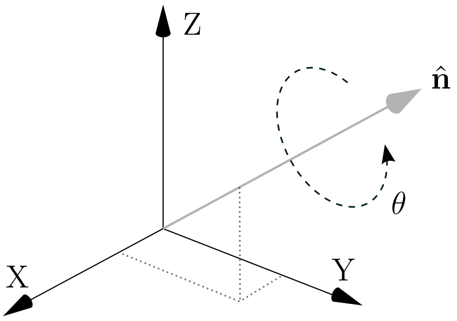


Figure 2.9 – Euler rotation through angle θ about Eigen-axis $\hat{\mathbf{n}}$

The merit of the axis-angle representation lies in its simplistic and intuitive interpretation. It resembles the way the human mind instinctively describes rotation in a three dimensional space; the rotation of an object through some angle about a specific point or axis. Of particular interest is the close resemblance between the axis-angle and quaternion representations (the latter detailed in section 2.4.3). Using these representations interchangeably consolidates the intuitive interpretation of angle-axis with the computational power of quaternions.

The axis-angle representation does however have two fundamental drawbacks. Firstly, the rotation axis is defined by the rotation itself. The absence of such rotation (i.e., an angle of rotation θ equal to zero) will therefore result in the rotation axis being undefined. Secondly, the axis-angle representation also mandates often undesirable trigonometrical computations.

2.4.2 Rotation Matrix and Euler Rotation Angles

Any vector in three dimensional space can be expressed as

$$\mathbf{v} = v_1\hat{\mathbf{e}}_1 + v_2\hat{\mathbf{e}}_2 + v_3\hat{\mathbf{e}}_3 \tag{2.4.1}$$

where orthonormal basis $\mathcal{E} : \{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3\}$ designates the abstract directions in three coordinate axes and vector components $\{v_1, v_2, v_3\}$ designate the components of arbitrary vector $\mathbf{v}$ along those axes [11]. Alternatively, the orthonormal basis $\mathcal{E}$ can be interpreted as a reference frame of vector $\mathbf{v}$.

A rotation matrix, otherwise known as a Direction Cosine Matrix (DCM), is a 3 × 3 matrix that defines the rotation (or transformation) between two such reference frames. Typical applications in the field of satellite attitude determination include rotations between inertial-, orbital- and body-fixed coordinate systems. Given arbitrary reference frames a and b, the rotation of vector $\mathbf{v}$ can be represented as

$$\mathbf{v}_b = \mathbf{A}^{ba}\mathbf{v}_a = \begin{bmatrix} \hat{\mathbf{u}}_1 & \hat{\mathbf{v}}_1 & \hat{\mathbf{w}}_1 \\ \hat{\mathbf{u}}_2 & \hat{\mathbf{v}}_2 & \hat{\mathbf{w}}_2 \\ \hat{\mathbf{u}}_3 & \hat{\mathbf{v}}_3 & \hat{\mathbf{w}}_3 \end{bmatrix} \mathbf{v}_a \tag{2.4.2}$$

where $\mathbf{A}^{ba}$ is the 3×3 matrix that describes the rotation from frame a to frame b. The columns of a rotation matrix is composed of three unit vectors $\hat{\mathbf{u}}$, $\hat{\mathbf{v}}$ and $\hat{\mathbf{w}}$, which form the rotated basis.

The rotation matrix representation exhibits no singularities or trigonometric functions, making for a robust and unequivocal method of representing satellite attitude. Unfortunately, since any three-dimensional rotation can be fully described using a minimum of three parameters, the rotation matrix representation (consisting of nine parameters) is somewhat less concise and efficient than its alternatives.

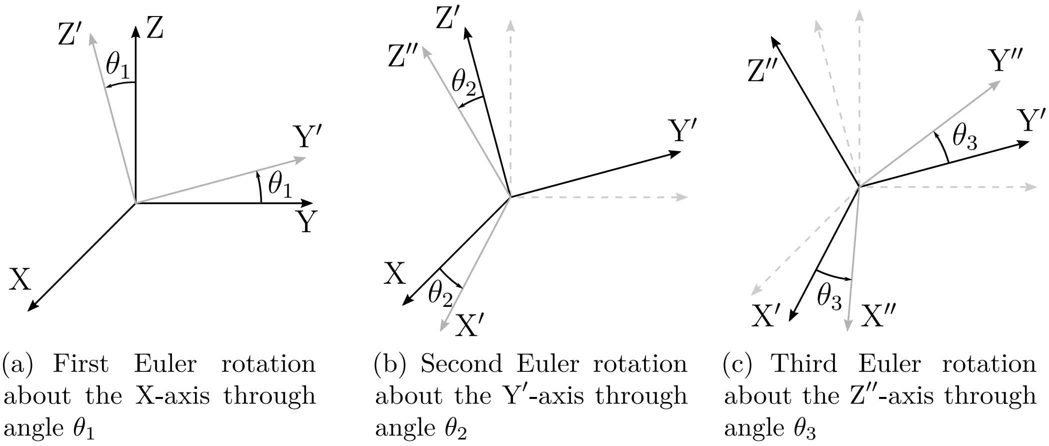
The single rotation described by such rotation matrix can also be broken down into three consecutive rotations about body-fixed specific orthogonal axes such that

$$\mathbf{A} = \mathbf{A}_3\mathbf{A}_2\mathbf{A}_1 = \mathbf{A}_\mathrm{Z}\mathbf{A}_\mathrm{Y}\mathbf{A}_\mathrm{X} \tag{2.4.3}$$

where

$$\mathbf{A}_\mathrm{X} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ 0 & -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \quad \mathbf{A}_\mathrm{Y} = \begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \quad \mathbf{A}_\mathrm{Z} = \begin{bmatrix} \cos\theta_3 & \sin\theta_3 & 0 \\ -\sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4.4}$$

The angles θ_1, θ_2 and θ_3 are referred to as Euler rotation angles. Unfortunately, the definition of Euler angles is not unique, seeing as the collective rotation represented by such angles is dependant on both the axes about which rotation takes place as well as the sequence of these rotations.



(a) First Euler rotation about the X-axis through angle θ_1

(b) Second Euler rotation about the Y′-axis through angle θ_2

(c) Third Euler rotation about the Z″-axis through angle θ_3

Figure 2.10 – Definition of the Euler 1–2–3 rotation

Figure 2.10 illustrates one such rotational sequence, namely Euler 1–2–3. The first rotation is defined as a roll about the X-axis through angle θ_1. The second rotation is defined as a pitch about the intermediate Y′-axis through angle θ_2. The third and final rotation is defined by a yaw about the Z″-axis through angle θ_3. Another particularly common representation used in the fields of science and engineering is the Euler 3–1–3 sequence.

One of the main advantages of Euler angles is its simple and intuitive interpretation. Furthermore, Euler angles exhibit no redundancy, since they describe rotations using only three parameters. Unfortunately, this efficiency comes at the cost of a heavy reliance on trigonometric computation and singularities at some angles.

2.4.3 Quaternions

Quaternion parametrisation is a method of representing rotation as a four-component vector. Unlike most attitude representations, quaternions are somewhat unintuitive. The quaternion representation of a rotation, parametrised as a versor (unit quaternion), is given by

$$\hat{\mathbf{q}} = \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 + q_4 = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^{\mathrm{T}} \tag{2.4.5}$$

Recall from section 2.4.1 that, according to Euler's rotation theorem, any rotation in three dimensional space can be expressed as a single rotation by some angle (θ) about a fixed axis ($\hat{\mathbf{n}}$). Quaternions extend this representation by combing the angle and axis of rotation to create a single four-component vector. A versor can therefore also be expressed as

$$\hat{\mathbf{q}} = \sin\frac{\theta}{2}(\mathbf{i}n_x + \mathbf{j}n_y + \mathbf{k}n_z) + \cos\frac{\theta}{2} \tag{2.4.6}$$

Regardless of their unintuitive interpretation, quaternions offer various advantages in terms of attitude representation. Similar to rotation matrices, quaternions exhibit no trigonometric functions or singularities. They are also more compact than rotation matrices (one redundant parameter as opposed to matrices' nine), and therefore less susceptible to rounding errors.

2.5 Related Work

The use of star trackers as primary on-orbit attitude determination devices has been the subject of various research endeavours since the mid 20th century. Due to sensitivity and bandwidth limitations, star trackers were traditionally used primarily as a method of drift compensation for physical gyroscopes [16]. Advancements in star sensor technology increasingly suggested using star trackers as autonomous attitude determination devices [17], capable not only of absolute referencing but also inertial referencing. The recent shift towards satellite miniaturization has further encouraged research specifically regarding the design of gyro-less spacecraft [18]. Stellar gyros are of particular interest with regard to such applications due to their exceptional accuracy capabilities [6]. Recent advancements in sensor technology (pioneered largely by the shift towards APS sensors) suggests that replacing traditional gyros with their stellar counterparts is becoming increasingly feasible [19].

Attitude Dependence

Rate estimation algorithms, as employed by stellar gyroscopes, can be separated into two categories: *attitude dependant* and *attitude independent* approaches.

Attitude dependant approaches approximate a dynamic model of the system through implementation of a Kalman filter that models the external torques acting on the spacecraft as a random process, using absolute attitude measurements to estimate angular rates [20]. The main disadvantage of using this approach is its heavy reliance on an accurate dynamic model, meaning that angular rate estimates are subject to errors in spacecraft attitude estimates.

Attitude independent approaches, on the other hand, estimate angular rates based on the time derivative of star tracker body measurements [20]. These methods are independent of star identification and attitude measurements, rendering angular rate estimates immune to poorly modelled system dynamics. Due to their reliability and robustness, attitude independent approaches has since become the norm for stellar gyro applications.

Another important aspect to consider with regard to attitude independent approaches is the angular rates at which the stellar gyro is expected to operate.

Low Angular Rates

When imaging at low angular rates, stars typically manifest themselves as point spread functions (PSFs). Various sophisticated centroiding algorithms exist that, when combined with deliberate optical defocussing, allow centroid locations to be determined with exceptional accuracy [21]. Once determined, stars must be matched between image frames. Ruocchio *et al.* [22] suggests incorporating traditional star tracker functionality via star identification. Once stars are identified, matching between image frames becomes a relatively straightforward task. Rawashdeh *et al.* [23, 24, 5, 25] suggests an alternative approach concerned with matching stars between image frames using a variation of the Random Sample Consensus (RANSAC) algorithm. Various image intrinsic parameters such as proximity and star brightness are considered in an attempt to improve the initial matching estimate, often greatly reducing the number of iterations required. Not only is RANSAC significantly more robust that the aforementioned star identification approach, but it also greatly reduces complexity [25]. Section 4.6.2 discusses the viability of implementing RANSAC on CubeStar.

Various algorithms exist that address the problem of rate estimation using the previously matched star vector pairs. Crassidis [26] suggests a linear least-squares approach based on first-order difference approximation. Liu *et al.* [27] expands upon the linear least-squares approach by incorporating dynamic estimation through adaptive Kalman filtering (specifically the implementation of a Sage-Husa filter). Rawashdeh *et al.* [24] suggests Davenport's q-method, which minimizes the sum of the square errors of all the star vector pairs. Simulations performed by the aforementioned authors suggest that each of these methods are capable of reliably estimating angular rates, albeit with varying degrees of accuracy.

High Angular Rates

At high angular rates, on the other hand, stars typically manifest themselves as star streaks. Accardo and Rufino [28] suggests an approach in which the endpoints of such star streaks are extracted from two successive images. These endpoints, which are assumed to represent the star locations, are then matched between images based on proximity. The angular rates can subsequently be estimated using methods similar to those used at low angular rates. Liebe *et al.* [19] suggests an alternative approach in which the star streaks are subjected to a spherical fit routine, responsible for identifying the axis of rotation associated with each star streak. Once these axes of rotation are known, the accompanying spherical angles (which are analogous to the spin angles of the stellar gyroscope) can be calculated using the lengths of the star streaks. With known sampling time between successive images, the angular rates can be determined based on the aggregate of these spherical angles.

When designing a stellar gyro to operate at high angular rates, there are a few important factors that must be taken into account. Consider a stellar gyro with fixed exposure time. When stationary or rotating at very low angular rates, the total incoming photons from any given star is focussed on a relatively small area. When rotating at higher angular rates, the equivalent number of photons is distributed over a much larger area, effectively reducing the projected star brightness. Stars are now much harder to identify above the noise floor, reducing the minimum detectable star brightness. Also consider the effect of high angular rates on projected star streak locations. Recall that stellar gyro algorithms designed to operate at high angular rates require information regarding both the length and endpoints of a projected star streak. Unfortunately, operating at such high angular rates increases the probability that a portion of a star streak will fall outside the image area. Consequently, neither the length nor endpoint(s) can be determined accurately, rendering that star streak invalid for the purpose of rate estimation.

Liebe *et al.* [19] suggests that these influences can be mitigated by using a low noise, high update rate image sensor with comparably large pixels. Such characteristics are typical of dedicated star sensors (e.g., ON Semiconductor's STAR family, most Mikro-Tasarim sensors). Note that CubeStar, the star tracker camera that serves as development platform for the work proposed by this thesis, was intended to operate exclusively at very low angular rates. In an attempt to reduce costs, CubeStar also uses a COTS sensor originally intended for automotive applications [7]. Unsurprisingly, CubeStar meets none of the aforementioned mitigation criteria, making it a poor choice for implementing high angular rate algorithms.

Hybrid Approach

Fasano *et al.* [29] recently proposed a novel hybrid approach capable of operating at both low and high angular rates. Using successive star images, the velocity vector field arising from the relative motion between the celestial sphere and the imaging solution is calculated using a region based matching technique. This is followed by computing the unit vectors and unit vector derivatives corresponding to the velocity vector field using a neural-network-based calibration procedure. With the unit vectors and their derivatives known, angular rates are estimated based on the

Poisson equation using a minimum-least-squares approach. Simulations suggest reliable operation at angular rates up to 5 deg/s. Unfortunately, this approach is still largely unoptimized, exhibiting comparably high noise levels while posing a significant computational burden to current generation small satellite technology. This algorithm was therefore not considered for implementation on CubeStar.

Chapter 3

Optical Analysis

One of the most important steps in the design and development of a star tracker/stellar gyro camera is a thorough optical analysis. Such analysis involves (a) characterizing lens and image sensor operation, (b) determining the optimal operational parameters given certain requirements and environmental conditions, and finally, (c) estimating the star detection capability of the assembled unit.

Erlank [7] performed an initial sensitivity analysis for CubeStar optics (which combines the Melexis MLX75412 image sensor and Lensation BL6012 lens), verifying reliable spaceborne operation. Considering the increased reliability demand of multi-algorithm functionality, a more in-depth analysis is performed here.

3.1 Spectral Irradiance and Photon Influx

This section aims to characterize the photon influx for an $M_V = 0$ reference star as a function of wavelength. The significance of such characterization with regard to star tracker detection capabilities will be clarified at a later stage.

The visual magnitude/spectral irradiance of two stars are related by [14]

$$M_{V(x)} - M_{V(ref)} = -2.5 \times \log_{10}\left(\frac{I_x}{I_{ref}}\right) \tag{3.1.1}$$

where $M_{V(x)}$ and $M_{V(ref)}$ represent the respective visual magnitudes of the investigated and reference stars, while I_x and I_{ref} represent their respective irradiance values.

The sun, a star of spectral classification G2 ($T = 5800\,\mathrm{K}$), exhibits a visual magnitude of $M_V = -26.7$ and a solar irradiance of approximately $1370\,\mathrm{W/m^2}$. The sun is a convenient reference for determining the spectral irradiance of other stars due to (a) its relatively close proximity to the earth, allowing solar irradiance values to be measured with reasonable accuracy, and (b) the fact that many bright stars have surface temperatures close to that of the sun.

Using the sun as reference, the total spectral irradiance of an $M_V = 0$ star can be determined as

$$I_x = I_{ref}\ 10^{(M_{V(x)} - M_{V(ref)})/-2.5} \tag{3.1.2}$$

$$= (1370 \times 10^{-6}\,\mathrm{W/mm^2})\ 10^{(-26.7/-2.5)}$$

$$I_{(M_V=0)} = 2.862 \times 10^{-14}\,\mathrm{W/mm^2} \tag{3.1.3}$$

With the total spectral irradiance known, its distribution per wavelength should be investigated.

The incident energy from an $M_V = 0$ star on an area of $1\,\mathrm{mm}^2$ has the same relative spectral characteristic curve as a blackbody radiator. A blackbody can be defined as an object or system that absorbs all radiation incident on its surface and re-emits radiation based on its temperature (thermal radiation). The spectral irradiance of a blackbody is given by Planck's radiation law as [13]

$$I(\lambda, T) = \frac{2\pi hc^2}{\lambda^5\left(e^{hc/\lambda k_b T} - 1\right)} \quad \left[\mathrm{W/m^3}\right] \tag{3.1.4}$$

where I is the spectral irradiance per unit wavelength, λ is the wavelength of radiation (in metre), T is the temperature of the blackbody (in Kelvin), h is Planck's constant at $6.626 \times 10^{-34}\,\mathrm{J\,s}$, c is the speed of light in a vacuum at $2.998 \times 10^8\,\mathrm{m/s}$ and k_b is Boltzmann's constant at $1.38 \times 10^{-23}\,\mathrm{J/K}$. Figure 3.1 shows typical blackbody spectral characteristic curves as a function of temperature and wavelength.

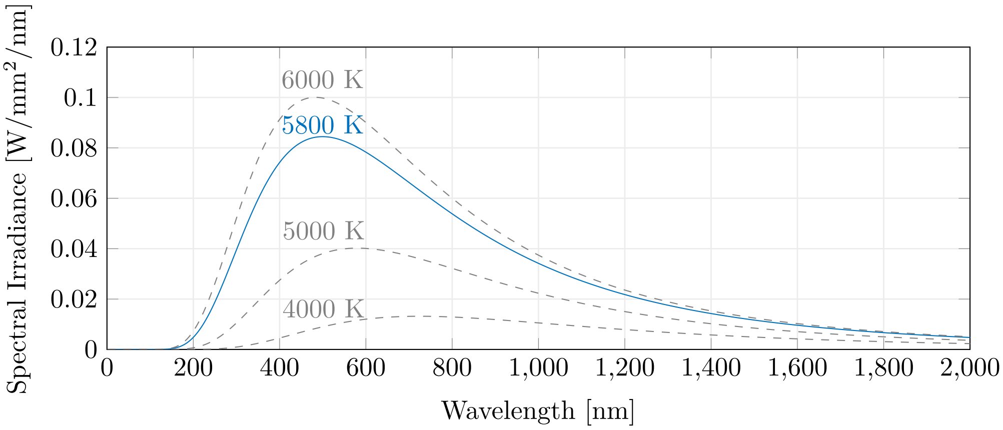


Figure 3.1 – Typical blackbody spectral characteristic curves

With the spectral irradiance of a blackbody known at any given temperature and wavelength, the total irradiance can be determined by integrating eq. (3.1.4) over the entire wavelength range. Recall that an $M_V = 0$ star has the same relative spectral characteristic as a blackbody. The constant of proportionality can therefore be determined by dividing the total irradiance of an $M_V = 0$ star by that of a blackbody. Figure 3.2 shows the resulting spectral characteristic curve for an $M_V = 0$ star. Note that the specified wavelength range is based on the spectral response of the MLX75412 image sensor (see section 3.3).

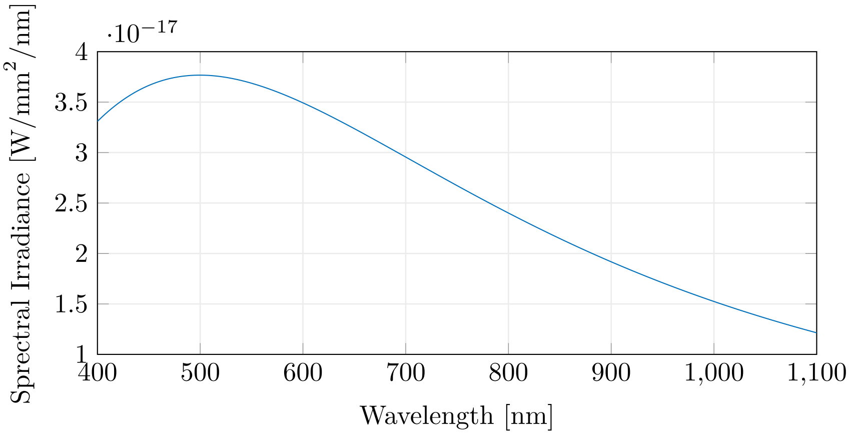


Figure 3.2 – Spectral irradiance from a 5800 K, $M_V = 0$ star

To determine a more quantitative representation, the spectral irradiance can be converted to an equivalent number of photons. Consider the energy of a single photon given by

$$E = \frac{hc}{\lambda} \quad [\mathrm{J}] \tag{3.1.5}$$

where h is Planck's constant, c is the speed of light and λ is the wavelength of the photon (in metre). By dividing the irradiance per wavelength by the energy of a single photon over the entire wavelength range, the spectral characteristic curve can be represented in terms of the number of photons per second as shown in fig. 3.3.

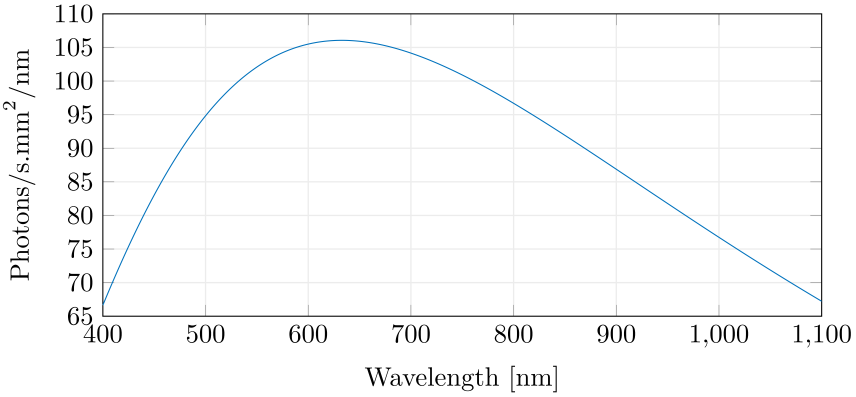


Figure 3.3 – Photon influx from a 5800 K, $M_V = 0$ star on the focal plane

Summating over the entire wavelength range shows that, for an $M_V = 0$ reference star, the total photon influx amounts to **63 623 photons/(s mm²)**.

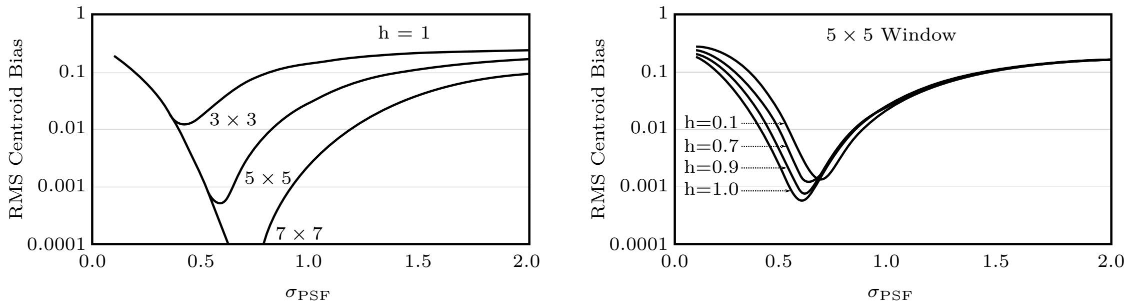
3.2 Optical Defocussing

From the perspective of a star tracker camera, a star is for all practical purposes a point source of light. Imaging with perfectly focussed optics will result in all incoming light being distributed over an area smaller than that of a single pixel. Practically, this introduces a number of complications.

Firstly, confining a star to a single pixel will render it indistinguishable from dead pixels. During spaceborne operation, various factors such as exposure to extreme temperatures, natural component degradation and even radiation can lead to single pixels becoming unresponsive, possibly stuck in either an inactive or active state. Inactive dead pixels will result in stars being masked while their locations, as projected to the image plane, coincide with that of dead pixels. Active dead pixels, on the other hand, could lead to the detection of false stationary stars. Since rates are estimated based exclusively on the displacement of star locations between successive image frames, the detection of such false stationary stars could severely impair stellar gyro functionality.

Secondly, for a single pixel star distribution, image plane searching algorithms (see section 4.1) will require a significant number of computations. In order to guarantee that all stars are extracted from a captured image, the searching algorithm will have to analyse each and every pixel. For the MLX75412 sensor, this amounts to $1024 \times 512 = 524\,288$ computations. Now consider star distribution over a 3×3 pixel grid. All stars can now be extracted by only analysing every third pixel. This reduces the required number computations to $1024/3 \times 512/3 = 58\,255$, nine times less than that of the single pixel distribution case. The number of computations can be decreased even further by increasing the grid size (through means of optical defocussing). Note, however, that an increase in grid size will also reduce the detectable star magnitude.

Finally, single pixel distribution will severely impair star centroiding. A star centroid is typically determined as a weighted function of the photon distribution over all pixels related to that star. Since no information is captured regarding sub-pixel photon distribution, the centroid location will always be identified as falling in the middle of the confining pixel.



(a) RMS centroid bias error as a function of spot size for various centroiding window sizes

(b) RMS centroid bias error as a function of spot size for different photosite sizes

Figure 3.4 – RMS centroid bias error with respect to Gaussian spread for various grid size and fill factor simulations (Data recompiled from [30])

Considering the aforementioned complications, star tracker optics are deliberately defocussed to improve accuracy and reliability. A research publication by Hancock *et al.* [30] determined that optimal star tracker performance can be achieved by defocussing stars over a 5×5 pixel grid, with a corresponding $\sigma_{PSF} = 0.7$ (see fig. 3.4(a)). These defocussing parameters will be used for any further analyses. It is also shown that, for a 5×5 pixel grid size, altering the fill factor has no significant impact on the RMS centroid bias characteristic curve (see fig. 3.4(b)). The fill factor of an image sensor describes the ratio between each pixel's light sensitive area and its total area. Since CMOS sensors (such as the employed Melexis MLX75412) generally exhibit much smaller fill factors than their CCD counterparts, elimination of fill factor as a sensor selection criteria further validates using the CMOS variety.

The theory behind how an image sensor reacts to incoming starlight is rather complex (a detailed discussion can be found in [31]). For the purpose of this thesis, it is sufficient to assume that a star PSF can be approximated by a two-dimensional Gaussian distribution. Figure 3.5 illustrates such approximation using the previously determined standard deviation of $\sigma_{PSF} = 0.7$.

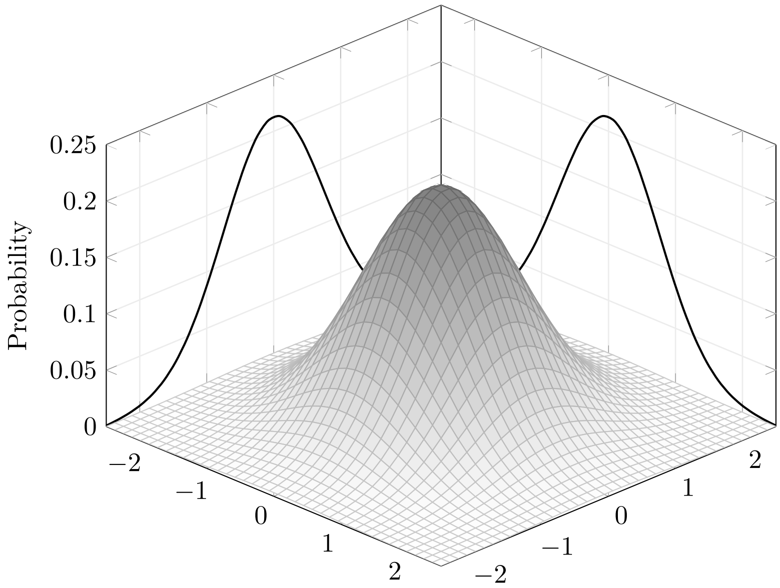
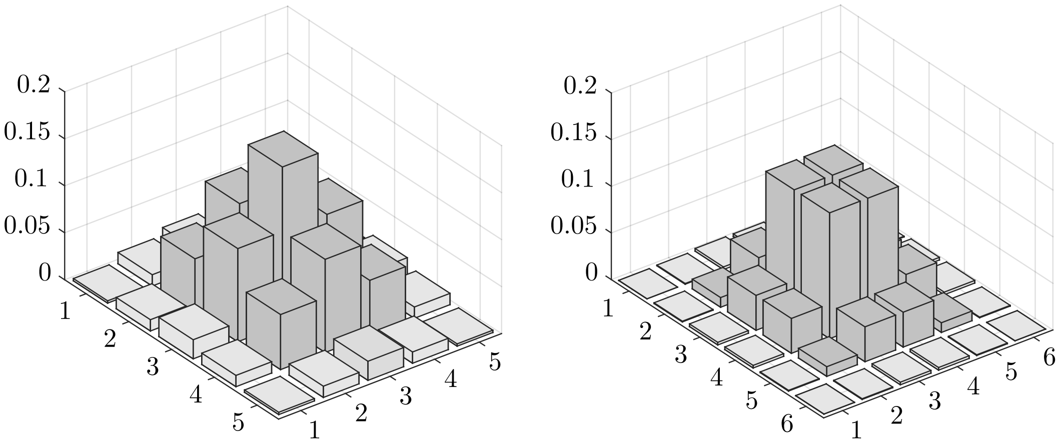


Figure 3.5 – Gaussian photon distribution over 5×5 pixel grid using $\sigma_{PSF} = 0.7$

Note, however, that the minimum resolvable element with respect to photon distribution is the size of a single pixel. In order to determine a meaningful representation of the overall photon distribution, two scenarios were considered. The first scenario, as seen in fig. 3.6(a), involves a projected star centroid falling precisely in the centre of a single pixel. As a result, the majority of the incoming photons will be confined to the centroid pixel, with neighbouring pixels exhibiting a gradual falloff. The second scenario, as seen in fig. 3.6(b), involves a projected star centroid falling on the boundary between four adjacent pixels. In this case, the majority of incoming photons will be evenly distributed over the four boundary pixels, with a much steeper falloff observed for neighbouring pixels. These scenarios represent the best and worst case distributions respectively. Designing star tracker optics based on this worst case distribution should ensure reliable operation throughout.



(a) Photon distribution for star centroid falling on a pixel center

(b) Photon distribution for star centroid falling on the boundary between four adjacent pixels

Figure 3.6 – 3D Visualization of the per pixel photon distribution

As mentioned earlier, one of the primary motivations behind optical defocussing is to enable the star tracker to distinguish between dead pixels/sensor anomalies and actual stars. The detection algorithm therefore also requires a predetermined threshold specifying the minimum number of adjacent pixels falling above the detection threshold that is required to classify the detected pixel group as a star. Iterative simulations showed that specification of a 9 pixel minimum, effectively a 3×3 grid, ensures optimal detection rates. This essentially means that at least 9 pixels per detected pixel group should fall above the detection threshold for a star to be identified. Figure 3.7 shows the per pixel percentile photon distribution.

	5.86	10.0	5.86	
	10.0	16.8	10.0	
	5.86	10.0	5.86	

(a) Photon distribution for star centroid falling on a pixel center

		3.7	3.7		
	3.7	13.5	13.5	3.7	
	3.7	13.5	13.5	3.7	
		3.7	3.7		

(b) Photon distribution for star centroid falling on the boundary between four adjacent pixels

Figure 3.7 – Per pixel percentile photon distribution

For the first scenario, fig. 3.7(a) shows that the photon quantity obtained through summation of 5.86 % of the total incoming photons per star must be greater than the minimum photon quantity specified by the detection threshold (see eq. (3.3.2). Similarly, for the second scenario, fig. 3.7(b) shows that a summation of 3.7 % of the incoming photons must be greater than the detection threshold photon quantity. This detection threshold is characterized in section 3.3.

3.3 Detection Threshold

The detection threshold is a sensor specific parameter, determined by characteristics such as spectral response, sensitivity and responsivity. Figure 3.8 shows the spectral response of the Melexis MLX75412 image sensor.

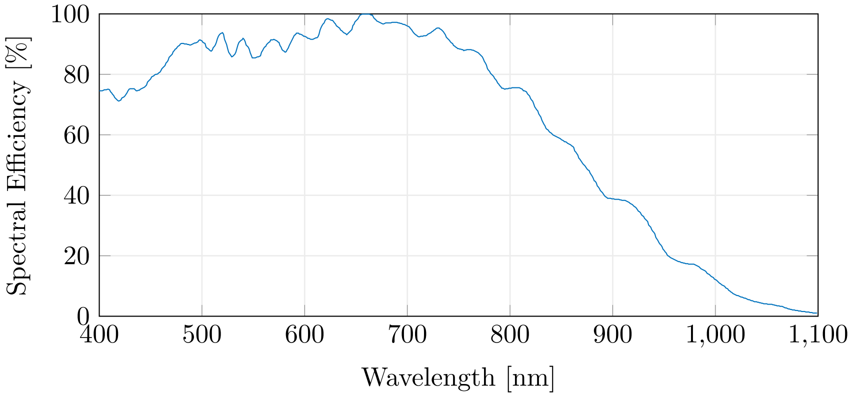


Figure 3.8 – Spectral response of the Melexis MLX75412 image sensor

The effective number of photons per wavelength (i.e., the sensor specific photon influx) can now be determined by multiplying the star photon influx (fig. 3.3) by the sensor specific spectral response (fig. 3.8). The result is shown in fig. 3.9.

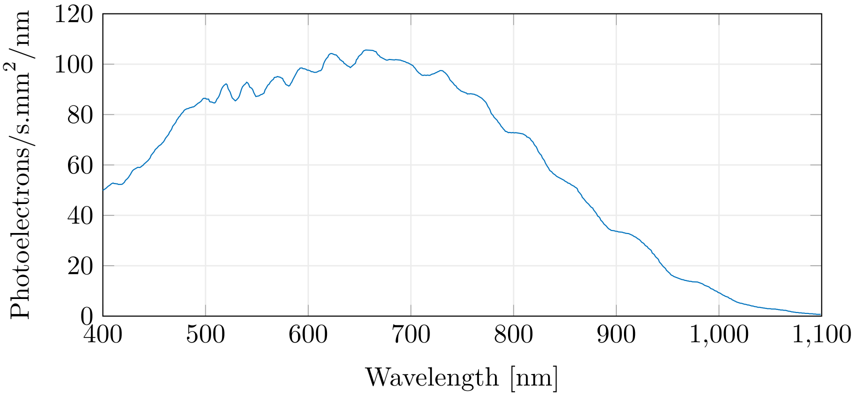


Figure 3.9 – Photons detected by an MLX75412 image sensor from a 5800 K, $M_V = 0$ star

With spectral efficiency now taken into consideration, summating over the entire wavelength range reveals a total photon influx of **42 627.34 photons/(s mm^2)**.

Notice the impact of the image sensor's spectral efficiency on the effective number of photons. Even for the Melexis sensor, which exhibits a comparatively high spectral efficiency, the effective number of photons from an $M_V = 0$ star amounts to a mere 67% of the total incoming photons. It is therefore paramount that the chosen image sensor exhibit a superior spectral efficiency.

In order to fully characterize the detection capabilities of the star tracker optics, it is necessary to determine the sensor specific detection sensitivity. The MLX75412 datasheet specifies a power requirement of 25 nW/cm² @ 535 nm in order to achieve a signal-to-noise ratio (SNR) of 10. In a research publication by Truesdale *et al.* [32] on the development of *DayStar*, a star tracker designed under the University of Colorado Aerospace Capstone Program, SNR6 is determined to be sufficient for detecting stars with visual magnitudes between 4 and 8. Assuming an estimated maximum detection capability of less than $M_V = 4$ for CubeStar (based on the sensitivity analysis performed by Erlank [7]), SNR10 is a conservative benchmark.

Since the output resolution of an image is determined by the number of pixels in the image sensor array, no meaningful information can be obtained by investigating the photon/energy distribution on a sub-pixel level. For the Melexis sensor, which exhibits a $(5.6 \times 5.6)\,\mu\text{m}^2$ pixel size, the image sensor power requirement can therefore be rewritten as

$$(25 \times 10^{-9}\,\text{W/cm}^2)(10^{-8})(5.6^2\,\mu\text{m}^2/\text{pixel}) = 7.84 \times 10^{-15}\,\text{W/pixel} \tag{3.3.1}$$

Using eq. (3.1.5), the energy of a single photon at the benchmark wavelength of 535 nm can be determined as 3.678×10^{-19} J. Dividing the power requirement in eq. (3.3.1) by the energy per photon determines a detection threshold of

$$\frac{7.84 \times 10^{-15}}{3.678 \times 10^{-19}} \, \frac{\cancel{W}\,\text{photons}}{\cancel{W}\,\text{s pixel}} = 21\,313.85\,\text{photons/(s pixel)} \tag{3.3.2}$$

Essentially, this detection threshold value specifies the minimum number of photons required (per second of exposure) to distinguish an illuminated pixel from background sensor noise.

3.4 Lens Selection

Apart from selecting a suitable image sensor, the accompanying lens should also be considered. Not only does the lens determine the achievable sky coverage of the star tracker (see section 3.5), but it also regulates the amount of light that reaches the image sensor array.

A star tracker's achievable sky coverage is dependant on its FOV which, in turn, is jointly dependant on the image sensor dimensions and the lens focal length. The Melexis MLX75412 image sensor, with its 2:1 aspect ratio (5.8016 mm × 2.912 mm), exhibits different horizontal and vertical FOV values. Considering the camera pinhole model illustrated by fig. 3.10, these values can be determined using eq. (3.4.1).

$$\text{FOV}_H = 2\tan^{-1}\left(\frac{d_x}{f}\right) \qquad \text{FOV}_V = 2\tan^{-1}\left(\frac{d_y}{f}\right) \tag{3.4.1}$$

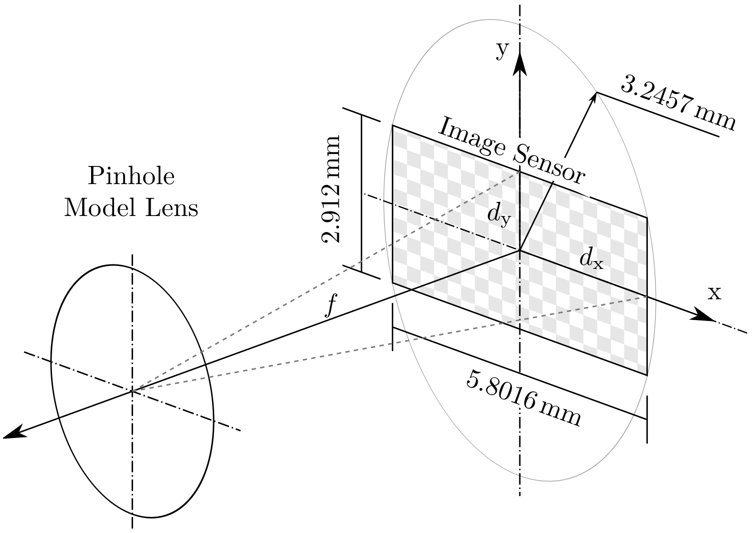


Figure 3.10 – Pinhole model of star tracker optics

Notice that the FOV values are inversely proportional to the lens focal length. Minimizing the focal length will therefore maximize the FOV, consequently also maximizing the average number of stars in the FOV. Theoretically, increasing the average number of stars will improve algorithm accuracy and reliability. Unconditional application of this approach however introduces a practical complication.

Space, as seen from the perspective of a star tracker camera, not only consists of a series of stars, but also of various visual obstructions such as the sun, earth, and space debris. These objects, often brightly illuminated by a relatively close light source or being a light source in itself, are much brighter than the observed stars. When such objects enter the camera FOV they risk saturating the image sensor, impairing star tracker operation in the process. To mitigate such complications, the minimum FOV that guarantees full sky coverage should be determined. This is discussed in more detail in section 3.5.

Combining the horizontal and vertical FOVs, a single value equivalent can be determined in the form of a circular FOV. This facilitates direct comparison between different lenses and image sensor form factors (with regard to sky coverage in particular). The circular FOV can be determined using eq. (3.4.2).

$$\mathrm{FOV}_{circ} = 2\sqrt{\frac{\mathrm{FOV}_H\ \mathrm{FOV}_V}{\pi}} \tag{3.4.2}$$

Due to CubeStar's strict budget limitations, only COTS lenses were considered. Furthermore, lens selection was restricted to a 1/3" optical format, corresponding to that of the MLX75412 sensor. Finally, only s-mount configurations (M12 × 0.5) were considered in an attempt to minimize the weight and size of the CubeStar module. Comprehensive research and comparison during the original CubeStar design procedure narrowed the selection down to two suitable lenses: the Marshall V–4305.7–1.3–A and the Lensation BL6012. The latter was approved for the original CubeStar design. Table 3.1 summarizes the key lens specifications.

Table 3.1 – COTS Lenses considered for implementation CubeStar

Manufacturer	Marshall	Lensation	Lensation
Model	V-4305.7-1.3-A	BL6012	BL4012DN
Image Format	1/3"	1/3"	1/3"
Mount Configuration	s-Mount	s-Mount	s-Mount
Focal Length	5.7 mm	6 mm	4 mm
FOV with MLX75412	54 × 28.6°	51.6 × 27.28°	71.9 × 40°
Circular FOV	44.34°	42.33°	60.513°
Aperture	f/1.4	f/1.2	f/1.2
Weight	5 g	32.5 g	21 g

The Lensation BL4012DN, a wider FOV and therefore more light sensitive solution, was also considered for comparative reasons. Since CubeStar already exhibits a relatively large FOV compared to most commercial alternatives, realistic consideration of such large FOV lens is unlikely. The BL4012DN lens should however be considered for future research, specifically with regard to a stellar gyro designed to operate at higher angular rates.

3.5 Sky Coverage Simulations

Sky coverage, a quantity based jointly on the star tracker FOV, detectable star magnitude and star distribution over the celestial sphere, is a measure of the reliability of star tracker operation. More specifically, sky coverage can be interpreted as the likelihood of a star tracker being able to detect the minimum number of stars required for accurate and reliable operation. Orientation algorithms, which form the foundation of CubeStar, require a minimum of 3 stars in its FOV for accurate and reliable operation. Stellar gyro algorithms, on the other hand, require only 2 stars. Since both sets of algorithms utilize the same hardware, component viability is assessed based on a 3 star coverage.

Figure 3.11 shows the celestial distribution plot for stars up to $M_V = 4$. Even though the magnitude detection threshold of CubeStar is expected to be slightly lower than the plotted maximum, a higher star density serves to emphasize distribution patterns over the celestial sphere. Note that stars are not evenly distributed. The detected number of stars, and therefore also the likelihood of detecting the required number of stars, will be dependant largely on the orientation of the star tracker camera with respect to the celestial sphere.

In order to determine accurate sky coverage estimates, Monte Carlo simulations consisting of 10 000 iterations per FOV were performed. Simulations considered stars in the Hipparcos, Yale Bright Star, and Gliese catalogues, with a minimum of 3 stars required to validate sky coverage. Results are shown in fig. 3.12. The

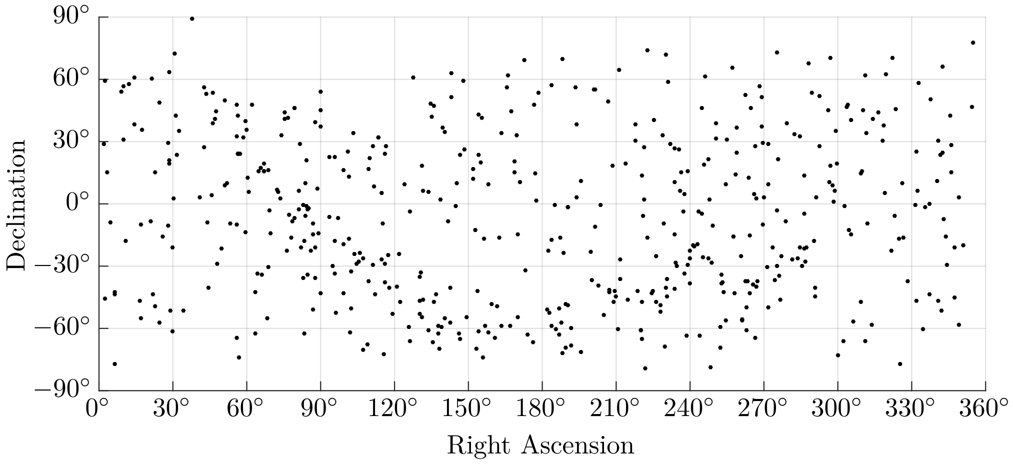


Figure 3.11 – Celestial distribution plot for stars up to $M_V = 4$

detection capability and resultant sky coverage of the assembled CubeStar unit (which combines the Melexis MLX75412 image sensor and Lensation BL6012 lens) is characterized in Section 3.6.

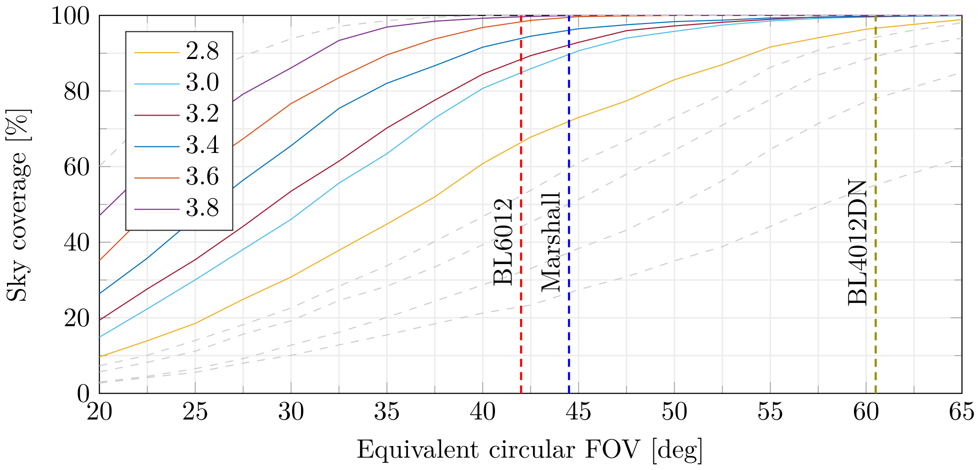


Figure 3.12 – Monte Carlo simulations showing the estimated sky coverage as a function of the FOV for various visual magnitudes

3.6 Detection Capability

With the desired level of optical defocusing and the detection threshold now determined, the sensor specific detection capability can be characterised. Equation (3.3.2) specifies a single pixel detection threshold of 21 313.85 photons/(s pixel). As determined in section 3.2, worst case operation necessitates that the sum of 3.7 % of the total incoming photons must fall above the detection threshold. This allows the total number of photons required for star detection to be determined as

$$\#\ \text{photons} = \frac{21\,313.85}{0.037} = 576\,050\,\text{photons/s} \tag{3.6.1}$$

Recall from section 3.3 that, for the Melexis MLX75412, the sensor specific photon influx from an $M_V = 0$ star amounts to $42\,627.34\,\text{photons/(s mm}^2)$. Since the number photons incident from a star is directly proportional to its spectral irradiance, the ratio given by eq. (3.1.1) can be used to determine the number of photons per unit area based on star magnitude. Taking the exposure time and effective lens area into consideration, the total number of photons reaching the image plane for a star of arbitrary magnitude can be determined as

$$\#\ \text{photons} = \left(42\,627.34\,\frac{\text{photons}}{\text{s mm}^2}\right)\left(10^{M_V/(-2.5)}\right)\left(t_x\,\frac{\text{s}}{\text{exposure}}\right)\left(A_r\ \text{mm}^2\right) \tag{3.6.2}$$

where M_V is the visual magnitude of the star in question, t_x is the image exposure time and A_r is the effective lens area. Figure 3.13 shows the detection threshold superimposed on the total number of photons per magnitude. Simulations assume a one second exposure time and an effective lens area of

$$A_r = \pi r^2 = \pi(7.95)^2 = 198.56\,\text{mm}^2$$

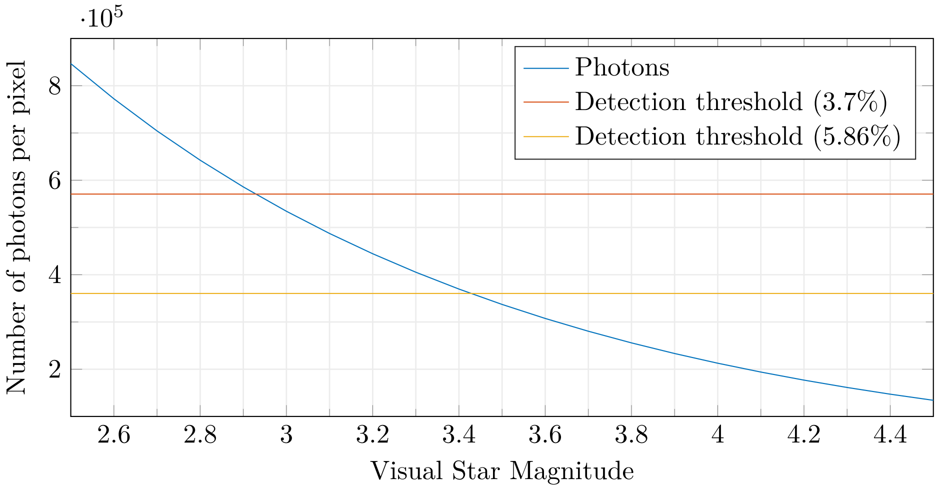


Figure 3.13 – Detection capability of CubeStar's original optical solution (Melexis MLX75412 image sensor with Lensation BL6012 lens

Simulations show that, assuming worst case operation, the detection capability of CubeStar's original optical solution is limited to $M_V = 2.92$ stars. From fig. 3.12, this corresponds to a sky coverage of merely 85 %. Note, however, that various conservative design choices were made, including an absolute worst case photon distribution, overly conservative SNR specification as well as the strict wavelength range limitation. Considering best case photon distribution (represented by the

5.86 % detection threshold) the detection capability is increased to $M_V = 3.42$ stars, corresponding to a sky coverage of 93 %. It is therefore safe to assume that, during spaceborne operation, CubeStar optics should be capable of achieving close to full sky coverage ($> 99\%$).

Remarks

As opposed to the various conservative design decisions, simulations assume exaggerated exposure times. Considering the relatively poor detection capabilities of the MLX75412 sensor, maximizing exposure time is crucial. Fortunately, the fast execution of both orientation (star tracker) and rate determination (stellar gyro) algorithms facilitate such long exposure times, but at the cost of increased power consumption. Since imagers tend to consume far more power during imaging than they do during inactive or standby modes, increased exposure times typically correspond to proportional increases in power consumption. Fortunately, CubeStar was designed with long exposure times in mind, sustaining acceptable levels of power consumption throughout.

Furthermore, simulations fail to account for the effects of rotation on star distribution patterns. Fortunately, due to the relatively low range of operation (angular rates of up to ±1 deg/s), the impact on detection capabilities should be minimal.

In light of these considerations (or lack thereof), future iterations of CubeStar would benefit from implementing a more light sensitive, similarly low powered image sensor. This would facilitate reduced exposure times, thereby also lessing the effects of rotation on star distribution patterns. Key considerations for selecting an alternative star tracker/stellar gyro image sensor are discussed appendix B.

Chapter 4

Algorithms

This chapter presents an overview of the algorithms used throughout the stellar gyro process. To aid understanding, algorithms are presented in order of execution.

4.1 Image Plane Search

Once an image is captured, star locations must be extracted from the raw image data. This is done by comparing pixel values to a predetermined detection threshold (see section 3.3). In order for a star to be extracted, only a single pixel related to that star must be identified as falling above the detection threshold.

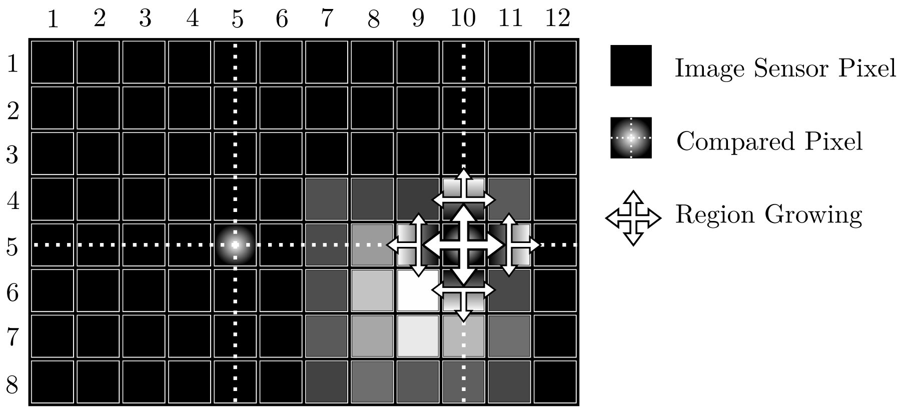


Figure 4.1 – Graphical representation of image plane search and region growing algorithms

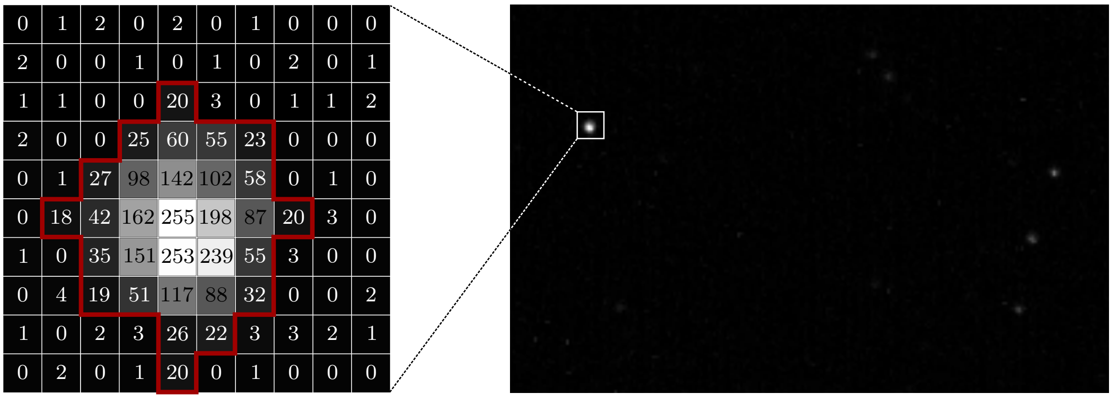
The coordinates of the compared pixels are determined based on the level of optical defocussing (see section 3.2). For star distribution over a 5×5 pixel grid (as illustrated by fig. 4.1), only every fifth pixel need be compared to the detection threshold in order to guarantee the extraction of each and every star. Once a pixel is identified as falling above the detection threshold, a recursive growing algorithm is called with that pixel as seed point. The region growing algorithm is responsible for extracting the remaining pixels associated with that star.

4.2 Region Growing Algorithm

The region growing algorithm is responsible for extracting all pixels related to a star using a region-based image segmentation approach. The process can be outlined as follow:

1. The algorithm is given the coordinates of a single pixel with a value exceeding the detection threshold. This pixel, referred to as a seed point, is added to the region identifying it as being part of a possible star.
2. The value of the seed pixel is set to zero to prevent it from being detected by further recursions of the algorithm.
3. Each pixel adjacent to the seed pixel is compared to the detection threshold (see fig. 4.1).
4. If the value of a pixel exceeds the detection threshold, the algorithm is called recursively with that pixel as seed point.
5. Recursion continues until all pixels related to the possible star have been added to the region.

Figure 4.2 illustrates a star region extracted from a captured image using a threshold value of 14. Note that CubeStar uses 8-bit parallel data communication, allowing for a resolution of $2^8 = 256$ different intensity values.



0	1	2	0	2	0	1	0	0	0
2	0	0	1	0	1	0	2	0	1
1	1	0	0	20	3	0	1	1	2
2	0	0	25	60	55	23	0	0	0
0	1	27	98	142	102	58	0	1	0
0	18	42	162	255	198	87	20	3	0
1	0	35	151	253	239	55	3	0	0
0	4	19	51	117	88	32	0	0	2
1	0	2	3	26	22	3	3	2	1
0	2	0	1	20	0	1	0	0	0

Figure 4.2 – Star region extracted from a night sky image captured by CubeStar

Once extracted, the region is subjected to a further verification step to prevent the detection of false stars. For a region to be representative of a star, the number of pixels within that region must fall within a predetermined allowable range. If the number of pixels fall below the allowable minimum, the region likely indicates either a dead pixel or a star that is too faint for reliable centroiding. If the number of pixels exceed the allowable maximum, the region likely indicates some form of brightly illuminated celestial object other than a star (e.g., the earth, sun or moon). The allowable pixel range is determined based on camera hardware as well as the level of optical defocussing (see section 3.2).

Since centroids are determined based only on the pixels contained within these star regions, the algorithm effectively serves both as a star extraction tool as well as background noise filter. The latter is especially significant where star centroiding is concerned.

4.3 Centroiding Algorithms

Once all star regions have been extracted, the exact star locations corresponding to these regions are determined through sub-pixel centroiding. Centroiding plays an important role in achieving the desired accuracy and reliability for both matching and rate estimation algorithms. Identifying the best suited centroiding method is therefore an important step in the stellar gyro design process.

The original CubeStar design draws on the centre of gravity (CoG) approach for centroiding purposes. Even though the existing implementation of this method proved to be largely successful, a more comprehensive weighted centre of gravity (WCoG) method is also considered in the interest of thoroughness. Other state-of-the-art centroiding methods do exist (e.g., Least Squares Gaussian Fit, and Gaussian Grid [33]), but their implementation tends to be significantly more processor intensive. Erlank [7] deduced that CubeStar should be capable of determining star centroids within 0.2 pixels using the CoG approach, which is sufficiently accurate for stellar gyro implementation.

4.3.1 Centre of Gravity (CoG) Method

The CoG method was originally chosen for implementation on CubeStar. The centroid location is given by

$$(x_c, y_c) = \left(\frac{\sum_{ij} I_{ij} x_{ij}}{\sum_{ij} I_{ij}}, \frac{\sum_{ij} I_{ij} y_{ij}}{\sum_{ij} I_{ij}} \right) \tag{4.3.1}$$

where (x_c, y_c) represents the estimated centroid coordinates, (x_{ij}, y_{ij}) represents the coordinates of each pixel in the star region and I_{ij} represents the respective intensities of those pixels.

The main advantages of the CoG method are its relative simplicity and fast execution. Note that CoG centroiding accuracy is contingent on optimal background noise filtration [21]. For more information regarding the implementation of the CoG algorithm on CubeStar, refer to [7].

4.3.2 Weighted Centre of Gravity (WCoG) Method

The weighted centre of gravity (WCoG) method expands upon the basic CoG approach by incorporating information regarding the point spread function (PSF) through means of a weighting factor. The star centroid (x_c, y_c) can be determined as

$$(x_c, y_c) = \left(\frac{\sum_{ij} W_{ij} I_{ij} x_{ij}}{\sum_{ij} W_{ij} I_{ij}}, \frac{\sum_{ij} W_{ij} I_{ij} y_{ij}}{\sum_{ij} W_{ij} I_{ij}} \right) \tag{4.3.2}$$

where W_{ij} is the approximate value of a two-dimensional Gaussian distribution at coordinates (x_{ij}, y_{ij}) given by

$$W_{ij} = V\ exp\left[-\left(\frac{(x_{ij}-x_c)^2}{2\sigma_x^2}+\frac{(y_{ij}-y_c)^2}{2\sigma_y^2}\right)\right] \tag{4.3.3}$$

The coefficient V is set equal to the maximum intensity value. Note that the weighting function requires an initial estimate of the centroid location. Assuming a largely undistorted PSF, the centroid location can be approximated as the coordinates of the brightest pixel.

Considering the low angular rates at which the proposed stellar gyro is expected to operate, assuming equal standard deviations for random variables x and y ($\sigma_x = \sigma_y = \sigma$) should provide a reasonable although not entirely accurate approximation of the actual distribution. The complexity associated with accurately characterizing the standard deviation values based on angular rates could not be justified at the desired range of operation. Resultantly, the joint standard deviation can be determined as

$$\sigma = \frac{fwhm}{2\sqrt{2\ log(2)}} \tag{4.3.4}$$

where the full width at half maximum ($fwhm$) represents the maximum width of the Gaussian function between points where the function value is equal to half its maximum value (i.e., half the intensity value of the brightest pixel). This value can be approximated by taking the square root of the number of pixels with a value higher than half the maximum value [33].

By incorporating information about the PSF, the WCoG method is theoretically capable determining star centroids more accurately than its CoG counterpart.

Note, however, the reliance of the weighting function on an initial centroid estimate. Recall from eq. (4.3.3) that the centroid coordinates are initially assumed to be the coordinates of the brightest pixel. Considering an ideal PSF, the validity of this assumption is maximised by a unique brightest pixel. Image sensor saturation will increase the number of pixels that exhibit this “brightest” intensity value, diminishing the accuracy of the estimated centroid location in the process. Even though CubeStar was determined to avoid saturation for exposure times shorter than 4.06 s [7], this reliance still makes for a worthwhile consideration where saturation is unavoidable (see section 6.2).

The reliability of the WCoG method is also contingent on the validity of the assumption regarding equal standard deviations. Centroiding accuracy is therefore inversely proportional to the magnitude of the angular rates.

4.4 Distortion Correction

Up to this point algorithms assumed an ideal pinhole lens model. Unfortunately, physical lenses all exhibit some form of optical distortion (i.e., lens error). The most commonly encountered form of optical distortion is radial distortion, which is radially symmetric with respect to the optical axis.

Radial distortion manifests itself as either *barrel* or *pincushion* distortion (see fig. 4.3). Barrel distortion is commonly associated with wide-angle lenses exhibiting relatively short focal lengths, where the FOV is wider than the size of the image sensor. The result is an apparent spherical mapping due to a decrease in image magnification proportional to the distance from the optical axis. Pincushion distortion is the exact opposite of barrel distortion, where image magnification increases proportional to the distance from the optical axis. The effect is an image that appears "pinched" at its centre. This form of distortion is commonly associated with optical solutions where the FOV is smaller than the size of the image sensor.

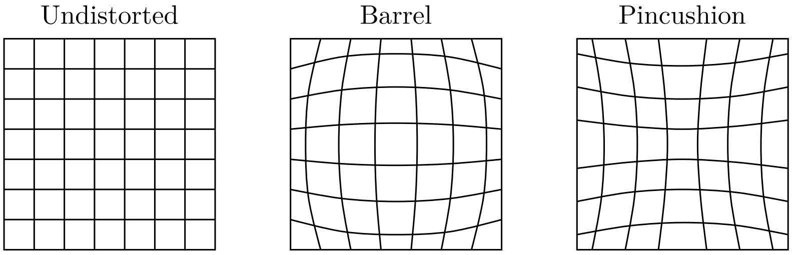


Figure 4.3 – Visual representations of barrel and pincushion distortion

Another form of optical distortion is tangential distortion, resulting from slight misalignments in a lens's optical design or between the lens and image sensor. Even though radial distortion typically dominates distortion models (the effects of radial distortion exceed those of tangential distortion by at least an order of magnitude [34]), accurate distortion models should ideally consider both forms of optical distortion.

A simplified version of Brown's distortion model, which only considers the first two distortion coefficients, was used to perform distortion correction on CubeStar [7]. Consideration of higher order distortion coefficients is only necessary for lenses that exhibit extreme levels of distortion (e.g., fisheye lenses).

Points on the image plane can be undistorted using eqs. (4.4.1) to (4.4.2).

$$x_u = x(1 + K_1r^2 + K_2r^4) + P_2(r^2 + 2x^2) + 2P_1xy \tag{4.4.1}$$

$$y_u = y(1 + K_1r^2 + K_2r^4) + P_1(r^2 + 2y^2) + 2P_2xy \tag{4.4.2}$$

with

$$x = \frac{x_d - x_c}{f} \qquad y = \frac{y_d - y_c}{f}$$

where

x_u, y_u = undistorted image points as projected by ideal pinhole camera

x_d, y_d = distorted image points as projected on image plane

x_c, y_c = distortion centre (assumed to be the principal point)

f = lens focal length

K_n = n^{th} radial distortion coefficient

P_n = n^{th} tangential distortion coefficient

$r = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$

Even though the distortion coefficients were determined for the original CubeStar optics, each and every lens exhibits slightly different intrinsic parameters. The calibration procedure used to determine these coefficients must therefore be repeated for the CubeStar unit used as development platform. The calibration procedure is detailed by Erlank [7].

4.5 Image Plane to Unit Vector Conversion

Once undistorted, star centroids are transformed from image plane to body-fixed coordinates using eq. (4.5.1). A full derivation of this equation is performed by Erlank [7].

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} (x_u - x_c)\frac{pp_x}{f_{mm}} \left[1 + \left((x_u - x_c)\frac{pp_x}{f_{mm}}\right)^2 + \left((y_u - y_c)\frac{pp_y}{f_{mm}}\right)^2\right]^{-\frac{1}{2}} \\ (y_u - y_c)\frac{pp_x}{f_{mm}} \left[1 + \left((x_u - x_c)\frac{pp_x}{f_{mm}}\right)^2 + \left((y_u - y_c)\frac{pp_y}{f_{mm}}\right)^2\right]^{-\frac{1}{2}} \\ \left[1 + \left((x_u - x_c)\frac{pp_x}{f_{mm}}\right)^2 + \left((y_u - y_c)\frac{pp_y}{f_{mm}}\right)^2\right]^{-\frac{1}{2}} \end{bmatrix} \tag{4.5.1}$$

where

u_x, u_y, u_z = components of a unit vector

x_u, y_u = undistorted centroid locations in pixels

x_c, y_c = coordinates of the principal point in pixels

pp_x, pp_y = pixel pitches[1]of the image sensor in mm

f_{mm} = focal length of the lens in mm

It is worth noting that even though star matching can also be performed using image plane coordinates, many rate estimation algorithms necessitate body-fixed unit vector representations of star locations.

4.6 Matching Algorithm

Rate estimation algorithms determine angular rates based on the displacement of a series of centroids between successive image frames. In order to determine such displacement, the algorithm requires information regarding the correspondence of centroid locations between images (i.e., centroid matching).

4.6.1 Star Identification Based Approach

One possibility for centroid matching is through star identification. Using CubeStar's existing star identification algorithms, centroid locations are matched to catalogue stars. Once the stars have been identified, the matching process becomes relatively

[1] The pixel pitch of an image sensor is the distance between the centre points of two adjacent pixels

simple. The efficiency of this method as a means of centroid matching is however questionable.

Star identification algorithms tend to be relatively complex. Orientation procedures necessitate star identification, justifying the execution of such complex algorithms. Rate estimation algorithms, on the other hand, require only centroid locations matched from one image frame to the next. The identity of the stars in question are therefore of no concern to such algorithms, rendering the execution of unnecessarily complicated star identification undesirable. Furthermore, star identification is much less reliable than simply matching centroids between images, as orientation algorithms mandate a high level of certainty with respect to star identity.

4.6.2 RANSAC (Random Sample Consensus)

Rawashdeh *et al.* [5] proposed an alternative solution to the star correspondence problem using a variation of the RANSAC (Random Sample Consensus) algorithm. The approach can be summarized as:

1. **Hypothesize**: The rotation matrix (describing the rotation between successive image frames) is computed using two randomly selected stars from each image frame.

2. **Test**: The rotation matrix is tested for consensus against all remaining stars. Consensus is registered if the hypothesized rotation matrix correctly describes the rotation of a star between image frames, given some measurement of deviation.

3. **Iterate**: The algorithm iterates between the aforementioned steps until consensus is achieved for at least 40% of stars in the first image frame. Stars which showed consensus are then matched.

The design of the RANSAC algorithm is such that it is capable of estimating the parameters of a mathematical model from a set of observed data contaminated by a large number of outliers that do not fit the model [35]. This not only identifies RANSAC as a robust and reliable method for star matching (even when images are particularly noisy), but it also eliminates the need to detect and discard false stars (see section 4.2).

The main disadvantage of RANSAC is that it requires a significantly large number of iterations to find a hypothesis that registers consensus. Rawashdeh and Lumpp [25] suggest that the number of iterations can be drastically reduced by first performing crude matching based on proximity or star brightness. In this case, random star centroids are still selected from the first image frame, but the corresponding centroids in the subsequent frame are selected such that (a) their locations correspond to where the transformed centroid is expected based on previous angular rate estimates, or (b) they exhibit similar brightness values.

Rawashdeh and Lumpp [25] motivate the use of RANSAC over more primative methods of establishing correspondance (e.g., matching purely based on proximity or similar brightness) by referring to the severely detrimental effect false correspondence can have on angular rate estimates. Since simulations based on CubeStar hardware determine flawless correspondence for angular rates up to ±1 deg/s in all axes when using the aforementioned primitive methods (see section 5.3), the complexity and computational burden RANSAC would add to CubeStar could not be justified.

4.6.3 Proximity Based Approach

A proximity based matching algorithm was developed for implementation on CubeStar. The remainder of this section is dedicated to providing a detailed algorithm breakdown.

Note that centroids are matched based on their body-fixed unit vector coordinates (see section 4.5). Even though slightly less intuitive than image plane coordinates with regard to star matching, the unit vector representation retains general applicability regardless of the chosen optical solution (i.e., the distances between unit vector star locations remain fixed, whereas the corresponding distances between their image plane locations are determined by the camera focal length). Any matching parameters determined through simulation (see section 5.3) therefore applies to stellar gyro solutions in general.

Once the centroid lists of two consecutive images have been constructed, centroid matching is initiated. For simplicity, the centroid lists corresponding to the first and second image frames will from here on be referred to as *listA* and *listB* respectively. The matching procedure can be described as follow:

Step 1: Initial centroid matching

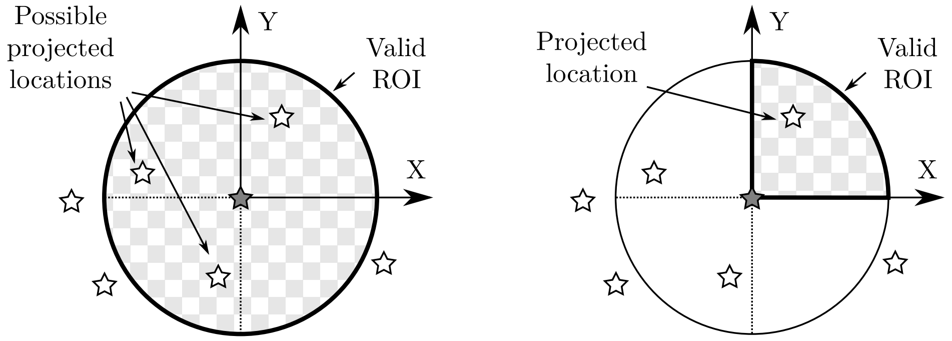
The algorithm uses each centroid location in *listA* as origin for a circular region of interest (ROI) in which its projected location in *listB* can be expected. The size of this ROI, defined as a *search radius*, is based on the maximum angular rates at which the stellar gyro is expected to operate. A more comprehensive analysis of the ROI origin and size is performed in section 5.3. Every centroid in *listB* that falls within the ROI for a centroid in *listA* is identified as a possible match (see fig. 4.4(a)). The indices of both centroids are preserved in the form of a match pair, which is then subjected to an additional verification step based on the availability of previous angular rate estimates. If

(a) no previous rate estimates are available,

a new entry is created in the match list for each possible centroid pair.

(b) previous rate estimates are available,

the algorithm determines the centroid's expected direction of displacement between image frames in both the X- and Y-directions. This reduces the valid ROI by a factor of four (see fig. 4.4(b)). The location of each potential matching centroid is then compared to the reduced ROI. Considering the distribution of stars visible to CubeStar (see section 5.3), comparison with this reduced ROI should eliminate the majority of conflicting matches. Since a body-fixed coordinate system is used, comparison in the Z-direction (which is representative of the camera boresight) will be relatively inaccurate and unreliable.



(a) ROI where no previous rate estimates are available

(b) ROI where previous rate estimates define $\Delta x \geq 0$ and $\Delta y \geq 0$

Figure 4.4 – ROI based on the availability of previous rate estimates

Matching centroids based on previous angular rate estimates introduces a significant complication when confronted with abrupt changes in the direction of rotation. To illustrate this point, consider a scenario where the employed rate estimation algorithm determined a small positive angular rate around either cross-boresight axis. Say now at the next sampling instance that the direction of rotation around that axis has changed since the positive estimate was made. The matching algorithm, unaware of this change in the direction of rotation, erroneously determines each centroid's direction of displacement based on the previous angular rate estimates. Each centroid's correct match will now be discarded due to this verification imposed by the availability of previous rate estimates. A similar complication could occur at almost negligibly small angular rates due to algorithm errors and inaccuracies.

Simulations show that, for angular rates up to 0.5 deg/s, the proposed matching algorithm is capable of flawless matching regardless of the availability of angular rates (see section 5.3). The aforementioned complication can therefore be avoided by assuming that rate information is unavailable if previous rate estimates fall below a predetermined threshold. Following a conservative design approach, a threshold of 0.2 deg/s was chosen. Consequently, the matching algorithm will assume that rate information is unavailable at any point where previous rate estimates fall between ±0.2 deg/s.

Step 2: Second matching phase

If the initial matching phase was unable to identify a possible match for any centroid in *listA*, the algorithm once again assesses the availability of previous rate estimates. If

(a) no previous rate estimates are available,

the algorithm abandons the matching process for the centroid in question.

(b) previous rate estimates are available,

the algorithm determines the expected location of the centroid as projected to the second frame based on these rate estimates. Rate estimation algorithms assume very small changes in angular rates from one sampling instance to the next. Previous rate estimates should therefore provide a reasonably accurate estimate of the projected centroid location. The matching process detailed in **Step 1** is repeated with the expected location as the new ROI origin. Furthermore, a larger search radius is also used during the second iteration to improve the likelihood of finding a match. The comprehensive verification procedure associated with the availability of angular rates ensure that matching reliability is not compromised through use of a larger search radius.

Figure 4.5 illustrates an example in which the projected ROI succeeds in identifying the correct match where the original ROI failed to do so. If the second iteration also fails to identify a possible match, the matching process is abandoned for the centroid in question.

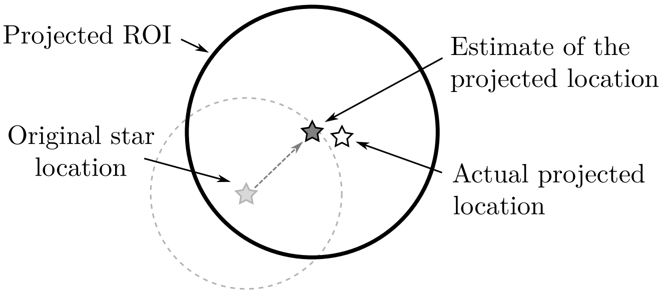


Figure 4.5 – ROI based on the estimate of the projected centroid location

Step 3: Resolve *listA* matching disputes

Once the matching process for a centroid in *listA* is completed, the number of entries in the match list associated with that centroid is determined. If the match list contains more than one entry for the centroid in question (i.e., one or more centroids from *listB* have been erroneously matched to the centroid in *listA*), the match dispute is resolved based on the Euclidean distance between the location of each matched centroid in *listB* and the best estimate of the current location of the centroid in *listA*. If

(a) no previous rate estimates are available,

the best estimate of the current location is the existing entry in *listA*.

(b) previous rate estimates are available,

the best estimate of the current location is the projected location of the centroid onto the second frame. As was explained in **Step 2**, previous rate estimates can be used to determine a reasonably accurate projection of a centroid onto the second frame.

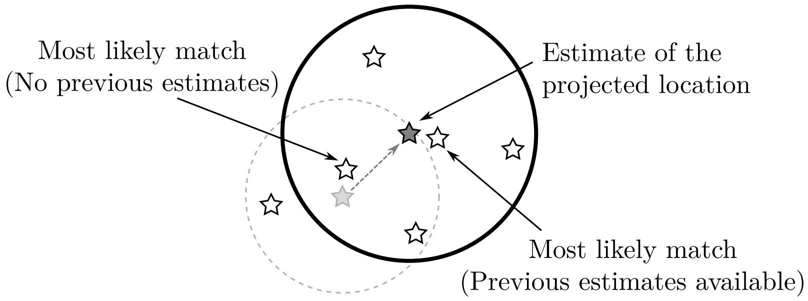


Figure 4.6 – Resolve *listA* matching dispute

As illustrated by fig. 4.6, the matched centroid closest to the best estimate of the current location is identified as the most likely match, after which the other less suitable entries are discarded.

Step 4: Resolve *listB* matching disputes

The match list has now been resolved to the extent that it contains a maximum of one match pair for each centroid in *listA*. The final step is to confirm that no centroid from *listB* has been matched to more than one centroid from *listA*. This is done by comparing the index value of each entry in *listB* to the index value of every other entry in an attempt to find a possible duplicate. If more than one match is identified for any centroid in *listB*, the match dispute is resolved in a manner similar to that detailed in **Step 3**. As shown in fig. 4.7, the match pair for which the centroid in *listB* is closest to the best estimate of the current location of the corresponding centroid in *listA* is identified as the most likely match, after which the other less suitable entries are discarded.

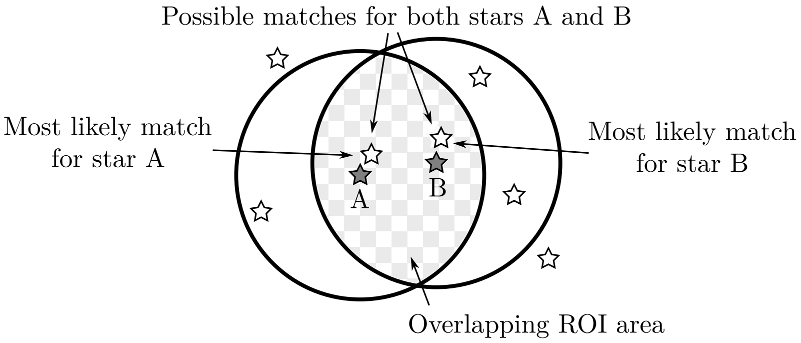


Figure 4.7 – Resolve *listB* matching dispute

The algorithm was tested for reliability through simulation in the MATLAB environment. Results are shown in chapter 5.

4.7 Significant Developments in the Field of Satellite Attitude Determination

Satellite attitude determination is an overarching term that describes the process of measuring and calculating vector data using a combination of physical sensors and mathematical models, after which satellite attitude is determined by subjecting said data to various estimation algorithms. Attitude is typically described using either rotation matrices, Euler angles or quaternions, each providing their own advantages in terms of analysis and application (see section 2.4). Considering the immense accuracy requirements and low tolerance for errors mandated by spaceborne operation, the importance of accurate and reliable attitude determination can not be overstated.

Significant amongst attitude determination algorithms are the rate estimation techniques employed by stellar gyroscopes. Comprehension of such techniques, however, requires some background regarding the development of attitude determination algorithms in general.

4.7.1 TRIAD Algorithm

The TRIAD algorithm, presented by Harold Black in 1964 [36], is one of the earliest and simplest solutions to the problem of spacecraft attitude determination. What makes the TRIAD algorithm distinct from its predecessors is its methodology, being the first algorithm to use body and reference observations for attitude determination. The interpretation presented here is based on a related textbook publication by Markley and Crassidis [37] entitled *Fundamentals of Spacecraft Attitude Determination and Control.*

Fundamentally, the TRIAD algorithm attempts to find the rotation matrix $\mathbf{A}$ that transforms a set of vectors from a reference frame to the spacecraft body frame, i.e.,

$$\mathbf{b}_i = \mathbf{A}\mathbf{r}_i \qquad \text{for} \qquad i = 1, 2 \tag{4.7.1}$$

where $\mathbf{b}_i$ and $\mathbf{r}_i$ denote the vectors in the reference and body frames respectively. Equation (4.7.1) implies that

$$\mathbf{b}_1 \bullet \mathbf{b}_2 = (\mathbf{A}\mathbf{r}_1) \bullet (\mathbf{A}\mathbf{r}_2) = \mathbf{r}_1^T \mathbf{A}^T \mathbf{A}\mathbf{r}_2 = \mathbf{r}_1 \bullet \mathbf{r}_2 \tag{4.7.2}$$

which is generally not possible in the presence of measurement noise. In order for eq. (4.7.1) to be true for both sets of vectors, eq. (4.7.2) must be obeyed. TRIAD proposes a solution to this problem by stating that, given the corresponding orthonormal vector triads $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ and $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ in the reference and spacecraft body frames respectively, the attitude matrix

$$\mathbf{A} \equiv \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \mathbf{w}_3 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix}^T = \sum_{i=1}^{3} \mathbf{w}_i \mathbf{v}_i^T \tag{4.7.3}$$

will transform $\mathbf{v}_i$ to $\mathbf{w}_i$ by

$$\mathbf{w}_i = \mathbf{A}\mathbf{v}_i \qquad \text{for} \qquad i = 1, 2, 3 \tag{4.7.4}$$

The triads $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ and $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ are formed from the reference and spacecraft body frame vector sets respectively according to

$$\mathbf{v}_1 = \mathbf{r}_1 \qquad \mathbf{v}_2 = \mathbf{r}_\times \equiv \frac{\mathbf{r}_1 \times \mathbf{r}_2}{||\mathbf{r}_1 \times \mathbf{r}_2||} \qquad \mathbf{v}_3 = \mathbf{r}_1 \times \mathbf{r}_\times \tag{4.7.5a}$$

$$\mathbf{w}_1 = \mathbf{b}_1 \qquad \mathbf{w}_2 = \mathbf{b}_\times \equiv \frac{\mathbf{b}_1 \times \mathbf{b}_2}{||\mathbf{b}_1 \times \mathbf{b}_2||} \qquad \mathbf{w}_3 = \mathbf{b}_1 \times \mathbf{b}_\times \tag{4.7.5b}$$

An estimate of the attitude matrix can be obtained by substituting eqs. (4.7.5a) to (4.7.5b) into eq. (4.7.3), giving

$$\hat{\mathbf{A}}_{\text{TRIAD}} = \mathbf{b}_1\mathbf{r}_1^T + (\mathbf{b}_1 \times \mathbf{b}_\times)(\mathbf{r}_1 \times \mathbf{r}_\times)^T + \mathbf{b}_\times\mathbf{r}_\times^T \tag{4.7.6}$$

Some algebraic manipulation shows that eq. (4.7.6) satisfies eq. (4.7.1) for both vector sets if and only if eq. (4.7.2) is obeyed. The derivation above highlights the simplicity of the TRIAD algorithm. Even though various algorithms have since surpassed TRIAD in terms of functionality and usability, its simplicity and efficiency still encourage further research regarding its application.

4.7.2 Whaba's Problem

The main disadvantage of the TRIAD algorithm is the limitation of only being able to combine the information of two measurements. Whaba's Problem, published by Grace Whaba in 1965 [38], improves upon the TRIAD algorithm by posing a three-axis attitude determination problem that seeks to find a proper orthogonal attitude matrix which brings two vector sets, each containing n independent vector entries, into least squares coincidence with one another. In other words, Whaba's Problem is concerned with finding the optimal rotation matrix between two reference frames that will minimize the cost function

$$\mathrm{L}(\mathbf{A}) \equiv \frac{1}{2}\sum_{i=1}^{N} a_i||\mathbf{b}_i - \mathbf{A}\mathbf{r}_i||^2 \tag{4.7.7}$$

Note that, unlike the TRIAD algorithm, eq. (4.7.7) facilitates the integration of an arbitrary number of measurements, as well as the arbitrary weighting of such measurements.

Considering the orthogonality of the attitude matrix $\mathbf{A}$ and the resultant unit norm of the body and reference unit vectors, the latter part of eq. (4.7.7) can be rewritten as

$$\begin{aligned} ||\mathbf{b}_i - \mathbf{A}\mathbf{r}_i||^2 &= ||\mathbf{b}_i||^2 + ||\mathbf{A}\mathbf{r}_i||^2 - 2\mathbf{b}_i \bullet (\mathbf{A}\mathbf{r}_i) && (4.7.8)\\ &= 2 - 2\mathbf{b}_i \bullet (\mathbf{A}\mathbf{r}_i) \\ ||\mathbf{b}_i - \mathbf{A}\mathbf{r}_i||^2 &= 2 - 2\mathrm{tr}(\mathbf{A}\mathbf{r}_i\mathbf{b}_i^T) && (4.7.9) \end{aligned}$$

The trace of the product of matrices functions similarly to the dot-product of vectors. This similarity, along with the inherent cyclic invariance of the matrix trace, validates eq. (4.7.9).

Substituting eq. (4.7.9) into eq. (4.7.7) gives

$$\begin{aligned} \mathrm{L}(\mathbf{A}) &\equiv \frac{1}{2}\sum_{i=1}^{N} a_i\Big(2 - 2\,\mathrm{tr}(\mathbf{A}\mathbf{r}_i\mathbf{b}_i^T)\Big) && (4.7.10) \\ &\equiv \sum_{i=1}^{N} a_i - \mathrm{tr}\left(\sum_{i=1}^{N} a_i\mathbf{A}\mathbf{r}_i\mathbf{b}_i^T\right) \\ \mathrm{L}(\mathbf{A}) &\equiv \lambda_0 - \mathrm{tr}(\mathbf{A}\mathbf{B}^T) && (4.7.11) \end{aligned}$$

where

$$\lambda_0 = \sum_{i=1}^{N} a_i \qquad \text{and} \qquad \mathbf{B} = \sum_{i=1}^{N} a_i\mathbf{b}_i\mathbf{r}_i^T \tag{4.7.12}$$

It is worth noting that the loss function presented in eq. (4.7.7) reveals no intuitive approach regarding its minimization, whereas the rewritten form of eq. (4.7.11) clearly shows that maximization of the term $\mathrm{tr}(\mathbf{A}\mathbf{B}^T)$ will minimize the loss function in question.

Assuming the availability of at least one observation, the weights can be normalized to give

$$\lambda_0 = \sum_{i=1}^{N} a_i = 1 \tag{4.7.13}$$

This allows for the simplification of eq. (4.7.11) to

$$\mathrm{L}(\mathbf{A}) \equiv 1 - \mathrm{tr}(\mathbf{A}\mathbf{B}^T) \tag{4.7.14}$$

In order to retain the general applicability of the proposed estimation algorithms, the cost function presented in eq. (4.7.11) will be used for further derivations unless otherwise specified.

4.8 Rate Estimation Algorithms

The fundamental concern of Whaba's problem is to find the optimal rotation matrix between two reference frames. Solutions to this problem typically focus on orientation, concerned with determining the rotation matrix between body-fixed and inertially referenced frames. Fortunately, due to the general applicability of such solutions, algorithms can easily be repurposed to determine angular rates. By specifying two consecutive star images as reference frames, the rotation matrix now describes the angular displacement between such images. Given the delay between successive captures, the process of determining angular rates becomes relatively simple. The algorithms considered for stellar gyro implementation are detailed here.

4.8.1 q-Method

Paul Davenport provided the first useful solution to Whaba's problem with respect to spacecraft attitude determination. His solution was first used onboard the NASA

High Energy Astronomy Observatory satellites (HEAO–1,2,3) launched in 1978 [39]. The q-method employs the quaternion representation

$$\mathbf{q}(\mathbf{e},\theta) = \begin{bmatrix} \mathbf{q}_{1:3} \\ q_4 \end{bmatrix} = \begin{bmatrix} \mathbf{e}\,\sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} \tag{4.8.1}$$

Equation (4.8.1) allows for the quaternion parametrization of the attitude matrix as

$$\mathbf{A}(\mathbf{q}) = (q_4^2 - ||\mathbf{q}_{1:3}||^2)\ \mathbf{I}_{3\times3} - 2q_4[\mathbf{q}_{1:3}\times] + 2\mathbf{q}_{1:3}\mathbf{q}_{1:3}^T \tag{4.8.2}$$

Substituting eq. (4.8.2) into eq. (4.7.11), followed by considerable matrix algebra [40], gives

$$\mathrm{L}(\mathbf{A}(\mathbf{q})) = \lambda_0 - \mathbf{q}^T\mathbf{K}(\mathbf{B})\mathbf{q} \tag{4.8.3}$$

where $\mathbf{K}(\mathbf{B})$ is the symmetric traceless matrix

$$\mathbf{K}(\mathbf{B}) \equiv \begin{bmatrix} \mathbf{B} + \mathbf{B}^T - (\mathrm{tr}\mathbf{B})\mathbf{I}_{3\times3} & \mathbf{z} \\ \mathbf{z}^T & \mathrm{tr}\mathbf{B} \end{bmatrix} \tag{4.8.4}$$

with

$$\mathbf{z} \equiv \sum_{i=1}^{N} a_i \mathbf{b}_i \times \mathbf{r}_i = \begin{bmatrix} B_{23} - B_{32} \\ B_{31} - B_{13} \\ B_{12} - B_{21} \end{bmatrix} \tag{4.8.5}$$

Using Lagrange multipliers, the optimal solution (which maximizes the term $\mathbf{q}^T\mathbf{K}(\mathbf{B})\mathbf{q}$) is shown to be a quaternion that is the normalized eigenvector of $\mathbf{K}(\mathbf{B})$ corresponding to the largest positive eigenvalue, i.e., the solution of

$$\mathbf{K}(\mathbf{B})\mathbf{q}_{opt} \equiv \lambda_{max}\mathbf{q}_{opt} \tag{4.8.6}$$

The nature of the q-Method is such that it determines the quaternion representation of the attitude matrix rather than the attitude matrix itself.

4.8.2 Singular Value Decomposition

In 1987, Markley [41] provided an alternative solution to Whaba's Problem based on the singular value decomposition (SVD) of a 3 × 3 matrix. Even though significantly more computationally expensive than the method proposed by Davenport, SVD provides valuable analytical insight through the determination of the eigenvalues and eigenvectors of the covariance matrix in addition to the desired attitude estimate. These values allow for a more in-depth analysis of attitude uncertainty.

The singular value decomposition of the matrix $\mathbf{B}$ is given by

$$\mathbf{B} = \mathbf{USV}^T \tag{4.8.7}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and

$$\mathbf{S} = \mathrm{diag}(s_1, s_2, s_3) \qquad \text{with} \qquad s_1 \geq s_2 \geq s_3 \geq 0 \tag{4.8.8}$$

a diagonal matrix containing the singular values of matrix $\mathbf{B}$.

At this point, it is convenient to define (see appendix A for the derivation of eq. (4.8.10))

$$\mathbf{U}_+ = \mathbf{U}[\mathrm{diag}(1, 1, |\mathbf{U}|)] \qquad \text{and} \qquad \mathbf{V}_+ = \mathbf{V}[\mathrm{diag}(1, 1, |\mathbf{V}|)] \tag{4.8.9}$$

with

$$\mathbf{W} \equiv \mathbf{U}_+^T\mathbf{A}\mathbf{V}_+ = \cos\theta\ \mathbf{I}_{3\times3} + (1 - \cos\theta)\mathbf{e}\mathbf{e}^T - \sin\theta[\mathbf{e}\times] \tag{4.8.10}$$

Also define the diagonal matrix

$$\mathbf{S}' = \mathrm{diag}(s_1, s_2, |\mathbf{U}||\mathbf{V}|s_3) \tag{4.8.11}$$

The singular value decomposition of matrix $\mathbf{B}$ (eq. (4.8.7)) can now be written as

$$\mathbf{B} = \mathbf{U}_+\mathbf{S}'\mathbf{V}_+^T \tag{4.8.12}$$

Substituting eq. (4.8.12) into Whaba's loss function (eq. (4.7.14)) gives

$$\mathrm{L}(\mathbf{A}) = 1 - \mathrm{tr}(\mathbf{A}\mathbf{U}_+^T\mathbf{S}'\mathbf{V}_+) \tag{4.8.13}$$

Using the cyclic invariance of the matrix trace gives

$$\mathrm{L}(\mathbf{A}) = 1 - \mathrm{tr}(\mathbf{S}'\mathbf{U}_+^T\mathbf{A}\mathbf{V}_+) \tag{4.8.14}$$

$$= 1 - \mathrm{tr}(\mathbf{S}'\mathbf{W})$$

$$\mathrm{L}(\mathbf{A}) = 1 - \mathrm{tr}\mathbf{S}' + (1 - \cos\theta)[s_2 + |\mathbf{U}||\mathbf{V}|s_3 + (s_1 - s_2)e_2^2 + (s_1 - |\mathbf{U}||\mathbf{V}|s_3)e_3^2] \tag{4.8.15}$$

Considering the singular value characterization of $s_1 \geq s_2 \geq s_3 \geq 0$, it is seen that the minimization of $\mathrm{L}(\mathbf{A})$ can be achieved by specifying $\theta = 0$, reducing eq. (4.8.15) to

$$\mathrm{L}(\mathbf{A}_{opt}) = 1 - \mathrm{tr}\mathbf{S}' = 1 - s_1 - s_2 - |\mathbf{U}||\mathbf{V}|s_3 \tag{4.8.16}$$

Substitution of the angle condition $\theta = 0$ into eq. (4.8.10) gives

$$\mathbf{W} \equiv \mathbf{U}_+^T\mathbf{A}_{opt}\mathbf{V}_+ = \cos(0)\ \mathbf{I} + (1 - \cos(0))\mathbf{e}\mathbf{e}^T - \sin(0)[\mathbf{e}\times] \tag{4.8.17}$$

$$\mathbf{U}_+^T\mathbf{A}_{opt}\mathbf{V}_+ = \mathbf{I}$$

$$\mathbf{A}_{opt} = \mathbf{U}_+\mathbf{V}_+^T = \mathbf{U}[\mathrm{diag}(1, 1, |\mathbf{U}||\mathbf{V}|)]\mathbf{V}^T \tag{4.8.18}$$

The resultant optimal rotation matrix $\mathbf{A}_{opt}$ represents the transformation from reference to body coordinates. Alternatively, from a stellar gyro perspective, $\mathbf{A}_{opt}$ represents the optimal rotation of set of points from one image frame to the next. Note that, unlike the preceding q-Method, SVD solves the attitude matrix directly.

4.8.3 Weighted Least-Squares Estimation

Crassidis [26] proposed a weighted least-squares approach to determine the satellite angular velocity directly from star tracker body measurements. Consider the following attitude measurement model at time t_k:

$$\tilde{\mathbf{b}}_i(k) = \mathbf{A}(k)\mathbf{r}_i + \mathbf{v}_i(k) \tag{4.8.19}$$

where $\widetilde{\mathbf{b}}_i(k)$ is the i^{th} body measurement vector, $\mathbf{A}(k)$ is the proper-orthogonal attitude matrix, $\mathbf{r}_i$ is the star reference vector (an inertially fixed, time independent representation), and $\mathbf{v}_i(k)$ is the measurement noise given by a zero-mean Gaussian white-noise process. Shuster [42] came to the conclusion that, from a practical standpoint, the probability density on a sphere is indistinguishable from the corresponding density on a tangent plane as far as star trackers are concerned. The measurement noise covariance can therefore be represented as

$$\boldsymbol{\Omega}_v = E\Big\{\mathbf{v}_i(k)\mathbf{v}_i^T(k)\Big\} = \sigma_i^2\mathbf{I}_{3\times3} \tag{4.8.20}$$

Taking the difference between successive measurements of eq. (4.8.19) gives

$$\widetilde{\mathbf{b}}_i(k+1) - \widetilde{\mathbf{b}}_i(k) = [\mathbf{A}(k+1) - \mathbf{A}(k)]\mathbf{r}_i + \mathbf{v}_i(k+1) - \mathbf{v}_i(k) \tag{4.8.21}$$

It can be shown that, for small rotation angles between successive measurements, the relation between successive attitude matrices is given by the first-order approximation

$$\mathbf{A}(k+1) \approx \Big(\mathbf{I}_{3\times3} - \Delta t[\boldsymbol{\omega}(k)\times]\Big)\mathbf{A}(k) \tag{4.8.22}$$

where

$$[\boldsymbol{\omega}(k)\times] = \begin{bmatrix} 0 & -\omega_3(k) & \omega_2(k) \\ \omega_3(k) & 0 & -\omega_1(k) \\ -\omega_2(k) & \omega_1(k) & 0 \end{bmatrix}$$

A comprehensive derivation of this approximation can be found in appendix A. Substituting eq. (4.8.22) into eq. (4.8.21) gives

$$\widetilde{\mathbf{b}}_i(k+1) - \widetilde{\mathbf{b}}_i(k) = -\Delta t[\boldsymbol{\omega}(k)\times]\mathbf{A}(k)\mathbf{r}_i + \mathbf{v}_i(k+1) - \mathbf{v}_i(k) \tag{4.8.23}$$

The next step is to determine an angular velocity estimate independent of attitude and reference vector. By eliminating the reference vector, the angular velocity can be estimated by comparing only the information available in successive images. This in turn eliminates the need for star matching (as well as the need for star catalogue data), greatly reducing algorithm complexity and execution time. As described in section 4.6, dedicated matching algorithms can also be much more reliable than star identification algorithms.

The reference vector can be eliminated by solving eq. (4.8.19) in terms of $\mathbf{A}(k)\mathbf{r}_i$ and substituting the result into eq. (4.8.23), defining the new measurement model as

$$\frac{1}{\Delta t}[\widetilde{\mathbf{b}}_i(k+1) - \widetilde{\mathbf{b}}_i(k)] = [\widetilde{\mathbf{b}}_i(k)\times]\boldsymbol{\omega}(k) + \mathbf{w}_i(k) \tag{4.8.24}$$

where $\mathbf{w}_i(k)$ is the new effective measurement noise given by

$$\mathbf{w}_i(k) \equiv [\boldsymbol{\omega}(k)\times]\mathbf{v}_i(k) + \frac{1}{\Delta t}[\mathbf{v}_i(k+1) - \mathbf{v}_i(k)] \tag{4.8.25}$$

Assuming a stationary noise process $\mathbf{v}_i$, the new measurement noise covariance expression can be derived as

$$\boldsymbol{\Omega}_w = E\{\mathbf{w}_i(k)\mathbf{w}_i^T(k)\} = \sigma_i^2[\boldsymbol{\omega}(k)\times][\boldsymbol{\omega}(k)\times]^T + \left(\frac{2\sigma_i^2}{\Delta t^2}\right)\mathbf{I}_{3\times3} \tag{4.8.26}$$

Recall that the first-order approximation of eq. (4.8.22) assumed small rotation angles between successive measurements. It is therefore safe to assume that the bandwidth of the attitude variations is well below the Nyquist frequency with a safety factor of 10, that is

$$||\boldsymbol{\omega}(k)|| < \frac{2\pi}{10} \cdot \frac{f_s}{2} \tag{4.8.27}$$

Normalization of the frequency content allows eq. (4.8.27) to be written as

$$||\boldsymbol{\omega}(k)|| < \frac{1}{\Delta t} \cdot \frac{2\pi(0.5)}{10} \tag{4.8.28}$$

Combining eq. (4.8.28) with the equality $||[\boldsymbol{\omega}(k)\times]|| = ||\boldsymbol{\omega}(k)||$ as defined by the two-norm of matrices results in the inequality

$$||\sigma_i^2[\boldsymbol{\omega}(k)\times][\boldsymbol{\omega}(k)\times]^T|| < \frac{\pi^2}{100} \cdot \frac{\sigma_i^2}{\Delta t^2} \ll \frac{2\sigma_i^2}{\Delta t^2} \tag{4.8.29}$$

From eq. (4.8.29) it is clear that the measurement noise covariance expression is dominated by the second term. Equation (4.8.26) can therefore be approximated as

$$E\{\mathbf{w}_i(k)\mathbf{w}_i^T(k)\} \approx \left(\frac{2\sigma_i^2}{\Delta t^2}\right)\mathbf{I}_{3\times 3} \tag{4.8.30}$$

The significance of this approximation will become clear at a later stage. We now seek to find the least-squares approximation corresponding to the measurement model derived in eq. (4.8.24). Note that this measurement model adheres to the overdetermined form

$$y_i = \sum_{j=1}^{N} X_{ij}\beta_j + \varepsilon_j \qquad (i = 1, 2, ..., m) \tag{4.8.31}$$

For simplicity, the equivalent matrix representation is given by

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{4.8.32}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{m1} & X_{m2} & \cdots & X_{mn} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix} \tag{4.8.33}$$

The general least-squares (GLS) approach is concerned with finding the coefficients β resulting in the best approximation of eq. (4.8.32) by minimizing the sum of the squared residuals, i.e., estimation via optimization of the minimization problem [43]

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 \tag{4.8.34}$$

It can be shown that the optimization problem has a unique solution, provided that the columns of matrix $\mathbf{X}$ are linearly independent, given by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{4.8.35}$$

A special case of the GLS method, called weighted least-squares (WLS), occurs when the off-diagonal entries of the measurement error covariance matrix Ω are zero [44]. The significance of eq. (4.8.30) now becomes clear, as it verifies the adherence of the measurement model to the aforementioned requirement. The WLS case concerns itself with the optimization of the minimization problem

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} ||\sqrt{\mathbf{W}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})||^2 \tag{4.8.36}$$

Note that the aforementioned condition for WLS classification is based on the assumption that the measurement errors (1) are uncorrelated with each other and with the independent variables and (2) have equal variances. The Gauss-Markov theorem [45] states that, for a linear regression model (as is the concern of the method of least-squares) in which the errors (1) have expectation zero, (2) are uncorrelated and (3) have equal variances, the best linear unbiased estimator of the coefficients is given by the ordinary least squares estimator ($\hat{\boldsymbol{\beta}}$). The measurement model in question adheres to the first two conditions, but generally exhibits different uncertainties for each measurement. The work of renown mathematician Alexander Aitken [46] addresses this problem by showing that, when the weighted sum of the residuals is minimized, the ordinary least-squares estimator ($\hat{\boldsymbol{\beta}}$) is a best linear unbiased estimator if the weights are equal to the reciprocal of the variances of the individual measurements.

Similar to the case of GLS, the optimization problem in eq. (4.8.36) has a unique solution given by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{y} \tag{4.8.37}$$

The measurement model of eq. (4.8.24) can now be cast into the WLS form of eq. (4.8.37), giving

$$\begin{aligned}\hat{\boldsymbol{\omega}}(k) = \frac{1}{\Delta t}\left\{\sum_{i=1}^{n}\bar{\sigma}_i^{-2}[\widetilde{\mathbf{b}}_i(k)\times]^T[\widetilde{\mathbf{b}}_i(k)\times]\right\}^{-1}& \\ \times\sum_{i=1}^{n}\bar{\sigma}_i^{-2}[\widetilde{\mathbf{b}}_i(k)\times]^T\Big(\widetilde{\mathbf{b}}_i(k+1) - \widetilde{\mathbf{b}}_i(k)\Big)&\end{aligned} \tag{4.8.38}$$

where $\hat{\boldsymbol{\omega}}(k)$ is the best linear unbiased estimate of $\boldsymbol{\omega}(k)$ and the weight $\bar{\sigma}_i^{-2} \equiv 2\sigma_i^2/\Delta t^2$ is the effective measurement error variance.

Chapter 5

Simulations

This chapter presents an evaluation of the proposed matching and rate estimation algorithms based on simulations performed in the MATLAB environment. In order to best approximate realistic behaviour, the simulation procedure was designed such as to resemble spaceborne stellar gyro operation as closely as possible.

5.1 Simulation Procedure

Prior to MCU implementation, various simulations were performed in the MATLAB environment to evaluate the validity, reliability and robustness of the proposed algorithms. Accurate simulation of both stellar gyro operation as well as its spaceborne surroundings provide an environment capable of estimating behaviour under various extreme conditions. The general MATLAB test procedure can be outlined as follow:

1: Import star data from an external star catalogue

This data includes the ID tag, celestial coordinates (right ascension and declination) and visual magnitude of each star.

2: Limit visual star magnitude

All stars that adhere to a predetermined maximum visual magnitude condition are extracted. In most cases, this maximum magnitude is representative of the stellar gyro's detection capability.

3: Transform star coordinates

The coordinates of each valid star[1] is then transformed from celestial to body-fixed unit vector coordinates (see section 2.3). Recall that the concern of stellar gyro algorithms is merely the displacement of a series of star centroids from one image frame to the next. The inertially referenced locations of stars are therefore irrelevant. Such transformation simplifies the process of accurately simulating stellar gyro behaviour.

4: Determine stars that fall within the camera FOV

This process can be simplified by first extracting all stars that fall within a

[1] a valid star in this context being a star that adheres to the aforementioned magnitude condition, and which should therefore be visible to the stellar gyro

circular FOV that covers the entirety of the image sensor area (see fig. 3.10 in section 3.4) based on their angular displacement from the camera boresight. Extracted stars are then transformed to image plane coordinates (see section 2.3.4), after which any stars falling outside the image sensor area are discarded.

5: Generate star image

A simulated night sky image is generated by projecting the extracted star locations onto a uniformly black bitmap with resolution equal to that of the image sensor. The plotted PSF of each star is jointly based on the level of optical defocussing, detection capabilities of the stellar gyro optics and the visual magnitude of the star itself.

6: Image processing

The generated images are subjected to image processing algorithms in an attempt to recover the centroids of the plotted stars. Once recovered, these centroids are stored as centroid list A.

7: Simulate angular rotation

To simulate angular rotation, each star extracted in **step 2** is rotated based on predetermined angular rates. **Steps 4-6** are repeated to determine a second set of star centroids, which is then stored as centroid list B. Lists A and B now contain the star centroids identified in two consecutive images.

8: Star matching

At this point centroid lists A and B merely contain the centroid locations extracted from successive images. In order to estimate angular rates, information regarding the correspondence of stars between such images is required. This is achieved through means of a dedicated star matching algorithm (as detailed in section 4.6), which ties corresponding star locations in the form of matched centroid pairs.

9: Estimate angular rates

Rate estimation algorithms use these matched centroid pairs to determine the angular displacement from one image frame to the next in three dimensional space. With the time lapse between successive images known, angular rates are determined from these angular displacement values.

10: Repeat

Once the angular rates have been estimated, the centroids stored in list B are transferred to list A, after which **steps 7-9** are repeated to determine the angular rates at the next sampling instance.

5.2 Validity of Generated Star Images

Accurate recreation and simulation of spaceborne stellar gyro surroundings rely heavily on the authenticity of generated star images. The two most important aspects to consider with regard to such authenticity are photon distribution and inertially referenced orientation.

5.2.1 Photon Distribution

As detailed in section 3.2, a two-dimensional Gaussian distribution serves as close approximation of an actual star PSF. Star images can therefore be generated by determining the value of each pixel in an arbitrarily large grid around each star centroid using eq. (5.2.1).

$$f(\mathbf{X}) = I \exp\left\{-\frac{1}{2}(\mathbf{X}-\bar{\mathbf{X}})^T\mathbf{\Sigma}^{-1}(\mathbf{X}-\bar{\mathbf{X}})\right\} \tag{5.2.1}$$

where

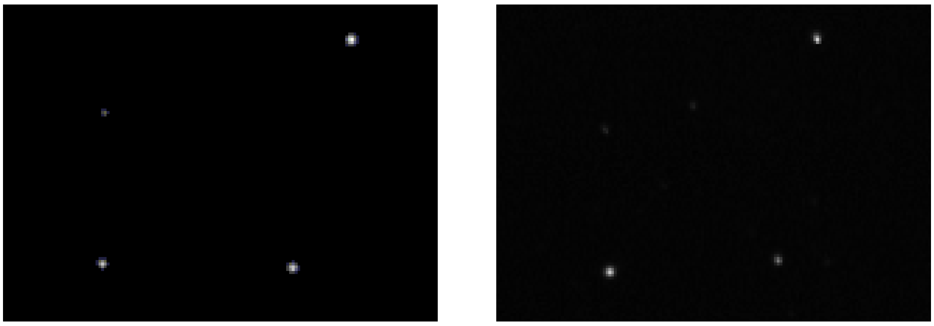
$\mathbf{X} = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$ → discrete coordinates of each pixel in the chosen grid

$\bar{\mathbf{X}} = \begin{bmatrix} x_c \\ y_c \end{bmatrix}$ → coordinates of the star centroid

$\mathbf{\Sigma} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$ → covariance matrix

The coefficient I scales the amplitude of the Gaussian distribution based on the visual magnitude of the star in question. Determining an accurate mathematical representation of such scaling factor would require a comprehensive and unnecessarily complicated optical analysis. Alternatively, an accurate approximation can be obtained by comparing actual images to their generated counterparts, iteratively adjusting the scaling factor until an acceptable level of similarity has been achieved. The latter method is more than suitable for the simulations presented here. Similarly, the scaling factor for the variance of the Gaussian distribution was also determined in an iterative manner.

Note that the random variables x and y were assumed to exhibit equal variances, while also being uncorrelated. Considering the low angular rates at which the proposed stellar gyro is expected to operate, these assumptions should allow for an accurate approximation of actual star images.



(a) Generated Image

(b) Captured Image

Figure 5.1 – Comparison of generated and captured partial star images of the Southern Cross from the constellation Crux

Figure 5.1 shows a comparison between generated and captured[1] partial star images of the Southern Cross from the constellation Crux. A clear similarity is observed between images. Generated stars are slightly brighter than their captured counterparts in an attempt to compensate for light pollution and refraction through the earth's atmosphere. Note, however, that the similarity of the PSF differ from star to star. This is due to the assumption of a linear scaling function, whereas actual stellar gyro operation introduces various unmodelled non-linearities. For the purpose of these simulations, the generated images exhibit a satisfactory level of similarity.

5.2.2 Orientation

Stellar gyroscope algorithms are concerned only with the propagation of a series of star centroids between successive image frames. It is therefore crucial to accurately project stars within the camera FOV onto the image plane. With orientation algorithms readily available, generated star images were uploaded to CubeStar in order to determine their boresight coordinates relative to the celestial sphere. Comparing these coordinates to those specified as generation parameters provide a rough estimate of star projection accuracy.

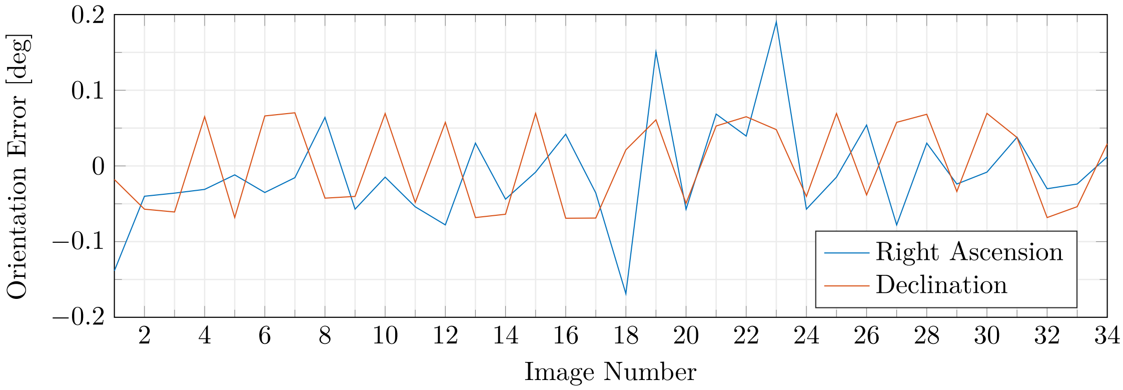


Figure 5.2 – Attitude determination error for simulated star images

Of a 50 image sample group, each generated with random boresight coordinates, CubeStar was able to successfully determine the orientation in 34 images. Note that the orientation algorithms used by CubeStar rely heavily on a *tracking* mode, in which the boresight coordinates can be approximated based on a previous attitude estimate. Orientation determination is also much more reliable when operating in *tracking* mode. Given that each image in the sample group was generated with random boresight coordinates, orientation algorithms were forced to operate in *lost* mode, justifying a 68% identification rate. For a more in-depth explanation regarding CubeStar modes of operation, refer to [7].

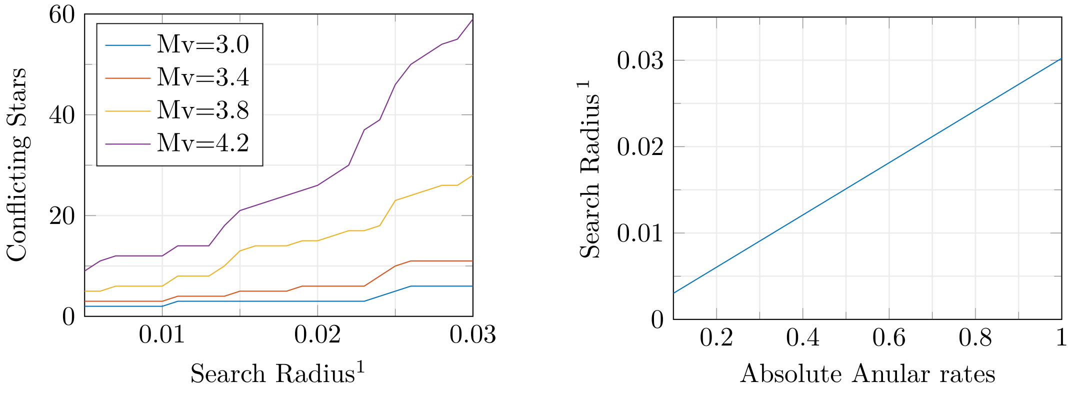
Figure 5.2 shows the orientation error for each successful iteration. Standard deviations of 0.0853° and 0.0597° were determined for right ascension and declination errors respectively, verifying the validity of the generated star images.

[1] Night sky images captured using CubeStar hardware

5.3 Matching Algorithm Reliability

The efficiency and reliability of the proposed matching algorithm are contingent on the selection of an optimal region of interest (ROI). This ROI, defined jointly by its origin and radius, specifies the area on the celestial sphere in which a star centroid transformed through rotation from one sampling instance to the next can be expected (see section 4.6). Alternatively, the size and origin of such ROI can be interpreted as the level of certainty regarding the location of the transformed centroid. Since this measure of certainty is dependant on the availability of attitude information, dynamically adjusting ROI parameters based on the availability of such information should theoretically increase matching efficiency and reliability. This hypothesis will be tested through simulation.

To fully understand the importance of determining the optimal ROI size, the consequences of a poor selection should be considered. On the one hand, specifying an excessively small radius could decrease the probability of identifying the correct match, especially since the projected centroids (which are subject to measurement noise and optical aberrations) might deviate from their expected locations. Specifying an excessively large radius, on the other hand, could result in the identification of multiple possible projections for a single star. This would not only slow down algorithm execution time, but also increase the likelihood of making an incorrect match. The radius should therefore be chosen such that the ROI (a) is large enough to facilitate the desired angular rates, while (b) also being small enough to contain only the correct match whenever possible.



(a) Number of conflicting stars over the celestial sphere for a given search radius at various visual magnitudes

(b) Maximum Euclidean distance between a star's location in successive iterations based on angular rates

Figure 5.3 – Simulations aimed at determining the optimal ROI size

Figure 5.3(a) shows the number of stars that, in the absence of additional angular rate information, will fail to provide a unique solution (i.e., the ROI will contain

[1] Search radius describes the maximum Euclidean distance between the ROI origin and any point within the area the ROI encompasses on the celestial sphere

more than one possible projection) for a given search radius. Such stars will from here on be referred to as *conflicting stars*.

Assuming a detection threshold of 3.4 (as determined for Cubestar hardware in section 3.6), search radii up to approximately 0.023 show no significant change in terms of the number of conflicting stars. Notice, however, that an increase in visual magnitude corresponds to an increase in the rate of change between the specified search radius and the number of conflicting stars for radii exceeding 0.012. Recall from chapter 3 that, due to various conservative assumptions, increased detection capabilities are expected during spaceborne operation. Consequently, the larger the specified ROI, the larger the corresponding increase in the number of conflicting stars expected during spaceborne operation. To ensure that matching efficiency is not compromised by a possible increase in detection capabilities, a conservative search radius of 0.012 was chosen.

Figure 5.3(b) shows the maximum angular rates at which reliable operation can be expected for any given search radius (i.e., the star centroid transformed through rotation will fall within the specified search radius). According to these simulations, the chosen search radius of 0.012 should allow for reliable operation at angular rates of up to 0.45 deg/s simultaneously in all three axes. Note that these are the maximum rates at which the algorithm can reasonably be expected to match *all* centroids in the *absence* of previous angular rate estimates. Since such estimates will usually be available, and rate estimation algorithms theoretically require only two stars for successful operation, accurate estimates can be expected at rates far exceeding 0.45 deg/s.

The availability of previous angular rate estimates provide an additional verification step for star matching, effectively reducing the valid ROI area by a factor of four (see section 4.6). Since such estimates greatly improve the level of certainty regarding the location of the transformed centroid, the matching algorithm can safely search for possible projections over a much larger area. Assuming that reliable operation is required at maximum angular rates of 1 deg/s in all axes, the search radius can be increased to 0.03 where previous rate estimates are available. Furthermore, the availability of such estimates allow the algorithm to determine the expected location of a transformed centroid. Positioning the ROI origin at this expected location should further increase the likelihood of finding the correct match.

Figure 5.4 illustrates the reliability of the matching algorithm using the previously determined radii. For each rate denomination, matching is performed both with (assisted) and without (unassisted) previous angular rate estimates. A few false stars were also generated in close proximity to either the original or transformed location of some centroids to further assess algorithm reliability. In all cases, the locations of the false stars were generated such as to be less suitable matches than the true star locations.

The algorithm is observed to reliably match all stars up to angular rates of approximately 0.5 deg/s regardless of the availability of previous rate estimates. Further increases in angular rates show a steady decline in the success rate of unassisted matching. For angular rates exceeding 0.7 deg/s, the average success rate for unassisted matching amounts to a mere 14.2%. Assisted matching, on the other hand,

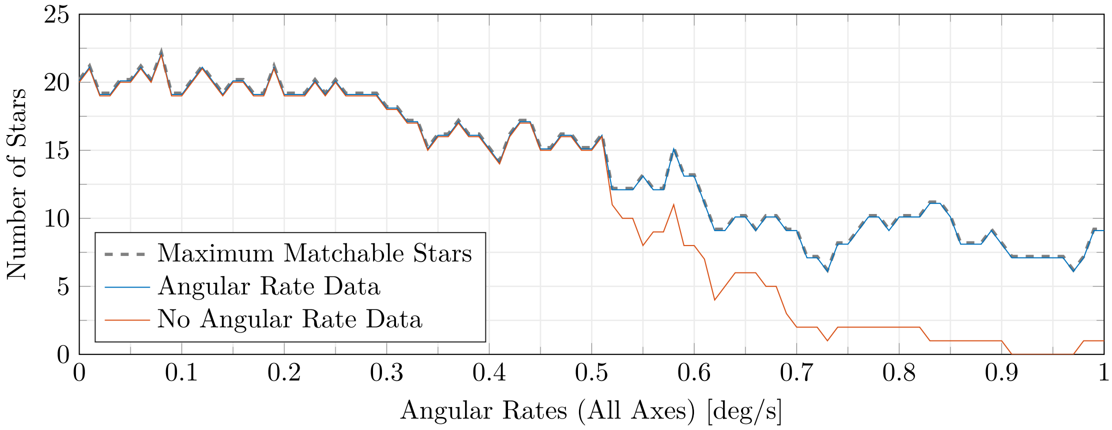


Figure 5.4 – Number of successfully matched stars for gradually increasing angular rates

maintains a 100% success rate for angular rates up to 1 deg/s. Commendably, none of the generated false stars were matched at any point. As hypothesised, dynamically adjusting ROI parameters based on the availability of previous rate estimates drastically improves matching reliability at higher angular rates.

It is important to remember that the preceding simulations serve only to characterize and determine the limitations and reliability of the matching algorithm itself. This being said, the current version of CubeStar is unlikely to operate reliably at angular rates exceeding ±1 deg/s.

CubeStar exhibits relatively long exposure times of up to 520 ms which, at high angular rates, introduces a practical complication. Long exposure times at high angular rates are likely to result in a star being smeared over the image plane (i.e., appearing as a line/star streak rather than a point source of light). A smeared star introduces two major complications:

1. The total number of photons from any given star is distributed over a much larger number of pixels. Consequently, pixels related to such star are now much more likely to fall below the detection threshold. Star smearing therefore reduces the detection capabilities of a stellar gyro.

2. Since stars are no longer observed as point sources of light, the symmetrical PSF approximation (see section 3.2) of the photon distribution is no longer valid. This impairs centroiding accuracy, since the chosen algorithms assume such symmetry.

Considering such complications, the purpose of these simulations are twofold: (a) to identify the optical solution as the limiting factor with respect to maximum angular rate capabilities rather than the matching algorithm itself, as well as (b) to verify the accuracy and reliability of the proposed matching algorithm. Implementing a more light sensitive image sensor would facilitate a shorter exposure time, increasing the maximum angular rates at which reliable operation can be expected.

5.4 Comparison of Rate Estimation Algorithms

Simulations were used to compare candidate rate estimation algorithms and ultimately to identify the best suited method for implementation on CubeStar. Given initial boresight coordinates[1] of

$$[\text{RA}, \text{Dec}, \text{Rotation}] = [-146.428°, -60.996°, -150°]$$

successive star images were generated over a 100 second interval using the angular rates specified in eq. (5.4.1).

$$\boldsymbol{\omega}(t) = \begin{array}{c} \\ \left[\begin{array}{ccc} \textbf{X} & \textbf{Y} & \textbf{Z} \\ 0.2\,\sin(8t) & 0.4 & 0.2\,\cos(8t) \end{array}\right] \end{array} \text{deg/s} \tag{5.4.1}$$

The remainder of the simulation procedure, as detailed in section 5.1, was performed for each method individually. The results are presented in fig. 5.5.

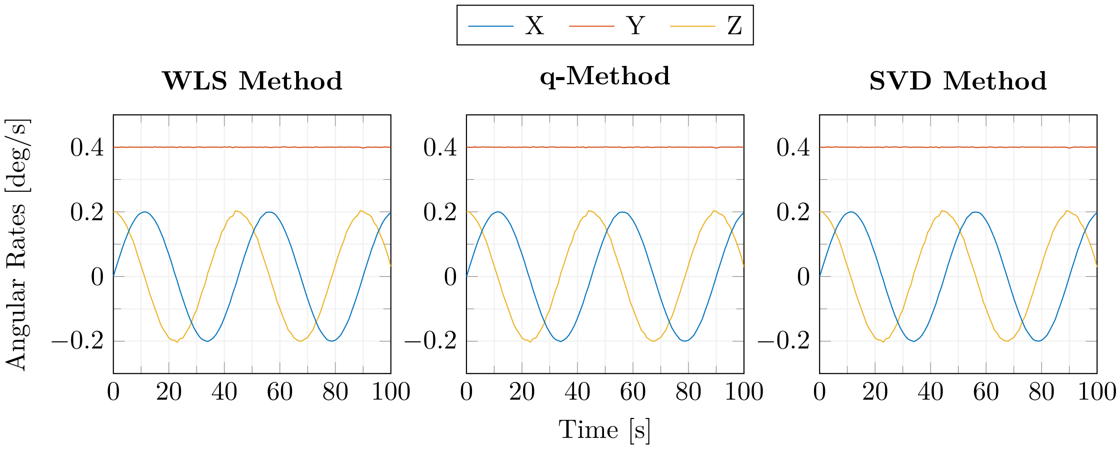


Figure 5.5 – Estimated angular rates for each candidate method

All three methods provide seemingly accurately estimates of the specified angular rates. Since these simulations identify no clear favourite, a more in-depth comparative analysis is performed based on the estimation errors of each method (as shown in fig. 5.6). Take note of the fact that results are analysed on a comparative basis *only*. A detailed analysis of the estimation error for the chosen method is presented in section 5.5.

Results are once again almost indistinguishable from one another. Choosing a method based on estimation accuracy would therefore be ill-advised, suggesting the consideration of alternative parameters. CubeStar imposes relatively strict limitations with respect to execution time and memory usage. Algorithm complexity therefore logically presents itself as the next most important consideration.

The MATLAB *Profiler* tool determines the execution time of a script based on CPU time. Even though execution times determined through MATLAB profiling in no

[1] the chosen coordinates point in the direction of the Southern Cross, an abundantly populated region on the celestial sphere

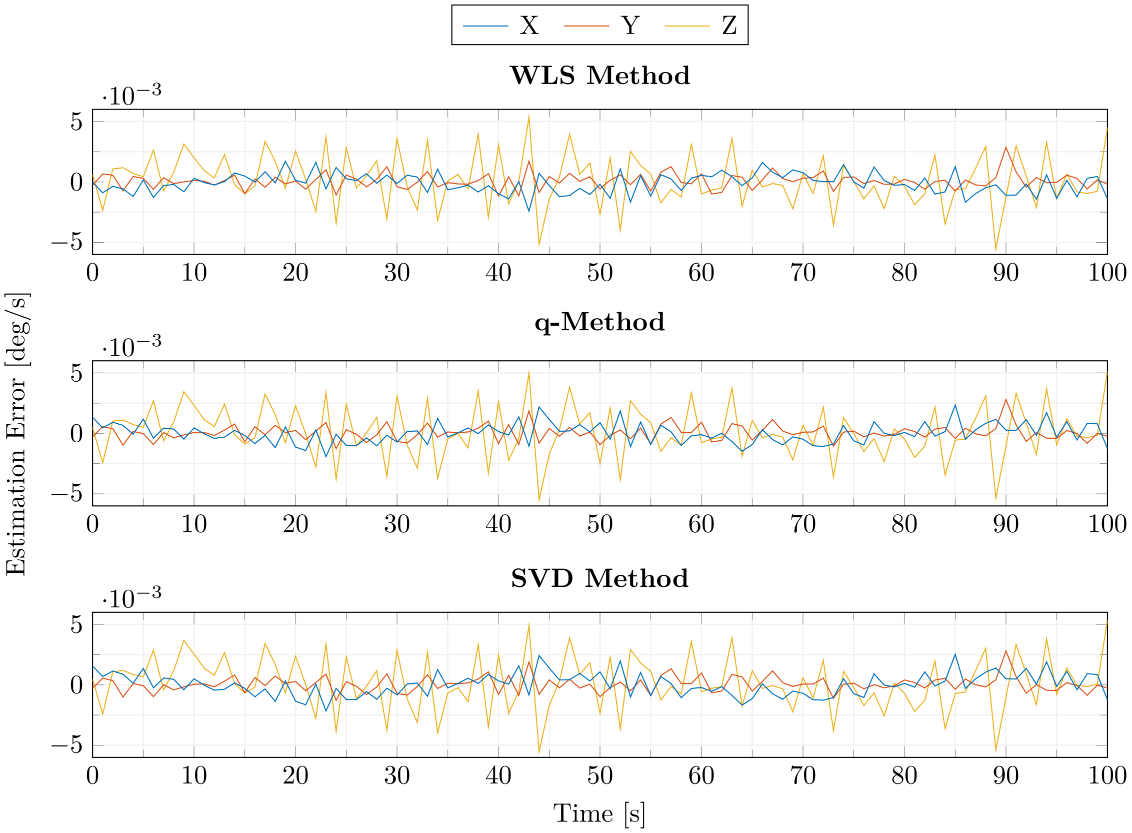


Figure 5.6 – Estimation error for each candidate method

way resemble those expected for MCU implementation, they provide a valuable comparative guideline for further analysis. Profiling over the aforementioned 100 second interval, execution times of 0.02 s, 0.017 s and 0.011 s were determined for the WLS-, q- and SVD-methods respectively. Even though a clear difference in execution time is observed, especially with SVD being almost twice as fast as WLS, the order of magnitude of these differences render them largely insignificant with respect to the overall execution time.

Overall, comparisons suggest remarkable similarity between methods in terms of both estimation accuracy and complexity. At his point it is worth considering that, based purely on the derivation of each algorithm, both the SVD- and q-methods would likely be much harder to implement and debug on CubeStar hardware than the WLS method. Since neither of the preceding comparisons favoured a particular method, WLS was chosen due to its simplicity and ease of implementation.

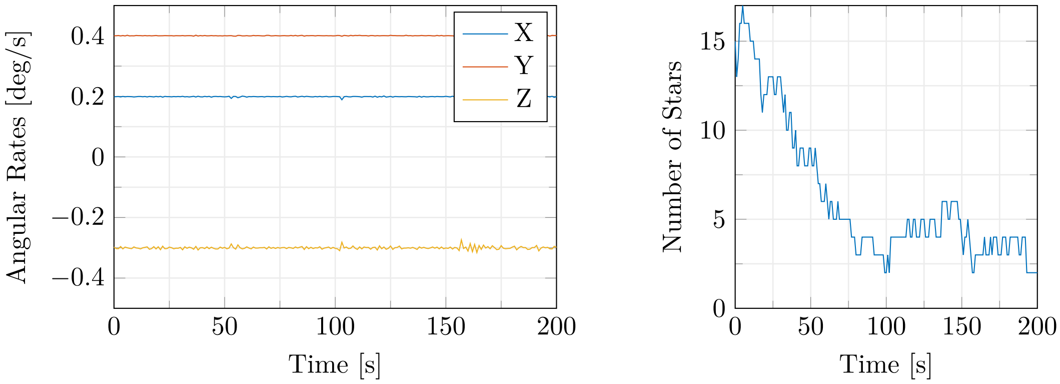
5.5 WLS Simulation Analysis

With the WLS algorithm identified as the best suited candidate for implementation on CubeStar, a more in-depth error analysis was performed. The stellar gyroscope process introduces three major inaccuracies: measurement noise, which is largely characterized by the image sensor, centroiding errors, which accompanies image processing, and WLS estimation bias, which is the result of various assumptions and approximations made during the derivation of the chosen WLS algorithm. In an attempt to identify the contribution of each inaccuracy towards the overall estimation error, simulations isolating each inaccuracy were performed. Note that since optical aberrations are characterized and accounted for (see section 4.4), they were not considered as inaccuracies.

Simulations were repeated using the constant angular rates specified in eq. (5.5.1).

$$\boldsymbol{\omega}(t) = \begin{array}{c} \\ \\ \end{array}\begin{array}{ccc} \textbf{X} & \textbf{Y} & \textbf{Z} \\ \left[\ 0.2 \right. & 0.4 & \left. -0.3\ \right] \end{array} \text{deg/s} \tag{5.5.1}$$

This was done to aid visual comparison between axes as well as to accentuate estimation errors. Figure 5.7 shows the estimated angular rates over a 200 second simulation period, as well as the number of stars detected at any point within this interval.

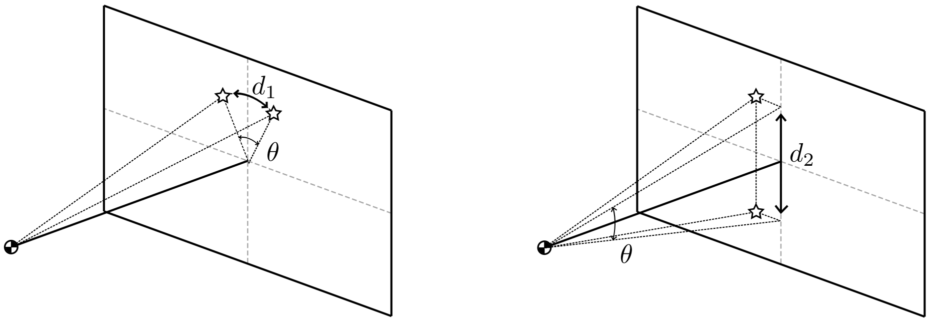


(a) Estimated angular rates

(b) Number of stars detected

Figure 5.7 – WLS estimation results for constant angular rate simulations

The most striking observation is the increased noise levels associated with rotation around the Z-axis. The convention by which the axes are defined in the body-fixed coordinate system dictate that rotations around either the X- or Y-axes will result in angular displacement values that are perpendicular to the camera boresight (i.e., a cross-boresight rotation). Rotations around the Z-axis, on the other hand, will result in angular displacement values that are representative of rotations around the camera boresight. Since the focal length of a stellar gyro is usually much larger than the dimensions of the image sensor, rotation around the boresight axis will induce a much smaller displacement relative to the image plane than a similar rotation around either of the cross-boresight axes (see fig. 5.8). Star tracker cameras (which utilize similar hardware to stellar gyros) are typically in the order of 6–16 times less accurate around the boresight axis [47]. Any noise or inaccuracies can therefore be expected to have a much more pronounced effect on rate estimation accuracy around the boresight axis than around its cross-boresight counterparts.



Rotation around the boresight axis

Rotation around the cross-boresight axis

Figure 5.8 – Visual comparison of the angular displacement resulting from a rotation through angle θ around either the boresight or cross-boresight axis

Another noteworthy observation is the correlation between estimation error levels and the number of stars detected. Section 4.8.3 illustrates that the nature of least-squares algorithms is such that estimation accuracy is proportional to the number of data samples available. Consequently, a larger number of stars can be expected to produce a more accurate estimate.

Figure 5.9 shows the simulation errors associated with each major inaccuracy.

The *Ideal Centroids* case resembles a simulation where (a) the centroids normally used to generate star images are fed directly into the matching algorithm (image processing is therefore eliminated), and (b) no measurement noise is induced. The entirety of the observed error can therefore be ascribed to the algorithm itself. Note that, in all three axes of rotation, the algorithm error is approximated by a near constant offset. Such behaviour is typical where assumptions and approximations are made to aid algorithm implementation. Since the aforementioned errors are by-products of the algorithm itself, further simulations where image processing and measurement noise are introduced should be centred around these offset values.

The *Measurement Noise* case resembles a simulation similar to that described by the *Ideal Centroids* case, the main difference being that the ideal centroid locations are now corrupted by measurement noise. Measurement noise, with respect to stellar gyro operation, is primarily associated with the optical solution. Since accurate characterization of such noise is a relatively complex process, the noise induced here is artificial in nature. A zero-mean Gaussian distribution with a standard deviation of 10^{-5} degrees should provide an adequate estimate of the measurement noise expected during operation.

The *Image Processing* case resembles a simulation where the ideal centroid locations were used to generate star images based on the method described in section 5.2.1.

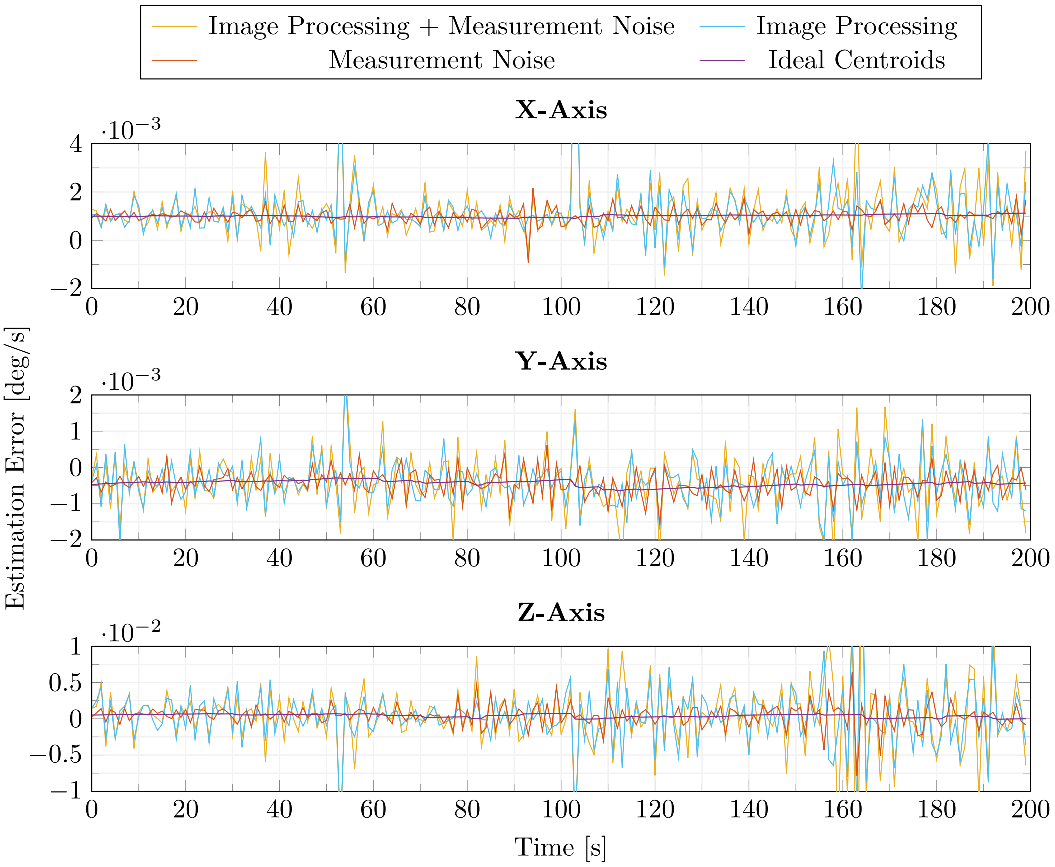


Figure 5.9 – Estimation errors associated with each major inaccuracy

These images were then subjected to a centroiding algorithm (see section 4.3) in an attempt to recover the plotted centroids, after which the extracted centroid locations were matched between image frames. The error associated with image processing can be jointly ascribed to the resolution of the image sensor and the chosen centroiding algorithm.

Comparing the isolated effects of the *Measurement Noise* and *Image Processing* cases, the latter is clearly identified as the more prominent of the two inaccuracies. As mentioned earlier, the error associated with image processing can be reduced either through implementation of a higher resolution image sensor, which would improve centroiding accuracy by distributing each star over a larger number of pixels, or through implementation of a more accurate and comprehensive centroiding method. Since changing the image sensor used on-board CubeStar would be a relatively complex and time consuming task, various centroiding methods were evaluated through means practical tests in an attempt to minimize the error associated with image processing.

Figure 5.10 illustrates the probability distribution of the estimation error in each axis of rotation. In all cases, the error could be fitted to a Gaussian distribution with reasonable accuracy. Table 5.1 shows the 3σ accuracies for each case.

As expected, rates in both cross-boresight axes are estimated with a much higher level of certainty than the corresponding rates in the boresight axis. Furthermore, notice

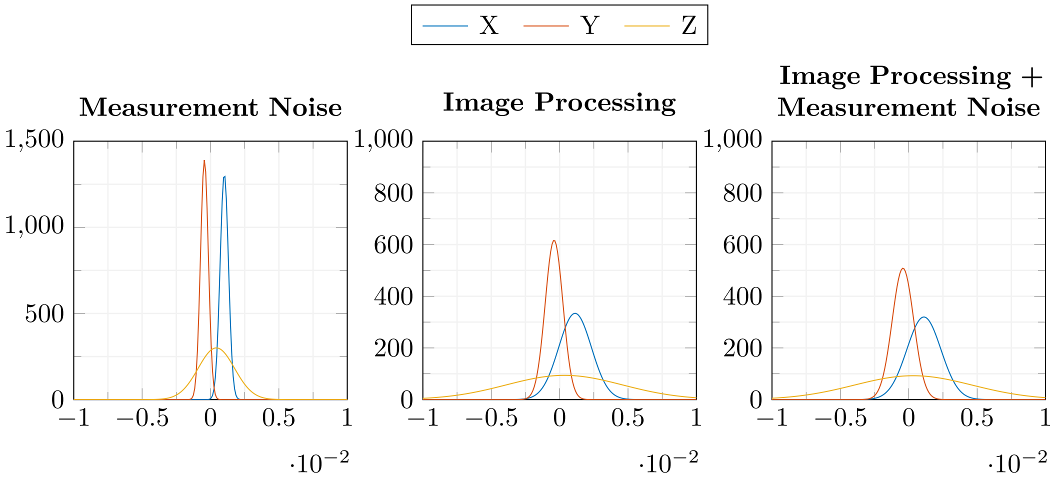


Figure 5.10 – Probability distribution functions of rate estimation errors for each inaccuracy

that the mean value of each distribution plot corresponds to the offset introduced by the algorithm itself. For the *Measurement Noise* case, angular rates in all three axes are estimated with at least twice the level of certainty of those corresponding to the *Image Processing* case. This confirms image processing as being the more prominent source of inaccuracy. Incorporating both measurement noise and image processing, which represents actual stellar gyro behaviour, shows very little change with respect to error distribution when compared to the isolated *Image Processing* case.

Table 5.1 – 3σ Accuracies determined for each simulated case

	3σ Accuracy [deg/s]		
Simulation Case	**X**	**Y**	**Z**
Measurement Noise	0.00091	0.00086	0.00399
Image Processing	0.00358	0.00194	0.01274
Measurement Noise + Image Processing	0.00374	0.00235	0.01295

Chapter 6

Testing and Results

This chapter presents the results of practical tests performed on the proposed stellar gyro algorithms. CubeStar hardware was used as testing platform. A detailed breakdown of the test setup is included, as well as an assessment of its authenticity as accurate reconstruction of the night sky.

6.1 Test Setup

Stellar gyro algorithms should ideally be tested using actual star images. Unfortunately, the propagation of light through the earth's atmosphere is somewhat erratic. Continual fluctuations in air density cause light to refract unpredictably as it passes through different layers of the atmosphere. The amount of light reaching a point of observation therefore differs from one moment to the next, causing stars to scintillate. Considering CubeStar's relatively limited detection capability (see section 3.6), such inconsistencies could jeopardize stellar gyro functionality. An accurate reconstruction of the night sky was therefore approved for testing purposes rather than the night sky itself. Figure 6.1 illustrates the proposed test setup.

Rotation is achieved through means of an Ideal Aerosmith Model 1270VS rate table, fundamentally a closed-loop motion control system consisting of a DC servo motor, controller and optical encoder. Key features include bidirectional rotation capabilities, a speed range of 1-216 000 deg/min and a rotation accuracy of ±0.1%. The rate table facilitates communication through means of an RS-232 port, which also provides access to some advanced functions only available through remote link. The stellar gyro is mounted atop the table surface. The nature of the rate table is such that it only facilitates rotation around a single axis. Practical tests will therefore be conducted for each body axis individually. Figure 6.2 illustrates the test setup orientation for each axis of rotation.

The night sky is simulated using a $(3.5 \times 2)\,\text{m}^2$ circularly arched surface with a 2 m radius. The rate table, with stellar gyro mounted atop, is placed at the arch centre point. The distance between the simulated night sky and the lens focal point therefore remains fixed throughout rotation. Note that the surface is only curved in the rotating plane, slightly deviating from the ideal spherical representation of

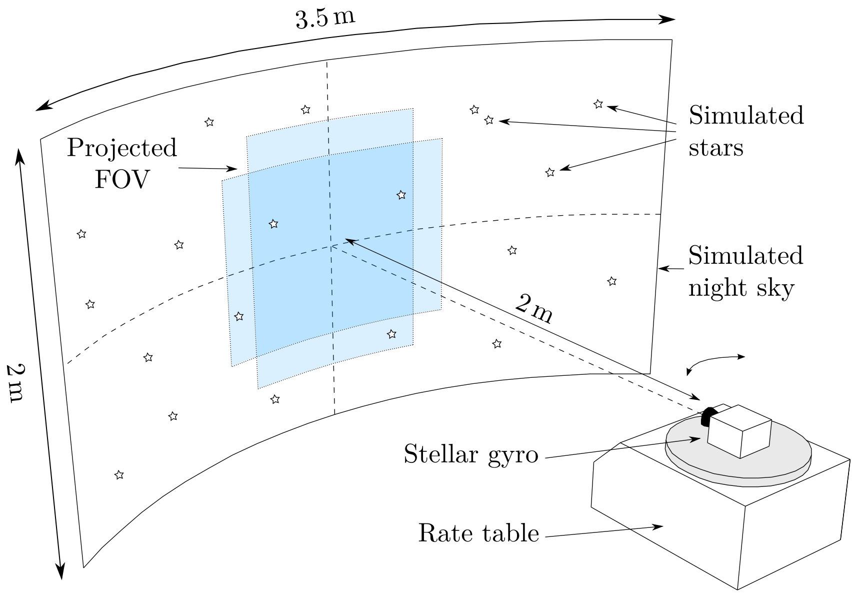


Figure 6.1 – Illustration of the proposed test setup [not to scale]

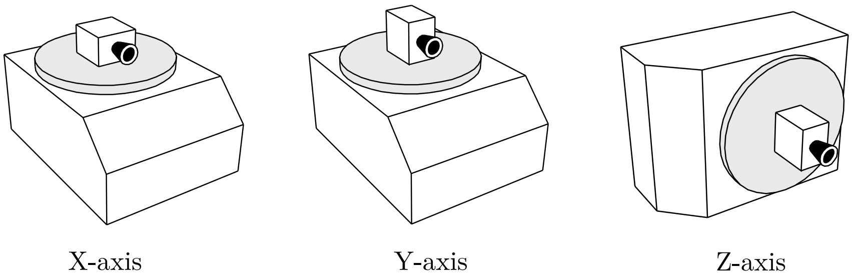


Figure 6.2 – Test setup orientation for each axis

the celestial sphere. The impact of such discrepancy on rate estimates should be relatively small.

Stars are simulated using Wah Wang Series 5A3 ultra bright white LEDs. Key specifications include a colour temperature of 5500 K, closely resembling that of most stars visible to CubeStar, as well as a strong directivity with narrow light distribution, approximating collimation as closely as possible (section 6.2 details the desired light distribution patterns). Stars are introduced to the simulated night sky by mounting the LEDs behind small holes in the arched surface.

In an attempt to replicate spaceborne operation as closely as possible, practical tests were performed using CubeStar's maximum exposure time of 520 ms (at which it was designed to operate).

6.2 Star Distribution Patterns

Centroiding algorithms assume a PSF approximated by an ideal two-dimensional Gaussian distribution over a 5×5 pixel grid (see section 3.2). In order to maximize centroiding accuracy, simulated stars should exhibit photon distribution patterns that closely resemble this ideal assumption. Figure 6.3 compares the per-pixel distribution patterns of ideal, night sky[1] and simulated stars.

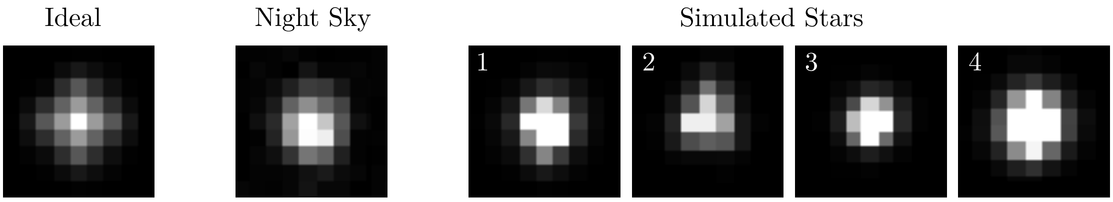


Figure 6.3 – Comparison between ideal, night sky and simulated star distribution patterns

Even though the ideal and simulated distributions are far from identical, clear similarities are observed with regard to intensity falloff. In all cases, the brightest pixels are located near the centre of the distribution, with a gradual falloff observed for neighbouring pixels. It is worth noting that the ideal distribution considers a star located precisely at the centre of a pixel, facilitating the observed symmetry. Furthermore, it also disregards any form of measurement error or optical aberration (see section 4.4).

A more pragmatic approach would be to compare simulated stars to those captured of the night sky using CubeStar hardware, both equally susceptible to the aforementioned optical and measurement errors. Even though simulated distributions still appear slightly more rugged and uneven than their night sky counterparts, they were deemed to exhibit sufficient similarity for reliable testing.

Note from fig. 6.3 that the fourth simulated star contains seven saturated pixels, despite being distributed over a mere 6×6 pixel area. This has to do with the distribution of light.

Stars are for all practical purposes infinitely far away from a stellar gyro in LEO. This implies that light rays incident on the image sensor surface can be considered collimated. The result is a decidedly focused light distribution that, when combined with optical defocussing, approximates the expected two-dimensional Gaussian PSF (see section 3.2). Stars should therefore ideally be simulated using collimated light sources.

Since collimating numerous simulated stars can be a somewhat expensive and arduous process, unaltered ultra bright LEDs were used as alternative (see section 6.1). Even though such LEDs exhibit strong directivity with narrow light distribution (the chosen LEDs exhibiting a mere 15° radiation angle), their close proximity to the stellar gyro prevent them from approximating a collimated distribution (see fig. 6.4). The result is that multiple pixels are now exposed to the maximum LED radiation, saturating them in the process. Unfortunately, in order to ensure that the region growing algorithm (see section 4.2) detects simulated stars (which requires that a star region contain at least 9 pixels), some saturation is unavoidable.

[1] Night sky stars were captured using CubeStar hardware

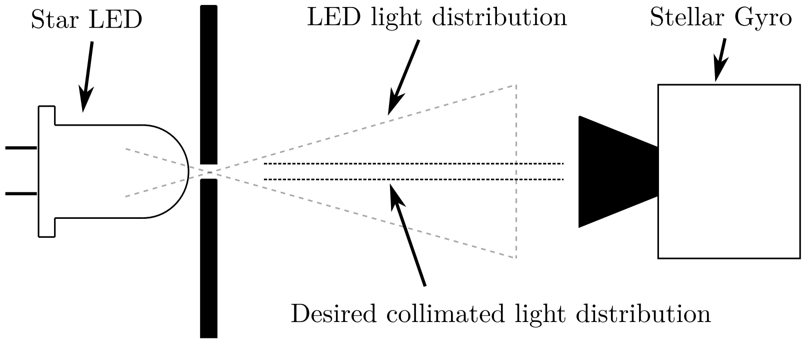


Figure 6.4 – Simulated and ideal light distribution patterns [not to scale]

6.3 Comparison of Centroiding Algorithms

Recall from section 4.3 that two different centroiding algorithms were proposed for implementation on CubeStar, namely the standard CoG approach and the more comprehensive WCoG approach.

Here it is important to understand how the CoG method is implemented. Since the algorithm expects a two-dimensional Gaussian distribution as PSF, it assumes that the majority of the distribution will be taken into consideration if the 5 × 5 pixel grid (dictating which pixels are considered during centroiding) is centred at the brightest pixel. Note from fig. 6.3, however, that simulated stars often contain multiple pixels that saturate the image sensor (the observed saturation is explained in section 6.2), resulting in multiple brightest pixels. Since the centroiding algorithm has no way of distinguishing between such pixels, it identifies the first pixel to saturate as the brightest pixel, after which the rest are discarded based on the fact that they don't actually exceed the saturation value (they merely match it). This approach should have little impact on smaller stars, but can severely impair centroiding accuracy for larger stars containing numerous saturated pixels (see fig. 6.5).

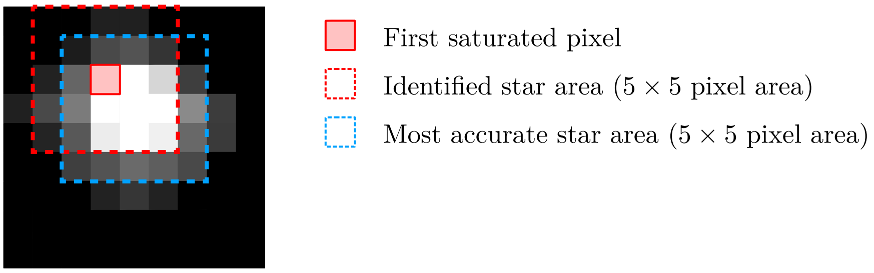

First saturated pixel
Identified star area (5 × 5 pixel area)
Most accurate star area (5 × 5 pixel area)

Figure 6.5 – Graphical illustration of the CoG centroiding algorithm facing numerous saturated pixels

To avoid such inaccuracies, the CoG algorithm was modified to determine the brightest pixel based not only on its value, but also on its location. This was done such that if the modified CoG algorithm encounters multiple saturated pixels, it identifies the pixel located closest to the centre of the star region as the brightest pixel.

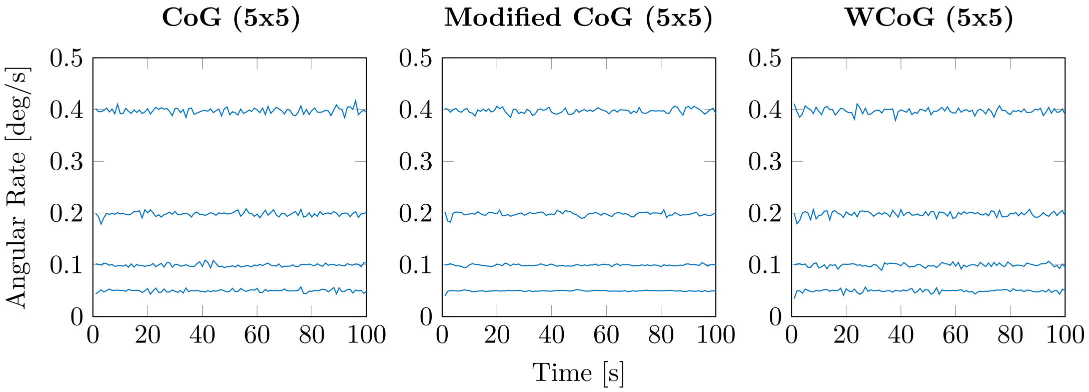


Figure 6.6 – Comparison of centroiding algorithms based on practical tests

In an attempt to accurately assess the efficiency of each centroiding method, specifically with regard to their implementation on CubeStar, practical tests were conducted at various angular rates (see fig. 6.6). Rotations were performed around the X-axis. All three methods correspond to reasonably accurate rate estimates. Specific accuracies are detailed in table 6.1.

Table 6.1 – Estimation accuracies associated with each centroiding method

	3σ Accuracy [deg/s]		
Angular Rate [deg/s]	**CoG**	**Modified CoG**	**WCoG**
0.05	0.0081	0.0036	0.0093
0.1	0.0087	0.0048	0.0102
0.2	0.0129	0.0114	0.0153
0.4	0.0168	0.0135	0.0168

As expected, modification of the CoG method showed significant improvements with regard to the accuracy of estimated rates. This improvement is particularly noticeable at low angular rates, where estimates corresponding to the modified method are more than twice as accurate. Notice that the more comprehensive WCoG method is less accurate than both variations of the standard CoG. This is due to the WCoG method's heavy reliance on an accurate, unsaturated Gaussian distribution (see section 4.3.2) which the simulated stars fail to provide.

Even though the original CoG and WCoG methods should prove to be reliable during spaceborne operation, the more robust modified CoG method was approved for further testing.

6.4 Constant Rate Tests

Stellar gyro design specifications stipulate reliable operation up to angular rates of ±1 deg/s in all three axes of rotation. To this end, the maximum angular rate capabilities were determined through means of constant rate tests. The modified CoG approach was used for centroiding purposes (see section 6.3). Test results are shown in fig. 6.7.

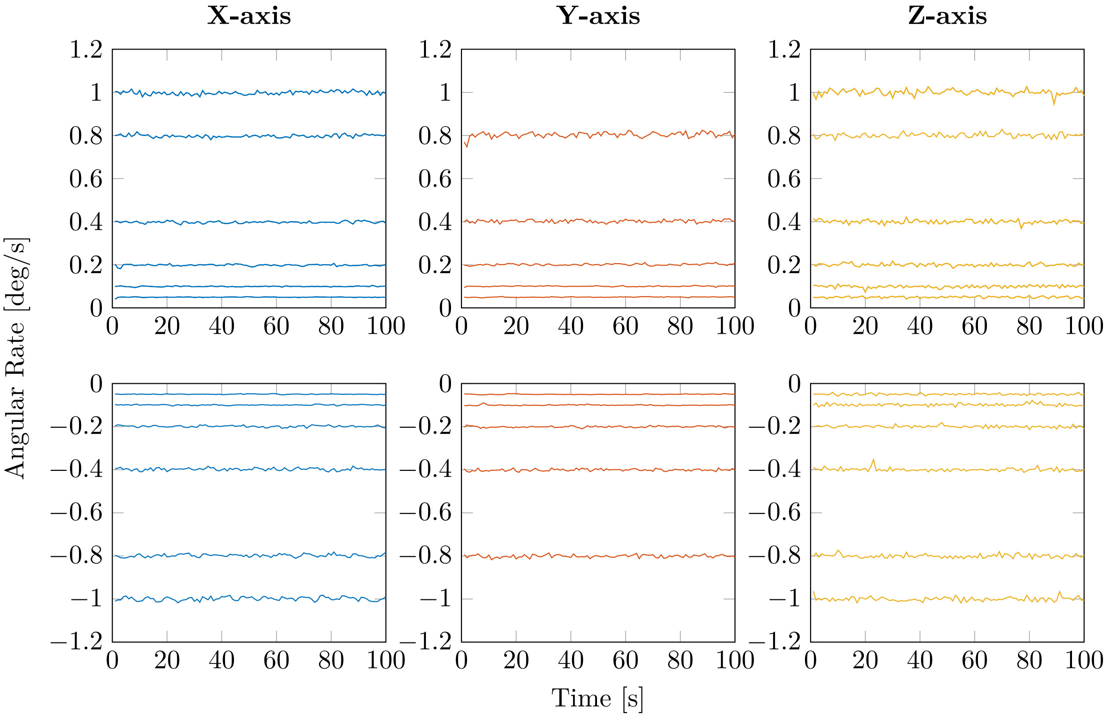


Figure 6.7 – Constant angular rate tests in all three axes of rotation

Results suggest reliable operation up to angular rates of ±0.8 deg/s in all three axes of rotation. Tests at a maximum angular rate of ±1 deg/s proved to be similarly reliable in the X- and Z-axes. Such tests were however omitted from the Y-axis. This was done due to test setup limitations rather than lacking stellar gyro capabilities.

Table 6.2 – Estimation accuracies for constant angular rate tests

	3σ Accuracy [deg/s]					
Angular Rate [deg/s]	X		Y		Z	
	+	-	+	-	+	-
0.05	0.0037	0.0033	0.0038	0.0036	0.0114	0.0128
0.1	0.0048	0.0054	0.0053	0.0058	0.0182	0.0174
0.2	0.0113	0.0116	0.0114	0.0097	0.0189	0.0161
0.4	0.0136	0.0170	0.0194	0.0140	0.0252	0.0232
0.8	0.0190	0.0196	0.0350	0.0188	0.0292	0.0241
1.0	0.0241	0.0258			0.0379	0.0280

Specific accuracies are detailed in table 6.2. Note that, in all three axes of rotation, increasing angular rates correspond to decreasing estimation accuracies. Such response can be expected, since increased angular rates cause star distribution patterns to deviate from the ideal symmetrical Gaussian distribution (see section 5.3). Such deviation impairs the centroiding process, meaning that star centroids are determined with less accuracy. This, in turn, reduces the accuracy of the estimated rates, which are proportional to the accuracy by which centroids are determined.

Rate estimates in the cross-boresight axes are also observed to be much more accurate than corresponding estimates in the boresight axis. This is particularly noticeable at lower angular rates, where estimates in the cross-boresight axes are more than three times as accurate. Such disparity is typical of star imaging solutions (see section 5.5).

All things considered, test results suggest reliable operation up to maximum angular rates of ± 1 deg/s in all three axes of rotation during spaceborne operation, verifying the designed stellar gyro's adherence to the maximum rate design specification.

6.5 Reliability Tests

Attitude control systems rely heavily on the availability of accurate attitude estimates. Stellar gyros should therefore, apart from adhering to strict accuracy requirements, be as robust and reliable as possible to ensure the availability of rate estimates throughout.

To evaluate stellar gyro reliability within the confines of its operational limitations, two rudimentary tests were devised that assess the urgency and reliability of the algorithms' response to either a constant angular acceleration/deceleration or abrupt changes in angular rates.

6.5.1 Constant Angular Acceleration/Deceleration

Satellites are often controlled to rotate at specific angular rates, usually in the interest of stabilization (e.g., a Y-Thomson mode of stabilization). Such rates are commonly achieved through gradual angular acceleration or deceleration. Apart from being required to operate at constant angular rates, stellar gyros should therefore also be capable of reliably estimating time varying angular rates.

Given an initial angular rate of ± 0.2 deg/s, the stellar gyro was subjected to a constant angular acceleration of $\pm 0.0015\,\mathrm{deg/s^2}$ for a period of 150 s. Due to rate table limitations, the angular rate can only be updated once every two seconds, resulting rather in a sampled approximation of the desired constant acceleration. Such approximation was deemed sufficiently accurate to evaluate stellar gyro reliability. Test results are shown in fig. 6.8.

Results are similar to those observed at constant angular rates. Table 6.3 details the 3σ accuracies over the entire interval. Once again, rates are estimated almost twice as accurately in the cross-boresight axes than in the boresight axis itself. The gradually increasing angular rates also correspond to steady declines in estimation accuracy. This is particularly noticeable in the X-axis. Specific accuracies are as anticipated based on constant angular rate tests, verifying the designed stellar gyro's reliance when faced with time varying angular rates.

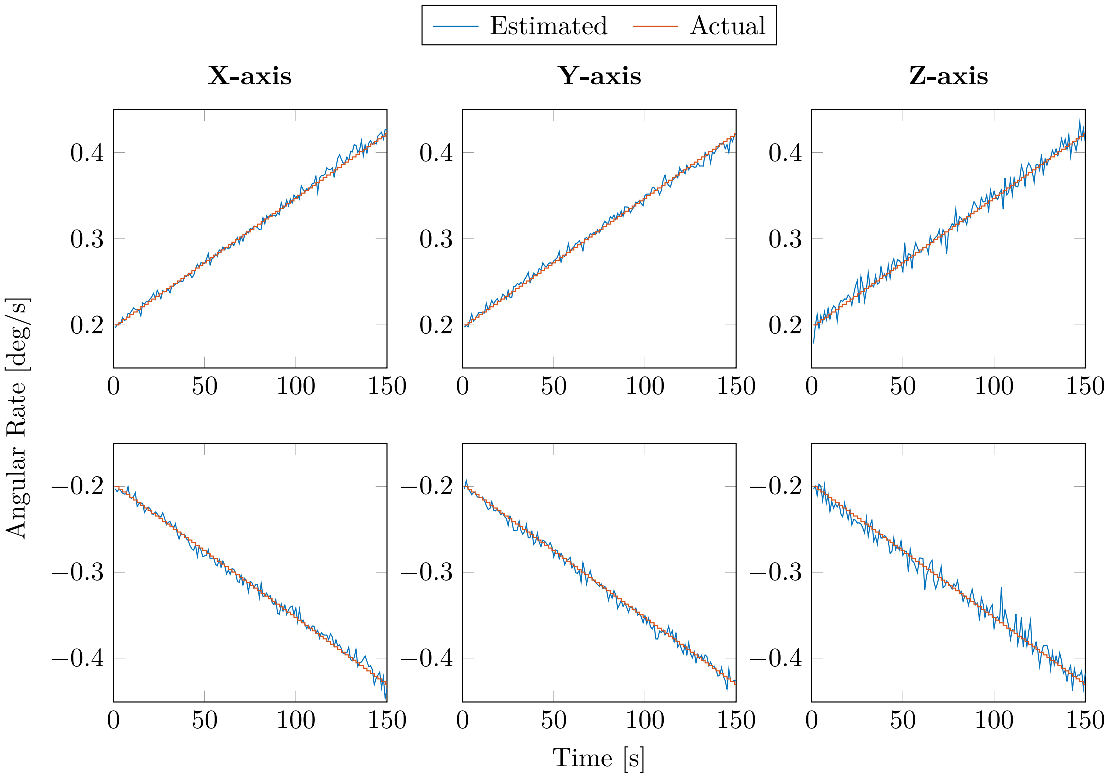


Figure 6.8 – Constant angular acceleration tests in all three axes of rotation

Table 6.3 – Estimation accuracies for constant angular acceleration/deceleration tests

	3σ Accuracy [deg/s]		
Acceleration [deg/s^2]	**X**	**Y**	**Z**
+ 0.0015	0.0140	0.0135	0.0247
− 0.0015	0.0168	0.0158	0.0294

6.5.2 Rate Steps

Preceding tests suggest that the designed stellar gyro is capable of reliably estimating both constant as well as gradually changing angular rates. In order to exhaustively evaluate its reliability, an assessment of the algorithms' response to abrupt changes in angular rates (approximating discontinuity) is required.

To this end, a test resemblant of bang-bang control was performed. From an initial standstill or slow rotation, angular rates were iteratively incremented and decremented throughout the testing process. Since the employed rate table is capable of accelerating at up to 500 deg/s^2, rates could be changed almost instantly.

The results shown in fig. 6.9 confirm that the designed stellar gyro is capable of operating reliably regardless of the abrupt changes in angular rates. This assessment is based solely on immediate observations, since comparative analyses of estimation accuracies were already performed in preceding tests. It's worth noting that, even

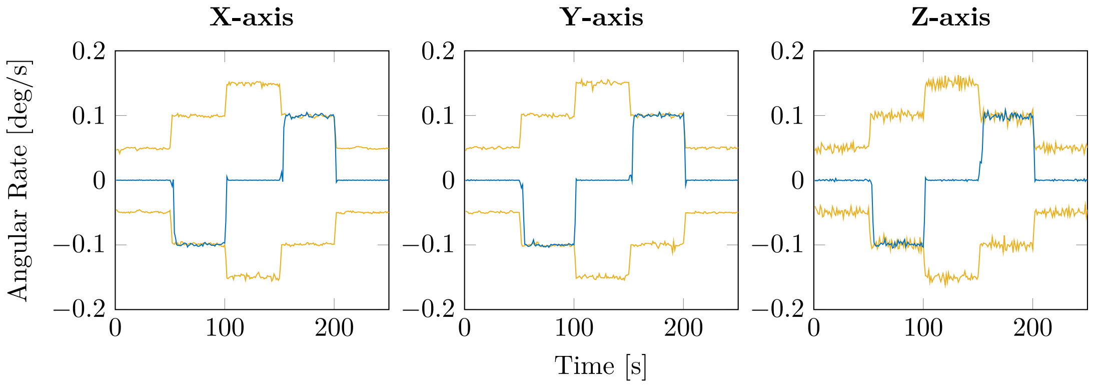


Figure 6.9 – Rate step tests in all three axes of rotation

though these tests provide a measure of conformation regarding algorithm robustness and reliability, such instantaneous rate “jumps” are unlikely to occur during spaceborne operation.

Chapter 7

Conclusion

This chapter summarizes the topics discussed throughout this thesis, draws conclusions based on those discussions that address the project aims, and ultimately provides recommendations for improvements and future work.

7.1 Summary and Conclusions

The project was aimed at the development and implementation of stellar gyro algorithms capable of operating interchangeably with orientation algorithms on existing CubeStar star tracker hardware. With a focus on low angular rates, the stellar gyro solution was expected to operate reliably at angular rates of up to ± 1 deg/s in all three axes of rotation.

7.1.1 Motivation

Small satellites have become increasingly popular for both commercial and educational space research and exploration missions. This success can largely be attributed to the CubeSat project, which presents a design standard for small scale satellites aimed at reducing development time and costs, as well as sustaining frequent launches and increasing accessibility to space. This standardization also provides a platform for international collaboration, as well as a support structure for less experienced and financially capable participants. This is epitomized by the upcoming QB50 mission, which will oversee the launch of 50 CubeSats from 27 different countries from all over the world with the goal of achieving affordable and sustainable access to space.

The size and power limitations imposed by the CubeSat design standard however places a strong emphasis on the miniaturization of satellite subsystems. To this end, CubeSats typically rely on MEMS gyros for attitude propagation purposes due to their affordability and compact design. One of the main drawbacks of MEMS sensors is bias drift, which causes errors to accumulate over time. This drift must in some way or another be compensated for, a task for which stellar gyros have proved to be exceptionally useful.

Stellar gyros are devices capable of inferring three-axis attitude propagation based on the displacement of a series of stars between successive image frames. Their design is analogous to that of star trackers, fundamentally consisting of a sensitive camera connected to a micro-controller unit (MCU). They also employ many of the same algorithms for star extraction and centroiding. The main difference being that star trackers apply the determined star vectors towards attitude *determination*, whereas in stellar gyros they are applied towards attitude *propagation*.

This analogy is particularly significant with regard to application, since it allows stellar gyro algorithms to be implemented alongside orientation algorithms on star tracker solutions. This adds a substantial measure of functionality to an existing subsystem without the need for any additional hardware (and very little additional power and execution time requirements). When used in combination with MEMS sensors it provides not only a means for drift compensation, but also a measure of functional redundancy with regard to attitude propagation.

7.1.2 Detection Capability

The viability of CubeStar as a platform for stellar gyro implementation was assessed based on its detection capabilities. It was determined that the RMS centroid bias error could be minimized by defocussing the camera optics to such an extent that a star PSF is distributed over a 5×5 pixel grid with a standard deviation of $\sigma_{PSF} = 0.7$. This necessitates that any pixel containing upwards of 3.7% of the total incoming photons should be detectable above the noise floor to ensure reliable operation throughout. By combining the star photon influx and sensor specific spectral response, the sensor specific photon influx was determined as a function of star magnitude. Taking into account the aforementioned detection requirement, CubeStar's absolute worst case detection threshold was determined as $M_V = 2.92$ stars. This corresponds to a sky coverage of approximately 80%. Fortunately, simulations assumed various overly conservative operational parameters. During spaceborne operation CubeStar is expected to reliabily detect stars up to $M_V = 3.8$, which corresponds an approximate full sky coverage ($> 99\%$).

7.1.3 Algorithms

The proposed stellar gyro retains CubeStar's original approach for extracting stars from raw image data. An image plane search algorithm is used to identify single pixels that could form part of a possible star, after which the remaining pixels associated with that star are extracted using a region based image segmentation method.

Once a star region is extracted, CubeStar employs a centre of gravity (CoG) approach to determine the corresponding star centroid. An alternative implementation of the standard CoG approach, termed "modified CoG", was suggested that aims to improve centroiding accuracy when faced with multiple brightest pixels. A more comprehensive weighted centre of gravity (WCoG) approach was also considered. Practical tests favoured the modified CoG approach, with average 3σ accuracies of 0.0116, 0.0083 and 0.0129 for the CoG, modified CoG and WCoG methods respectively.

Three distinct methods were considered to determine star correspondence between image frames. The RANSAC and star identification based approaches, even though advantageous in some respects, were deemed unnecessarily complex and processor intensive for implementation on CubeStar. Instead, star correspondence was determined using a proximity based approach, which dynamically adjusts the region of interest based on the availability of previous rate estimates. Simulations suggested flawless matching up to angular rates of ±1 deg/s in all axes whenever previous rate estimates were available.

Three rate estimation approaches were considered, namely singular value decomposition (SVD), Davenport's q-Method and weighted least-squares (WLS) estimation. Simulations suggested remarkable similarity between these methods in terms of accuracy and reliability. This suggested comparison based on alternative parameters, for which algorithm execution time was deemed most relevant. The timing analysis was performed using the MATLAB Profiler tool. Even though clear differences in execution time were observed, the order of magnitude of these differences rendered them largely insignificant with respect to overall execution time. Since simulations failed to identify a clear favourite, the WLS method was chosen due to its simplicity and ease of implementation.

7.1.4 Practical Tests

Simulation results were verified through means of practical testing. Considering the somewhat erratic propagation of light through the earth's atmosphere, tests were performed on simulated stars in a controlled environment rather than the night sky itself.

Rotation was achieved through means of an Ideal Aerosmith Model 1270VS rate table, which exhibits a rotational accuracy of ±0.1%. The night sky was simulated using a $(3.5 \times 2)\,\mathrm{m}^2$ circularly arched surface with a 2 m radius. The rate table, with stellar gyro mounted atop, was placed at the arch centre point. Stars were simulated using ultra bright white LEDs with a colour temperature of 5500 K, closely resembling that of many visible stars. Stars were introduced to the simulated night sky by mounting the LEDs behind small holes in the arched surface. Regardless of slight deviations from the ideal night sky model, the observed distribution patterns of simulated stars closely approximated their night sky counterparts.

The maximum angular rate capabilities were evaluated through means of constant rate tests. Results suggested reliable operation up to angular rates of ±1 deg/s in all three axes of rotation, with average 3σ accuracies of 0.0133, 0.0127 and 0.0219 deg/s in the X-, Y- and Z-axes respectively. Constant angular acceleration tests were performed to assess the stellar gyro's ability to reliably estimate time varying angular rates. Results were similar to those observed at constant angular rates, with average 3σ accuracies of 0.0154, 0.0147 and 0.0270 deg/s in the X-, Y-, Z-axes respectively. This verified the stellar gyro's performance when faced with time varying angular rates. In order to exhaustively evaluate the reliability of the proposed algorithms, the stellar gyro's response to abrupt changes in angular rates (approximating discontinuity) was assessed. Once again, angular rates were estimated successfully, verifying the proposed stellar gyro's reliability over the desired range of operation (rates of up to ±1 deg/s).

7.2 Recommendations for Future Work

This section discusses potential improvements to the proposed stellar gyro, as well as to the CubeStar star tracker module (with a focus specifically on improved stellar gyro functionality).

Consider alternative image sensors

Worthwhile considerations include sensor technology (CMOS or CCD), sensitivity, colour (or mono), resolution, pixel size, aspect ratio and shutter type. The implementation of a higher resolution, more light sensitive image sensor is of particular interest for CubeStar, since it would allow (and justify) using a more accurate and comprehensive centroiding method such as Gaussian Fit or Grid. Appendix B presents a detailed breakdown of significant image sensor selection criteria. The majority of recommendations that follow assume the implementation of an alternative image sensor (significant improvement to the CubeStar module mandates an alternative image sensor).

Implement a smaller FOV lens

One of the main disadvantages of a large FOV lens (such as the Lensation BL6012 used on-board CubeStar) is that it increases the risk of brightly illuminated celestial objects (such as the sun, earth, moon, etc.) falling within the camera FOV. Such objects often saturate the image sensor, disrupting stellar gyro operation in the process. The implementation of a more light sensitive image sensor would increase detection capabilities, allowing for full sky coverage to be achieved using a smaller FOV lens.

Operate at higher angular rates

Increased detection capabilities would also reduce the exposure time necessary to guarantee detection of the required minimum number of stars. Shorter exposure times would in effect reduce star smear, increasing the maximum angular rates at which reliable stellar gyro operation can be expected. This would encourage supplementing the existing stellar gyro with a high angular rate mode, transitioning between algorithms based on previous angular rate estimates (recall that stellar gyro algorithms differ drastically depending on the desired range of operation). This would require a significant amount of work, but could revolutionize the multifunctional star tracker/stellar gyro concept.

Consider alternative matching parameters

The proposed matching algorithm would benefit from an in-depth comparative analysis of the various parameters on which correspondence can be based. These parameters include proximity (favoured by the current design), number of pixels per star region, detected star brightness and even colour data (which would of course require a colour image sensor). Of interest is a publication by Enright and McVittie [48] that proposes a method for star identification based on a combination of geometric and colour data. The potential benefits of colour data for both star tracker and stellar gyro applications encourage further research regarding colour sensor implementation (see appendix B for a more in-depth discussion).

Consider the RANSAC approach for star validation and matching

Appendix B illustrates how an increase in detectable star magnitude would correspond to an almost exponential upsurge in the number of stars detected. Depending on the extent of such increase, it could call into question the reliability and efficiency of the somewhat primitive proximity based matching approach. Future iterations of CubeStar would therefore benefit from considering the more comprehensive and robust RANSAC approach for star validation and matching purposes.

Incorporate windowing

One of the advantages of CMOS sensors (over their CCD counterparts) is the ability to read image data from only specific areas of interest, a feature referred to as “windowing”. Recall that the proximity based matching approach is capable of estimating the projected location of a star based on previous angular rates, effectively determining such areas of interest. The image read time, as well as the memory required to store image data, can be reduced by reading only (a) these areas of interest and (b) the image bounding area (of which the size can also be determined based on previous angular rates). All stars within an image frame can reasonably be expected to fall within these areas.

Improved calibration techniques

CubeStar would benefit from a more accurate and comprehensive calibration procedure. Various publications [47, 49, 50] discuss the optical characterization and calibration of star trackers in great detail. Of particular interest to CubeStar is the publication by Dzamba and Enright [50], which discusses various calibration techniques specifically for low-cost star trackers. Furthermore, it might be worth investigating (autonomous) on-orbit calibration using captured star images. This would provide the means for compensating for any misalignments caused by vibration during launch. The concept is discussed by Samaan *et al.* [51].

Further consolidate star tracker and stellar gyro functionality

Even though the work presented in this thesis confirms the proposed stellar gyro’s compatibility with the CubeStar unit, it would nevertheless benefit from further consolidation of star tracker and stellar gyro algorithms. Suggested improvements include dynamic adjustment of image sensor parameters (such as frame rate, exposure time, etc.) based on the mode of operation (star tracker or stellar gyro), as well as a possible streamlined/simplified star extraction approach that incorporates windowing (see above). It might also be worth investigating the possibility of inferring the attitude quaternion using angular rate estimates whenever rates exceed the star tracker’s operational range.

Appendix A

Derivation of Crassidis' Method of Least-Squares Estimation

The following revision of vector algebra and attitude representation serves specifically as derivation for the first order approximation used in Crassidis's method of weighted least-squares (WLS) estimation. Other references do exist, but not to the derivation in its entirety. Equations (A.1) to (A.18) highlight the derivation of the rotation matrix representation specified by the SVD method. The majority of the information contained in this derivation is based on a publication by Shuster [11] regarding attitude representation.

Any vector in three dimensional space can be represented as

$$\mathbf{v} = v_1 e_1 + v_2 e_2 + v_3 e_3 \tag{A.1}$$

with base $\mathcal{E} : \{e_1, e_2, e_3\}$ designating the *abstract*[1] directions of the three coordinate axes. Of particular interest is the representation of the base with respect to itself

$$(e_1)_{\mathcal{E}} = \begin{bmatrix}1\\0\\0\end{bmatrix} = \mathbf{1} \qquad (e_2)_{\mathcal{E}} = \begin{bmatrix}0\\1\\0\end{bmatrix} = \mathbf{2} \qquad (e_3)_{\mathcal{E}} = \begin{bmatrix}0\\0\\1\end{bmatrix} = \mathbf{3} \tag{A.2}$$

Equation (A.2) is true for every base by definition. This results in numerical values **1**, **2** and **3** being independent of the base itself.

Arbitrarily rotating around the Z-axis (see fig. A.1), the equations of rotation are given by

$$e_1^{'} = \cos\theta\ e_1 + \sin\theta\ e_2 \tag{A.3}$$

$$e_2^{'} = -\sin\theta\ e_1 + \cos\theta\ e_2 \tag{A.4}$$

$$e_3^{'} = e_3 \tag{A.5}$$

Using eqs. (A.3) to (A.5), the general rotation matrix for a rotation about the Z-axis can be determined as

$$\mathbf{R}(\mathbf{3}, \theta) = \begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix} \tag{A.6}$$

[1] abstract in this context implies that meaning can be attached to these values without having to attach a numerical/concrete value

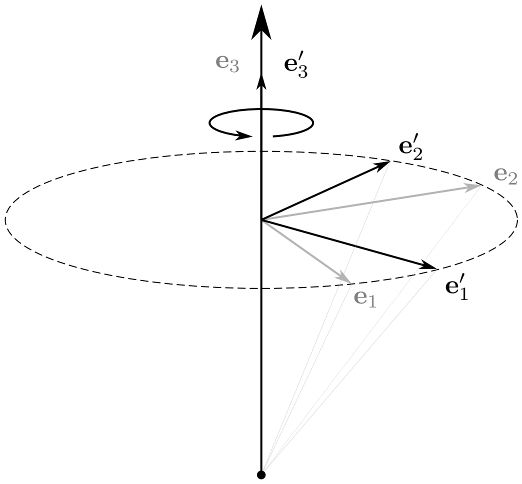


Figure A.1 – Arbitrary rotation of a three dimensional coordinate system about the Z-axis

The result corresponds with general rotation matrix theory. Note that eq. (A.6) denotes the rotation about a specific axis. In order to derive an expression for a rotation matrix about an arbitrary axis of rotation, further inspection is required

$$\begin{aligned}\mathbf{R}(\mathbf{3},\theta)\ \mathbf{1} &= \cos\theta \begin{bmatrix}1\\0\\0\end{bmatrix} - \sin\theta \begin{bmatrix}0\\1\\0\end{bmatrix} = \cos\theta\ \mathbf{1} - \sin\theta\ \mathbf{2}\\ &= \cos\theta\ \mathbf{1} - \sin\theta\ (\mathbf{3}\times\mathbf{1}) \qquad \text{(A.7)}\\ \mathbf{R}(\mathbf{3},\theta)\ \mathbf{2} &= \sin\theta \begin{bmatrix}1\\0\\0\end{bmatrix} + \cos\theta \begin{bmatrix}0\\1\\0\end{bmatrix} = \cos\theta\ \mathbf{2} + \sin\theta\ \mathbf{1}\\ &= \cos\theta\ \mathbf{2} - \sin\theta\ (\mathbf{3}\times\mathbf{2}) \qquad \text{(A.8)}\\ \mathbf{R}(\mathbf{3},\theta)\ \mathbf{3} &= \mathbf{3} \qquad \text{(A.9)}\end{aligned}$$

Recall that the numerical quantities **1**, **2** and **3** are independent of basis. Consequently, the general form of eqs. (A.7) to (A.9) can be applied to any basis. For $\mathbf{n}$ an arbitrary unit vector and $\mathbf{v}_\perp$ the projection of the vector $\mathbf{v}$ onto the plane perpendicular to $\mathbf{n}$, then

$$\begin{aligned}\mathbf{R}(\mathbf{n},\theta)\ \mathbf{v}_\perp &= \cos\theta\ \mathbf{v}_\perp - \sin\theta\ (\mathbf{n}\times\mathbf{v}_\perp)\\ &= \cos\theta\ \mathbf{v}_\perp + \sin\theta\ [\mathbf{n}\times]\ \mathbf{v}_\perp \qquad \text{(A.10)}\end{aligned}$$

Also note that $\mathbf{v}_\parallel$, the projection of $\mathbf{v}$ along $\mathbf{n}$, is not changed by rotation (see fig. A.1), i.e.,

$$\mathbf{R}(\mathbf{n},\theta)\ \mathbf{n} = \mathbf{n} \qquad \text{(A.11)}$$

thus

$$\mathbf{R}(\mathbf{n},\theta)\ \mathbf{v} = \mathbf{v}_\parallel + \cos\theta\ \mathbf{v}_\perp + \sin\theta\ [\mathbf{n}\times]\ \mathbf{v}_\perp \qquad \text{(A.12)}$$

Using Lagrange's formula, the vector projections $\mathbf{v}_{\|}$ and $\mathbf{v}_{\perp}$ can be rewritten in terms of know vector values $\mathbf{v}$ and $\mathbf{n}$ as

$$\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{b}(\mathbf{a} \bullet \mathbf{c}) - \mathbf{c}(\mathbf{a} \bullet \mathbf{b}) \tag{A.13}$$
$$\mathbf{n} \times (\mathbf{n} \times \mathbf{v}) = \mathbf{n}(\mathbf{n} \bullet \mathbf{v}) - \mathbf{v}(\mathbf{n} \bullet \mathbf{n}) \tag{A.14}$$
$$\mathbf{v} = \mathbf{n}(\mathbf{n} \bullet \mathbf{v}) - \mathbf{n} \times (\mathbf{n} \times \mathbf{v})$$
$$\mathbf{v} = \mathbf{v}_{\|} + \mathbf{v}_{\perp} = \mathbf{n}\mathbf{n}^T\mathbf{v} - [\mathbf{n}\times]^2\mathbf{v} \tag{A.15}$$

Using eq. (A.15), the rotation matrix can be expressed as

$$\mathbf{R}(\mathbf{n},\theta)\ \mathbf{v} = \mathbf{n}\mathbf{n}^T\mathbf{v} + \cos\theta(-[\mathbf{n}\times]^2\mathbf{v}) + \sin\theta([\mathbf{n}\times]^3\mathbf{v}) \tag{A.16}$$
$$= \mathbf{n}\mathbf{n}^T\mathbf{v} - \cos\theta[\mathbf{n}\times]^2\mathbf{v} - \sin\theta[\mathbf{n}\times]\mathbf{v} \qquad ([\mathbf{u}\times]^3 = -|\mathbf{u}|^2[\mathbf{u}\times])$$
$$\mathbf{R}(\mathbf{n},\theta) = \mathbf{n}\mathbf{n}^T - \cos\theta[\mathbf{n}\times]^2 - \sin\theta[\mathbf{n}\times]$$
$$= \mathbf{I} + [\mathbf{n}\times]^2 - \cos\theta[\mathbf{n}\times]^2 - \sin\theta[\mathbf{n}\times] \qquad ([\mathbf{u}\times][\mathbf{v}\times] = -(\mathbf{u} \bullet \mathbf{v})\mathbf{I} + \mathbf{v}\mathbf{u}^T)$$
$$\mathbf{R}(\mathbf{n},\theta) = \mathbf{I} - \sin\theta[\mathbf{n}\times] + (1-\cos\theta)[\mathbf{n}\times]^2 \tag{A.17}$$

Substitution of the equality ($\mathbf{I} = \mathbf{n}\mathbf{n}^T - [\mathbf{n}\times]^2$) into eq. (A.17) results in the rotation matrix representation specified by the SVD method

$$\mathbf{R}(\mathbf{n},\theta) = \cos\theta\ \mathbf{I} + (1-\cos\theta)\mathbf{n}\mathbf{n}^T - \sin\theta[\mathbf{n}\times] \tag{A.18}$$

Suppose the rotation vector representation $\boldsymbol{\theta} = \theta\ \mathbf{n}$. For θ infinitesimally small (frequently written as $\Delta\theta$), then $\sin\Delta\theta \approx \Delta\theta$ and $\cos\Delta\theta \approx 1$. Euler's formula (eq. (A.17)) becomes

$$\mathbf{R} = \mathbf{I} - [\boldsymbol{\Delta\theta}\times] + \mathrm{O}(|\boldsymbol{\Delta\theta}|^2) \qquad \text{where} \qquad \boldsymbol{\Delta\theta} = \Delta\theta\ \mathbf{n} \tag{A.19}$$

$\mathrm{O}(x^c)$ in eq. (A.19) denotes a quantity that becomes proportional to x^c as x tends to zero.

Assuming that the attitude is changing with time, the relation between the rotation matrices at successive sampling instances is given by

$$\mathbf{R}(t+\Delta t) = \boldsymbol{\Phi}(t+\Delta t, t)\mathbf{R}(t) \tag{A.20}$$
$$\boldsymbol{\Phi}(t+\Delta t, t) = \mathbf{R}(t+\Delta t)\mathbf{R}(t)^{-1} \tag{A.21}$$

where $\boldsymbol{\Phi}(t+\Delta t, t)$, also a rotation matrix, describes the rotation of $\mathbf{R}$ between sampling points t and $t+\Delta t$.

Considering the general form for an attitude rotation matrix as derived in eq. (A.19), $\boldsymbol{\Phi}(t+\Delta t, t)$ can be represented as

$$\boldsymbol{\Phi}(t+\Delta t, t) = \mathbf{I} - [\boldsymbol{\Delta\xi}(t)\times] + \mathrm{O}(|\boldsymbol{\Delta\xi}(t)|^2) \tag{A.22}$$
$$\mathbf{R}(t+\Delta t)\mathbf{R}(t)^{-1} = \mathbf{I} - [\boldsymbol{\Delta\xi}(t)\times] + \mathrm{O}(|\boldsymbol{\Delta\xi}(t)|^2)$$
$$\mathbf{R}(t+\Delta t) = \mathbf{R}(t) - [\boldsymbol{\Delta\xi}(t)\times]\mathbf{R}(t) + \mathrm{O}(|\boldsymbol{\Delta\xi}(t)|)\Delta t$$
$$\frac{1}{\Delta t}\Big(\mathbf{R}(t+\Delta t) - \mathbf{R}(t)\Big) \approx -\frac{1}{\Delta t}[\boldsymbol{\Delta\xi}(t)\times] \tag{A.23}$$

Note that $\boldsymbol{\omega}(t)$, the body-referenced angular velocity, is defined as

$$\boldsymbol{\omega}(t) \equiv \lim_{\Delta t \to 0} \frac{\boldsymbol{\Delta\xi}(t)}{\Delta t} \tag{A.24}$$

Substituting eq. (A.24) into eq. (A.23) gives

$$\begin{aligned} \frac{1}{\Delta t}\Big(\mathbf{R}(t+\Delta t) - \mathbf{R}(t)\Big) &\approx -[\boldsymbol{\omega}(t)\times]\mathbf{R}(t) && \text{(A.25)} \\ \mathbf{R}(t+\Delta t) &\approx -\Delta t[\boldsymbol{\omega}(t)\times]\mathbf{R}(t) + \mathbf{R}(t) \\ &\approx \Big(\mathbf{I} - \Delta t[\boldsymbol{\omega}(t)\times]\Big)\mathbf{R}(t) && \text{(A.26)} \end{aligned}$$

Recall that, in order to derive eq. (A.26), it was assumed that (a) the rotation θ – and therefore the body-referenced angular rate $\boldsymbol{\omega}$ – between successive measurements is infinitesimally small, and (b) terms higher than first order (hence the first-order approximation classification) in $\boldsymbol{\omega}\Delta t$ is ignored.

Appendix B

Image Sensor Selection

This appendix elaborates on various key image sensor characteristics, as well as their significance towards star tracker/stellar gyro capabilities.

Sensor Technology

Star tracker cameras traditionally employ one of two sensor types: CCD (charge-coupled device) or CMOS (complementary metal-oxide semiconductor). The decision of which to use is largely based on application, since both options feature distinct advantages over the other. Until recently, the digital vision sensor market was dominated by the CCD array. Technological advancements in terms of CMOS performance and efficiency has however reduced this divide significantly.

Fundamentally, CCD and CMOS sensors are based on similar principles. Incoming photons strike the image sensor array on light sensitive elements called *pixels*, which consist of specially doped silicon semiconductor substrates that release electrons proportional to the photon influx on their surface. The released electrons, collected over the course of a predetermined integration period, are processed based on the chosen technology.

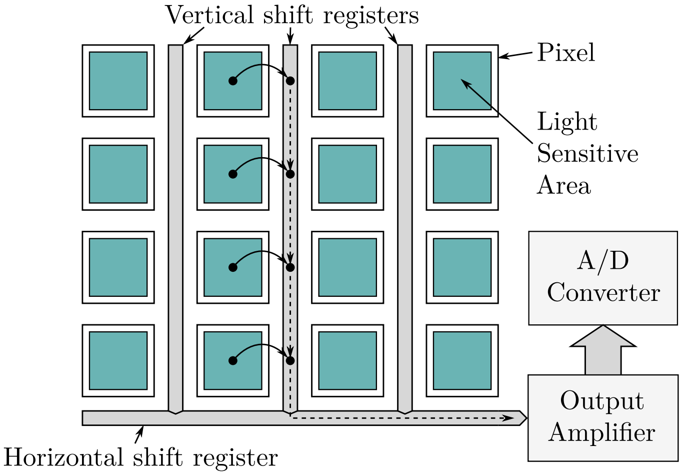


Figure B.1 – CCD sensor principles of operation

CCD sensors, as illustrated by fig. B.1, employ shift registers to carry the accumulated electron charge to external amplification and conversion circuitry. Such externalization facilitates an increased fill factor, a quantity that describes the ratio of a pixel's light sensitive area to its overall area. Since CCD sensors use a single charge-to-voltage converter for the entire pixel array, they also tend to exhibit a high pixel-to-pixel consistency. The main drawback of CCD sensors is speed, since the transfer and conversion of electron charge is limited by the capabilities of the shift registers and external conversion circuitry.

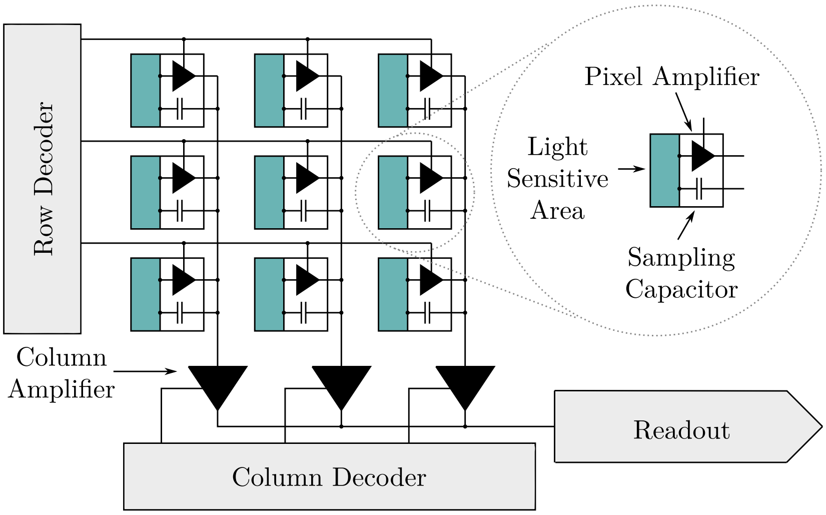


Figure B.2 – CMOS sensor principles of operation

CMOS sensors, as illustrated by fig. B.2, incorporate the amplification and charge-to-voltage conversion circuitry into each pixel. This facilitates much higher speeds than what can be achieved by CCD sensors, since the accumulated electron charge is converted at the pixel itself. The lessened flow of electrons also translate to much lower power consumption and dissipation. The per-pixel circuitry also enables "windowing", which refers to the selective addressing and readout of certain areas of interest. Unfortunately, the greatest strength of CMOS sensors also give rise to its most critical weaknesses. Since the necessary circuitry is incorporated into each pixel, CMOS sensors generally exhibit much smaller fill factors than their CCD counterparts. Furthermore, fabrication inconsistencies in the various charge-to-voltage conversion circuits introduce a measure of fixed-pattern noise to the overall image. Fortunately, such noise can easily be compensated for.

Modern day star trackers, specifically those designed for use in small form factor satellites such as CubeSats, strongly favour the CMOS variety due to its increased speed, smaller size and reduced power consumption. Even though CMOS sensors still exhibit smaller fill factors than their CCD counterparts, the difference is much less substantial than it once was.

Sensitivity

One of the most important characteristics of an image sensor is its sensitivity which, with regard to star tracker/stellar gyro applications, dictates the detectable star magnitude. The data presented in fig. B.3 shows the total number of stars within a certain visual magnitude, considering all stars in the Hipparcos, Yale Bright Star, and Gliese catalogues. This accentuates how an increase in sensitivity, and therefore in maximum detectable star magnitude, could correspond to an almost exponential upsurge in the number of detectable stars.

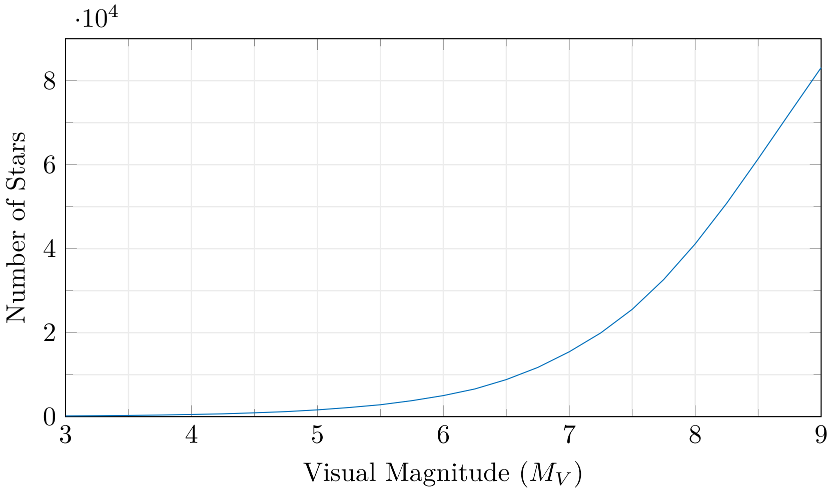


Figure B.3 – Total number of stars within a certain visual magnitude

The sensitivity is largely characterized by the image sensor's spectral response, which describes the relative efficiency of detection as a function of wavelength. Ideally, image sensor selection should seek to maximize the spectral response over the entire visible spectrum. Figure B.4 illustrates typical image sensor spectral response curves. This underlines another important consideration: choosing between monochromatic and colour configurations.

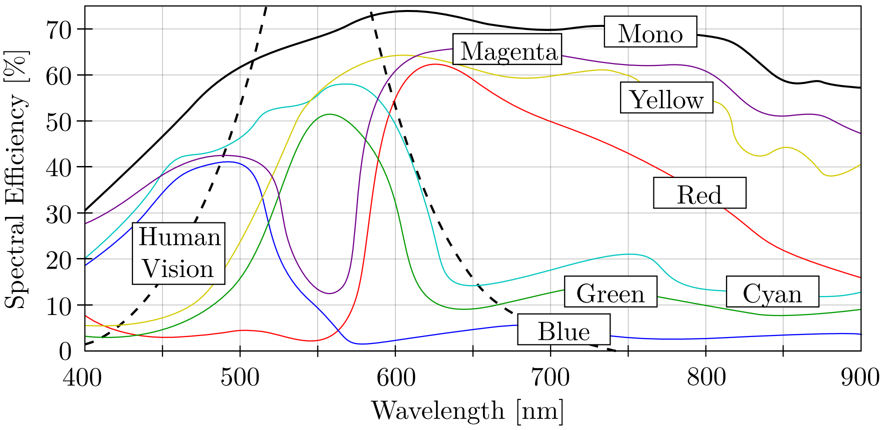


Figure B.4 – Typical image sensor spectral response curves

Monochrome vs Colour

Both CMOS and CCD sensors are inherently monochromatic. The electron charge converted to an equivalent voltage is only capable of capturing luminance information. To determine chrominance information, the sensor area is typically overlaid with a mosaic colour filter array (CFA) which assigns colour tones to each pixel.

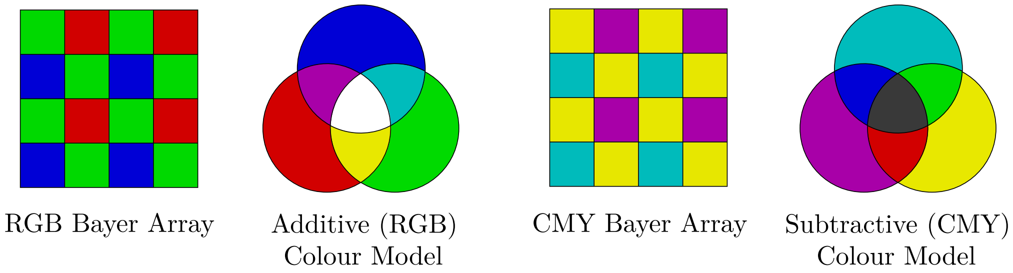


Figure B.5 – Illustration of the RGB and CMY Bayer colour filter patterns alongside their respective colour models

Figure B.5 illustrates the most commonly used CFA arrangement namely the Bayer array, which suggests two colour registration methods: RGB (Red-Green-Blue) and CMY (Cyan-Magenta-Yellow). The RGB pattern consists of 25% red, 50% green and 25% blue colour filters. The CMY pattern employs a similar ratio, with 25% cyan, 50% yellow and 25% magenta colour filters. The Bayer array prioritises green/yellow filters to mimic the physiology of the human eye, which is most sensitive to light in those portions of the visible spectrum.

One of the biggest disadvantages of employing such colour arrays is a loss of luminance information (and therefore sensitivity). Not only does it limit the photon influx on each pixel to a specific portion of the visible spectrum (see fig. B.4), but it also absorbs some of the colour it is intended to pass due to the non-ideal nature of the colour dyes. This is where the CMY pattern poses a significant advantage over its RGB counterpart. The RGB CFA is created by overlaying two layers of dye: yellow and magenta for red, yellow and cyan for green, and magenta and cyan for blue. The CMY pattern, on the other hand, consists of only a single dye per colour. The result is that the CMY CFA absorbs a much smaller portion of the light it is intended to pass. As illustrated by fig. B.4, this leads to a much better signal-to-noise (SNR) ratio and spectral response.

From a pure sensitivity perspective, star trackers would benefit from employing the monochromatic image sensor variety. Since if forgoes any form of filtration, each pixel is subjected to the entire spectrum of light incident on its surface. This maximizes both the achievable SNR as well as the spectral response over the entire visible spectrum. Furthermore, colour sensors also require various sophisticated interpolation techniques to reconstruct the final image from raw data. This is referred to as "demosaicing". Monochrome sensors, which avoid the need for such reconstruction, therefore produce slightly higher resolution images that more accurately depict the actual photon distribution on the sensor array.

It is however worth investigating the application of colour data (and therefore colour sensors) towards star identification and matching. Of interest is a publication by Enright and McVittie [48] that proposes a method for star identification based on a combination of geometric information (as favoured by traditional star identification algorithms) and amplitude ratios calculated from the red, green and blue colour channels. They suggest that, by incorporating chromatic dat,a the proposed algorithms are capable of finding matches with only two imaged stars in most regions of the sky (in contrast to traditional identification algorithms that require a minimum of three stars for reliable matching). The detection capabilities of such a design could potentially be bolstered even further by opting for a CMY colour arrangement. Considering the stellar gyro project at hand, colour data could also be used to match stars between image frames. All things considered, the multitude of possible applications strongly encourage further research with regard to incorporating colour data into star tracker/stellar gyro functionality.

Resolution and Pixel Size

Recent advancements in image sensor technology has shown a gradual increase in the imaging resolution of small form factor star trackers, with resolutions in the multi-megapixel range not uncommon. The key benefit of a larger resolution image sensor is that a star PSF is distributed over a much larger number of pixels. Considering the CubeStar module, not only would this greatly increase the centroiding accuracy for existing CoG methods (recall that image resolution and centroiding accuracy was determined as the limiting factors with regard to the achievable rate estimation accuracy), but it would also justify the use of more comprehensive centroiding methods such as Gaussian Grid or Fit.

Another important consideration is pixel size. Larger pixels tend to capture more photons, increasing the image sensor's sensitivity (and therefore detection capabilities). Terrestrial applications such as automotive or surveillance cameras prioritize high resolutions since sensitivity is often not an issue. For star tracker/stellar gyro applications, on the other hand, these considerations are of almost equal significance. Assuming strict limitations with regard to image sensor dimensions (as is usually the case for small form factor star trackers/stellar gyros), the pixel size must be large enough to attain the desired sensitivity (detectable star magnitude) without reducing the resolution to such an extent that it compromises centroiding accuracy. Furthermore, it is worth noting that high resolution image sensors tend to consume more power than their similarly sized, low resolution counterparts.

Aspect Ratio

The Melexis MLX75412 image sensor (as employed by CubeStar) features a 2:1 aspect ratio, which means that a large portion of the lens's FOV is projected outside the image sensor area. This is to some extent compensated for by using an exceptionally large FOV lens. Future iterations of CubeStar, which might opt for a smaller FOV lens, would benefit from implementing an image sensor approximating a square form factor (aspect ratio of 1:1).

Shutter Type

The MLX75412 employs a rolling shutter mechanism, in which rows are exposed sequentially. The global (frame) shutter mechanism, on the other hand, exposes all pixels simultaneously. Consequently, image sensors that employ a rolling shutter mechanism are more prone to certain distortion effects, specifically those related to movement (such as motion blur). This is of little significance to star trackers, which are designed to operate exclusively near zero angular rates. Stellar gyros, on the other hand, rely on the continuation of rotation (movement). Using a rolling shutter mechanism could therefore contribute to image distortion, impairing the accuracy of rate estimates.

Image Sensors Considered for CubeStar

(see next page)

Table 1

Manufacturer	Model	Resolution			Pixel Size (um)	Physical Size (mm2)		Optical Format (inch)	Power Consumption (mW)		Chroma	Shutter	Output		Operating Temperature		Package	Documentation
		Ratio		Megapixel														
		(H)	(V)			Package	Sensor Area		Specified	Conditions			Interface	bits	Min	Max		
Melexis	MLX75412	1024	512	0.52	5.6	9,56 x 9,77	5,80 x 2,91	1/3	250		iRGB, Mono	Rolling	TWI (Master, Slave)	8/10/12	-40	85	55-pin Glass-BGA	Full
E2V	EV76C560	1280	1024	1.31	5.3	12,7 x 12,7	6,90 x 5,50	1/1.8	200	@ 60fps, full format	RGB, Mono	Global	SPI	8/10	-30	65	48-pin CLCC	Full
Clairpixel	CP5112	1280	960	1.23	3.65		4,64 x 3,72	1/3			RGB	Rolling	TWI (Master, Slave)	8/10	-40	105		-
Fairchild	HWK1910A	1920	1080	2.07	5.0			2/3	500	@30fps, 20degC	RGB	Rolling		16	-30	70	64-pin CLCC	Preliminary Brief
Hamamatsu	S11661	1280	1024	1.31	7.4	16,5 x 16,5	9,47 x 7,58		475	@ 8fps, 25degC (self calculated - unsure)		Rolling/Global	SPI	12	-10	65	Ceramic	Comprehensive Brief Datasheet
Dynamax Imaging	DYN0021	1920	1080	2.07	5.0		9,60 x 5,40	2/3	760	@ 60fps, 70degC, dark	RGB, Mono	Rolling/Global	SPI	8/10/12			98-pin CLCC	Brief Datasheet
CMOSIS	NanEyeGS	620	640	0.40	3.6	3,40 x 3,40	2.30 x 2.30		100		RGB, Mono	Global		10	0	60		-
New Imaging Technologies (NIT)	NSC0902	768	576	0.44	5.6		4,30 x 3,20	1/4 (1/3)	230	@ 50fps, full frame	Mono	Rolling		8	-40	90	48-pin CLCC	Flyer
	NSC1005	1280	720	0.92	5.6		7,10 x 4,00	1/2	260	@ 50fps, full frame	RGB, Mono	Rolling		10	-40	90	48-pin CLCC	Flyer
	NSC1003	1280	1024	1.31	6.8		8,70 x 6,90	2/3	250	@ 50fps, full frame	RGB, Mono	Rolling/Global			-40	90	48-pin CLCC	Flyer
SiOnyx	XQE-0570	872	654	0.57	5.6	14,2 x 14,2	4,88 x 3,66	1/3	300		RGB, Mono	Rolling	I2C	12	-20	45	48-pin CLCC	Brief Datasheet
	XQE-0920	1280	720	0.92	5.6	16,5 x 16,5	7,17 x 4,03	1/2	400		RGB, Mono	Rolling	I2C	12	-20	45	48-pin CLCC	Brief Datasheet
	XQE-1310	1280	1024	1.31	10	22,0 x 23,0	12,8 x 10,2	1	400		RGB, Mono	Rolling	I2C	12	-20	45	60-pin COB/CLCC	Brief Datasheet
Luxima Technology	LUX 1310	1280	1024	1.31	6.6	16,9 x 16,9		2/3	2000	@ 1070fps full resolution, adjustable with lower frame rates (eg. 670mW @ 250fps)	RGB, Mono	Global		16			88-pin CLCC	Brief Datasheet
	LUX 1310-S	1280	1024	1.31	6.6	16,9 x 16,9		2/3	670	@ 270fps full resolution	RGB, Mono	Global		16			88-pin CLCC	Brief Datasheet
OmniVision	OV10133	1280	720	0.92	4.2	11,0 X 11,0	5,51 x 3,42	1/3	532		Mono	Rolling	SCCB (Serial Camera Control Bus) - Should work with I2C	8/10	-20	85	96-pin CLGA	Product Brief
	OV9211-V28A	1280	800	1.02	3	5,43 x 4,43	3,89 x 2,43	1/4	110		RGB, Mono		SCCB Slave	8/10	-30	70	28-pin CSP2	Product Brief
	OV09215-F48Y	1280	800	1.02	3	12,0 x 12,0	3,89 x 2,43	1/4	110		RGB, Mono		SCCB Slave	8/10	-30	95	48-pin QFP2	Product Brief
Aptina	MT9M034	1280	960	1.23	3.75	10,0 x 10,0	4,80 x 3,60	1/3	270	@ 60fps, full resolution	RGB, Mono	Rolling	TWI (Master, Slave)	12	-30	70	48-pin iLCC	Full
	MT9M001C12STM	1280	1024	1.31	5.2	14,2 x 14,2	6,66 x 5,32	1/2	363		Mono	Rolling	TWI (Master, Slave)	10	0	70	48-pin CLCC	Full
	MT9M021/31	1280	960	1.23	3.75	9,00 x 9,00	4,80 x 3,60	1/3	270		RGB, Mono	Global	TWI (Master, Slave)	12	-30	70	64-ball iGBA / 48-pin iLCC	Full
	MT9M024	1280	960	1.23	3.75	9,00 x 9,00	4,80 x 3,60	1/3	270	@ 60fps, 720p, linear	RGB, Mono	Rolling	TWI (Master, Slave)	12	-30	70	64-ball iGBA	Full
	AR0134	1280	960	1.23	3.75	9,00 x 9,00	4,80 x 3,60	1/3	400		RGB, Mono	Global	TWI (Master, Slave)	12	-30	70	64-ball iGBA / 48-pin iLCC	Full
	AR0130	1280	960	1.23	3.75	10,0 x 10,0	4,87 x 3,67	1/3	270		RGB, Mono	Rolling	TWI (Master, Slave)	12	-30	70	48-pin iLCC	Full
ViiMagic	9211	1920	1080	2.07	5	29,2 x 29,2	10,2 x 5,74	2/3	500	@ 60fps	Mono	Rolling		12	-20	75	84-pin LCC	Flyer
	9215	1920	1080	2.07	5	25,0 x 20,0	10,2 x 5,74	2/3	500	@ 60fps	Mono	Rolling		12	-20	75	141-pin uPGA Ceramic	Flyer
ON Semiconductor	PYTHON2000	1920	1200	2.30	4.8				1500	in 10-bit mode	RGB, Mono	Global	SPI	8/10	-40	85	84-pin LCC	Flyer
	VITA2000	1920	1200	2.30	4.8	19,1 x 19,1	9,25 x 5,79	2/3	385		RGB, Mono	Rolling/Global	SPI	8/10	<0	70	52-pin LCC	Full
	PYTHON1300	1280	1024	1.31	4.8			1/2	375		RGB, Mono	Rolling/Global	SPI	8/10	-40	85	48-pin LCC	Product Brief
	PYTHON500	800	600	0.48	4.8			1/3.6	375		RGB, Mono	Rolling/Global	SPI	8/10	-40	85	48-pin LCC	Product Brief
	VITA1300	1280	1024	1.31	4.8	14,2 x 14,2	6,18 x 4,95	1/2	290		RGB, Mono	Rolling/Global	SPI	8/10	-40	85	48-pin LCC	Full
	STAR1000	1024	1024	1.05	15			1	400		RGB, Mono	Rolling		10			84-pin Ceramic JLCC	Comprehensive Brief Datasheet
Sony	IMX238LQJ	1305	1049	1.37	3.75			1/3			Mono		I2C, 4 Wire Serial	12			94-pin LGA	Flyer
Mikro Tasarim	MT6410DA	640	512	0.33														
	MT12810DA	1280	720	0.92														

Appendix C

Support Software

C-Functions

The stellar gyro algorithms were implemented on CubeStar in the C programming language. An overview of the stellar gyro specific functions is provided.

```
void matchStars (void)
```

The *matchStars* function oversees the matching process. This involves calling auxiliary functions (as detailed below), specifying ROI parameters, as well as resolving match disputes that arise from multiple *matchID* index entries that point to a single centroid in the *oldCentroidList* structure.

```
uint8_t matchingProcess (int i, float search_radius, uint8_t
    ratecheck)
```

The *matchingProcess* function is responsible for matching stars between successive images. It accepts an index integer *i* that points to a specific centroid in the *oldCentroidList* structure, a float *search_radius* that defines the maximum Euclidean distance between a star centroid and the boresight coordinates that should register consensus, as well as an 8-bit unsigned integer *ratecheck* that specifies the availability of angular rates. The centroid at location *i* in the *oldCentroidList* structure is compared to each centroid in the *centroidList* structure in an attempt to find a possible match. The index of each centroid in the *centroidList* structure that registers consensus is tied to the index value *i* and stored in a *matchID* structure for further processing.

```
void clearMatchedStars (void)
```

Clear all entries in the *matchID* structure. This ushers in each new matching iteration.

```
void calcRates (void)
```

Estimates angular rates using the unit vector coordinates of matched star centroids as tied by their index values in the *matchID* structure.

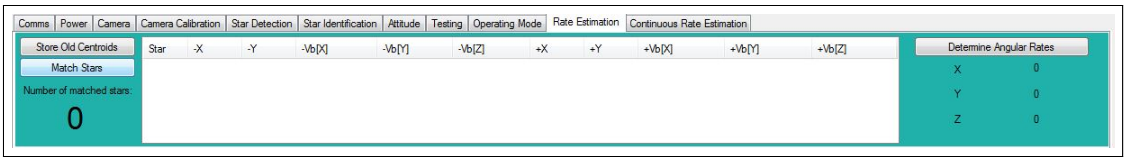
```
void resolveDuplicate (int i, int j, uint8_t ratecheck)
```

The *resolveDuplicate* function is responsible for resolving match disputes that arise from duplicate *matchID* index entries that point to a single centroid in the *centroidList* structure. It accepts integers *i* and *j*, which represent the *matchID* index values for the duplicate entries, as well as an 8-bit unsigned integer *ratecheck* that specifies the availability of previous angular rate estimates.

CubeStar Ground Support – Additions

The CubeStar ground support software package is a standalone application, written in the C# programming language, that provides the means for interfacing with the CubeStar unit through either UART or I2C interfaces. Additions to the ground support package, specifically those related to stellar gyro operation, are detailed here.

Rate Estimation Mode



Number of matched stars: 0

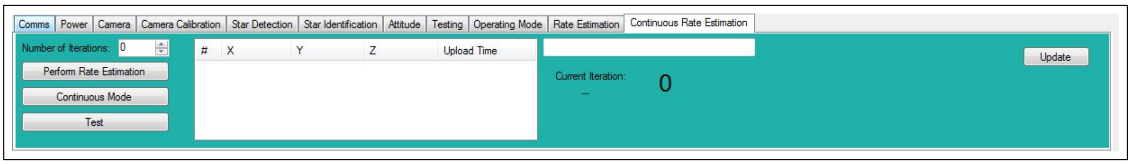
Star	-X	-Y	-Vb[X]	-Vb[Y]	-Vb[Z]	+X	+Y	+Vb[X]	+Vb[Y]	+Vb[Z]

X 0
Y 0
Z 0

Figure C.1 – Rate estimation mode for the CubeStar ground support package

The rate estimation mode, as illustrated by fig. C.1, facilitates star matching between two manually uploaded images. The image plane and body-fixed unit vector coordinates, as determined from both image frames, are provided for each matched star. The three axis angular rates are estimated based on the matched stars.

Continuous Rate Estimation Mode



Number of Iterations: 0

#	X	Y	Z	Upload Time

Current Iteration: 0
--

Figure C.2 – Continuous rate estimation mode for the CubeStar ground support package

The continuous rate estimation mode, as illustrated by fig. C.2, automates the stellar gyro process. Star images, either captured or generated (using a simulation environment), are sequentially uploaded to the CubeStar unit. Star centroids are extracted and matched between images, after which angular rates are estimated. The process iterates until all images in the specified directory has been accounted for.

MATLAB Scripts

This section presents an overview of selected MATLAB scripts that proved relevant to the stellar gyro design procedure.

Star Catalogue Generator
The script reads the HYG star database[1] (which contains all stars in the Hipparcos, Yale Bright Star and Gliese catalogues) provided in either *xls* or *xlsx* file format. It extracts the ID, right ascension (which it converts from hours to degrees), declination, and visual magnitude of each star. The extracted data is stored in a *mat* file for future use.

Plot Magnitude Distribution
Uses the data extracted via the star catalogue generator to determine the number of stars within a certain visual magnitude. Extracted stars are then plotted in three-dimensional space to illustrate distribution patterns over the celestial sphere. The script also plots a 2D projection of the celestial sphere (i.e., a star map).

Practical Test Scripts
Provides a means for interfacing with both the Aerosmith Model 1270VS rate table and the CubeStar star tracker unit via the MATLAB environment (includes image download functionality). Scripts used to evaluate the stellar gyro algorithms as implemented on CubeStar are also included.

Random Image Generator
Generates an arbitrary number of images, each with random boresight coordinates. Can be used to evaluate the reliability of CubeStar's orientation algorithms.

Sky Coverage Simulation
Performs Monte Carlo simulations that determine the sky coverage (for both a two and three star minimum) as a function of the circular FOV for various visual magnitudes.

Stellar Gyro Simulation
Encompasses the entire stellar gyro simulation process as detailed in section 5.1. The procedure is also visualized.

[1] Available at: `http://www.astronexus.com/hyg` [25 September 2015]

Appendix D

Important Datasheet Extracts

ESL
Electronic Systems Laboratory

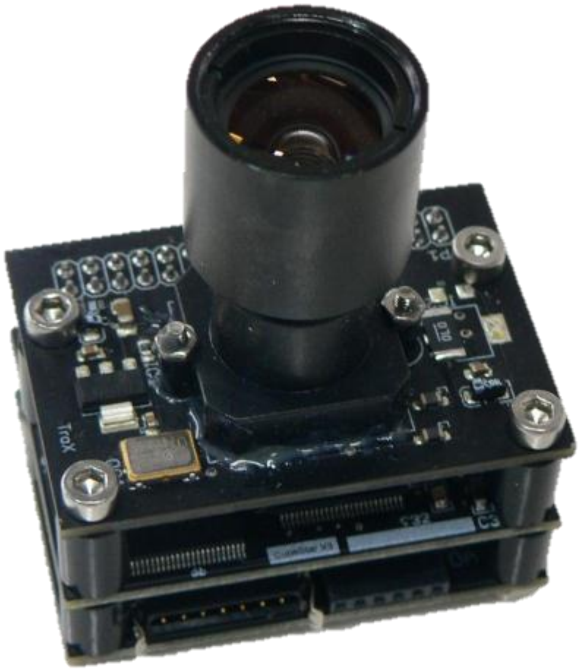
CubeStar

Nano Star Tracker

Description

CubeStar is a CubeSat compatible nano star tracker. CubeStar can operate in full autonomous mode, outputting attitude estimates in inertial quaternions at a rate of 1Hz.

CubeStar builds largely on the proven hardware designs of CubeSense and CubeComputer.



Features

- Output attitude quaternions or raw images.
- Onboard current monitors and power switches safeguard against radiation induced latchups.

Specifications (without Baffle)

Weight (g)	56g
Dimensions (mm)	50.1 x 35.2 x 64
Power (mW)	320 nominal 500 peak
Operating Voltage (V)	3.3
Data Interface	I^{2}C / UART
Field of View (deg)	52 x 27
Star Catalogue Size	415

Performance

Sensitivity Range (Star Mag)	< 4.0
Number of Stars Tracked	15
Accuracy (deg, 1σ)	0.01 (cross bore) 0.03 (roll)
Update Rate (Hz)	1
Max Tracking Rate (deg/s)	0.3
Max Acquisition Time (ms)	1000

- CubeStar has been tested in a lab environment and under the night sky.
- Vacuum, thermal and vibration testing to be completed.

Electronics Systems Laboratory
Engineering Faculty
University of Stellenbosch
Private Bag X1 Matieland 7602
South Africa

Contact Us:
Phone: +27-21-808-4926
Fax: +27-21-808-4981
Email: whsteyn@sun.ac.za
URL: http//www.esl.sun.ac.za

UNIVERSITEIT
STELLENBOSCH
UNIVERSITY

Features and Benefits

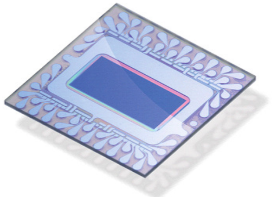
- ❑ 1024 x 512 pixels CMOS image sensor
- ❑ 154dB Extended HDR
- ❑ Low noise, low power rolling shutter
- ❑ 1/3” Optical format for 1024x512
- ❑ 1/4" Optical format with VGA subwindow
- ❑ Monochrome, standard Bayer, RCCC and special Bayer RGBi
- ❑ Parallel data output
  8/10/12 bits + CLK/HSYNC/VSYNC
- ❑ Operating Temperature Range:
  -40C to +85℃ full performance
  -40C to +115℃ reduced performance
- ❑ Storage Temperature Range:
  -40℃ to +125℃
- ❑ Automotive Qualified AEC-Q100

MLX75412 Only

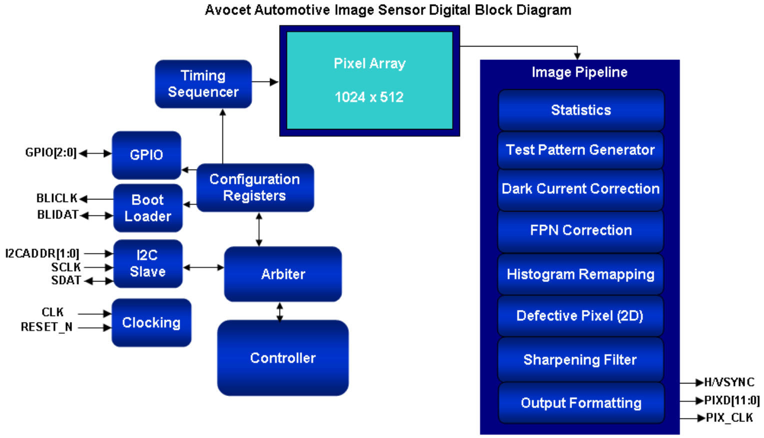
- ❑ Autobrite® auto-exposure (AE) and auto-HDR. This high performing function automatically sets the best exposure and HDR setting on a frame by frame basis.
- ❑ Autoview^{TM} histogram remapping. This function enhances display viewing performance.

Application Examples

- ❑ Automotive Driver Assistance Systems (ADAS)
  - ❑ Lane departure warning (LDW)
  - ❑ Forward collision warning (FCW)
  - ❑ Automatic high-beam assist
  - ❑ HDR rear-view
- ❑ Cameras on trucks, trains, busses, emergency vehicles, agricultural vehicles, autonomous vehicles, heavy off-road vehicles
- ❑ Night vision cameras
- ❑ HDR surveillance cameras
- ❑ Traffic monitoring cameras
- ❑ Fleet Safety/ Black-box cameras



1 *Functional Diagram*



.

2 *General Description*

The Avocet image sensor integrates a high-sensitivity array, a feature-rich digital processor for monochrome images and camera control functions into a single chip.

In a monochrome system, image processing is performed on the Avocet as a single chip solution.

In a color system Avocet acts as a slave in a system controlled by a separate Image Signal Processor chip through one of the external interfaces delivering raw single images or video-like streams of color Bayer-patterned images. The sensor captures the images at the required speed, exposure, and gain, and the external device is responsible for all further image processing such as color demosaicing and white balance.

7 Sensor Specific Specifications

DC Operating Parameters $T_A = 25^{o}C$, $V_{DDIO} = 3.3V \pm 5\%$

Parameter	Symbol	Test Conditions	Min	Typ	Max	Units
Sensitivity	SNR10	@535nm		25		nW/cm^2
Responsivity		@535nm		2.28		e-/DN12
Dark Leakage		T=65 °C		1008		DN12/sec
Number of barriers					6	
Dynamic Range					154	dB
Chief ray angle the sensor has been optimized for.	CRA			10		degrees

Table 6: Specifications: Optical Characteristics

8 Device Overview

8.1 Avocet Sensor Overview

The Avocet image sensor integrates a high-sensitivity array, a feature-rich digital processor for monochrome images and camera control functions into a single chip.

Specification	Avocet	Comments
Active Resolution	1024 x 512	Wider horizontal resolution for next generation ADAS
Optical format	~1/3“(6.45mm)	Center 1/4“ can be used for VGA resolution
Pixel size	5.6µ square	Optimized for sensitivity at 1024x512 resolution
Max frame rate	60fps	At full resolution
Input clock range	20 – 54 MHz	Options for clocking: Crystal input, Oscillator
Exposure time range @ 60fps	12.5µs – 16.7 ms	At 54MHz and 60fps, at full resolution and speed. Minimum barrier time 1.22us.
Control interface	2 Wire Boot Loader Interface	Used to load register settings on recovery from a reset. Must be accessible (for programming of serial PROM) via other control interfaces.
	2 Wire Serial Slave	2-Wire, low speed, serial control interface used for short distances. Supports broadcast writes for writing to multiple imagers.
Video interface	LVTTL	12-bit Monochrome or Raw Color. Pixel clock, vsync and hsync compatible with DSPs. (i.e. TI DaVinci or ADI Blackfin)
Signal processing	Defect pixel interpolation FPN correction Histogram Optimization Dark Current correction Sharpening	Sensor provides on-chip processing required for vision applications or monochrome display applications. (Color processing is not included in on-chip functions.)
Scanning modes	Progressive	Required to support machine vision applications
	Subsample	2x and 4x vertical subsampling
	Subwindow	Single rectangular region. The starting point of the x- and y-address is programmable, as well as the window size.
Sensitivity	SNR10	25nW/cm2 @ 25degC @ 535nm
Responsivity	2.28 e-/DN12	In dark at 25°C 535nm
Dark leakage	1008 DN12/sec	At 65°C

Table 4 - Avocet image sensor highlights

8.3 Pixel Array Description & Operation

The pixel array is composed from 5.6um square sensing elements. The image sensor operates using an electronic rolling shutter. This maximizes the amount of integration time available by overlapping the readout with the “reset” of a pixel to begin integration.

Avocet operates in an adaptable, programmable Extended High Dynamic Range (HDR) mode maximizing sensitivity while providing uncompromising dynamic range.

The response curve is stretched towards the NIR region, featuring high sensitivity up to 850nm.

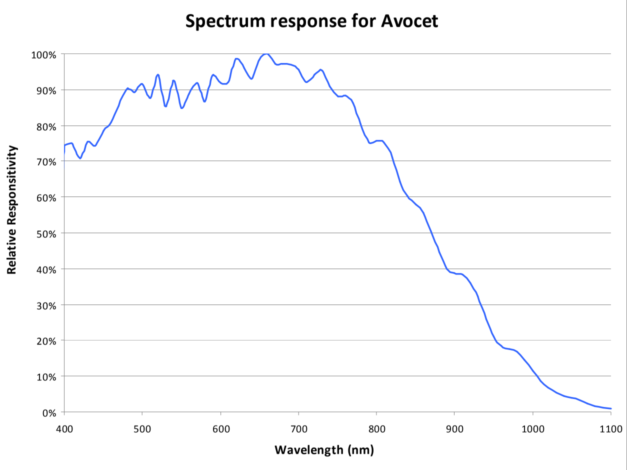


Figure 2 – Avocet spectral response: relative responsitivity

9 55-pin Glass-BGA package information

The following sections detail the information about the pinout and physical dimensions of Avocet in the 55-pin GBGA package. Please refer to the “Printed Circuit Board (PCB) Layout” for information concerning use of Avocet on a circuit board.

9.1 GBGA55 Mechanical Drawings

The following sections contain mechanical drawings and dimensional information for the GBGA55 package.

9.1.1 Package mechanical drawings

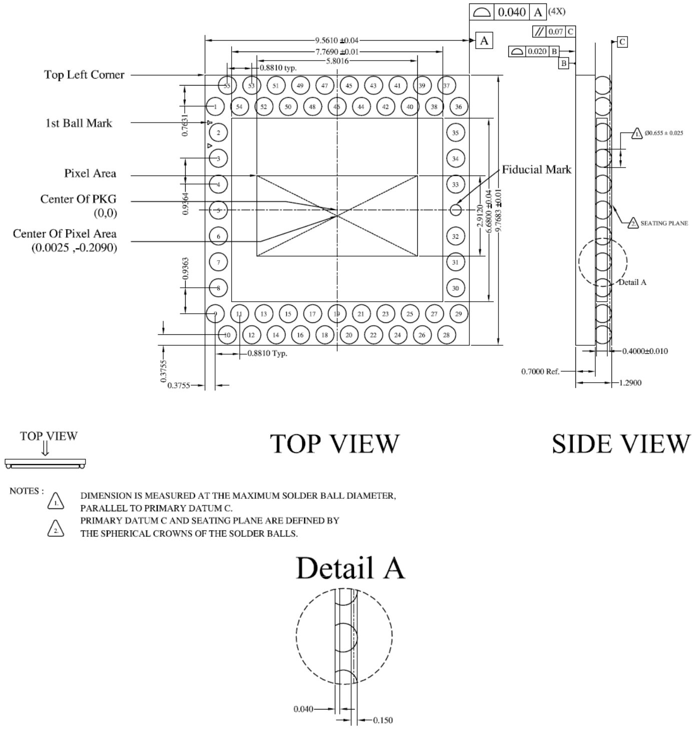


Figure 13 – 55-pin GBGA package mechanical drawings

EFM®32

... the world's most energy friendly microcontrollers

EFM32GG280 DATASHEET

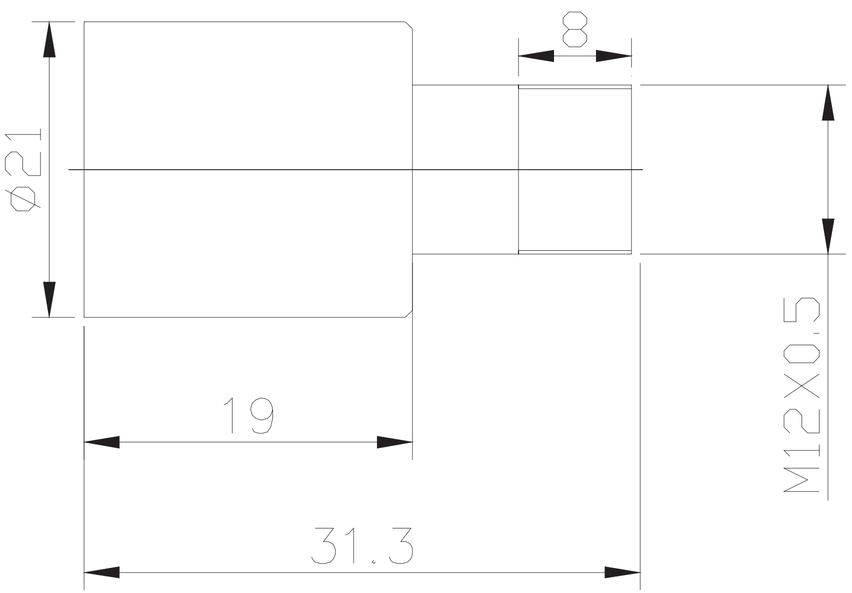
F1024/F512

- **ARM Cortex-M3 CPU platform**
  - High Performance 32-bit processor @ up to 48 MHz
  - Memory Protection Unit
- **Flexible Energy Management System**
  - 20 nA @ 3 V Shutoff Mode
  - 0.4 µA @ 3 V Shutoff Mode with RTC
  - 0.8 µA @ 3 V Stop Mode, including Power-on Reset, Brown-out Detector, RAM and CPU retention
  - 1.1 µA @ 3 V Deep Sleep Mode, including RTC with 32.768 kHz oscillator, Power-on Reset, Brown-out Detector, RAM and CPU retention
  - 80 µA/MHz @ 3 V Sleep Mode
  - 219 µA/MHz @ 3 V Run Mode, with code executed from flash
- **1024/512 KB Flash**
  - Read-while-write support
- **128 KB RAM**
- **85 General Purpose I/O pins**
  - Configurable push-pull, open-drain, pull-up/down, input filter, drive strength
  - Configurable peripheral I/O locations
  - 16 asynchronous external interrupts
  - Output state retention and wake-up from Shutoff Mode
- **12 Channel DMA Controller**
- **12 Channel Peripheral Reflex System (PRS) for autonomous inter-peripheral signaling**
- **Hardware AES with 128/256-bit keys in 54/75 cycles**
- **Timers/Counters**
  - 4× 16-bit Timer/Counter
    - 4×3 Compare/Capture/PWM channels
    - Dead-Time Insertion on TIMER0
  - 16-bit Low Energy Timer
  - 1× 24-bit Real-Time Counter and 1× 32-bit Real-Time Counter
  - 3× 16/8-bit Pulse Counter with asynchronous operation
  - Watchdog Timer with dedicated RC oscillator @ 50 nA
- **Backup Power Domain**
  - RTC and retention registers in a separate power domain, available in all energy modes
  - Operation from backup battery when main power drains out
- **External Bus Interface for up to 4x256 MB of external memory mapped space**
  - TFT Controller with Direct Drive
- **Communication interfaces**
  - 3× Universal Synchronous/Asynchronous Receiver/Transmitter
    - UART/SPI/SmartCard (ISO 7816)/IrDA/I2S
  - 2× Universal Asynchronous Receiver/Transmitter
  - 2× Low Energy UART
    - Autonomous operation with DMA in Deep Sleep Mode
  - 2× I^2C Interface with SMBus support
    - Address recognition in Stop Mode
- **Ultra low power precision analog peripherals**
  - 12-bit 1 Msamples/s Analog to Digital Converter
    - 8 single ended channels/4 differential channels
    - On-chip temperature sensor
  - 12-bit 500 ksamples/s Digital to Analog Converter
  - 2× Analog Comparator
    - Capacitive sensing with up to 16 inputs
  - 3× Operational Amplifier
    - 6.1 MHz GBW, Rail-to-rail, Programmable Gain
  - Supply Voltage Comparator
- **Low Energy Sensor Interface (LESENSE)**
  - Autonomous sensor monitoring in Deep Sleep Mode
  - Wide range of sensors supported, including LC sensors and capacitive buttons
- **Ultra efficient Power-on Reset and Brown-Out Detector**
- **Debug Interface**
  - 2-pin Serial Wire Debug interface
    - 1-pin Serial Wire Viewer
  - Embedded Trace Module v3.5 (ETM)
- **Pre-Programmed UART Bootloader**
- **Temperature range -40 to 85 ºC**
- **Single power supply 1.98 to 3.8 V**
- **LQFP100 package**

32-bit ARM Cortex-M0+, Cortex-M3 and Cortex-M4 microcontrollers for:

- Energy, gas, water and smart metering
- Health and fitness applications
- Smart accessories
- Alarm and security systems
- Industrial and home automation

1. Product Name: Light sensitive board lens
2. Model No.: BL6012
3. Sensing Area: 1/3” CCD
4. Max Image Height: 3.6mm
5. Focal length: 6mm
6. Back Focal Length: 6.96mm
7. Aperture: F1.2
8. Iris: Fixed
9. Lens Construction: 6 Glasses
10. Horizontal Field: 44° 30″
11. M.O.D: 0.2m
12.Focus Extent: 25cm-∞
13.Weight: 32.5g



TOLERANCE		UNIT mm	MATERIAL	SCALE 5:1	SHEET 1 OF 1	QUANTITY	TYPE
DIMENSION	TOLERANCE	LENSATION smart lenses. smart solutions.		BORDER A4	NAME	DATE	PART NO. / REV.N(A
X.XX X.X X.	ø 0.05 ø 0.10 ø 0.30			DRAWING BY			DATA BASE
		TITLE	Lensagon BL6012	CHECKED BY			
DO NOT SCALE DRAWING				APRROVED BY			CONFIDENTIAL

Bibliography

[1] Mehrparvar, A. and Pignatelli, D.: Cubesat design specification rev. 13. *The CubeSat Program, California Polytechnic State University*, 2014.

[2] Cape Peninsula University of Technology [Online]: ZACUBE-1 launch makes history. Available at: http://www.cput.ac.za/academic/faculties/engineering/zacube1, [2015, September 17].

[3] Daalen, C.V.: New funding for QB50 gives SU space project the lift-off. Available at: http://www.ee.sun.ac.za/new-funding-qb50-gives-su-space-project-lift-off/, [2015, September 17], April 2015.

[4] QB50 Project [Online]: Facilitating access to space. Available at: https://www.qb50.eu/index.php/project-description-obj/mission-objectives, [2015, April 22], 2015.

[5] Rawashdeh, S., Lumpp Jr, J., Barrington-Brown, J. and Pastena, M.: A stellar gyroscope for small satellite attitude determination. *26th Annual AIAA/USU Conference on Small Satellites*, 2012.

[6] Liebe, C.C.: Star trackers for attitude determination. *Aerospace and Electronic Systems Magazine, IEEE*, vol. 10, no. 6, pp. 10–16, 1995.

[7] Erlank, A.O.: *Development of CubeStar: A CubeSat-Compatible Star Tracker*. Master's thesis, ESL, University of Stellenbosch, Stellenbosch, South Africa, December 2013.

[8] Bridges, C., Kenyon, S., Underwood, C. and Sweeting, M.: STRaND: Surrey training research and nanosatellite demonstrator. In: *Proceedings of the1st IAA Conference on University Satellite Missions and CubeSat Workshop*. 2011.

[9] Visagie, L.: QB50: Surrey ADCS documents. PDF available at: https://www.qb50.eu/index.php/tech-docs/category/2-adcs?download=57:adcs-icd-v3, [2015, August 12], June 2014.

[10] Underwood, C., Pellegrino, S., Lappas, V.J., Bridges, C.P. and Baker, J.: Using cubesat/micro-satellite technology to demonstrate the autonomous assembly of a reconfigurable space telescope (AAReST). *Acta Astronautica*, vol. 114, pp. 112–122, 2015.

[11] Shuster, M.D.: A survey of attitude representations. *Journal of the Astronautical Sciences*, vol. 41, pp. 439–517, 1993.

[12] Shipman, J., Wilson, J., Higgins, C. and Torres, O.: *An Introduction to Physical Science*, chap. 18.3 Classifying Stars. Cengage Learning, 2015.

[13] Serway, R. and Jewett, J.: *Physics for Scientists and Engineers with Modern Physics*. Physics for Scientists and Engineers. Cengage Learning, 2007. ISBN 9780495112938.

[14] Sterken, C. and Manfroid, J.: *Astronomical Photometry: A Guide*. Astrophysics and Space Science Library. Springer Netherlands, 2012. ISBN 9789401124768.

[15] Grewal, M.S., Weill, L.R. and Andrews, A.P.: *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley-Interscience, 2007. ISBN 0470041900.

[16] Gai, E., Daly, K., Harrison, J. and Lemos, L.: Star-sensor-based satellite attitude/attitude rate estimator. *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 560–565, 1985.

[17] Fisher, H., Shuster, M. and Strikwerda, T.: Attitude determination for the star tracker mission. *Astrodynamics 1989*, pp. 139–150, 1990.

[18] Bar-Itzhack, I.Y.: Classification of algorithms for angular velocity estimation. *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 214–218, 2001.

[19] Liebe, C.C., Gromov, K. and Meller, D.M.: Toward a stellar gyroscope for spacecraft attitude determination. *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 91–99, 2004.

[20] Singla, P., Crassidis, J.L. and Junkins, J.L.: Spacecraft angular rate estimation algorithms for star tracker-based attitude determination. *Advances in the Astronautical Sciences*, vol. 114, pp. 1303–1316, 2003.

[21] Stone, R.C.: A comparison of digital centering algorithms. *The Astronomical Journal*, vol. 97, pp. 1227–1237, 1989.

[22] Ruocchio, C., Accardo, D., Rufino, G., Mattei, S. and Moccia, A.: Development and testing of a fully autonomous star tracker. In: *2nd IAA Symposium on Small Satellites for Earth Observation, Berlin, Germany*. 1999.

[23] Rawashdeh, S.: Development of a drift-free stellar gyroscope. *25th Annual AIAA/USU Conference on Small Satellites*, 2011.

[24] Rawashdeh, S., Danhauer, W.C., Lumpp Jr, J.E. *et al.*: Design of a stellar gyroscope for visual attitude propagation for small satellites. In: *Aerospace Conference, 2012 IEEE*, pp. 1–9. IEEE, 2012.

[25] Rawashdeh, S.A. and Lumpp, J.E.: Image-based attitude propagation for small satellites using RANSAC. *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 50, no. 3, pp. 1864–1875, 2014.

[26] Crassidis, J.L.: Angular velocity determination directly from star tracker measurements. *Journal of guidance, control, and dynamics*, vol. 25, no. 6, pp. 1165–1168, 2002.

[27] Liu, H.-b., Yang, J.-c., Yi, W.-j., Wang, J.-q., Yang, J.-k., Li, X.-j. and Tan, J.-c.: Angular velocity estimation from measurement vectors of star tracker. *Applied Optics*, vol. 51, no. 16, pp. 3590–3598, 2012.

[28] Accardo, D. and Rufino, G.: A procedure for three-dimensional angular velocity determination using a star sensor in high-rate rotation modes. *Acta Astronautica*, vol. 48, no. 5, pp. 311–320, 2001.

[29] Fasano, G., Rufino, G., Accardo, D. and Grassi, M.: Satellite angular velocity estimation based on star images and optical flow techniques. *Sensors*, vol. 13, no. 10, pp. 12771–12793, 2013.

[30] Hancock, B.R., Stirbl, R.C., Cunningham, T.J., Pain, B., Wrigley, C.J. and Ringold, P.G.: CMOS active pixel sensor specific performance effects on star tracker/imager position accuracy. In: *Symposium on Integrated Optics*, pp. 43–53. International Society for Optics and Photonics, 2001.

[31] Greyling, B.C.: *A charge coupled device star sensor system for a low earth orbit microsatellite*. Ph.D. thesis, University of Stellenbosch, 1995.

[32] Truesdale, N.A., Dinkel, K.J., Dischner, Z.J., Diller, J.H. and Young, E.F.: Daystar: Modeling and test results of a balloon-borne daytime star tracker. In: *Aerospace Conference, 2013 IEEE*, pp. 1–12. IEEE, 2013.

[33] Delabie, T., Vandenbussche, B. and Schutter, J.: An accurate and efficient gaussian fit centroiding algorithm for star trackers. In: *AAS/AIAA Space Flight Mechanics Meeting (475)*. 2013.

[34] Hugemann, W.: Correcting lens distortions in digital photographs. *Ingenieurbüro Morawski+ Hugemann: Leverkusen, Germany*, 2010.

[35] Fischler, M.A. and Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[36] Black, H.D.: A passive system for determining the attitude of a satellite. *AIAA Journal*, vol. 2, no. 7, pp. 1350–1351, 1964.

[37] Markley, F.L. and Crassidis, J.L.: *Fundamentals of Spacecraft Attitude Determination and Control*. 1st edn. Microcosm Press and Springer, New York, 2014.

[38] Farrell, J., Stuelpnagel, J., Wessner, R., Velman, J. and Brook, J.: A least squares estimate of satellite attitude (Grace Wahba). *SIAM Review*, vol. 8, no. 3, pp. 384–386, 1966.

[39] Wertz, J.R.: *Spacecraft attitude determination and control*, vol. 73. Springer Science & Business Media, 1978.

[40] Keat, J.E.: Analysis of least–squares attitude determination routine DOAOP. Paper CSC/TM–77/6034, National Aeronautics and Space Administration, Goddard Space Flight Center, 1977.

[41] Markley, F.L.: Attitude determination using vector observations and the singular value decomposition. *The Journal of the Astronautical Sciences*, vol. 36, no. 3, pp. 245–258, 1988.

[42] Shuster, M.D.: Maximum likelihood estimation of spacecraft attitude. *Journal of the Astronautical Sciences*, vol. 37, pp. 79–88, 1989.

[43] The City College of New York, Division of Science [Online]: Least squares. Available at: `http://www.sci.ccny.cuny.edu/~szlam/2013-fall-366/least_squares.pdf`, [2014, December 2], 2013.

[44] Strutz, T.: *Data Fitting and Uncertainty: A practical introduction to weighted least squares and beyond*, chap. 3. ISBN 978-3-8348-1022-9. Vieweg and Teubner, 2010.

[45] Kailath, T., Sayed, A.H. and Hassibi, B.: *Linear Estimation*, vol. 1. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[46] Aitken, A.C.: On least squares and linear combination of observations. *Proceedings of the Royal Society of Edinburgh*, vol. 55, pp. 42–48, 1936.

[47] Liebe, C.C.: Accuracy performance of star trackers – a tutorial. *IEEE Transactions on Aerospace and Electronic Systems*, pp. 587–599, 2002.

[48] Enright, J. and McVittie, G.R.: Color star tracking ii: matching. *Optical Engineering*, vol. 52, no. 1, 2013.

[49] Sun, T., Xing, F. and You, Z.: Optical system error analysis and calibration method of high-accuracy star trackers. *Sensors*, vol. 13, no. 4, pp. 4598–4623, 2013.

[50] Dzamba, T. and Enright, J.: Calibration techniques for low-cost star trackers. *23rd Annual AIAA/USU Conference on Small Satellites*, 2009.

[51] Samaan, M.A., Griffith, T., Singla, P. and Junkins, J.L.: Autonomous on-orbit calibration of star trackers. In: *Core Technologies for Space Systems Conference (Communication and Navigation Session)*. 2001.